



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

“SIMULADOR NUMÉRICO DE ACEITE CON
INTERFAZ DE VISUALIZACIÓN PARA USO
ACADÉMICO”

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO PETROLERO
P R E S E N T A N :

ORLANDO HERNÁNDEZ ROBLES
PAUL DEL VALLE MARTÍNEZ

DIRECTOR: DR. VICTOR HUGO ARANA ORTÍZ



MEXICO, D.F.

ABRIL, 2005

m. 343212



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA
DIRECCIÓN
60-I-1194

SR. ORLANDO HERNÁNDEZ ROBLES
Presente

En atención a su solicitud, me es grato hacer de su conocimiento el tema que propuso el profesor Dr. Víctor Hugo Arana Ortiz y que aprobó esta Dirección para que lo desarrolle usted como tesis de su examen profesional de Ingeniero Petrolero:

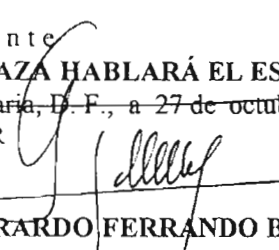
**SIMULADOR NUMÉRICO DE ACEITE CON INTERFAZ DE VISUALIZACIÓN PARA
USO ACADÉMICO**

	RESUMEN
	CONTENIDO
	TABLAS Y FIGURAS
I	INTRODUCCIÓN
II	FORMULACIÓN MATEMÁTICA DEL PROBLEMA
III	SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE FLUJO
IV	INTERFAZ DE VISUALIZACIÓN
V	APLICACIÓN DEL SIMULADOR
VI	DISCUSIÓN DE RESULTADOS
VI	CONCLUSIONES Y RECOMENDACIONES
	NOMENCLATURA
	REFERENCIAS
	APÉNDICES

Ruego a usted cumplir con la disposición de la Dirección General de la Administración Escolar en el sentido de que se imprima en lugar visible de cada ejemplar de la tesis el título de ésta.

Asimismo, le recuerdo que la Ley de Profesiones estipula que se deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito para sustentar examen profesional.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cd. Universitaria, D. F., a 27 de octubre de 2004
EL DIRECTOR


M. en C. GERARDO FERRANDO BRAVO
GFB*JAGC*gtg



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA
DIRECCIÓN
60-I-1195

SR. PAUL DEL VALLE MARTÍNEZ
Presente

En atención a su solicitud, me es grato hacer de su conocimiento el tema que propuso el profesor Dr. Víctor Hugo Arana Ortiz y que aprobó esta Dirección para que lo desarrolle usted como tesis de su examen profesional de Ingeniero Petrolero:


**SIMULADOR NUMÉRICO DE ACEITE CON INTERFAZ DE VISUALIZACIÓN PARA
USO ACADÉMICO**

	RESUMEN
	CONTENIDO
	TABLAS Y FIGURAS
I	INTRODUCCIÓN
II	FORMULACIÓN MATEMÁTICA DEL PROBLEMA
III	SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE FLUJO
IV	INTERFAZ DE VISUALIZACIÓN
V	APLICACIÓN DEL SIMULADOR
VI	DISCUSIÓN DE RESULTADOS
VI	CONCLUSIONES Y RECOMENDACIONES
	NOMENCLATURA
	REFERENCIAS
	APÉNDICES

Ruego a usted cumplir con la disposición de la Dirección General de la Administración Escolar en el sentido de que se imprima en lugar visible de cada ejemplar de la tesis el título de ésta.

Asimismo, le recuerdo que la Ley de Profesiones estipula que se deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito para sustentar examen profesional.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cd. Universitaria, D. F., a 27 de octubre de 2004
EL DIRECTOR


M. en C. GERARDO FERRANDO BRAVO
GFB*JAGC*gtg



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA EN CIENCIAS DE LA TIERRA

“SIMULADOR NUMÉRICO DE ACEITE CON INTERFAZ DE VISUALIZACIÓN
PARA USO ACADÉMICO”

TESIS PRESENTADA POR:
HERNÁNDEZ ROBLES ORLANDO
DEL VALLE MARTÍNEZ PAUL

DIRIGIDA POR:
DR. VICTOR HUGO ARANA ORTÍZ

JURADO DEL EXAMEN PROFESIONAL:

PRESIDENTE: ING. MANUEL VILLAMAR VIGUERAS
VOCAL: DR. VICTOR HUGO ARANA ORTÍZ
SECRETARIO: ING. GUILLERMO TREJO REYES
1ER SUPLENTE: M. C. JOSÉ MANUEL REYES AGUIRRE
2DO. SUPLENTE: ING. MARTÍN C. VELÁZQUEZ FRANCO

Ciudad Universitaria, México, D. F. marzo del 2005

Agradecimientos:

A mi querida Madre Amparito, que sin sus palabras de aliento y benevolencia para que concluyera esta meta que algún día me tracé me ayudo a concluir.

A Paul y Lilia Alejandra, por su paciencia, amor y cariño de hijos, siempre estarán en mi corazón, Dios los bendiga!

A mi amigo Orlando por compartir conmigo la culminación de esta meta.

Al Doctor Victor Hugo Arana Ortiz por su tiempo, apoyo, consejos y dedicación a realizar este humilde, sencillo y poderoso trabajo y sobre todo por tú amistad.. Gracias.

ÍNDICE.

RESUMEN.	1
FIGURAS.	2
I. INTRODUCCIÓN.	4
II. FORMULACIÓN MATEMÁTICA DEL PROBLEMA.	16
II.1. Ecuación de continuidad.	17
II.2. Ecuación de momento: Ley de Darcy.	22
II.3. Ecuación de estado: fluido ligeramente compresible.	24
II.4. Ecuación general de flujo de fluidos a través de medios porosos.	27
III. SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE FLUJO.	30
III.1. Características de mallas numéricas.	31
III.2. Método de aproximación mediante diferencias finitas.	35
III.2.1. Serie de Taylor.	35
III.2.2. Aproximaciones en espacio.	37
III 2.3. Aproximaciones en tiempo.	40
III.3. Condiciones iniciales y de frontera.	44
III.4. Conceptos de estabilidad, error y convergencia.	48
III.5. Solución numérica de las ecuaciones para fluido monofásico en tres dimensiones.	50
III.6. Solución numérica de las ecuaciones para fluido multifásico en tres dimensiones.	60
III.6.1. Ecuaciones para flujo multifásico.	60
III.6.2. Aproximación mediante diferencias finitas.	62
III.6.3. Planteamiento de la solución.	63
III.7. Modelado de pozos.	65
III.7.1. Modelo de Peaceman.	66
IV. INTERFAZ DE VISUALIZACION.	68
IV.1. Descripción de la Interfaz Winteracter.	68
IV.2. Estructura de la interfaz desarrollada.	75
IV.3. Descripción de subrutinas desarrolladas.	76

IV.3.1. Subrutina de menú de entrada.	77
IV.3.2. Subrutina de visualización.	79
IV.3.3. Subrutina de graficación en 3D.	81
IV.3.4. Subrutina de graficación en 2D.	83
V. APLICACION DEL SIMULADOR.	85
V.1. Validación del modelo numérico.	85
V.1.1. Coordenadas Cartesianas.	86
V.1.2. Coordenadas Radiales.	89
V.2. Comportamiento de un solo pozo productor.	94
V.3. Comportamiento de cuatro pozos productores.	96
V.4. Comportamiento de un solo pozo productor horizontal.	98
VI. DISCUSION DE RESULTADOS.	100
VI.1. Comportamiento de un solo pozo productor.	100
VI.2. Comportamiento de cuatro pozos productores.	102
VI.3. Comportamiento de un solo pozo productor horizontal.	104
CONCLUSIONES Y RECOMENDACIONES.	106
NOMENCLATURA.	108
REFERENCIAS.	110
APENDICES A. Principio de conservación de la materia. Balance de masa.	113
APENDICES B. Subrutina de visualización gráfica (Código).	116
APENDICES C. CD de aplicación.	129

RESUMEN

El objetivo de realizar este trabajo es poseer una herramienta didáctica para facilitar la comprensión de la explotación de un yacimiento de aceite bajo diferentes esquemas de explotación.

Este trabajo presenta un simulador numérico en tres dimensiones cartesianas para aceite en su etapa de bajosaturación. El simulador está equipado con una interfaz de visualización gráfica para facilitar el análisis y la comprensión del comportamiento de yacimientos de hidrocarburos con estas características. El trabajo propone el uso de la interfaz de visualización como una herramienta de aprendizaje que se podrá usar tanto con problemas con características académicas y con problemas reales de campo.

El trabajo presenta una breve pero suficiente teoría para entender la construcción de un simulador para yacimientos de aceite. El simulador es desarrollado en Fortran 90 manejando memoria dinámica y estructura modular. La interfaz de visualización es desarrollada con Winteracter, la cual es compatible con Fortran 90. El simulador tiene implementado un interfaz para manejar n número de pozos de una forma práctica, así como también la información necesaria para alimentar al simulador.

El simulador fue validado con soluciones analíticas en una dimensión tanto en coordenadas cartesianas como cilíndricas.

Los casos muestran que el simulador es de uso práctico y puede ser usado con casos reales con cualquier número de pozos.

FIGURAS

1.1a	Modelo de Diferencias Finitas	8
1.1b	Modelo de Volumen Finito	8
1.2a	Yacimiento en 1D	10
1.2b	Yacimiento en 2D	10
1.2c	Yacimiento en 3D	10
1.3a	Geometría cilíndrica de flujo en una sola dirección: r .	11
1.3b	Geometría cilíndrica de flujo en dos direcciones: r y θ .	11
1.3c	Geometría cilíndrica de flujo en tres direcciones: r , θ y z .	11
1.4a	Geometría esférica de flujo en una sola dirección: r .	12
1.4b	Geometría esférica de flujo en dos direcciones: r y θ .	12
1.4c	Geometría esférica de flujo en tres direcciones: r , θ y ϕ .	12
2.1	Volumen de control elemental	17
3.1	Malla cartesiana informe de nodos distribuidos	31
3.2	Malla cartesiana informe de nodos centrados	32
3.3	Malla en coordenadas cilíndricas	33
3.4	Malla radial informe de nodos distribuidos	33
3.5	Análisis de la Serie de Taylor	36
3.6a	Análisis de la Serie de Taylor. Aproximaciones Progresivas	40
3.6b	Análisis de la Serie de Taylor. Aproximaciones Regresivas	40
3.7	Análisis de la Serie de Taylor. Aproximaciones Centrales	42
3.8	Estabilidad de un algoritmo numérico	49
3.9	Modelo tridimensional de núcleo típico de malla cartesiana de cálculo.	55
4.1	El Ambiente de Desarrollo Winteracter, WiDE.	69
4.2	Consola de Errores, WiDE.	71
4.3	Editor de Texto Winteracter, WED.	73
4.4	Menú de Entrada de la subrutina de visualización.	77
4.5	Consola de Visualización Gráfica.	79

FIGURAS

4.6	Gráfica 3D.	81
4.7	Gráficas 2D.	83
5.1	Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en una sola dimensión (caso I).	87
5.2	Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en una sola dimensión (caso II).	88
5.3	Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en coordenadas radiales.	93
5.4	Simulación Pozo Productor Vertical al tiempo total de simulación.	95
5.5	Coordenadas. Caso de Simulación Pozo Productor Vertical al tiempo total de simulación.	95
5.6	Simulación Cuatro Pozos Productores Verticales al tiempo total de simulación.	96
5.7	Coordenadas. Caso de Simulación Cuatro Pozos Productores Verticales al tiempo total de simulación.	97
5.8	Simulación Pozo Productor Horizontal al tiempo total de simulación.	98
5.9	Coordenadas. Caso de Simulación Pozo Productor Horizontal al tiempo total de simulación.	99
6.1	Primer caso. Comportamiento de un pozo productor vertical.	100
6.2	Segundo caso. Comportamiento de cuatro pozos productores verticales.	102
6.3	Tercer caso. Comportamiento de un pozo productor horizontal.	104

CAPITULO I. INTRODUCCIÓN.

Definición y Objetivo de la Simulación Numérica de Yacimientos.

Los modelos son usados para describir procesos que suceden en todas las ramas de la ciencia y tecnología. Diferentes tipos de modelos son usados en todas las áreas de la Industria Petrolera incluyendo la Ingeniería de Yacimientos. Los modelos pueden ser físicos o matemáticos. Los primeros son aquellos que a una escala apropiada se construyen para analizar algún fenómeno y estudiarlo, por ejemplo, un tubo de acrílico para modelar el flujo de fluidos a través de tuberías o una celda PVT para modelar el comportamiento de un fluido en el yacimiento. En un modelo matemático el sistema o fenómeno a ser modelado es expresado en términos de ecuaciones. Estas ecuaciones deberán de reproducir el comportamiento del sistema a diferentes condiciones. Obviamente que algunos sistemas o fenómenos no es posible reproducirlos mediante modelos físicos, ya sea por que son muy costosos o simplemente es imposible. En este caso se debe recurrir a modelos matemáticos. Este es el caso del flujo de fluidos en los yacimientos petroleros cuando se desea analizarlos a gran escala. En general las ecuaciones que gobiernan el flujo de fluidos en medios poroso son no lineales y requieren para su solución el uso de métodos numéricos. Un programa de cómputo que resuelva las ecuaciones de flujo para el flujo de fluidos es llamado Modelo Numérico o Simulador Numérico de Yacimientos.

De esta manera la Simulación Numérica de Yacimientos (SNY) combina la física, matemáticas e ingeniería de yacimientos para obtener algoritmos que deben ser programados para desarrollar una herramienta que sea capaz de predecir el comportamiento de un yacimiento de hidrocarburos bajo diferentes condiciones de explotación. Coats (1969) define a un simulador numérico como un conjunto de ecuaciones diferenciales parciales que expresan la conservación de masa y/o energía.

El objetivo de la simulación de yacimientos es proporcionar al ingeniero de diseño de explotación una herramienta confiable para predecir el comportamiento de los yacimientos de hidrocarburos bajo diferentes condiciones de operación. El modelar el comportamiento de un yacimiento de hidrocarburos bajo diferentes esquemas de producción reduce el riesgo asociado a la elección del plan de explotación y por lo tanto minimiza los flujos de efectivo negativos.

Matax y Dalton (1990) mencionan que la simulación es la única forma de describir cuantitativamente el flujo de fluidos en un yacimiento heterogéneo, cuya producción se determina no solamente por las propiedades del yacimiento, sino también por la demanda del mercado, las estrategias de inversión y las políticas gubernamentales.

Los simuladores numéricos de yacimientos son usados principalmente porque son capaces de resolver problemas que no pueden ser resueltos analíticamente o de alguna otra manera.

Filosofía y Metodología del simulador numérico.

En la simulación numérica, el flujo de fluidos es representado por la ecuación de Darcy mientras que la ecuación de continuidad se encarga de la conservación de masa. Además que los pozos inyectores y productores deberán ser considerados. Las propiedades volumétricas y termodinámicas son modeladas a través de un análisis PVT y/o ecuaciones de estado. Las condiciones iniciales deberán ser definidas, i.e. presión inicial, saturaciones de fluidos y en su caso composiciones. Asimismo, las condiciones de frontera deberán ser impuestas al dominio del yacimiento para ser consideradas y así ser simuladas lo mas cercano posible.

De esta manera la incorporación de la ley de Darcy en la ecuación de continuidad y tomando en cuenta las propiedades de los fluidos resulta en una ecuación con la capacidad de describir el comportamiento del yacimiento. Esta ecuación es conocida como la ecuación de Difusividad.

Como se discutirá posteriormente, la ecuación de Difusividad es una ecuación en derivadas parciales de segundo orden. En su forma mas general es altamente no-lineal y por lo tanto para su solución se debe de recurrir a técnicas numéricas. Lo anterior es la razón del nombre Modelo o Simulador Numérico. La solución numérica de la ecuación de Difusividad comprende la discretización en espacio del dominio (i.e. dividir el volumen del yacimiento en celdas contiguas en las cuales se realizara un balance materia implícitamente). El tiempo es también discretizado en una serie de pasos de tiempo en donde se determinará la solución.

Un procedimiento resumido de una simulación numérica es el siguiente,

- 1. Definición del Modelo Geológico.** Distribución de las propiedades de la roca y la geometría de la estructura del yacimiento.

2. **Especificación de las Propiedades** Termodinámicas de los Fluidos. Distribución en el espacio de las propiedades de los fluidos contenidos en el yacimiento.
3. **Selección de la Malla de Simulación.** Considerar la geometría del yacimiento para elegir la malla acorde a la forma del yacimiento.
4. **Inicialización.** Asignar las propiedades estáticas y dinámicas necesarias a cada celda numérica en que se dividió el yacimiento. Así como también saturaciones y presiones iniciales.
5. **Ajuste de Historia.** Reproducir la historia de presión-producción del yacimiento hasta el tiempo presente. Este es un aspecto importante del modelo de simulación. El éxito del ajuste de la historia de producción repercutirá directamente en los escenarios de producción que se pronostiquen.
6. **Predicción del Comportamiento del Yacimiento.** Partiendo del modelo “ajustado” se realizan corridas con diferentes alternativas de producción incluyendo: pozos de relleno, sistemas artificiales de producción, proyectos de recuperación secundaria, EOR y IOR.

Un aspecto importante que se debe recordar es que el ajuste de historia no es único. Esto es, pueden existir varios modelos que proporcionen prácticamente el mismo ajuste de historia. Solo uno de ellos será correcto. Por lo tanto en el ajuste de historia es de gran importancia la experiencia, capacidad y criterio del ingeniero.

Clasificación de los simuladores numéricos.

Clasificación basada en la manera en que las ecuaciones de flujo son discretizadas.

Existen dos maneras principales de discretización la EDP que gobiernan el flujo de fluidos a través de un medio poroso: el método de diferencias finitas (MDF) y el método de elemento finito (MEF). Debido a que la discretización es muy importante en la SNY, una breve explicación es presentada.

El MDF está limitado en la geometría a un bloque en forma de paralelepípedo (Ortogonal o no-Ortogonal). El MEF permite gran flexibilidad, incluyendo formas de polígonos y prismas.

En el flujo en dos dimensiones (2D), es muy fácil diseñar mallas numéricas en zonas del yacimiento donde se requiere un conocimiento con MDF (Fig. 1.1a). Por otro lado, con el MEF, la *refinación* puede llegar a ser complicada y costosa (Fig. 1.1b).

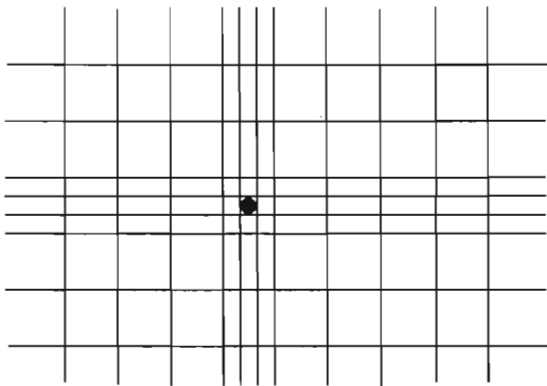


Fig. 1.1a Modelo de Diferencias Finitas

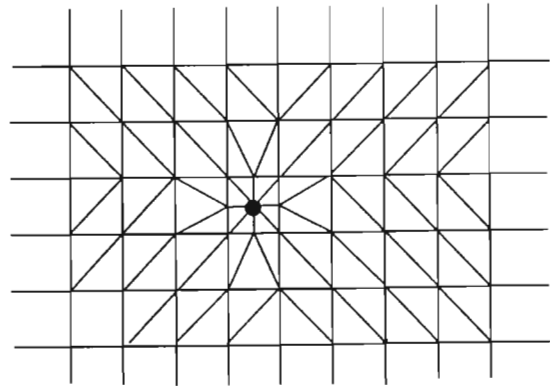


Fig. 1.1.b Modelo de Volumen Finito

A pesar de las mejores ventajas que ofrece el MEF, este tiene sus dificultades para manejar cambios bruscos en las incógnitas. Por esta razón, este método no ha sido de gran utilidad en la SNY.

Se han desarrollado técnicas mas especializadas para le refinación de la malla cerca de los pozos (asumiendo régimen permanente dentro de los bloque) haciendo posible el uso de MDF en una forma mas efectiva.

Clasificación basada en la geometría del yacimiento.

Los modelos numéricos pueden también clasificarse de acuerdo a la geometría del yacimiento, estos pueden ser: una dimensión (1D) (Fig 1.2a)., dos dimensiones (2D) (Fig 1.2b). y tres dimensiones (3D), (Fig 1.2c).

En coordenadas cartesianas, se tiene las siguientes geometrías:



Fig. 1.2a. Yacimiento en 1D

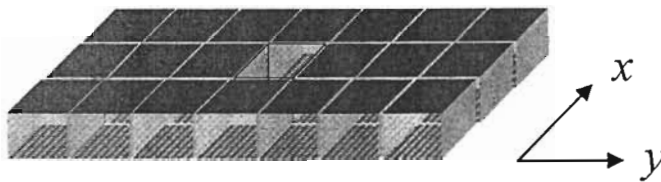


Fig. 1.2b. Yacimiento en 2D

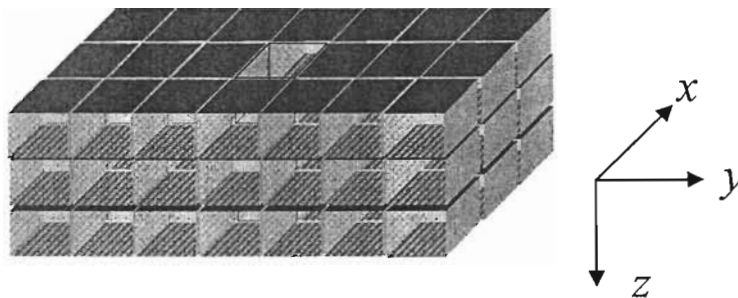


Fig. 1.2c. Yacimiento en 3D (Ertekin et al., 2001).

En coordenadas cilíndricas, se tiene las siguientes geometrías:

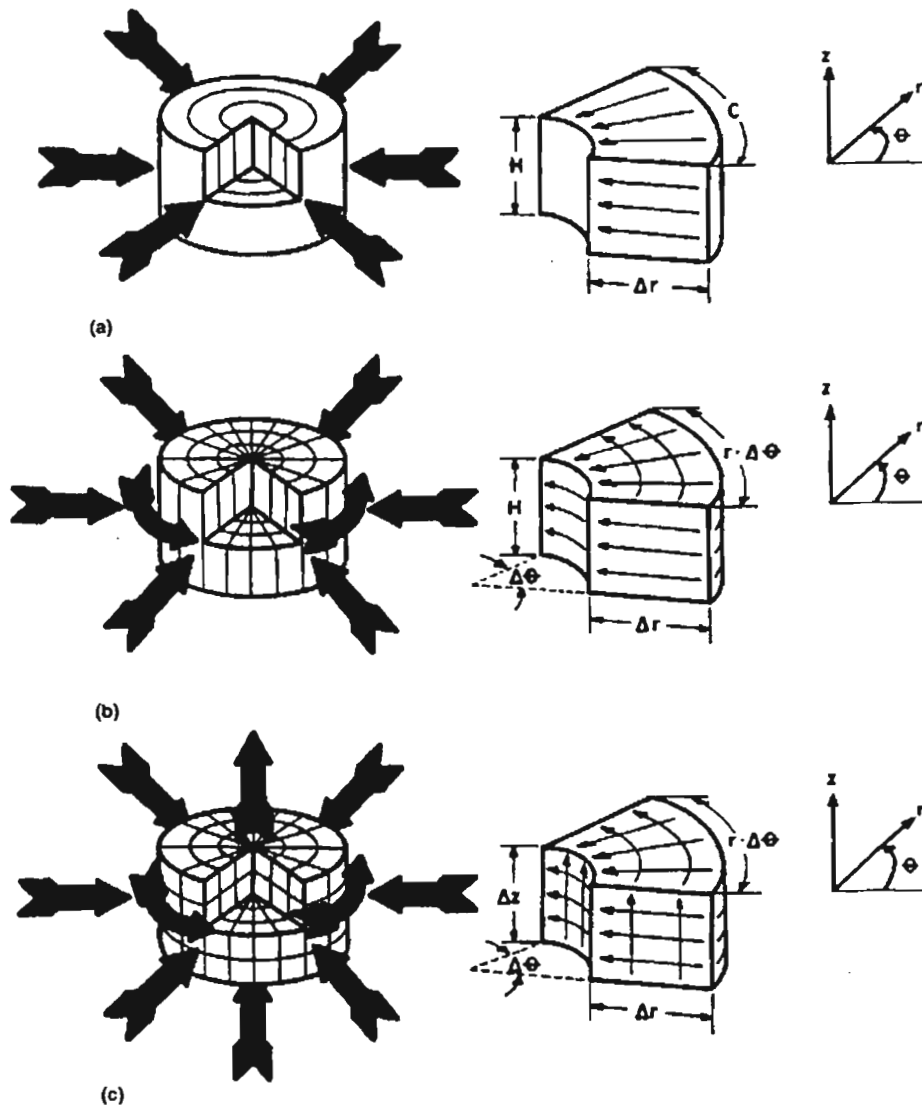


Fig. 1.3 Geometría cilíndrica de flujo. a) flujo en una sola dirección: r . b) flujo en dos direcciones: r y θ . c) flujo en tres direcciones: r , θ y z (Ertekin et al., 2001).

En coordenadas esféricas, se tiene las siguientes geometrías:

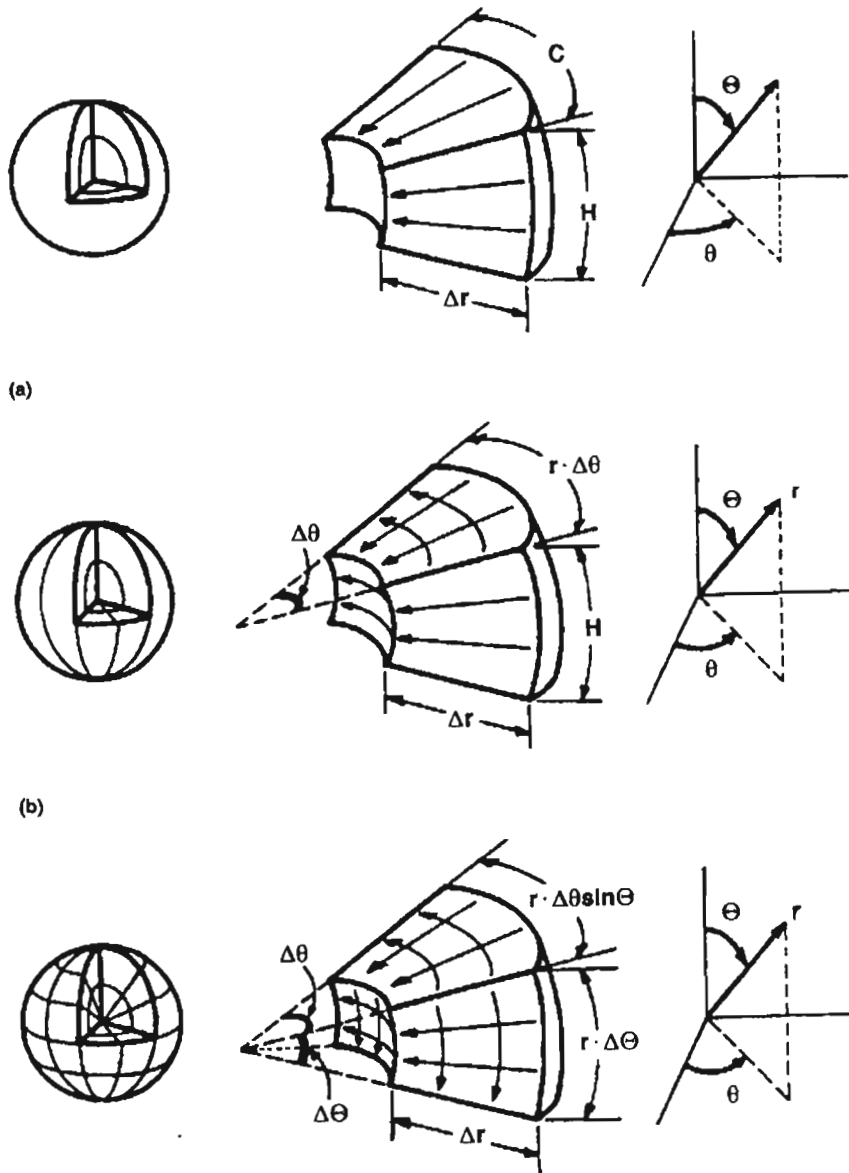
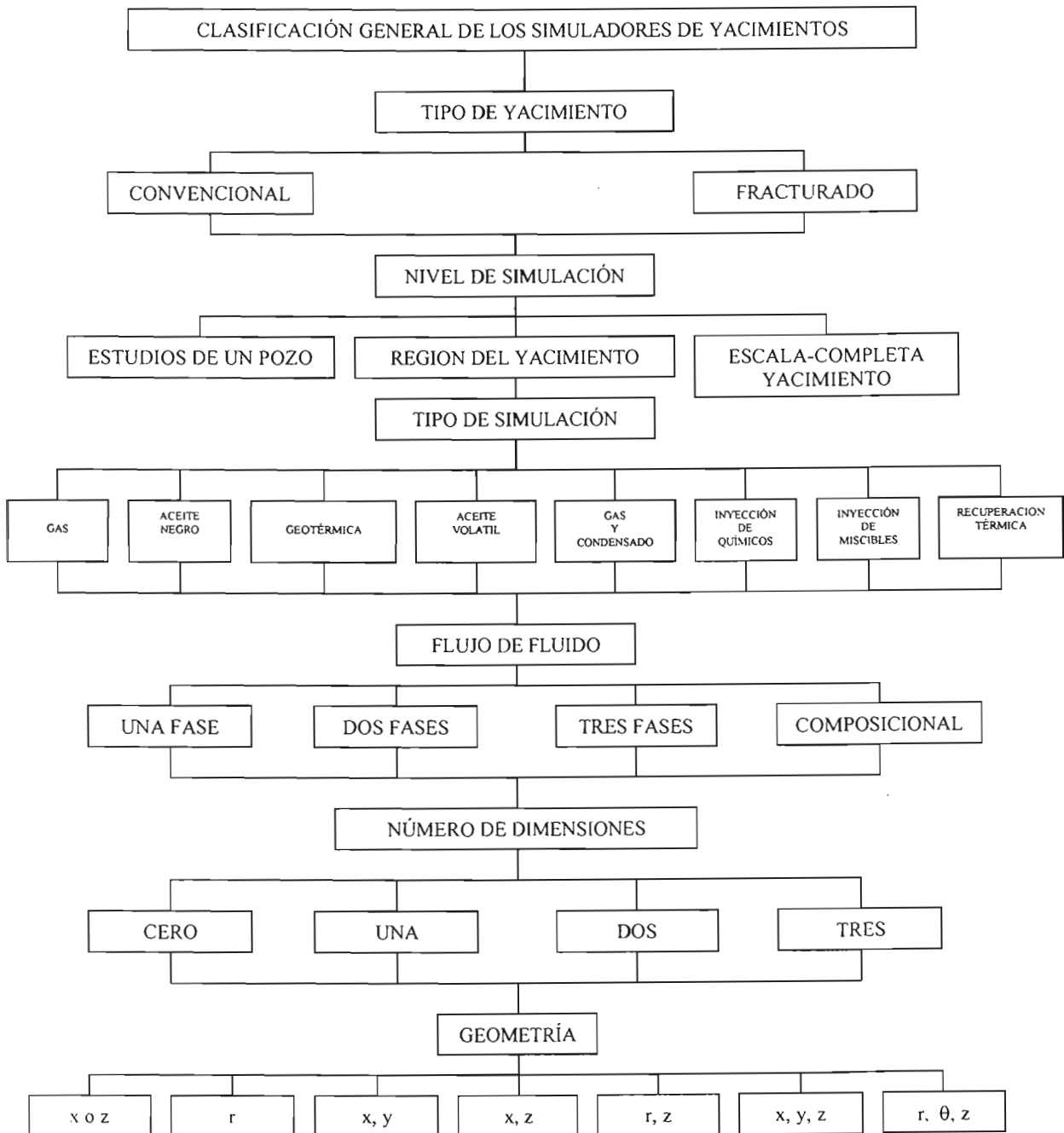


Fig. 1.4 Geometría esférica de flujo. a) flujo en una sola dirección: r . b) flujo en dos direcciones: r y θ . c) flujo en tres direcciones: r , θ y Θ . (Ertekin et al., 2001).

Clasificación General de los Simuladores.

La siguiente es una clasificación general de los simuladores incluyendo el número de fases móviles en el yacimiento (Chilingarian et al., 1992):



Beneficios de la Simulación Numérica de Yacimientos.

Los beneficios que se obtienen al usar la simulación de yacimientos para planear la explotación de un yacimiento de hidrocarburos son básicamente dos: beneficios económicos y técnicos (**Chrichlow, H.B., 1977**).

Beneficio Económicos.

El principal beneficio del uso de la simulación de yacimientos es económico. Esto se obtiene con el uso de la administración de los yacimientos disminuyendo el flujo negativo de efectivo y por supuesto incrementando la recuperación final de hidrocarburos. La administración de yacimientos es el método que busca maximizar el valor de un activo petrolero. La simulación de yacimientos es una herramienta importante para alcanzar este objetivo. Con la simulación numérica es posible obtener pronósticos, es decir, es posible simular el comportamiento del yacimiento bajo un infinito número de esquemas de producción. Al hacer esto es posible seleccionar la mejor alternativa de producción considerando la mayor ganancia para el activo.

Beneficio Técnicos.

Aunque cualquier beneficio es traducido automáticamente a beneficios económicos, es importante mencionar las ventajas técnicas que se obtienen al usar la simulación de yacimientos. La labor del Ingeniero de diseño se aligera y se sustenta grandemente. El monitoreo se facilita porque se anticipa el comportamiento del yacimiento. A medida que se obtenga información nueva se podrá actualizar el modelo de simulación para modelar el yacimiento lo más real posible. La comunicación entre el personal que conforma el equipo de trabajo se mejorará notablemente.

La relación de la Simulación Numérica con balance de materia.

El método de balance de materia es básicamente un modelo de tanque con propiedades del fluido y la roca constantes. Con el balance de materia se puede calcular volumen original y agotamiento del tanque sin considerar otros parámetros que los anteriormente descritos. Como se vio anteriormente en la simulación numérica, el yacimiento es discretizado en bloques. Conforme se ejecuta la simulación, las ecuaciones que representan el comportamiento del flujo realizan un balance de materia en cada bloque, es decir, se cumple la ecuación de continuidad: masa que sale – masa que entra = masa que se acumula.

Por lo tanto, se puede generalizar que una simulación de yacimientos con una sola celda colapsa a un balance de materia. En otras palabras, la simulación hace un balance de materia a detalle, donde esto es lo importante, y es posible cuantificar fenómenos que con el método de balance de materia tradicional no es posible realizar. Ertekin et. al. 2001 demuestra matemáticamente esta simplificación.

CAPITULO II. FORMULACIÓN MATEMÁTICA DEL PROBLEMA.

Para entender el flujo de fluidos a través de medios porosos, se tiene que postular algún sistema de ecuaciones el cual gobierne el comportamiento de los fluidos (**Chrichlow, H.B., 1977**).

Las ecuaciones básicas son obtenidas combinando varios principios físicos:

- Conservación de masa (Ecuación de continuidad)
- Conservación de momento (Ley de Darcy)
- Ecuación de estado
- Conservación de energía

Debido a que el flujo de fluidos a través de medios porosos es considerado a temperatura constante, la ecuación de conservación de energía no es considerada en el análisis. Las ecuaciones que controlan el flujo de fluidos a través de medio porosos junto con las condiciones iniciales y de frontera conforman el modelo matemático de nuestro sistema a analizar. Para resolver este modelo matemático, es necesario determinar los valores de las variables independientes, i.e. incógnitas (que en lo sucesivo llamaremos variables primarias) los cuales satisfacen todas las ecuaciones y las condiciones iniciales y de frontera simultáneamente. En general, se tienen dos opciones: modelos matemáticos y numéricos. Debido a que el sistema de ecuaciones resultante es altamente *no-lineal*, es necesario recurrir a los algoritmos numéricos.

El proceso para la derivación de las ecuaciones es básicamente el siguiente:

1. Elegir el volumen de control representativo del sistema, VC.
2. Identificar los flujos másicos que entran y salen del VC en un intervalo de tiempo.
3. Realizar un balance de masa, i.e. verificar la conservación de masa
4. Tomar el límite cuando el VC tiende a cero y el intervalo de tiempo tiende a cero.

II.1. Ecuación de continuidad.

La ecuación de continuidad es una expresión matemática del principio de conservación de masa. Esta ecuación será derivada para el caso general: flujo en tres fases en tres dimensiones.

Se usará las coordenadas cartesianas (x,y,z) con z positiva hacia abajo. Se considerará flujo laminar. La velocidad en el medio poroso es representada por la ecuación de Darcy.

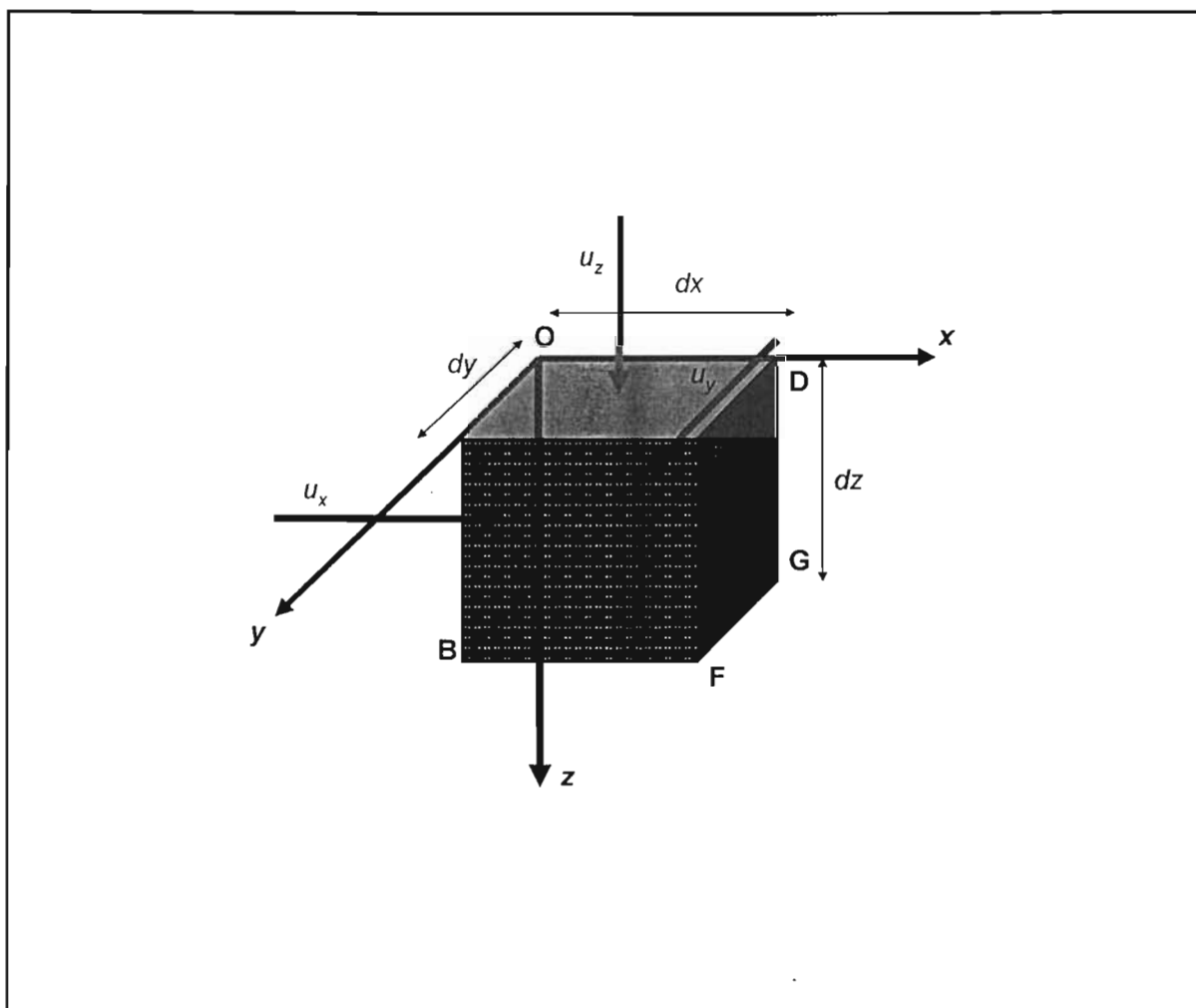


Fig. 2.1 Volumen de control elemental

La Fig. 2.1 representa el volumen de control el cual se usara para la derivación de la ecuación de continuidad.

Si u_x , u_y , y u_z representan las componentes de la velocidad en x , y , y z respectivamente, y ρ_p es la densidad del fluido p (a p y T) *entrando* a través de la cara OABC del paralelepípedo. Entonces el gasto másico por unidad de área, \tilde{m} esta dada por:

$$\tilde{m}_x = (\rho u)_x \quad (2.1)$$

$$\tilde{m}_{,x} \left[\frac{M}{tL^2} \right] = \rho_x \left[\frac{M}{L^3} \right] u_x \left[\frac{L}{t} \right]$$

El correspondiente gasto másico *saliendo* por la cara (área) DEFG es:

$$\tilde{m}_{x+\Delta x} = (\rho u)_{x+\Delta x} \quad (2.2)$$

Si S es la saturación del fluido en el medio poroso, y ϕV_b el volumen poroso en el VC, entonces la masa en el volumen de control es;

$$m = S\rho\phi V_b$$

Por lo tanto, El cambio de masa con el tiempo será:

$$\frac{[(S\rho\phi)_{t+\Delta t} - (S\rho\phi)_t] V_b}{\Delta t} \quad (2.3)$$

La conservación de masa es:

$$\{\text{masa entrando}\} - \{\text{masa saliendo}\} = \{\text{acumulación de masa}\} \quad (2.4)$$

Substituyendo las ecs. 2.1, 2.2, y 2.3 en la ec. 2.4, se tiene:

$$(\tilde{m}A)_x - (\tilde{m}A)_{x+\Delta x} = \frac{[(S\rho\phi)_{t+\Delta t} - (S\rho\phi)_t]V_b}{\Delta t}$$

Considerando la masa que se puede inyectar o producir del VC por unidad de tiempo, q_m y reorganizando, i.e. dividiendo y multiplicando el primer termino por Δx , se tiene lo siguiente,

$$\frac{[(\tilde{m}A)_x - (\tilde{m}A)_{x+\Delta x}]\Delta x}{\Delta x} \pm q_m = \frac{[(S\rho\phi)_{t+\Delta t} - (S\rho\phi)_t]V_b}{\Delta t}$$

Substituyendo ec. 2.1 y tomando límites cuando Δx y Δt tienden a cero se tiene lo siguiente:

$$\frac{\partial(\rho u)_x}{\partial x} \Delta x \pm q_m = -V_b \frac{\partial(S\rho\phi)}{\partial t} \quad (2.5)$$

Si A_x es independiente a x , entonces $V_b = A_x \Delta x$. Por lo tanto, la ec. 2.5 es

$$\frac{\partial(\rho u)_x}{\partial x} \pm \frac{q_m}{V_b} = -\frac{\partial(\phi S\rho)}{\partial t} \quad (2.6)$$

La ecuación 2.6 es la ecuación de continuidad. Similarmente para las direcciones y y z se tiene lo siguiente:

$$\frac{\partial(\rho u)_x}{\partial x} + \frac{\partial(\rho u)_y}{\partial y} + \frac{\partial(\rho u)_z}{\partial z} \pm \frac{q_m}{V_b} = -\frac{\partial(\phi S\rho)}{\partial t}$$

Utilizando la notación del operador nabla:

$$\vec{\nabla} \cdot (\rho \vec{u}) \pm \tilde{q}_m = -\frac{\partial(\phi S \rho)}{\partial t} \quad (2.7)$$

donde $\tilde{q}_m = \frac{q_m}{V_b}$ representa el gasto másico de fluido extraído por unidad de volumen de roca.

La ec. 2.7 se lee “la divergencia de ρu (Flujo másico)”. La definición del operador Nabla (Divergencia) para diferentes geometrías es:

Coordenadas cartesianas,

$$\nabla \mathbf{F} = \left(\frac{\partial F_x}{\partial x} \right) + \left(\frac{\partial F_y}{\partial y} \right) + \left(\frac{\partial F_z}{\partial z} \right)$$

Coordenadas Cilíndricas,

$$\nabla \mathbf{F} = \frac{1}{r} \frac{\partial}{\partial r} (r \cdot F_r) + \frac{1}{r} \left(\frac{\partial F_\theta}{\partial \theta} \right) + \left(\frac{\partial F_z}{\partial z} \right)$$

Coordenadas Esféricas,

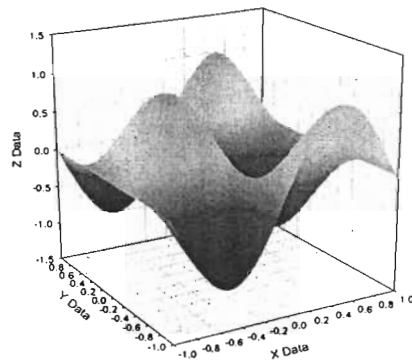
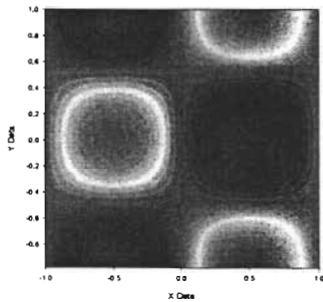
$$\nabla \mathbf{F} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \cdot F_r) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \cdot F_\theta) + \frac{1}{r \sin \theta} \left(\frac{\partial F_\phi}{\partial \phi} \right)$$

Ejemplo,

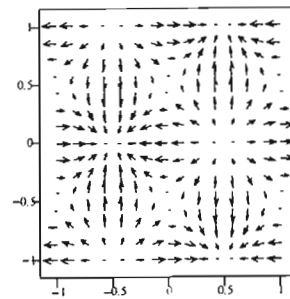
Un campo de concentraciones esta dado por $C(x,y)=-\sin(\pi x)\cos(\pi y)$. Calcular el gradiente del campo (Divergencia). Hacer un mapa del campo de concentraciones C .

$$\nabla C \equiv \left(\frac{\partial C}{\partial x}\right) \cdot i + \left(\frac{\partial C}{\partial y}\right) \cdot j = -\pi(\cos(\pi x)\cos(\pi y)) \cdot i - \sin(\pi x)\sin(\pi y) \cdot j$$

Campo de concentraciones



Magnitud y dirección del Gradiente



II.2. Ecuación de momento: Ley de Darcy.

La ley de Darcy para flujo multifásico relaciona la velocidad y el potencial:

$$\vec{u}_p = -\frac{kk_{rp}}{\mu_p} \left(\vec{\nabla} p_p - \gamma_p \vec{\nabla} D \right) \quad (2.8)$$

Donde p significa la fase, $p = o, g, w$.

Entonces, substituyendo la ec. 2.8 en la ec. 2.7, se llega a lo siguiente:

$$\vec{\nabla} \cdot \left(-\frac{kk_{rp}\rho_p}{\mu_p} \left(\vec{\nabla} p_p - \gamma_p \vec{\nabla} D \right) \right) \pm \tilde{q}_{mp} = -\frac{\partial(\phi S_p \rho_p)}{\partial t} \quad (2.9)$$

donde,

$$\tilde{q}_{mp} \left[\frac{m}{tL^3} \right] = \frac{q_{mp} \left[\frac{m}{t} \right]}{V_b \left[L^3 \right]}$$

y

$$q_{mp} \left[\frac{m}{t} \right] = q_{p,sc} \left[\frac{L^3}{t} \right] \rho_{p,sc} \left[\frac{m}{L^3} \right] \quad (2.9a)$$

La ec. 2.9 es la ecuación generalizada de flujo de un fluido a través de un medio poroso.

Para flujo monofásico de cualquier fase p , $S_p = 1$ y $k_{rp} = 1.0$, entonces la ec. 2.9 es:

$$\vec{\nabla} \cdot \left(\frac{k\rho}{\mu} \left(\vec{\nabla} p - \gamma \vec{\nabla} D \right) \right) \pm \tilde{q}_m = \frac{\partial(\phi\rho)}{\partial t} \quad (2.10)$$

II.3. Ecuaciones de estado: fluido ligeramente compresible.

La relación que existe entre la densidad y la presión en cualquier punto es especificada por una ecuación de estado. Primero se considerará el caso donde el fluido es líquido. Si la densidad se asume constante, entonces la ecuación de estado es:

$$\frac{\partial \rho}{\partial p} = 0 \quad (2.11)$$

Si el líquido es considerado ideal con compresibilidad constante, c :

$$c = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \quad (2.11a)$$

Integrando resulta en $\rho = \rho_o e^{[c(p-p_o)]}$ donde el subíndice o refiere al nivel de referencia.

Si el líquido es ligeramente compresible, entonces por la expansión de Taylor:

$$\rho = \rho_o \{1 + c(p - p_o)\} \quad (2.12)$$

Si el fluido es compresible, i.e. gas, la ecuación de estado que se emplea es la ley de los gases:

$$\begin{aligned} pV &= ZnRT \\ pV &= Z \frac{m}{M} RT \\ pM &= Z\rho RT \\ \rho &= \frac{Mp}{ZRT} \end{aligned} \quad (2.12a)$$

si se substituye $\frac{M}{RT} = \frac{\rho_s Z_s}{p_s}$, la densidad del gas esta dada por

$$\rho = \frac{\rho_s Z_s p}{p_s Z} \quad (2.13)$$

Si la compresibilidad esta dada por:

$$c = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \quad (2.14)$$

entonces,

$$\frac{\partial \rho}{\partial p} = \frac{M}{RT} \frac{\partial}{\partial p} \left(\frac{p}{Z} \right) \quad (2.15)$$

Substituyendo la ecs. 2.12a y 2.15 en la ec, 2.14;

$$c = \frac{Z}{p} \frac{\partial}{\partial p} \left(\frac{p}{Z} \right) \quad \text{desarrollándola, se tiene lo siguiente:}$$

$$c = \frac{1}{p} - \frac{1}{Z} \frac{\partial Z}{\partial p} \quad (2.16)$$

Ec. 2.16 es usada para un gas real (no ideal). Para un gas que se comporta idealmente, $Z=1$, entonces:

$$c = \frac{1}{p} \quad (2.17)$$

Ecuaciones constitutivas del medio poroso.

La variación de la porosidad con la presión se muestra con la definición de la compresibilidad de la roca:

$$c_r = \frac{1}{\phi} \left(\frac{\partial \phi}{\partial p} \right)_T \quad (2.18)$$

Por lo tanto,

$$c_r \phi = \left(\frac{\partial \phi}{\partial p} \right) \quad (2.19)$$

La ec. 2.18 asume que el volumen de la roca es constante.

II.4. Ecuación general de flujo de fluidos a través de medios porosos.

Expandiendo la ec. 2.10 en una dimensión, una sola fase fluyendo horizontalmente:

$$\vec{\nabla} \cdot \left(\frac{k\rho}{\mu} \left(\vec{\nabla} p - \gamma \vec{\nabla} D \right) \right) \pm \tilde{q}_m = \frac{\partial(\phi\rho)}{\partial t} \quad (2.10)$$

con las consideraciones anteriores ec. 2.10 es:

$$\frac{\partial}{\partial x} \left(\frac{k\rho}{\mu} \frac{\partial p}{\partial x} \right) \pm \tilde{q}_m = \frac{\partial(\phi\rho)}{\partial t} \quad (2.20)$$

La ec. 2.20 gobierna el flujo de una sola fase en una dimensión en forma horizontal asumiendo que aplica la ecuación de Darcy.

Desarrollando la ec. 2.20 (aplicando la derivación de un producto de funciones),

$$\frac{\partial(uv)}{\partial x} = u \frac{\partial(v)}{\partial x} + v \frac{\partial(u)}{\partial x}$$

Entonces,

$$\frac{k\rho}{\mu} \frac{\partial}{\partial x} \frac{\partial p}{\partial x} + \frac{k}{\mu} \frac{\partial p}{\partial x} \frac{\partial \rho}{\partial x} \pm \frac{\tilde{q}_m}{\rho_s} = \left[\phi \frac{\partial \rho}{\partial t} + \rho \frac{\partial \phi}{\partial t} \right]$$

Agrupando y aplicando la regla de la cadena,

$$\frac{k\rho}{\mu} \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) + \frac{k}{\mu} \frac{\partial p}{\partial x} \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial x} \pm \tilde{q}_m = \left[\phi \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} + \rho \frac{\partial \phi}{\partial p} \frac{\partial p}{\partial t} \right]$$

$$\frac{k\rho}{\mu} \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) + \frac{k}{\mu} \frac{\partial \rho}{\partial p} \left(\frac{\partial p}{\partial x} \right)^2 \pm \tilde{q}_m = \left[\phi \rho \left(\frac{1}{\rho} \frac{\partial \rho}{\partial p} + \frac{1}{\phi} \frac{\partial \phi}{\partial p} \right) \frac{\partial p}{\partial t} \right]$$

El gradiente de presión al cuadrado es muy pequeño y puede ser eliminado. También, la compresibilidad del fluido y formación pueden ser substituidos, entonces la ecuación final es:

$$\frac{k\rho}{\mu} \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) \pm \tilde{q}_m = \left[\phi \rho c_t \frac{\partial p}{\partial t} \right] \quad (2.21)$$

donde $c_t = c_r + c_f$

Dividiendo la ec. 2.21 por la densidad a condiciones estándar, se tiene lo siguiente:

$$\frac{k\rho}{\mu\rho_s} \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) \pm \frac{\tilde{q}_m}{\rho_s} = \left[\phi \frac{\rho}{\rho_s} c_t \frac{\partial p}{\partial t} \right] \quad (2.22)$$

Si el factor de volumen del fluido esta dado por:

$$B = \frac{\rho_s}{\rho} \quad (2.23)$$

Substituyendo la ec. 2.23 en la ec. 2.22:

$$\frac{k}{\mu B} \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) \pm \tilde{q}_s = \left[\frac{\phi c_t}{B} \frac{\partial p}{\partial t} \right] \quad (2.24)$$

El término \tilde{q}_s es el gasto volumétrico a condiciones estándar por unidad de volumen de roca:

$$\tilde{q}_s \left[\frac{L^3}{tL^3} \right] = \frac{\tilde{q}_m \left[\frac{m}{tL^3} \right]}{\rho_s \left[\frac{m}{L^3} \right]}$$

Si k, μ son constantes, la ec, 2.24 pasa ser la siguiente (sin considerar fuente o sumidero):

$$\frac{\partial^2 p}{\partial x^2} \pm \tilde{q}_s = \left[\frac{\phi \mu c_t}{k} \frac{\partial p}{\partial t} \right] \quad (2.24a)$$

La ec. 2.24a puede ser extendida a tres dimensiones de la siguiente manera:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \pm \tilde{q}_s = \left[\frac{\phi \mu c_t}{k} \frac{\partial p}{\partial t} \right] \quad (2.25)$$

CAPITULO III. SOLUCIÓN NUMÉRICA DE LAS ECUACIONES DE FLUJO.

Las ecuaciones que gobiernan el flujo de fluidos a través de un medio poroso son muy complejas para ser resueltas analíticamente, i.e. son altamente no-lineales. Solo en casos ideales el flujo en una sola fase puede ser solucionado analíticamente, i.e. la última parte del capítulo previo. En consecuencia, se debe usar algoritmos numéricos para su solución. Los algoritmos numéricos funcionan en un dominio discreto en lugar de un dominio continuo como lo hacen las soluciones analíticas.

La discretización de una ecuación diferencial puede ser hecha por diferentes métodos. Los métodos más comunes incluyen: Método de Diferencias Finitas (MDF), Método de Elemento Finito (MEF), y el Método de Volumen de Control (MVC). En este trabajo de tesis se usará el MDF.

El principal objetivo del MDF es generar valores de la función $p(x,t)$ en los nodos de una malla que cubre el dominio de solución, en una secuencia de niveles de tiempo que son separados un Δt .

La malla numérica para el MDF puede ser definida en las diferentes geometrías vistas en el capítulo I. Estas pueden ser: coordenadas cartesianas, cilíndricas o esféricas, Fig 1.2-1.4. La elección de coordenadas es guiada por la geometría del dominio del problema y es hecha con el propósito de facilitar la implantación de las condiciones de frontera.

El procedimiento para derivar ecuaciones en diferencias finitas (EDF) consiste en aproximar las derivadas en la ecuación diferencial mediante una serie de Taylor truncada, a este proceso se le denomina discretización.

III.1. Características de mallas numéricas.

Malla Cartesiana.

Este tipo de malla es más comúnmente usada en simulación de yacimientos. Los puntos de la malla son las intersecciones de los planos coordenados. En dos dimensiones un punto de malla tiene cuatro puntos vecinos.

La malla cartesiana puede ser uniforme de nodos distribuidos o centrados. Para una dimensión, por ejemplo en x , la malla uniforme con nodos distribuidos será como se muestra a continuación (Galindo, N.A., Tesis, UNAM 1998).

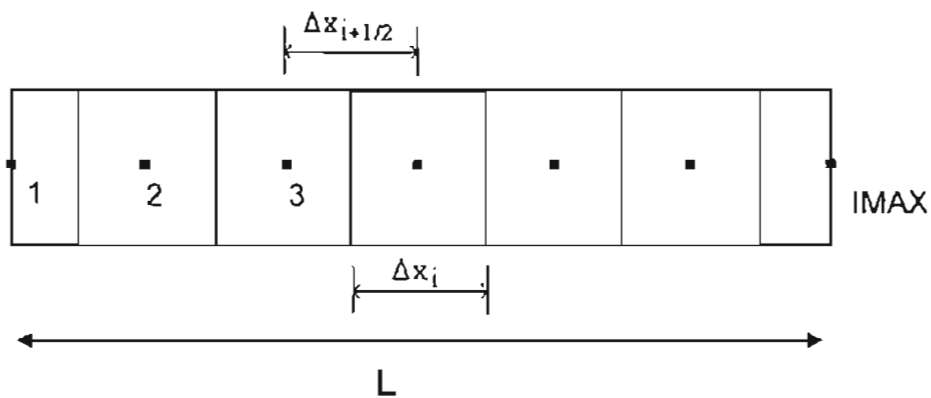


Fig. 3.1 Malla cartesiana uniforme de nodos distribuidos

Si el total de nodos es $IMAX$ y su espaciado es Δx , este último es definido como,

$$\Delta x = \frac{L}{IMAX - 1}$$

En las fronteras el volumen de las celdas es la mitad del volumen de las celdas internas, esto es,

$$Vr_i = \begin{cases} \frac{A\Delta x_i}{2} & i = 1, IMAX \\ A\Delta x_i & i = 2, 3, \dots, IMAX - 1 \end{cases}$$

Considerando lo anterior a posición de los nodos es,

$$x_i = (i - 1)\Delta x_i \quad i = 1, 2, \dots, IMAX$$

La posición de la frontera de las celdas es:

$$x_{i-1/2} = (i - \frac{1}{2})\Delta x_i \quad i = 1, 2, \dots, IMAX$$

Una malla uniforme con nodos centrados será como la siguiente figura,

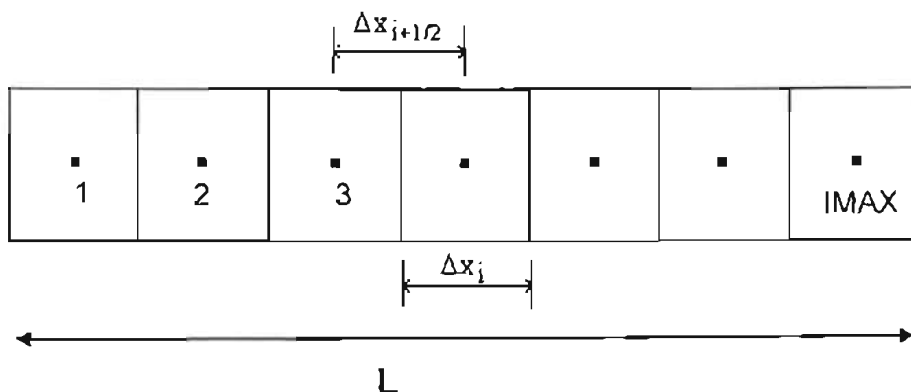


Fig. 3.2 Malla cartesiana uniforme de nodos centrados

Si el total de nodos es $IMAX$ y su espaciamento es Δx , este último es definido como,

$$\Delta x_i = \frac{L}{IMAX}$$

El volumen de cada celda es idéntico para todos los bloques,

$$V_i = A\Delta x_i \quad i = 1, 2, 3, \dots, IMAX$$

La posición de los nodos (centro de cada celda) es,

$$x_i = (i - \frac{1}{2})\Delta x_i \quad i = 1, 2, \dots, IMAX$$

y las posiciones de las fronteras de cada celda será:

$$x_{i+\frac{1}{2}} = i\Delta x_i \quad i = 1, 2, \dots, IMAX$$

Malla Radial.

Estas mallas son confiables para modelar el flujo de fluidos a un solo pozo. Para el caso isotrópico, la malla corresponde a un sistema en coordenadas cilíndricas (Galindo, N.A., Tesis, UNAM 1998).

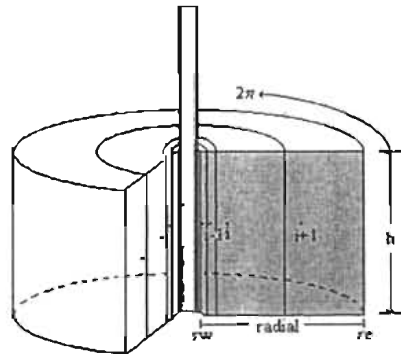


Fig. 3.3 Malla en coordenadas cilíndricas

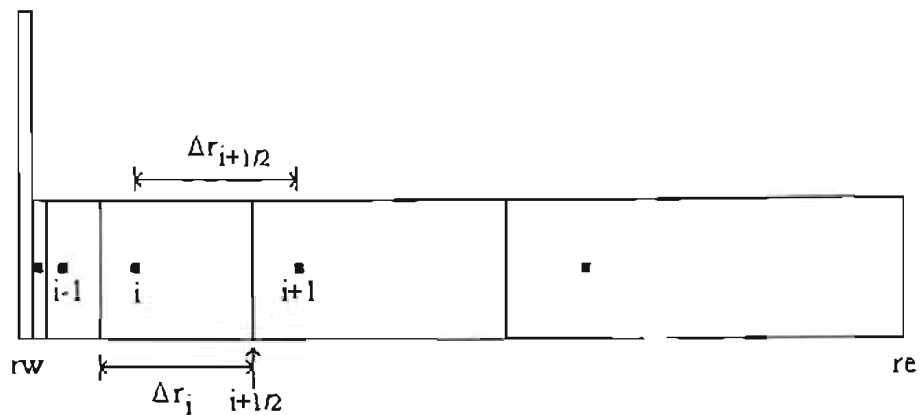


Fig. 3.4 Malla radial informe de nodos distribuidos

III.2. Método de aproximación mediante diferencias finitas.

III.2.1. Serie de Taylor.

La justificación para aproximar una ecuación en derivadas parciales usando el MDF esta basado en el análisis de una serie de Taylor (Chapra,S.C. y Canale,R.P., 1989):

Considérese la función $p(x)$ en la figura 3.5. Supóngase que el valor de $p(x)$ es conocido en el punto x . También supóngase que todas las derivadas de $p(x)$ son conocidas en el mismo punto. Por lo tanto se puede aproximar el valor de $p(x+\Delta x)$ en el punto $x+\Delta x$ con una serie de Taylor:

$$p(x + \Delta x) = p(x) + \frac{\Delta x}{1!} p'(x) + \frac{(\Delta x)^2}{2!} p''(x) + \frac{(\Delta x)^3}{3!} p'''(x) + \dots + \frac{(\Delta x)^n}{n!} p^n(x) \quad (3.1)$$

donde :

p^n = enésima derivada de p .

Esta serie infinita es en teoría exacta para un infinito número de términos. Sin embargo si se trunca la serie en el término k , un error e_t de truncamiento es introducido, i.e. los términos restantes que no son incluidos. Este error de truncamiento está dado por :

$$e_t = \frac{p(x)^{k+1} \Delta x^{k+1}}{(k+1)!} + \frac{p(x)^{k+2} \Delta x^{k+2}}{(k+2)!} + \dots + \frac{p(x)^n \Delta x^n}{n!} \quad (3.2)$$

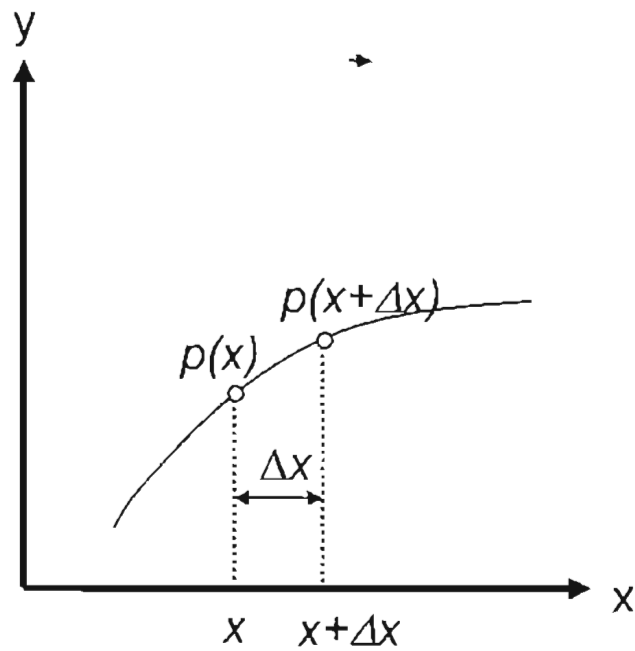


Fig. 3.5 Análisis de la Serie de Taylor

Simplificando la notación, i.e. haciendo,

$$p_i = p(x), \quad p_{i+1} = p(x + \Delta x) \quad \text{y} \quad p_{i-1} = p(x - \Delta x) \quad (3.3)$$

y considerando la ec. 3.3, la ec. 3.1 puede ser escrita como:

$$p_{i+1} = p_i + \frac{\Delta x}{1!} \frac{\partial p}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 p}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 p}{\partial x^3} + \dots + \frac{(\Delta x)^n}{n!} \frac{\partial^n p}{\partial x^n} p''(x) \quad (3.4)$$

Notese el uso de derivadas parciales debido a que $p(x,t)$.

La ecuación 3.4 sirve como base en la aproximación de las derivadas que constituyen las ecuaciones de flujo de fluidos en medios porosos que nos ocupan, como se verá a continuación.

III. 2.2. Aproximaciones en espacio.

Se tienen dos maneras de aproximar la $\frac{\partial p}{\partial x}$: con diferencias hacia delante (progresivas) y diferencias hacia atrás (regresivas).

Considerando hasta el segundo término en la ec. 3.4, se tiene lo siguiente:

$$p_{i+1} = p_i + \frac{\Delta x}{1!} \frac{\partial p}{\partial x} + O_p(\Delta x) \quad (3.5)$$

Entonces se tiene,

$$\left. \frac{\partial p}{\partial x} \right|_i = \frac{p_{i+1} - p_i}{\Delta x} - O_p(\Delta x) \quad (3.6)$$

donde el error de aproximar (error de primer orden) la derivada con solo los dos primeros términos de la serie de Taylor es :

$$O_p(\Delta x) = -\frac{\Delta x}{2!} \frac{\partial^2 p}{\partial x^2} \quad (3.7)$$

La ecuación 3.6 es la aproximación de la primera derivada mediante diferencias hacia delante y aplica para la malla de la fig. 3.2.

Similarmente, considerando las diferencias hacia atrás o regresivas, la aproximación de la primera derivada es:

$$\left. \frac{\partial p}{\partial x} \right|_i = \frac{p_i - p_{i-1}}{\Delta x} - O_p(\Delta x)$$

donde el error es:

$$O_r(\Delta x) = \frac{\Delta x}{2!} \frac{\partial^2 p}{\partial x^2} \quad (3.8)$$

Considerando las tres primeras derivadas en la ec. 3.4 y escribiendo la función $p(x)$ en $x = x_i + \Delta x$ y $x = x_i - \Delta x$ como se muestra a continuación:

$$p_{i+1} = p_i + \frac{\Delta x}{1!} \frac{\partial p}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 p}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 p}{\partial x^3} \quad (3.9)$$

$$p_{i-1} = p_i - \frac{\Delta x}{1!} \frac{\partial p}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 p}{\partial x^2} - \frac{(\Delta x)^3}{3!} \frac{\partial^3 p}{\partial x^3} \quad (3.10)$$

Restando la ec. 3.10 de la ec. 3.9 se obtiene la aproximación en diferencias centrales :

$$\left. \frac{\partial p}{\partial x} \right|_i = \frac{p_{i+1} - p_{i-1}}{2\Delta x} - O_c(\Delta x^2)$$

donde el error de segundo orden esta dado por:

$$O_c(\Delta x^2) = \frac{\Delta x^2}{6} \left[\left. \frac{\partial^3 p}{\partial x^3} \right|_p + \left. \frac{\partial^3 p}{\partial x^3} \right|_r \right] \quad (3.11)$$

Comparando el error de truncamiento de la aproximación en diferencias centrales, $O_c(\Delta x^2)$, con los obtenidos previamente para diferencias progresivas y regresivas, $O_p(\Delta x)$, y $O_r(\Delta x)$ se nota que

$$\lim_{\Delta x \rightarrow 0} O_c(\Delta x^2) < \lim_{\Delta x \rightarrow 0} O_p(\Delta x) \quad (3.12)$$

Indicando que el error de truncamiento de la aproximación de diferencias centrales, para la primera derivad, es menor que el correspondiente a diferencias progresivas o regresivas.

Con el propósito de simplificar la discretización y los sub-indices se sugiere la siguiente notación (para Δx constantes):

$$\begin{aligned}x_{i+1} &= x_i + \Delta x \\x_{i-1} &= x_i - \Delta x \\p_i &= p(x_i) \\p_{i+1} &= p(x_{i+1}) \\p_{i-1} &= p(x_{i-1})\end{aligned}\tag{3.13}$$

Aproximación de la segunda derivada.

Combinando las ecuaciones 3.9 y 3.10 (sumándolas), se puede obtener la aproximación de la segunda derivada como se muestra a continuación:

$$\frac{\partial^2 p}{\partial x^2} \Big|_i \approx \frac{p_{i+1} - 2p_i + p_{i-1}}{(\Delta x)^2}\tag{3.14}$$

donde el error es de segundo orden (Δx^2).

III. 2.3. Aproximaciones en tiempo.

Considerando la función $p(x,y,z,t)$ para la cual deseamos obtener aproximaciones en diferencias finitas de su derivada parcial con respecto al tiempo. Se puede aplicar las aproximaciones de diferencias progresivas, regresivas y centrales en tiempo. Antes de mostrar la forma de las aproximaciones, es conveniente resaltar que para representar puntos en el dominio del espacio discretizado, es común el uso de los subíndices i,j,k para representar x,y,z . Por otra parte, para representar puntos en el tiempo se emplea los superíndices n y $n+1$, que indican los niveles donde se conoce y se desconoce la solución del problema de interés.

La figura 3.6 muestra interpretación geométrica de aproximar la derivada de la función $p(x,y,z,t)$ con respecto al tiempo t .

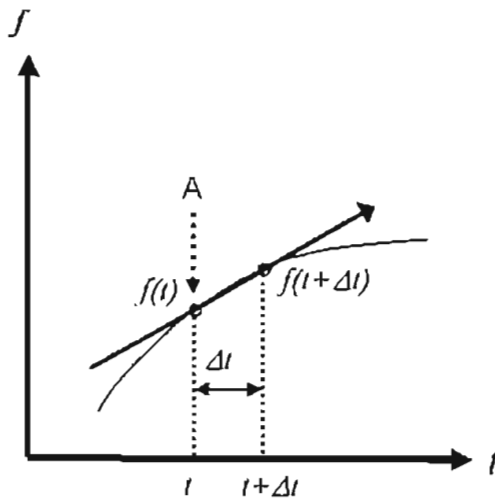


Fig. 3.6a Analisis de la Serie de Taylor. Aproximaciones Progresivas

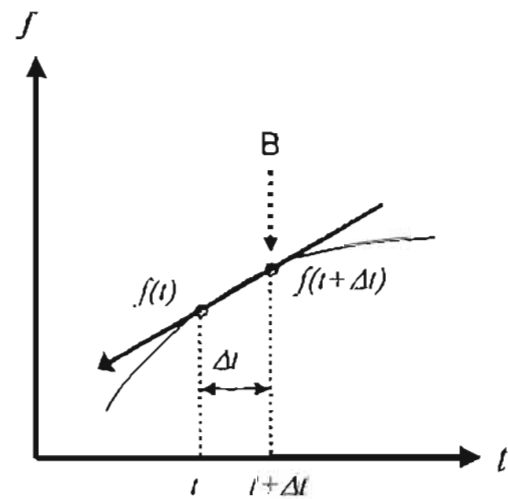


Fig. 3.6b Analisis de la Serie de Taylor. Aproximaciones Regresivas.

De esta manera la discretización de $\frac{\partial p}{\partial t}$ utilizando la serie de Taylor en diferencias progresivas y omitiendo los subíndices y, z , es :

$$p(x, t + \Delta t) = p(x, t) + \frac{\Delta t}{1!} p'(x, t) + \frac{(\Delta t)^2}{2!} p''(x, t) + \frac{(\Delta t)^3}{3!} p'''(x, t) + \dots + \frac{(\Delta t)^n}{n!} p^n(x, t) \quad (3.15)$$

En la ec. 3.15 el punto donde se parte es mostrado en la fig. 3.6a como punto A, i.e. t y el punto donde se expande la función f es $t + \Delta t$.

Considerando hasta la primera derivada, se tiene:

$$\left. \frac{\partial p}{\partial t} \right|_i^n = \frac{p_i^{n+1} - p_i^n}{\Delta t} + O(\Delta t) \quad (3.15a)$$

En diferencias regresivas es:

$$p(x, t) = p(x, t + \Delta t) + \frac{-\Delta t}{1!} p'(x, t + \Delta t) + \frac{(-\Delta t)^2}{2!} p''(x, t + \Delta t) + \dots + \frac{(-\Delta t)^n}{n!} p^n(x, t + \Delta t) \quad (3.16)$$

En la ec. 3.16 el punto donde se parte es mostrado en la fig. 3.6b como punto B, i.e. $t + \Delta t$, y el punto donde se expande la función f es t , o sea un $-\Delta t$.

Considerando hasta la primer derivada en la ec. 3.16, se tiene

$$\left. \frac{\partial p}{\partial t} \right|_i^{n+1} = \frac{p_i^{n+1} - p_i^n}{\Delta t} + O(\Delta t) \quad (3.16a)$$

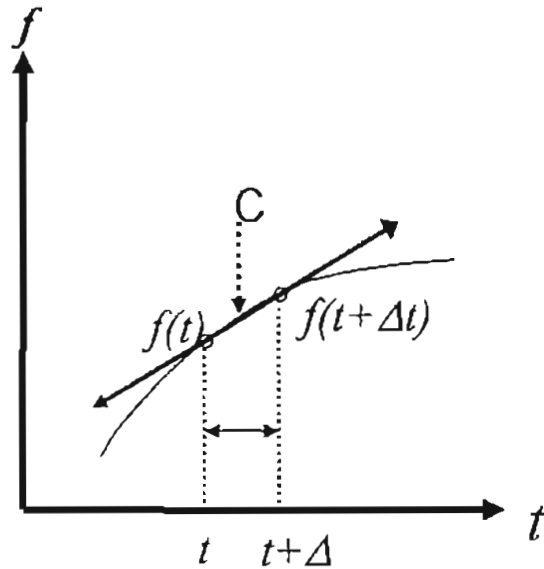


Fig. 3.7 Análisis de la Serie de Taylor. Aproximaciones Centrales

Finalmente expandiendo en ambas direcciones, i.e. en el punto medio entre t y $t+\Delta t$, o sea $t + \frac{\Delta t}{2}$, se tiene la aproximación mediante diferencias centrales:

Para la aproximación en diferencias progresivas, el punto donde se obtendrá la derivada es $t + \frac{\Delta t}{2}$ y el delta donde se expandirá es $\frac{\Delta t}{2}$ o sea $t + \frac{\Delta t}{2} + \frac{\Delta t}{2} = t + \Delta t$,

$$p(x, t + \Delta t) = p(x, t + \frac{\Delta t}{2}) + \frac{\Delta t}{1!} p'(x, t + \frac{\Delta t}{2}) + \frac{(\frac{\Delta t}{2})^2}{2!} p''(x, t + \frac{\Delta t}{2}) + \frac{(\frac{\Delta t}{2})^3}{3!} p'''(x, t + \frac{\Delta t}{2}) + \dots + \frac{(\frac{\Delta t}{2})^n}{n!} p^{(n)}(x, t + \frac{\Delta t}{2}) \quad (3.17)$$

Para la aproximación en diferencias regresivas, el punto donde se obtendrá la derivada es

$$t + \frac{\Delta t}{2} \text{ y el delta donde se expandirá es } \frac{-\Delta t}{2} \text{ o sea } t + \frac{\Delta t}{2} - \frac{\Delta t}{2} = t,$$

$$p(x, t) = p\left(x, t + \frac{\Delta t}{2}\right) + \frac{-\Delta t}{1!} p'\left(x, t + \frac{\Delta t}{2}\right) + \frac{\left(-\frac{\Delta t}{2}\right)^2}{2!} p''\left(x, t + \frac{\Delta t}{2}\right) + \frac{\left(-\frac{\Delta t}{2}\right)^3}{3!} p'''\left(x, t + \frac{\Delta t}{2}\right) + \dots + \frac{\left(-\frac{\Delta t}{2}\right)^n}{n!} p^{(n)}\left(x, t + \frac{\Delta t}{2}\right) \quad (3.18)$$

Restando ecs. 3.18 y 3.17 se tiene la aproximación en diferencias centrales:

$$\frac{\partial p}{\partial t} \Big|^{n+\frac{1}{2}} = \frac{p_i^{n+1} - p_i^n}{\Delta t} + O(\Delta t^2) \quad (3.18a)$$

Como se vio previamente, la aproximación de la primera derivada mediante diferencias centrales es más exacta que las aproximaciones mediante diferencias regresivas y progresivas. El empleo de diferencias centrales en tiempo no es común en la solución de problemas de flujo de fluidos en medios porosos. El empleo de diferencias regresivas se descarta por razones de estabilidad numérica. Como veremos posteriormente el empleo de estas aproximaciones resultará en diferentes esquemas numéricos.

Usando el caso más simple de la ecuación de flujo, i.e. Flujo lineal horizontal de una sola fase con propiedades de la roca y fluidos constantes, se procederá a escribir en diferencias finitas para los tres casos mostrados anteriormente, ec. 3.15a, 3.16a y 3.18a.

La ecuación 2.24a del capítulo II es :

$$\frac{\partial^2 p}{\partial x^2} = \left[\frac{\phi \mu c_v}{k} \frac{\partial p}{\partial t} \right] \quad (2.24a)$$

El valor de p^n es conocido al tiempo t^n en cada celda, se requiere conocer p^{n+1} al tiempo t^{n+1}

III.3. Condiciones iniciales y de frontera.

Para que un problema este planteado completamente, además de la Ecuación en Derivadas Parciales (EDP) es necesario establecer condiciones iniciales (C.I.) y de frontera (C.F.). Estas condiciones reciben este nombre porque estas pueden ser conocidas ya sea inicialmente o en las fronteras del dominio del yacimiento. Por ejemplo, para la ec. 24 una C.I. es necesaria (debido a la primer derivada con respecto al tiempo) y dos C.F. (debido a la segunda derivada con respecto a la variable x).

Condiciones iniciales.

Las condiciones iniciales especifican el estado inicial de las variables primarias (incógnitas) del sistema. Para el mismo problema de la ec. 24 las C.I. podrían ser:

$$p(x, t = 0) = p_o$$

Si el dominio del problema fuera una sección transversal (no horizontal), donde la presión inicial en cada profundidad deber de ser especificada la C.I. sería:

$$p(z, t = 0) = p_{ref} + (z - z_{ref})\gamma$$

Condiciones de frontera.

Las condiciones de frontera representan la interacción de nuestro sistema con el exterior, es decir, que condiciones en la frontera debe nuestro sistema simular y representar lo mas real posible. Consideremos nuestro ejemplo de una barra de medio poroso homogéneo.

Las condiciones de frontera pueden ser internas o externas, entre ellas existen por ejemplo:

- C.F. Interna: presión o gasto constantes
- C.F. Externa: Yacimiento infinito, no-flujo o frontera a presión constante etc.

Las condiciones iniciales pueden ser:

- C.I. : presión inicial conocida

Básicamente existen dos tipos de C.F.:

1. Condiciones Dirichlet (condiciones de presión)
2. Condiciones Neumann (condiciones de gasto)

Condiciones Dirichlet.

Cuando las condiciones de presión son especificadas, normalmente se especifican las condiciones de presión en las fronteras del dominio del problema. Por ejemplo, las condiciones Dirichlet para el problema de la ec. 24 podrían ser:

$$p(x = 0, t > 0) = p_L$$
$$p(x = L, t > 0) = p_R$$

Condiciones Neumann.

Alternativamente, se podría especificar el gasto de producción en los extremos del dominio del problema. Usando la ecuación de Darcy, las condiciones Neumann son:

$$q_L = -\frac{kA}{\mu} \left(\frac{\partial p}{\partial x} \right)_{x=0}$$

$$q_R = -\frac{kA}{\mu} \left(\frac{\partial p}{\partial x} \right)_{x=L}$$

Para el flujo en un yacimiento, la condición de gasto puede ser especificada como un gasto de producción o inyección en un pozo, en alguna posición en el yacimiento, o puede ser especificado como una frontera impermeable (cero flujo).

Aproximando con la formulación implícita la ecuación que gobierna el flujo de fluidos para un medio homogéneo en una dimensión se tiene lo siguiente.

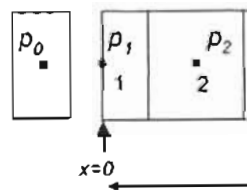
$$\frac{\partial^2 p}{\partial x^2} = \frac{\phi \mu c_i}{k} \frac{\partial p}{\partial t} \Rightarrow \left[\frac{p_{i+1}^{n+1} - 2p_i^{n+1} + p_{i-1}^{n+1}}{\Delta x^2} \right] = \left(\frac{\phi \mu c_i}{k} \right) \frac{(p_i^{n+1} - p_i^n)}{\Delta t} \quad (3.19)$$

Para este caso, las condiciones de frontera son las siguientes,

$$\begin{aligned} p(x = 0, t > 0) &= 0 \\ p(x = L, t > 0) &= 0 \end{aligned} \quad (3.20a \text{ y } 20b)$$

Por lo tanto, los nodos que intervendrán para reproducir estas condiciones de frontera serán únicamente los extremos de nuestro dominio. Es decir, los nodos 1 e IMAX.

Escribiendo la ec. 3.19 para estos nodos se tiene lo siguiente:



$$\gamma p_{i-1}^{n+1} - [2\gamma + \alpha] p_i^{n+1} + \gamma p_{i+1}^{n+1} = -\alpha p_i^n \quad (3.21)$$

donde

$$\gamma = \frac{\Delta t}{\Delta x^2} \quad \text{y} \quad \alpha = \frac{\phi \mu c_i}{k}$$

Para el nodo 1, la ec. 3.21 es la siguiente:

$$\gamma p_0^{n+1} - [2\gamma + \alpha] p_1^{n+1} + \gamma p_2^{n+1} = -\alpha p_1^n \quad (3.22)$$

Escribiendo la ec. 3.22 de la siguiente manera,

$$a p_0^{n+1} + b p_1^{n+1} + c p_2^{n+1} = d \quad (3.23)$$

En la ec. 3.23 p_0 no existe entonces $a = 0$. Para que p_1 sea siempre igual a la condición de frontera se tiene que forzar a que $b = 1$, $c = 0$, y $d = 0$ (por condición 1a). De esta manera la p_1 siempre será igual a la condición de frontera.

III.4. Conceptos de estabilidad, error y convergencia.

La exactitud de un algoritmo numérico basado en el método de diferencias finitas depende del tamaño del espaciamiento de la malla y del incremento de tiempo al cual se quiere calcular el nuevo valor de la incógnita. Si ambos parámetros son reducidos simultáneamente pero en una manera arbitraria en diferentes ordenes de magnitud y si la ecuación en diferencias finitas se aproxima con incremento en la exactitud a la ecuación en derivadas parciales, entonces el método en diferencias finitas es consistente.

Para que una formulación o esquema sea consistente, la aproximación en diferencias finitas debe ser idéntica a la EDP original a medida que la malla numérica tienda a cero. Esto es,

$$\lim_{\Delta x \rightarrow 0} (EDP - EDF) = 0$$

La estabilidad es un concepto que aplica a problemas dependientes del tiempo. Un algoritmo o esquema numérico es estable si cualquier error, introducido en alguna etapa de los cálculos, no se amplifica en cálculos subsecuentes. Supongamos que la solución de alguna ecuación en derivadas parciales, como las que surgen del flujo de fluidos en medios porosos, sujeta a condiciones iniciales y de frontera no crece si no que permanece constante o decrece en cada punto. Es razonable esperar que la solución con diferencias finitas reproduzca este comportamiento, i.e. que produzca un comportamiento libre de oscilaciones, Fig. 3.8 Si lo hace se dice que el método es estable, caso contrario el algoritmo es inestable. Una formulación o esquema es inestable cuando la combinación del error de redondeo y truncamiento domina la solución y causa resultados incorrectos.

El error de redondeo es el proveniente del número de dígitos y precisión que maneje cada computadora. El error de truncamiento proviene de la aproximación que se hace de la EDP con la serie de Taylor truncada (i.e., es la parte que no se considera en la aproximación).

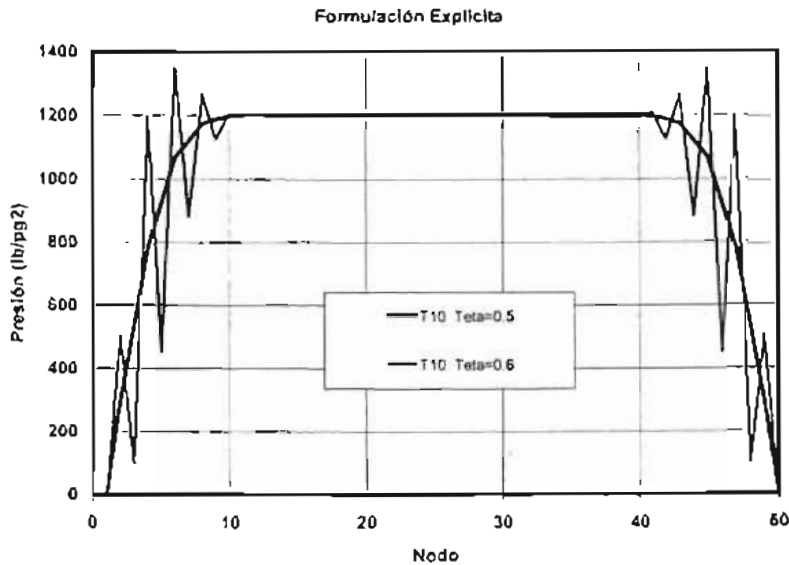


Fig. 3.8 Estabilidad de un algoritmo numérico

Con una formulación explícita la aproximación numérica de la ecuación que gobierna el flujo de fluidos para un medio homogéneo en una dimensión, se resuelve para una incógnita p^{n+1} para un nuevo tiempo $n+1$. El resto de las presiones al tiempo n son conocidas. Debido a que esta formulación tiene limitada estabilidad, raramente es usada.

Por lo tanto se tiene que el método es estable únicamente cuando $\theta \approx \frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2}$ donde

$$\alpha = \frac{k}{\phi \mu c} \text{ tal como se puede apreciar en la figura 3.8.}$$

Se debe asegurar que a medida que el tamaño de la malla y el lapso de tiempo (Δt) son reducidos, la solución numérica converja a la solución exacta. El teorema de equivalencia de Lax garantiza que si una solución numérica de una ecuación en derivadas parciales obtenida usando una aproximación consistente de diferencias finitas es estable, entonces si el espaciamiento de la malla y el incremento de tiempo tienden a cero, la solución numérica convergerá a la solución exacta (analítica). Por lo tanto, consistencia y estabilidad aseguran convergencia y viceversa. La estabilidad es una condición necesaria y suficiente para que exista la convergencia, cuando la aproximación es consistente.

III.5. Solución numérica de las ecuaciones para fluido monofásico en tres dimensiones.

A partir de la ec. 2.10 del capítulo II, se tiene que la ecuación general que describe el flujo de fluidos en medios porosos es

$$\vec{\nabla} \cdot \left(\frac{k\rho}{\mu} \left(\vec{\nabla} p - \gamma \vec{\nabla} D \right) \right) \pm \tilde{q}_m = \frac{\partial(\phi\rho)}{\partial t} \quad (2.10)$$

La ec. 2.10 para tres dimensiones y una sola fase es:

$$\frac{\partial}{\partial x} \left[\frac{k}{\mu B} \frac{\partial p}{\partial x} \right] + \frac{\partial}{\partial y} \left[\frac{k}{\mu B} \frac{\partial p}{\partial y} \right] + \frac{\partial}{\partial z} \left[\frac{k}{\mu B} \frac{\partial (p + \gamma D)}{\partial z} \right] + \tilde{q}_v = \frac{\partial}{\partial t} \left[\frac{\phi}{B} \right] \quad (3.24)$$

Siguiendo el mismo procedimiento de discretización visto en el capítulo II, la ec. 3.24, en forma diferencias finitas es la siguiente:

Para la dirección x:

$$\frac{\partial}{\partial x} \left(\frac{k}{\mu B} \frac{\partial p}{\partial x} \right) = \frac{1}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i+\frac{1}{2},j,k} \left(\frac{p_{i-1,j,k}^{n+1} - p_{i,j,k}^n}{\Delta x_{i+\frac{1}{2},j,k}} \right) - \frac{1}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i-\frac{1}{2},j,k} \left(\frac{p_{i,j,k}^{n-1} - p_{i-1,j,k}^{n+1}}{\Delta x_{i-\frac{1}{2},j,k}} \right) \quad (3.25)$$

Para la dirección y:

$$\frac{\partial}{\partial y} \left(\frac{k}{\mu B} \frac{\partial p}{\partial y} \right) = \frac{1}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j+\frac{1}{2},k} \left(\frac{p_{i,j+1,k}^{n+1} - p_{i,j,k}^{n+1}}{\Delta y_{i,j+\frac{1}{2},k}} \right) - \frac{1}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j-\frac{1}{2},k} \left(\frac{p_{i,j,k}^{n-1} - p_{i,j-1,k}^{n+1}}{\Delta y_{i,j-\frac{1}{2},k}} \right) \quad (3.26)$$

Para la dirección z:

$$\frac{\partial}{\partial z} \left(\frac{k}{\mu B} \frac{\partial (p + \gamma D)}{\partial z} \right) = \frac{1}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k+\frac{1}{2}} \left(\frac{(p_{i,j,k+1}^{n+1} + \gamma D_{i,j,k+1}) - (p_{i,j,k}^{n+1} + \gamma D_{i,j,k})}{\Delta z_{i,j,k+\frac{1}{2}}} \right) - \frac{1}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k-\frac{1}{2}} \left(\frac{(p_{i,j,k}^{n+1} + \gamma D_{i,j,k}) - (p_{i,j,k-1}^{n+1} + \gamma D_{i,j,k-1})}{\Delta z_{i,j,k-\frac{1}{2}}} \right) \quad (3.26a)$$

Para el término de acumulación:

$$\frac{\partial}{\partial t} \left(\frac{\phi}{B} \right) = \frac{\partial}{\partial t} (\phi b) = \phi \frac{\partial b}{\partial p} \frac{\partial p}{\partial t} + b \frac{\partial \phi}{\partial p} \frac{\partial p}{\partial t}$$

Entonces

$$\phi \frac{\partial b}{\partial p} \frac{\partial p}{\partial t} + b \frac{\partial \phi}{\partial p} \frac{\partial p}{\partial t} = [\phi b' + b \phi'] \frac{\partial p}{\partial t}$$

Y en forma discreta :

$$\phi b' = \phi_{i,j}^n (b_o c_f)_{i,j}$$

$$b \phi' = b_{i,j}^n (\phi_o c_r)_{i,j}$$

Por lo tanto,

$$[\phi b' + b \phi'] \frac{\partial p}{\partial t} = [\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k}] \left[\frac{p_{i,j,k}^{n+1} - p_{i,j,k}^n}{\Delta t} \right] \quad (3.27)$$

Sustituyendo las ecuaciones 3.25, 3.26, 3.26a y 3.27 en 3.24, se tiene lo siguiente:

$$\begin{aligned}
& \frac{1}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i+\frac{1}{2},j,k} \left(\frac{p_{i+1,j,k}^{n+1} - p_{i,j,k}^n}{\Delta x_{i+\frac{1}{2},j,k}} \right) - \frac{1}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i-\frac{1}{2},j,k} \left(\frac{p_{i,j,k}^{n+1} - p_{i-1,j,k}^n}{\Delta x_{i-\frac{1}{2},j,k}} \right) + \\
& \frac{1}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j+\frac{1}{2},k} \left(\frac{p_{i,j+1,k}^{n+1} - p_{i,j,k}^n}{\Delta y_{i,j+\frac{1}{2},k}} \right) - \frac{1}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j-\frac{1}{2},k} \left(\frac{p_{i,j,k}^{n+1} - p_{i,j-1,k}^n}{\Delta y_{i,j-\frac{1}{2},k}} \right) + \\
& \frac{1}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k+\frac{1}{2}} \left(\frac{(p_{i,j,k+1}^{n+1} + \gamma D_{i,j,k+1}) - (p_{i,j,k}^n + \gamma D_{i,j,k})}{\Delta z_{i,j,k+\frac{1}{2}}} \right) - \\
& \frac{1}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k-\frac{1}{2}} \left(\frac{(p_{i,j,k}^{n+1} + \gamma D_{i,j,k}) - (p_{i,j,k-1}^n + \gamma D_{i,j,k-1})}{\Delta z_{i,j,k-\frac{1}{2}}} \right) \\
\bar{q}_{i,j,k} &= \left[\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_e)_{i,j,k} \right] \left[\frac{p_{i,j,k}^{n+1} - p_{i,j,k}^n}{\Delta t} \right]
\end{aligned} \tag{3.28}$$

Multiplicando la ec. 3.28 por el volumen de roca $V_{r,i,j,k} = \Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}$,

$$\begin{aligned}
& \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i+\frac{1}{2},j,k} \left(\frac{p_{i+1,j,k}^{n+1} - p_{i,j,k}^n}{\Delta x_{i+\frac{1}{2},j,k}} \right) - \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta x_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i-\frac{1}{2},j,k} \left(\frac{p_{i,j,k}^{n+1} - p_{i-1,j,k}^n}{\Delta x_{i-\frac{1}{2},j,k}} \right) + \\
& \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j+\frac{1}{2},k} \left(\frac{p_{i,j+1,k}^{n+1} - p_{i,j,k}^n}{\Delta y_{i,j+\frac{1}{2},k}} \right) - \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta y_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j-\frac{1}{2},k} \left(\frac{p_{i,j,k}^{n+1} - p_{i,j-1,k}^n}{\Delta y_{i,j-\frac{1}{2},k}} \right) + \\
& \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k+\frac{1}{2}} \left(\frac{(p_{i,j,k+1}^{n+1} + \gamma D_{i,j,k+1}) - (p_{i,j,k}^n + \gamma D_{i,j,k})}{\Delta z_{i,j,k+\frac{1}{2}}} \right) - \\
& \frac{\Delta x_{i,j,k} \Delta y_{i,j,k} \Delta z_{i,j,k}}{\Delta z_{i,j,k}} \left(\frac{k}{\mu B} \right)_{i,j,k-\frac{1}{2}} \left(\frac{(p_{i,j,k}^{n+1} + \gamma D_{i,j,k}) - (p_{i,j,k-1}^n + \gamma D_{i,j,k-1})}{\Delta z_{i,j,k-\frac{1}{2}}} \right) + \\
q_{i,j,k} &= \left[\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_e)_{i,j,k} \right] V_{r,i,j,k} \left[\frac{p_{i,j,k}^{n+1} - p_{i,j,k}^n}{\Delta t} \right]
\end{aligned} \tag{3.29}$$

Simplificando,

$$\begin{aligned}
 & T_{i+\frac{1}{2},j,k}^{n+1} (p_{i+1,j,k} - p_{i,j,k})^{n+1} - T_{i-\frac{1}{2},j,k}^{n+1} (p_{i,j,k} - p_{i-1,j,k})^{n+1} + \\
 & T_{i,j+\frac{1}{2},k}^{n+1} (p_{i,j+1,k} - p_{i,j,k})^{n+1} - T_{i,j-\frac{1}{2},k}^{n+1} (p_{i,j,k} - p_{i,j-1,k})^{n+1} + \\
 & T_{i,j,k+\frac{1}{2}}^{n+1} \left([p_{i,j,k+1} + \gamma D_{i,j,k+1}] - [p_{i,j,k} + \gamma D_{i,j,k}] \right)^{n+1} - T_{i,j,k-\frac{1}{2}}^{n+1} \left([p_{i,j,k} + \gamma D_{i,j,k}] - [p_{i,j,k-1} + \gamma D_{i,j,k-1}] \right)^{n+1} + \\
 & + q_{i,j,k}^{n+1} = \frac{V_{r,i,j,k}}{\Delta t} \left[\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k} \right] (p_{i,j,k}^{n+1} - p_{i,j,k}^n)
 \end{aligned}$$

Donde:

$$\begin{aligned}
 T_{i+\frac{1}{2},j,k}^{n+1} &= \left(\frac{\Delta y \Delta z}{\Delta x} \right)_{i+\frac{1}{2},j,k} \left(\frac{k}{\mu B} \right)_{i+\frac{1}{2},j,k} & T_{i-\frac{1}{2},j,k}^{n+1} &= \left(\frac{\Delta y \Delta z}{\Delta x} \right)_{i-\frac{1}{2},j,k} \left(\frac{k}{\mu B} \right)_{i-\frac{1}{2},j,k} \\
 T_{i,j+\frac{1}{2},k}^{n+1} &= \left(\frac{\Delta x \Delta z}{\Delta y} \right)_{i,j+\frac{1}{2},k} \left(\frac{k}{\mu B} \right)_{i,j+\frac{1}{2},k} & T_{i,j-\frac{1}{2},k}^{n+1} &= \left(\frac{\Delta x \Delta z}{\Delta y} \right)_{i,j-\frac{1}{2},k} \left(\frac{k}{\mu B} \right)_{i,j-\frac{1}{2},k} \\
 T_{i,j,k+\frac{1}{2}}^{n+1} &= \left(\frac{\Delta x \Delta y}{\Delta z} \right)_{i,j,k+\frac{1}{2}} \left(\frac{k}{\mu B} \right)_{i,j,k+\frac{1}{2}} & T_{i,j,k-\frac{1}{2}}^{n+1} &= \left(\frac{\Delta x \Delta y}{\Delta z} \right)_{i,j,k-\frac{1}{2}} \left(\frac{k}{\mu B} \right)_{i,j,k-\frac{1}{2}}
 \end{aligned}$$

Agrupando para las incógnitas:

$$\begin{aligned}
 & T_{i,j,k-\frac{1}{2}}^{n+1} p_{i,j,k-1}^{n+1} + T_{i,j-\frac{1}{2},k}^{n+1} p_{i,j-1,k}^{n+1} + T_{i-\frac{1}{2},j,k}^{n+1} p_{i-1,j,k}^{n+1} \\
 & - \left[T_{i,j,k-\frac{1}{2}}^{n+1} + T_{i,j-\frac{1}{2},k}^{n+1} + T_{i-\frac{1}{2},j,k}^{n+1} + T_{i+\frac{1}{2},j,k}^{n+1} + T_{i,j+\frac{1}{2},k}^{n+1} + T_{i,j,k+\frac{1}{2}}^{n+1} + \frac{V_{r,i,j,k}}{\Delta t} \left\{ \phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k} \right\} \right] p_{i,j,k}^{n+1} + \\
 & T_{i+\frac{1}{2},j,k}^{n+1} p_{i+1,j,k}^{n+1} + T_{i,j+\frac{1}{2},k}^{n+1} p_{i,j+1,k}^{n+1} + T_{i,j,k+\frac{1}{2}}^{n+1} p_{i,j,k+1}^{n+1} \\
 & q_{i,j,k} = \\
 & \frac{V_{r,i,j,k}}{\Delta t} \left[\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k} \right] p_{i,j,k}^n + \\
 & T_{i,j,k+\frac{1}{2}}^{n+1} \gamma_{i,j,k+\frac{1}{2}} (D_{i,j,k+1} - D_{i,j,k}) - T_{i,j,k-\frac{1}{2}}^{n+1} \gamma_{i,j,k-\frac{1}{2}} (D_{i,j,k} - D_{i,j,k-1})
 \end{aligned} \tag{3.30}$$

La ecuación 3.30 también puede ser expresada como:

$$l_{i,j,k} p_{i,j,k-1}^{n+1} + s_{i,j,k} p_{i,j-1,k}^{n+1} + o_{i,j,k} p_{i-1,j,k}^{n+1} + c_{i,j,k} p_{i,j,k}^{n+1} + e_{i,j,k} p_{i+1,j,k}^{n+1} + n_{i,j,k} p_{i,j+1,k}^{n+1} + u_{i,j,k} p_{i,j,k+1}^{n+1} + q_{i,j,k} = d_{i,j,k}$$

Donde:

$$l_{i,j,k} = T_{i,j,k-\frac{1}{2}}^{n+1}$$

$$s_{i,j,k} = T_{i,j-\frac{1}{2},k}^{n+1}$$

$$o_{i,j,k} = T_{i-\frac{1}{2},j,k}^{n+1}$$

$$c_{i,j,k} = \left[T_{i,j,k-\frac{1}{2}}^{n+1} + T_{i,j-\frac{1}{2},k}^{n+1} + T_{i-\frac{1}{2},j,k}^{n+1} + T_{i+\frac{1}{2},j,k}^{n+1} + T_{i,j+\frac{1}{2},k}^{n+1} + T_{i,j,k+\frac{1}{2}}^{n+1} + \frac{V_{r,i,j,k}}{\Delta t} \left\{ \phi_{i,j,k}^n (b_o c_l)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k} \right\} \right]$$

$$e_{i,j,k} = T_{i+\frac{1}{2},j,k}^{n+1}$$

$$n_{i,j,k} = T_{i,j+\frac{1}{2},k}^{n+1}$$

$$u_{i,j,k} = T_{i,j,k+\frac{1}{2}}^{n+1}$$

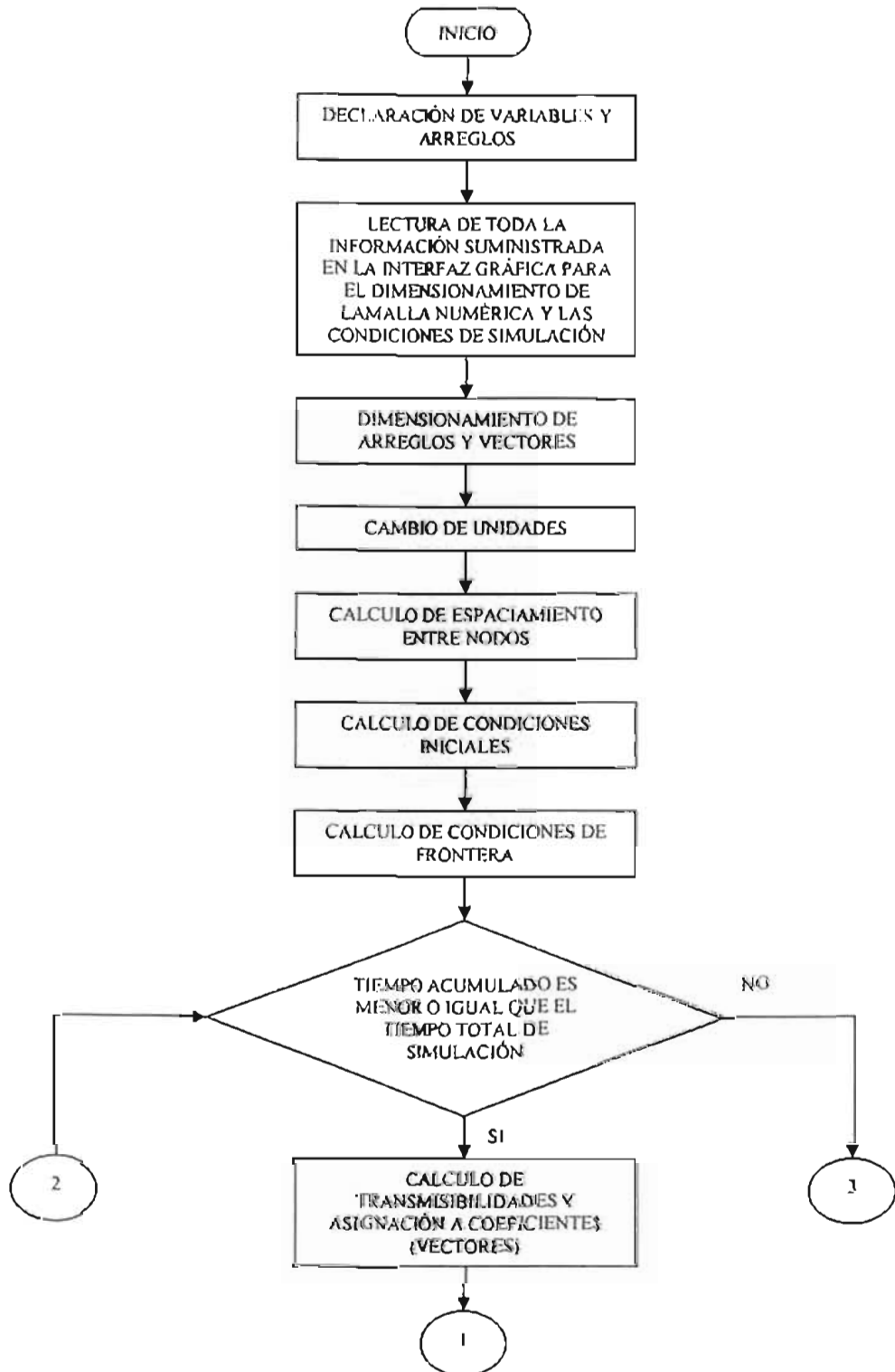
$$d_{i,j,k} = -\frac{V_{r,i,j,k}}{\Delta t} \left[\phi_{i,j,k}^n (b_o c_f)_{i,j,k} + b_{i,j,k}^n (\phi_o c_r)_{i,j,k} \right] p_{i,j,k}^n + T_{i,j,k+\frac{1}{2}}^{n+1} \gamma_{i,j,k+\frac{1}{2}} (D_{i,j,k+1} - D_{i,j,k}) - T_{i,j,k-\frac{1}{2}}^{n+1} \gamma_{i,j,k-\frac{1}{2}} (D_{i,j,k} - D_{i,j,k-1})$$

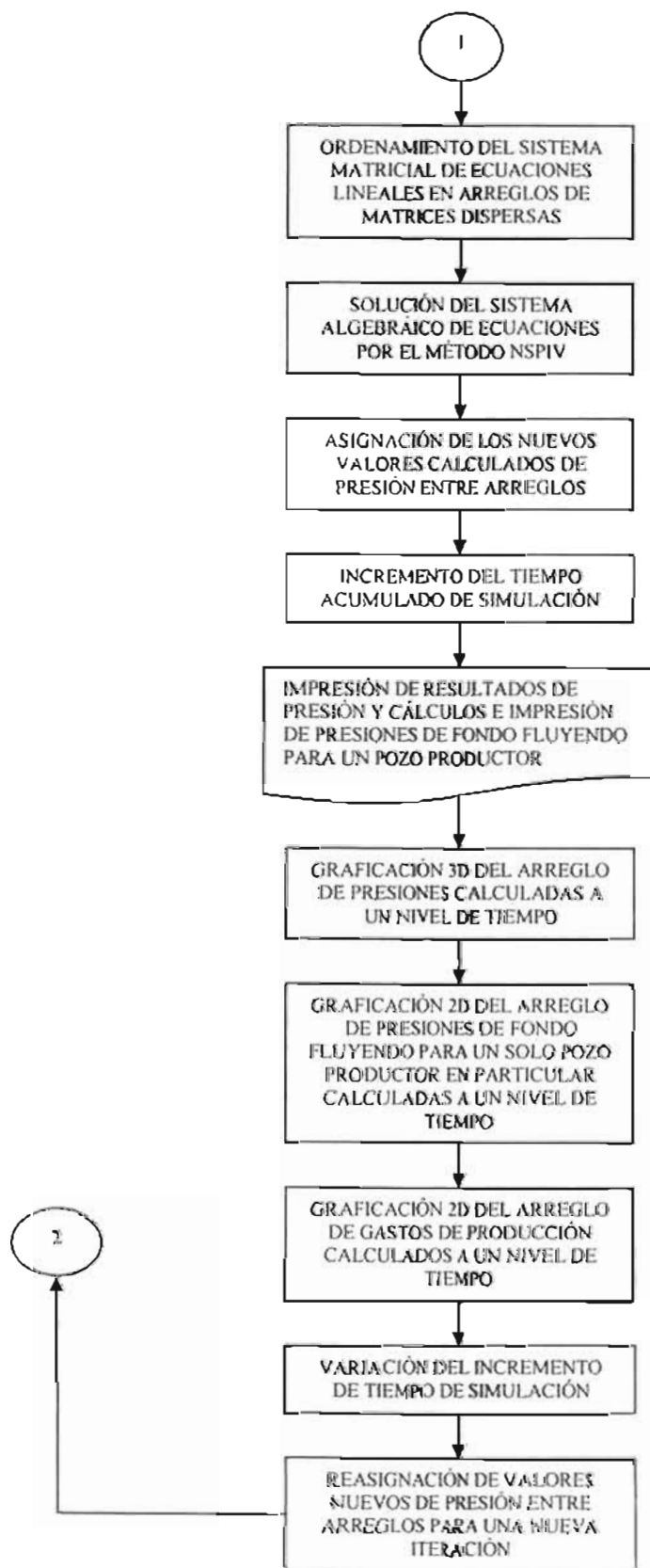
La ec. 3.30 es escrita para cada bloque o celda en que es discretizado el dominio, Fig. 3.9. Esto produce un conjunto de ecuaciones que deben resolverse simultáneamente.

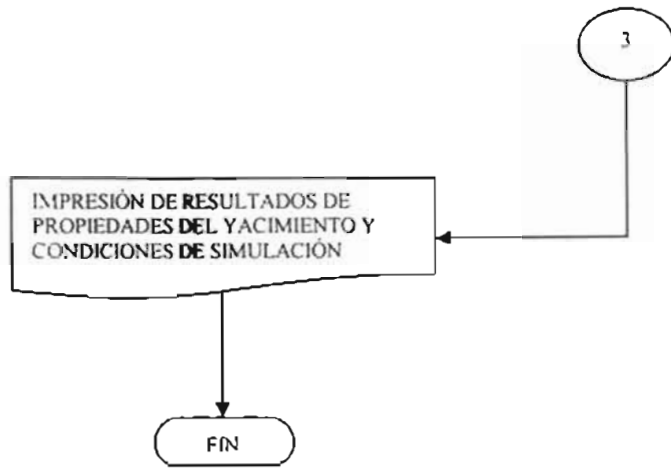
El sistema de ecuaciones en diferencias que describe el comportamiento de flujo de fluidos en el yacimiento generado a partir de la ecuación 3.30, constituye en cada iteración del simulador numérico propuesto, para cada nivel de tiempo diferente, un sistema algebraico de ecuaciones lineales que es resuelto en este trabajo empleando el método NSPIV (Sherman, A. H., 1978), especializado en la solución de sistemas de ecuaciones de matrices dispersas, que corresponde a una de las técnicas directas analizadas para resolver eficientemente sistemas matriciales pequeños o medianos debido a la relación directa, entre los requerimientos del sistema de cómputo, memoria y tiempo de ejecución, y el número de incógnitas que se resuelven simultáneamente. Entre mayor es este número, mayores son los requerimientos del sistema y la eficiencia disminuye drásticamente, por lo que en algunos casos se puede recurrir a métodos iterativos, tales como los métodos de sobre-relajación en línea LSOR, y en bloque BSOR, para resolver sistemas de ecuaciones de gran tamaño, aunque el grado de efectividad de los métodos iterativos se ve afectado por el grado de heterogeneidad y anisotropía del yacimiento, provocando problemas de convergencia (Carnahan, B. et al., 1969).

En forma general, la estructura del simulador numérico tridimensional de flujo de fluidos en medios porosos para yacimientos de aceite bajosaturado, está constituida por un programa principal acoplado a la interfaz de visualización gráfica, cuatro módulos principales o subprogramas externos con subprogramas internos y dos archivos de declaración y dimensionamiento de variables y arreglos, en donde se efectúan las diferentes etapas de cálculo requeridas por el simulador.

A continuación se presenta un diagrama de flujo simplificado de la secuencia de cálculo del simulador numérico propuesto.







III.6. Solución numérica de las ecuaciones para fluido multifásico en tres dimensiones.

III.6.1. Ecuaciones para flujo multifásico.

Las ecuaciones que gobierna el flujo multifásico en medios porosos se muestra a continuación. Su obtención es similar al procedimiento utilizado para una sola fase, capítulo II.

Fase Aceite,

$$\frac{\partial}{\partial x} \left[\frac{kk_{ro}}{\mu_o B_o} \left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial D}{\partial x} \right) \right] + \frac{\partial}{\partial y} \left[\frac{kk_{ro}}{\mu_o B} \left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial D}{\partial y} \right) \right] + \frac{\partial}{\partial z} \left[\frac{kk_{ro}}{\mu_o B} \left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial D}{\partial z} \right) \right] = \frac{\partial}{\partial t} \left(\frac{\phi S_o}{B_o} \right) \quad (3.31)$$

Fase Gas,

$$\begin{aligned} & \frac{\partial}{\partial x} \left[\frac{kk_{rg}}{\mu_g B_g} \left(\frac{\partial p_g}{\partial x} - \gamma_g \frac{\partial D}{\partial x} \right) \right] + \frac{\partial}{\partial x} \left[\frac{kk_{ro}}{\mu_o B_o} R_s \left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial D}{\partial x} \right) \right] + \\ & \frac{\partial}{\partial y} \left[\frac{kk_{rg}}{\mu_g B_g} \left(\frac{\partial p_g}{\partial y} - \gamma_g \frac{\partial D}{\partial y} \right) \right] + \frac{\partial}{\partial y} \left[\frac{kk_{ro}}{\mu_o B_o} R_s \left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial D}{\partial y} \right) \right] + \\ & \frac{\partial}{\partial z} \left[\frac{kk_{rg}}{\mu_g B_g} \left(\frac{\partial p_g}{\partial z} - \gamma_g \frac{\partial D}{\partial z} \right) \right] + \frac{\partial}{\partial z} \left[\frac{kk_{ro}}{\mu_o B_o} R_s \left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial D}{\partial z} \right) \right] + \\ & = \frac{\partial}{\partial t} \left(\frac{\phi S_g}{B_g} + R_s \frac{\phi S_o}{B_o} \right) \end{aligned} \quad (3.32)$$

Fase Agua,

$$\frac{\partial}{\partial x} \left[\frac{kk_{rw}}{\mu_w B_w} \left(\frac{\partial p_w}{\partial x} - \gamma_w \frac{\partial D}{\partial x} \right) \right] + \frac{\partial}{\partial y} \left[\frac{kk_{rw}}{\mu_w B_w} \left(\frac{\partial p_w}{\partial y} - \gamma_w \frac{\partial D}{\partial y} \right) \right] + \frac{\partial}{\partial z} \left[\frac{kk_{rw}}{\mu_w B_w} \left(\frac{\partial p_w}{\partial z} - \gamma_w \frac{\partial D}{\partial z} \right) \right] = \frac{\partial}{\partial t} \left(\frac{\phi S_w}{B_w} \right) \quad (3.33)$$

Con las siguientes restricciones:

$$S_o + S_g + S_w = 1 \quad (3.34)$$

$$P_{C_{go}} = p_g - p_o \quad (3.35)$$

y

$$P_{C_{wo}} = p_o - p_w \quad (3.36)$$

Las condiciones iniciales son las siguientes:

$$p_p(x, y, z, 0) = p_o ; \quad (3.37)$$

$$S_p(x, y, z, 0) = S_{p,0}; \quad (3.38)$$

donde $p = o, g, w$

Las condiciones de frontera se expresan, en $x=0$, and $x=L$, como cero flujo para todos los fluidos, $p=o, g, w$:

$$\left(\frac{\partial p_p}{\partial x} - \gamma_p \frac{\partial D}{\partial x} \right)_{x=0, x=L} = 0; \quad t > 0 \quad (3.39)$$

Expresiones idénticas son expresadas para la dirección y y z .

III.6.2. Aproximación mediante diferencias finitas.

La aproximación en diferencias finitas de las Ec. 3.31 a 3.33 en el nodo ijk , mediante un esquema implícito se expresa, a través de operadores en diferencias, como:

Fase Aceite,

$$\Delta \left[T_o (\Delta p_o - \gamma_o \Delta D) \right]_{i,j,k}^{n+1} + q_{n,i,j,k} = \frac{Vr_{i,j,k}}{\Delta t} \Delta_i \left(\frac{\phi(1 - S_g - S_w)}{B_o} \right)_{i,j,k} \quad (3.40)$$

Fase Gas,

$$\Delta \left[T_g (\Delta p_g + \Delta P C_{g^o} - \gamma_g \Delta D) \right]_{i,j,k}^{n+1} + \Delta \left[T_o R_i (\Delta p_o - \gamma_o \Delta D) \right]_{i,j,k}^{n+1} + q_{g,i,j,k}^{n+1} = \frac{Vr_{i,j,k}}{\Delta t} \Delta_i \left(\frac{\phi S_g}{B_g} + \frac{\phi R_i (1 - S_g - S_w)}{B_w} \right)_{i,j,k} \quad (3.41)$$

Fase Agua,

$$\Delta \left[T_w (\Delta p_w - \Delta P C_{w^o} - \gamma_w \Delta D) \right]_{i,j,k}^{n+1} = \frac{Vr_{i,j,k}}{\Delta t} \Delta_i \left(\frac{\phi S_w}{B_w} \right)_{i,j,k} \quad (3.42)$$

donde

$$i = 1, 2, \dots, I \quad j = 1, 2, \dots, J \quad k = 1, 2, \dots, K \quad n = 0, 1, 2, \dots$$

Donde relaciones adicionales, 3.40 a 3.42 fueron acopladas en las ecuaciones de flujo en diferencias. Con la sustitución anterior se han eliminado 3 incógnitas, i.e. (p_g, p_w, S_o) . El sistema a resolver será de tres ecuaciones con tres incógnitas, i.e. $(p_o, S_g, S_w)_{ijk}$, $i=1, 2, \dots, I$; $j=1, 1, 2, \dots, J$; and $k=1, 2, \dots, K$

III.6.3. Planteamiento de la solución.

Debido a que el sistema de ecuaciones 3.39 a 3.41 es altamente no lineal, se tiene que recurrir a un método iterativo para linealizarlo, por ejemplo, el método de Newton. Posteriormente el sistema resultante se puede resolver con cualquier algoritmo que resuelva sistema de ecuaciones lineales dispersos. (Sherman, A. H., 1978).

El sistema a resolverse tendrá la siguiente estructura,

$$l_{i,j,k} x_{i,j,k}^{n+1} + s_{i,j,k} x_{i,j,k}^{n+1} + o_{i,j,k} x_{i-1,j,k}^{n+1} + c_{i,j,k} x_{i,j,k}^{n+1} + e_{i,j,k} x_{i+1,j,k}^{n+1} + n_{i,j,k} x_{i,j,k+1}^{n+1} + u_{i,j,k} x_{i,j,k+1}^{n+1} + q_{i,j,k}^{n+1} = d_{i,j,k} \quad (3.43)$$

donde l, s, o, c, e, n, u son matrices de 3×3 y contienen las derivadas de las ecuaciones 3.40 a 3.42 con respecto a todas las incógnitas $x^T = (p_o, S_g, S_w)_{jk}$. Por ejemplo, considerando una sola dirección, x , el sistema es el siguiente:

$$\begin{bmatrix} O_{oo} & O_{og} & O_{ow} \\ O_{go} & O_{gg} & O_{gw} \\ O_{wo} & O_{wg} & O_{ww} \end{bmatrix}^{(v)} \begin{bmatrix} C_{oo} & C_{og} & C_{ow} \\ C_{go} & C_{gg} & C_{gw} \\ C_{wo} & C_{wg} & C_{ww} \end{bmatrix}^{(v)} \begin{bmatrix} e_{oo} & e_{og} & e_{ow} \\ e_{go} & e_{gg} & e_{gw} \\ e_{wo} & e_{wg} & e_{ww} \end{bmatrix}^{(v)} \begin{bmatrix} \delta p_o \\ \delta S_g \\ \delta S_w \end{bmatrix}_{i-1}^{(v+1)} = - \begin{bmatrix} R_o \\ R_g \\ R_w \end{bmatrix}_i^{(v)} \quad (3.44)$$

Los elementos se definen de la siguiente manera:

$$c_{p_o, i} = \frac{\partial R_{p, i}}{\partial p_{o, i}}$$

$$c_{pg,i} = \frac{\partial R_{p,i}}{\partial S_{g,i}}$$

y,

$$c_{pw,i} = \frac{\partial R_{p,i}}{\partial S_{w,i}}$$

donde. $p=o,g,w$

$\sigma_{pp,i}$ y $e_{pp,i}$ se definen similarmente. Fácilmente el sistema (3.44) es extendido a tres dimensiones para su solución.

III.7. Modelado de pozos.

El problema de representar pozos en simulación numérica es fundamentalmente cuestión de resolución y escala, (Peaceman, D. W., 1977). Debido a que los bloques son dos o tres ordenes de magnitud más grande que el diámetro del pozo, los pozos son raramente modelados explícitamente. Por lo tanto, la presión del bloque es diferente a la presión del pozo. Una relación entre la presión del bloque y la presión del pozo puede ser introducida al simulador mediante el uso del Índice de productividad, WI. Este índice captura la relación entre la presión en el pozo y el yacimiento. Específicamente, para cada bloque (i,j,k) interceptado por cada pozo, la magnitud de WI es requerida para relacionar el gasto del pozo ya sea de inyección o producción con la presión del bloque, $p_{b,ijk}$ y la presión del pozo, $p_{wf,ijk}$:

$$q_{w,ijk} = \frac{WI}{\mu} (p_{b,ijk} - p_{wf,ijk}) \quad (3.45)$$

Donde μ es la viscosidad del fluido. Esta definición es válida para una sola fase. Para flujo multifásico, la ec. 1 se generaliza como:

$$q_{p,ijk} = WI \frac{k_{r,p}}{\mu_p} (p_{b,p,ijk} - p_{wf,ijk}) \quad (3.46)$$

Donde $p = o, g, w$ y $k_{r,p}$ es la permeabilidad relativa a la fase p .

III.7.1. Modelo de Peaceman.

En este modelo se realizaron soluciones para presión considerando varios tamaños de celdas, (Peaceman, D. W., 1977). Los resultados muestran que un radio efectivo del bloque donde la presión calculada numéricamente iguala a la presión fluyendo del pozo es:

Para un yacimiento isótropo:

$$r_o = 0.14(\Delta x^2 + \Delta y^2) \quad (3.47)$$

Para un yacimiento Anisótropo:

$$r_o = 0.28 \frac{\left[\left(\frac{k_y}{k_x} \right)^{\frac{1}{2}} \Delta x^2 + \left(\frac{k_x}{k_y} \right)^{\frac{1}{2}} \Delta y^2 \right]^{\frac{1}{2}}}{\left(\frac{k_y}{k_x} \right)^{\frac{1}{4}} + \left(\frac{k_x}{k_y} \right)^{\frac{1}{4}}} \text{ para } 0.5 \leq \frac{\Delta x}{\Delta y} \leq 2 \quad (3.48)$$

Para un pozo drenando verticalmente el Índice de productividad es:

$$WI = \left[\frac{2\pi \sqrt{k_x k_y} \Delta z}{\ln \left(\frac{r_o}{r_w} \right)} \right] \quad (3.49)$$

Por lo tanto, considerando la ec. 3.47, 3.48, y 3.49 la presión de fondo fluyendo es:

$$p_{wf} = p_b - \frac{q\mu}{WI} \quad (3.50)$$

Expresándolo para el bloque ijk :

$$p_{wf,ijk} = p_{b,ijk} - \frac{q_{ijk} \mu_{ijk}}{WI_{ijk}} \quad (3.51)$$

CAPITULO IV. INTERFAZ DE VISUALIZACIÓN.

IV.1. Descripción de la Interfaz Winteracter.

La interfaz de desarrollo *Winteracter* es una subrutina o serie de subrutinas desarrolladas sobre una plataforma de ambiente gráfico que proporciona una extensa selección de herramientas de desarrollo visual en compatibilidad con el lenguaje de programación Fortran 90/95, (Interactive Software Services Ltd. 1997-2004).

En el desarrollo de un simulador numérico de yacimientos basado en la solución de un sistema matricial de ecuaciones, una interfaz de visualización provee las herramientas necesarias para simplificar el diseño y el desarrollo de los procesos de graficación y entrada/salida de datos.

La interfaz desarrollada en este trabajo esta basada en el lenguaje de programación Fortran 90, (Chapman, S. J., 1998). En esta se incluye el modulo de visualización WINTERACTER que permite acoplar nuestro modelo numérico de simulación a las subrutinas de graficación y de entrada/salida de datos.

```
PROGRAM SIMBOIL
USE WINTERACTER
USE VARIABLES
USE IDENTIFIER_MOD
USE constants
USE my_library
USE input_data
IMPLICIT NONE
TYPE(WIN_MESSAGE)::MESS
INTEGER::ITYPE,I_status
INTEGER,DIMENSION(2)::ISUB
INTEGER,DIMENSION(3)::IHANDLE
CALL WinInitiat()
CALL WindowOpen(FLAGS=SysMenuOn+MinButton+FixedSizeWin,MENUID=IDM_MENU1, &
                TITLE='Simulador de aceite negro',DIALOGID=DIALOG001)
CALL WDialogLoad(DIALOG002)
CALL WDialogLoad(AYUDA)
```

WiDE: El Ambiente de Desarrollo *Winteracter*

El Ambiente de Desarrollo *Winteracter*, WiDE, por sus siglas en inglés, es una herramienta independiente de compilación el cuál permite crear editar y construir, programas de tipo *Winteracter*. WiDE permite crear archivos múltiples que pueden ser administrados dentro de un mismo proyecto. También provee acceso a otras herramientas de visualización, documentación y herramientas de aplicación. Las opciones específicas de este compilador y sus librerías pueden ser definidas, (Interactive Software Services Ltd. 1997-2004).

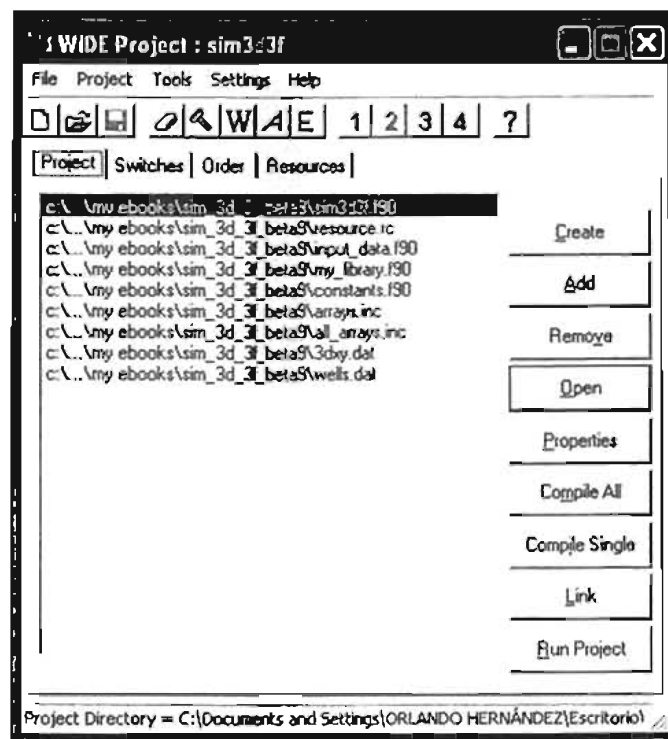


Fig. 4.1 El Ambiente de Desarrollo *Winteracter*, WiDE.

Proyectos

WiDE agrupa archivos dentro de proyectos. Usualmente un proyecto puede consistir de varios archivos fuente en lenguaje de programación fortran y un archivo de tipo resource propio del programa. Sin embargo, un proyecto puede consistir de varios tipos de archivos, no necesariamente en código fuente. Por lo tanto, la documentación, bitmaps, etc. Pueden ser administrados vía WiDE. Mientras esto podría ser normalmente más conveniente para organizar todos los archivos del proyecto bajo el mismo directorio, *Winteracter* permite añadir nuevos archivos al proyecto a partir de múltiples directorios. Todos los archivos del proyecto actual deberán ser listados en el directorio principal del proyecto. Esto es mostrado automáticamente en una ventana cuando un proyecto es creado por primera vez o cuando se ejecuta un proyecto ya creado.

La herramienta *Wizard*

Un nuevo proyecto puede ser creado por medio de la herramienta “*Wizard*” que permite crear nuevas aplicaciones mediante una serie de 5 diálogos que muestra simples preguntas acerca del tipo de aplicación que se requiera. Los proyectos generados mediante esta herramienta pueden ser inmediatamente compilados, ligados y ejecutados.


La herramienta *Skeleton*

Un proyecto desarrollado en la herramienta *Skeleton* consiste de un conjunto básico de archivos de aplicación *Winteracter*. Estos proveen un programa *Winteracter* con mínima compilación para usuarios más experimentados. Mediante un cuadro de dialogo se asigna el nombre del proyecto y el directorio de trabajo. Este tipo de proyectos constan de tres archivos principales, que son: un archivo fuente en lenguaje básico de programación Fortran 90, un módulo Fortran y un archivo de tipo resource script.

La herramienta *Bare*

En un proyecto "*Bare*" no se crea ningún archivo de proyecto con excepción del archivo básico de descripción del proyecto WiDE. Los archivos de aplicación existentes pueden ser añadidos al proyecto según sean requeridos. La opción *Bare* es principalmente usada cuando se requiere de alguna aplicación que ha sido creada fuera de WiDE. Un cuadro de dialogo permite asignar el nombre y el directorio de trabajo al proyecto. Posteriormente se tiene entonces la posibilidad de añadir archivos nuevos al proyecto, existentes en el directorio seleccionado (algunos archivos con determinadas extensiones pueden ser ignorados).

Construcción de un Proyecto

Existen varios caminos para construir un proyecto. El camino fácil es simplemente presionar el botón  de la barra de herramientas en la ventana de dialogo WiDE. Esto tiene el mismo efecto que seleccionar la instrucción Project→Build. Al realizar esto se podría compilar cada archivo del proyecto, seleccionando el compilador apropiado. Si la compilación resulta exitosa, el proyecto podrá ser enlazado automáticamente. Sin embargo, si ocurrieran algunos errores durante la compilación o el enlace, este sería reportado en una ventana de dialogo WiDE.

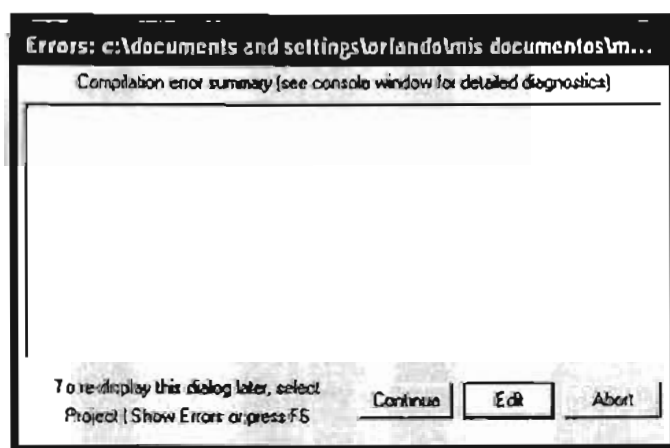



Fig. 4.2 Consola de Errores, WiDE.

Alternativamente, el archivo actual seleccionado, puede ser compilado individualmente por medio del botón de compilación *compile button* de la ventana de diálogo WiDE. Esto puede ser especialmente empleado cuando se trabaja en algún archivo particular de un proyecto con múltiples archivos fuente. Si por el contrario se han seleccionado varios archivos en la lista del proyecto, el botón de compilación también puede ser usado para compilar cada archivo por separado. En cada situación, cuando la compilación se haya completado, se deberá presionar el botón de enlace *link button* para crear un archivo ejecutable. De la misma manera el botón de compilación completa *Compile All button* seguido del botón de enlace, lograría el mismo efecto que la opción que proporciona la instrucción *Build* al presionar el botón  de la barra de herramientas o al presionar la tecla de función F4.

Cuando se desarrolla un proyecto, el cuál contiene múltiples módulos en Fortran. algunos de los cuales emplean módulos en otros archivos fuente en el mismo proyecto, el orden de compilación se vuelve muy importante. Cuando son escogidas las opciones de compilación *compile button*, de compilación completa *Compile All button* o de creación completa del proyecto *Build*, WiDE compila los archivos del proyecto en el orden que aparecen en la lista del proyecto de la ventana de diálogo WiDE. Para reordenar la lista de archivos del proyecto, se deberá seleccionar la instrucción *Order tab* en la ventana de dialogo principal WiDE.

Como una alternativa a las opciones de compilación completa y creación completa del proyecto, WiDE proporciona la posibilidad de ejecutar ambas instrucciones mediante el empleo de la opción *Make* del menú de la barra de herramientas en la ventana de dialogo WiDE. Esto optimiza la recompilación selectiva de archivos fuente modificados y reduce el tiempo de compilación cuando se trabaja con múltiples archivos fuente del proyecto. Sin importar el método elegido para desarrollar un proyecto, una vez compilado y enlazado, puede ser ejecutado por medio del botón de ejecución *Run Project button* de la barra de herramientas en la ventana de dialogo WiDE.

Herramientas

WiDE integra un número de herramientas de desarrollo. Algunas de estas son construidas dentro del propio ambiente WiDE y algunas son ejecutadas como procesos externos.

Editor de Texto

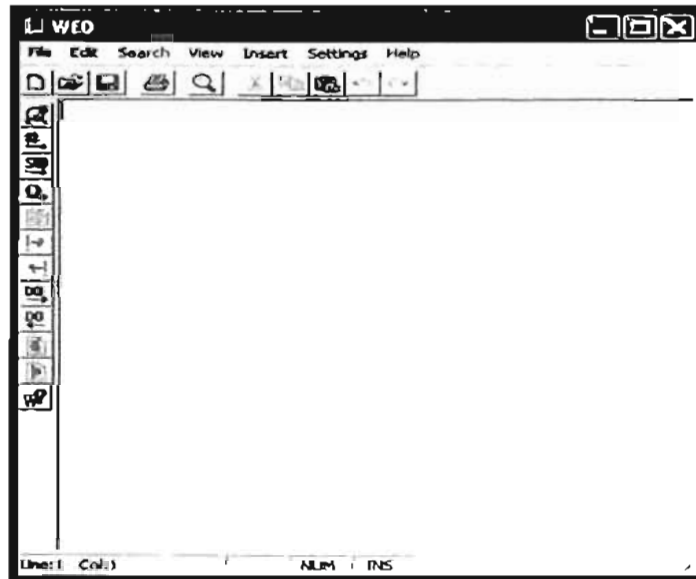


Fig. 4.3 Editor de Texto *Winteracter*, WED.

Existen varias opciones *WiDE* que permiten cargar algunos tipos de archivo del proyecto en un editor de texto. De manera predeterminada *WiDE* provee una versión de edición de texto WED por sus siglas en inglés. Este es un editor de texto multi-almacenamiento, el cuál proporciona muchas características de productividad tanto en lenguaje Fortran como en el propio de *Winteracter*. Notablemente, la opción de inserción de este editor puede ser empleada para añadir rápidamente llamadas a librerías *Winteracter*, identificadores de tipo *resource* propios de *Winteracter*, mensajes de procesamiento, etc. a cualquier código fuente en lenguaje Fortran. Alternativamente al editor de texto WED, se puede seleccionar otro editor por medio de la instrucción `Settings→Text Editor`. Aunque este editor provee una integración ligeramente inferior con *WiDE*, proporciona la comodidad de un editor de mayor familiaridad para quienes no están preparados a entender el editor WED.

Editor Resource

Los componentes de la interfaz de usuario (diálogos, menús, barras de herramientas e imágenes) son creados y mantenidos por medio del editor de archivos resource *ResEdit* de *Winteracter*. Este editor permite la selección de componentes a partir del menú principal del cuadro de diálogo *ResEdit* que despliega una lista de elementos que el usuario puede elegir dentro de sus proyectos para desarrollar alguna determinada aplicación.

Make

WiDE provee dos alternativas a las opciones de compilación completa y creación completa del proyecto, cada una de las cuales pueden ser usadas para crear un proyecto con la mínima recompilación.

- **WiDE make:** Es la herramienta interna de creación de proyectos, propia de *Winteracter*.
- **Automake:** Es un programa externo desarrollado por Polyhedron Software para el desarrollo de proyectos.

Un proyecto puede ser creado, por medio de las herramientas Make (de WiDE) o Make (de Automake) del menú principal del cuadro de diálogo WiDE. Estas herramientas Make proporcionan una alternativa opcional a los procesos de Compilar (*Compile*), Enlazar (*Link*) y Crear (*Build*) archivos de proyecto a partir del menú principal del cuadro de diálogo WiDE.

Ambas opciones Make son accesibles por medio de los botones **W** y **A** de la barra de herramientas del menú principal del cuadro de diálogo WiDE.

Ajustes

WiDE proporciona numerosas propiedades para el ajuste de configuración de las características de usuario para desarrollar un proyecto. Estas pueden ser modificadas a través de las opciones de ajuste del menú WiDE y por medio de las instrucciones *Switches tab* del cuadro de diálogo WiDE.

IV.2. Estructura de la interfaz desarrollada.

La interfaz consta de dos partes, una parte de diseño y otra de ejecución, la parte de diseño se refiere al ambiente sobre del cual se va a trabajar, esto es, a la programación de los eventos a los que va a responder el programa, en la parte de diseño se programan los eventos que rigen el comportamiento de la interfaz y estos se refieren a todas acciones que se realizan al correr el programa como oprimir un botón, mover el mouse, ocultar o dimensionar una ventana, oprimir un botón del mouse, etc., además se diseña el ambiente de acuerdo a nuestras necesidades tal que nuestro proyecto sea lo mas amigable posible.

Por otro lado se desarrolla el programa en cuanto a la ejecución, incluyendo las subrutinas de los procesos relacionados a los cálculos que realiza el modelo numérico del simulador incluyendo la lectura de datos, la ejecución donde se procesan estos datos y la salida de resultados los cuales serán mostrados por la misma interfaz, ambas partes (tanto diseño como ejecución) están diseñadas para trabajar en conjunto.

IV.3. Descripción de subrutinas desarrolladas.

El programa está estructurado en módulos, posee un programa principal (simboil.f90) y maneja 5 módulos, así como un archivo resource.rc en el que se guarda la información relacionada al ambiente. Se tiene un programa principal desde el cual se realiza la programación en cuanto a diseño del ambiente de trabajo y está formado también con subrutinas útiles para la interfaz de visualización. Se tiene un módulo para la declaración de variables que serán empleadas tanto en la parte de diseño como en la ejecución (identifier_mod). En el programa principal se emplea el módulo que contiene las variables y también se hace uso de los módulos input data, constants y my_library así como el módulo de resource que contiene la declaración como variables de todos los objetos empleados en la interfaz (este se crea automáticamente por Winteracter), se inicia el programa abriendo una ventana principal o raíz donde se encuentran todos los datos que se requieren como información para el simulador y en el cual se leen todas las variables.

El programa lee primero los datos del yacimiento se dimensionan los vectores de los pozos y posteriormente lee los datos relacionados a los pozos, para posteriormente abrir una nueva ventana para desplegar la visualización de los resultados en tiempo real. La declaración de las variables se encuentra en el módulo de identifier_mod y ahí se declaran con el archivo arrays.inc las variables del programa y dentro de la subrutina surfplot se dimensionan los arreglos.

IV.3.1. Subrutina de menú de entrada.

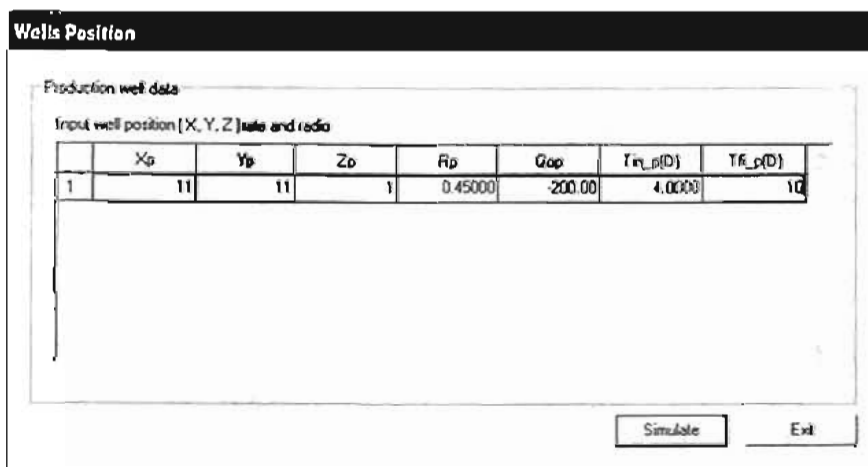
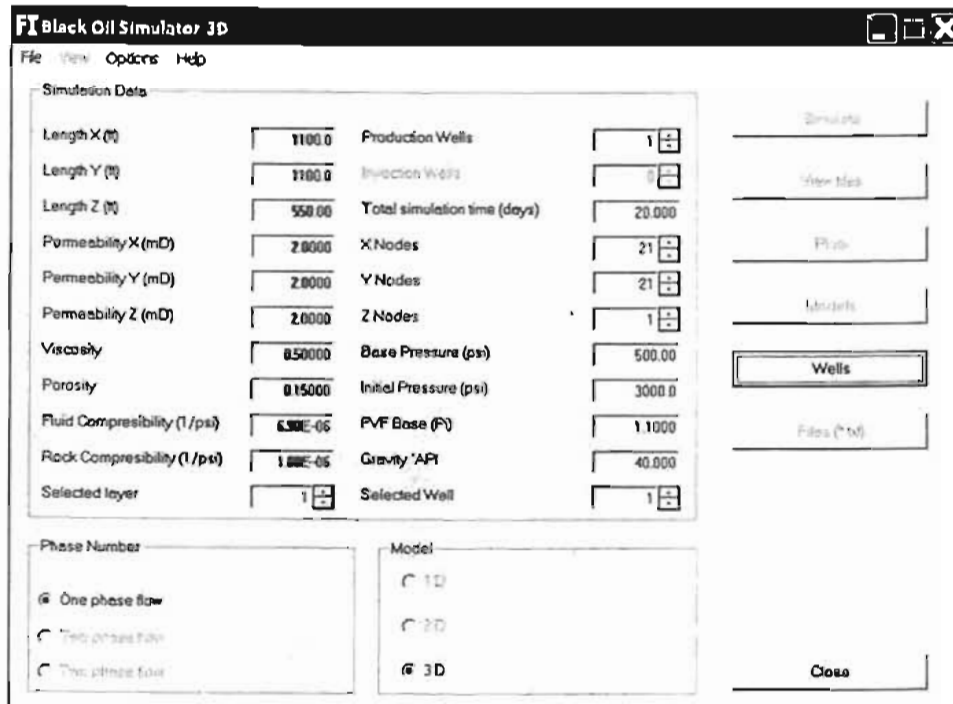


Fig. 4.4 Menú de Entrada de la subrutina de visualización..

La interfaz gráfica desarrollada para un modelo numérico de simulación de yacimientos en una sola fase, en coordenadas cartesianas y en tres dimensiones, se apoya de un modulo o subrutina de control del Menú Principal de Entrada de Datos, que contiene las instrucciones básicas para manipular y controlar la entrada y salida de datos, con base en sencillas líneas de código denominadas identificadores, que pueden ser empleadas en cualquier parte de la interfaz de acuerdo a las necesidades del programa o del proyecto que se esté llevando a cabo.

```

CALL WInitialise()
CALL WindowOpen(FLAGS=SysMenuOn+MinButton+FixedSizeWin,MENUID=IDM_MENU1, &
                TITLE='Simulador de aceite negro',DIALOGID=DIALOG001)
CALL WDialogLoad(DIALOG001)
CALL WDialogLoad(AYUDA)
DO
CALL WMessage(ITYPE,MESS)
SELECT CASE(ITYPE)
    CASE(MenuSelect)
        SELECT CASE(MESS%VALUE1)
            CASE(MENNEW)
                CALL LIMPIAR()
            CASE(MENOPEN)
                CALL LIMPIAR()
            CASE(MENSAVE)
                CALL WMessageBox(3,3,1,'SAVE','GRAFICA')
            CASE(MENPRINT)
                CALL WMessageBox(3,3,1,'PRINT','GRAFICA')
            CASE(MENSALIR)
                EXIT
            CASE(MENSIM)
                CALL WMessageBox(3,3,1,'SIMULACION','GRAFICA')
            CASE(MENHELP)
                CALL WDialogSelect(AYUDA)
                CALL WDialogShow(ITYPE=Modal)
                CALL WDialogSelect(DIALOG001)
        END SELECT
    CASE (PushButton)
        SELECT CASE (MESS%VALUE1)
            CASE (IDCLOSE)
                EXIT
        CASE (BUTTON05)
            CALL WDialogSelect(DIALOG001)
            CALL WDialogGetInteger(ENTERO001,wpro)
            CALL WDialogGetInteger(ENTERO002,winj)
            CALL WDialogGetInteger(ENTERO003,lux)

```

La subrutina completa se muestra en el apéndice B.

IV.3.2. Subrutina de visualización.

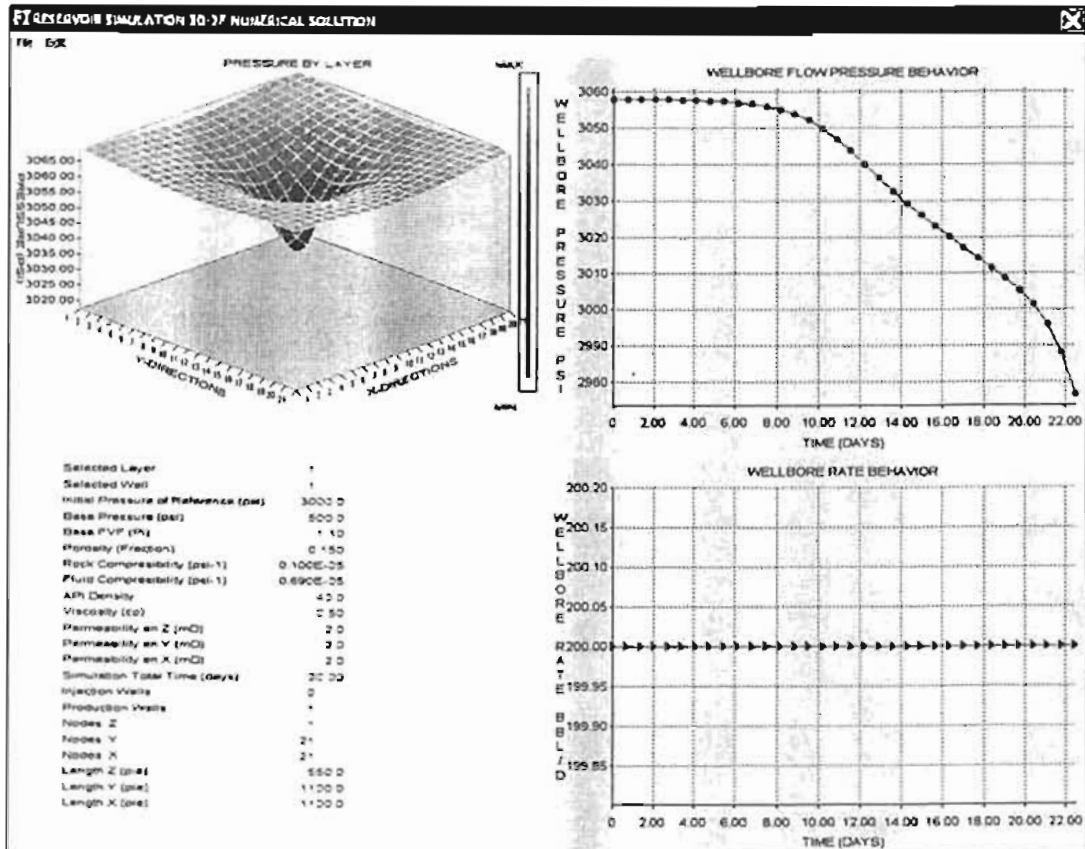


Fig. 4.5 Consola de Visualización Gráfica.

Una segunda parte de la interfaz esta compuesta por una subrutina de visualización o de control de gráficos (colores, escalas, diseños de gráficos, etc.) que esta conformada por una serie de comandos principales encargados de las operaciones más sencillas de graficación (llamadas a procesos) durante la ejecución de la interfaz.

```

!
! Open main (root) window, with:
! - Status bar non_enabled
! - Width & height 90% of full screen
! - Default position (i.e. centered)
! - menu
!
CALL WindowOpenChild(EDITWIN,FLAGS =SysMenuOn+FixedSizeWin+StatusBar+OwnedByParent, &
    WIDTH =INT(WInfoScreen(ScreenWidth)*0.90), &
    HEIGHT=INT(WInfoScreen(ScreenHeight)*0.90), &

```

```
MENUID=IDR_MENU1,TITLE="RESERVOIR SIMULATION 3D-3F NUMERIC SOLUTION")
CALL WindowOutStatusBar(1,"Click and drag the separator between sub-windows to resize")
CALL WDialogLoad(IDD_OPTIONS)
CALL WDialogLoad(IDD_TIME)
!
! Split vertically into sub-windows. (50/50 split)
! Split second sub-window horizontally into two further sub-windows. (Also 50/50 split)
! Use proportional resizing of sub-windows when main window is resized.
! Suppress Titles on sub-windows.
```

La subrutina completa se muestra en el apéndice B.

IV.3.3. Subrutina de graficación en 3D.

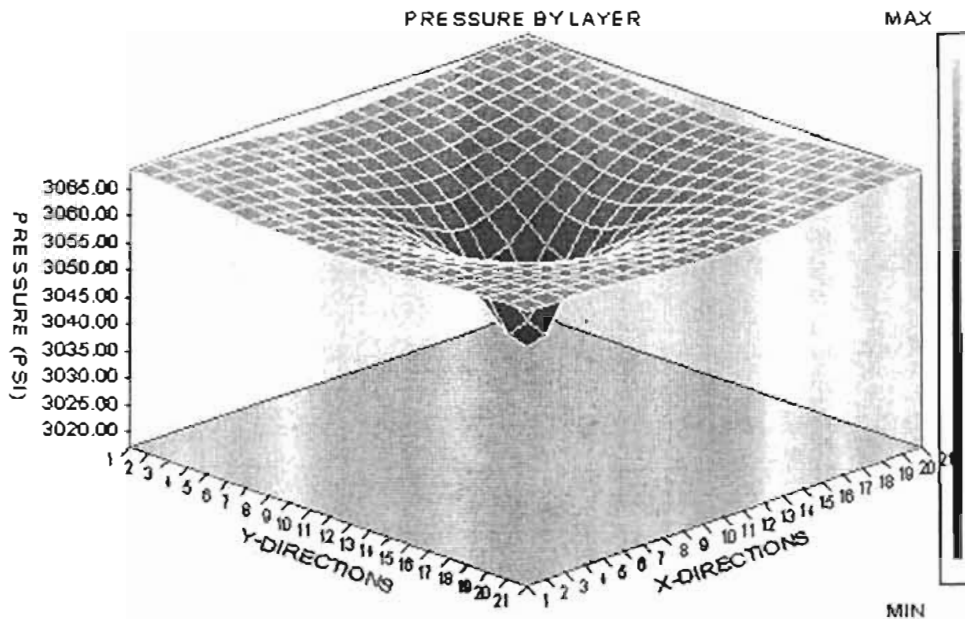


Fig. 4.6 Gráfica 3D.

Una tercera parte de la interfaz está conformada por las subrutinas de graficación de datos en dos y en tres dimensiones respectivamente que son llamadas a proceso por la subrutina de control de gráficos. Esta subrutina permite la visualización de las gráficas en tres dimensiones a partir de los datos generados por el modelo numérico de simulación, y con base en los datos obtenidos por la subrutina de control de entrada de datos.

```
! Request a 3D surface plot with height-dependent colours.  
! Here we plot a 50x50 grid stored in a 100x100 array to illustrate  
! that the whole of the array need not necessarily be used.  
:  
NGRIDX = nx  
NGRIDY = ny  
NVAL(1) = NGRIDX  
NVAL(2) = NGRIDY  
CALL IP2NewPlot(PgZarrayPlot,NSET,NVAL,LAYOUT)  
!  
! Convert the (x,y,z) triplets to a z(m,n) array  
! The 'search box' size is limited to a rectangle which
```

```

! is one tenth the width/height of the overall area to be
! plotted. Increasing this vale may give better interpolation.
! but at the expense of increased CPU time.
!
!
BOXSIZEX = (XMAX-XMIN)/10.0
BOXSIZY = (YMAX-YMIN)/10.0
CALL IPgXYZSearchBox(BOXSIZEX,BOXSIZY)
CALL IPgXYZToGrid(NIPT,YIPT,ZIPT,NPTS,ZDATA,NGRIDX,NGRIDY,NXDIM,NYDIM)
!
! Now plot the surface
!
! Define PG area and units
!
CALL IPgAren(.13,.58,.93,.98)
!
! define angle of elevation/rotation
!
CALL IPgElevation(ELEV)
CALL IPgRotation(ROTA)
!
! Modify x/y scale tick mark angles & lengths
!
SELECT CASE (MOD(INT((ROTA+45.)/90.0),4))
CASE (0) ! Angle = 315-44 degrees
XTICKANGLE = 315.0 - ROTA
YTICKANGLE = 45.0 - ROTA
CASE (1) ! Angle = 45-134 degrees
XTICKANGLE = ROTA - 135.
YTICKANGLE = ROTA - 45.
CASE (2) ! Angle = 135-224 degrees
XTICKANGLE = 135.0 - ROTA
YTICKANGLE = 225.0 - ROTA
CASE (3) ! Angle = 225-314 degrees
XTICKANGLE = ROTA - 315.
YTICKANGLE = ROTA - 225.
END SELECT
CALL IPgXScaleAngle(XTICKANGLE,0.0)
CALL IPgYScaleAngle(YTICKANGLE,0.0)

```

La subrutina completa se muestra en el apéndice B.

IV.3.4. Subrutina de graficación en 2D.

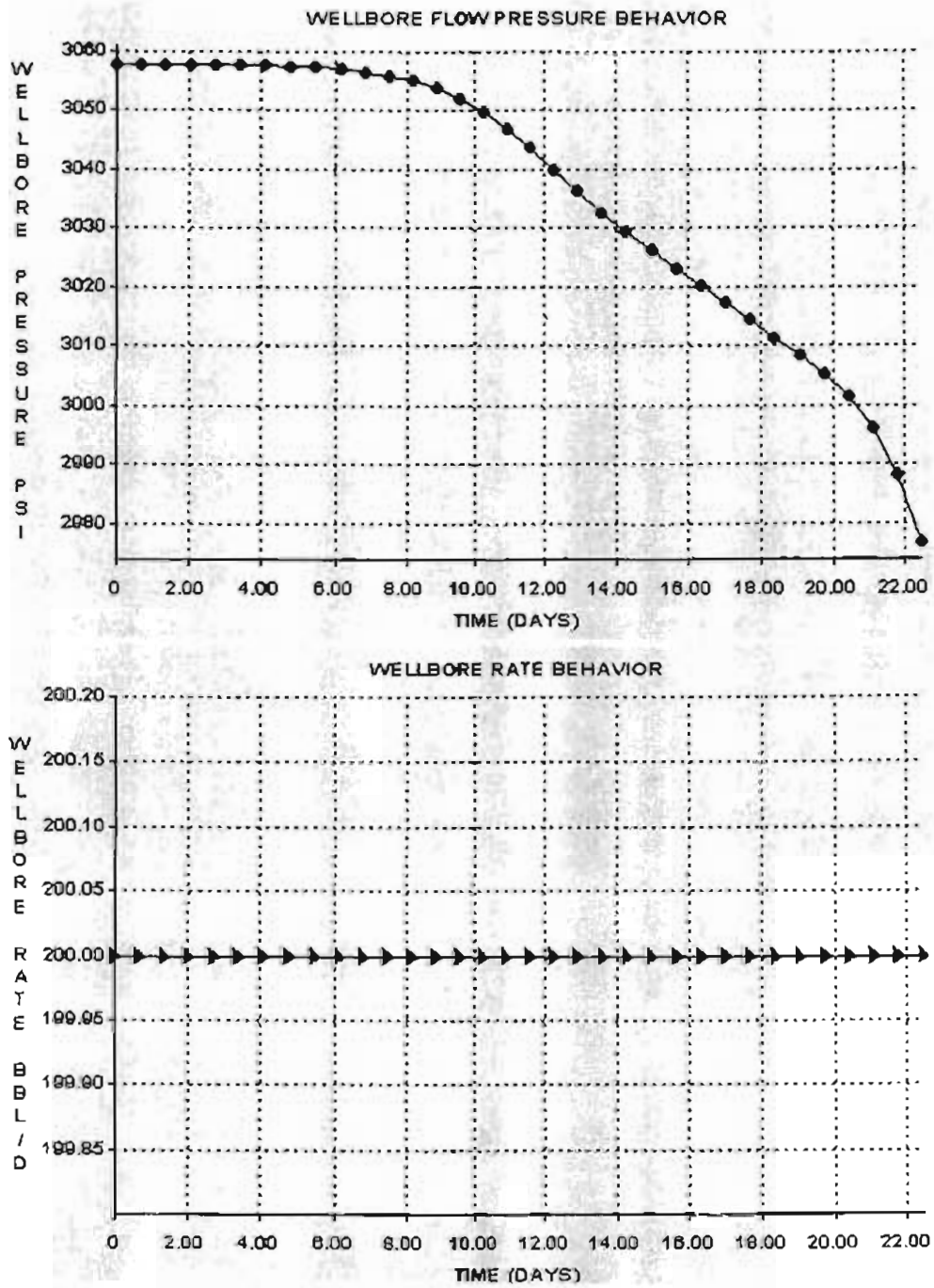


Fig. 4.7 Gráficas 2D.

Esta subrutina conforma la parte complementaria de la subrutina de visualización gráfica y esta diseñada con base en una serie de comandos de graficación en dos dimensiones que además de permitir la visualización de gráficas a color, incluye al igual que la subrutina de graficación en tres dimensiones, el manejo de memoria dinámica que permiten acoplarla tanto a la subrutina de visualización de gráficos como a la subrutina de control del Menú Principal de Entrada de Datos.

```

:
: Draw initial window contents
:
: CALL WindowSelect((HANDLE(2)) : select window 2
:
: call DRAW_BACKGROUND()
:
: Call IPgNewPlot to start a new plot
:
NVALUE = T_xcum/sec_day
IVALUE = IVALUE + 1
VALUES(IVALUE) = pwf(BLCKSLCTD)/pa_psi
IF (Qop(BLCKSLCTD)<=0.0) THEN
YMIN = 0.999*pwf(BLCKSLCTD)/pa_psi
YMAX = 1.001*VALUES(1)
ELSE
YMAX = 1.001*pwf(BLCKSLCTD)/pa_psi
YMIN = 0.999*VALUES(1)
END IF
CALL IPgNewPlot(PgLinePlot,1,IVALUE,LAYOUT=Pgl_ayStacked)

:
: Set style options
:
CALL IPgStyle(1,SolidLine,PgMarker,Outline,159,31)
CALL IPgMarker(1,14)
:
: Define PG area and units
:
CALL IPgArea(.11,.12,.98,.92)
:
: Set appropriate Presentation Graphics units for the plot
:
CALL IPgUnits(0.0,YMIN,NVALUE,YMAX)

```

La subrutina completa se muestra en el apéndice B.

CAPITULO V. APLICACIÓN DEL SIMULADOR.

V.1. Validación del modelo numérico.

Si consideramos una solución analítica para el flujo de fluidos en una sola fase y en una sola dimensión ya sea en coordenadas **cartesianas** o en coordenadas radiales; y su solución numérica por el Método de Diferencias Finitas, podemos validar el modelo numérico unidimensional comparando las curvas **obtenidas** al aplicar las dos diferentes soluciones, y así de esta manera extender esta validación a un modelo numérico en tres dimensiones.

V.1.1. Coordenadas Cartesianas.

La solución analítica ec. 2.24 para la ecuación de flujo de fluidos en medios porosos en una sola fase y en una sola dimensión en coordenadas cartesianas, y sin considerar pozos (fuentes o sumideros) es la siguiente (Crank, J., 1975):

$$\frac{\partial^2 p}{\partial x^2} = \left[\frac{\phi \mu c_t}{k} \frac{\partial p}{\partial t} \right] \quad (2.24a)$$

Bajo las condiciones iniciales y de frontera siguientes:

$$\begin{aligned} P(x, t = 0) &= 1200 \text{ psia} & 0 \leq x \leq 500 \\ P(x = 0, t > 0) &= 0 \\ P(x = L, t > 0) &= 0 \end{aligned}$$

La solución analítica es la siguiente:

$$P(x, t) = \frac{4P_i}{\pi} \sum_{n=0}^{\infty} \frac{1}{2n+1} \exp\left(-\frac{(2n+1)^2 \pi^2}{L^2} \frac{k}{\phi \mu c} t\right) \sin\left(\frac{(2n+1)\pi x}{L}\right) \quad (5.1)$$

Por otra parte si consideramos las siguientes condiciones iniciales y de frontera:

$$\begin{aligned} P(x, t = 0) &= 1200 & 0 \leq x \leq 500 \\ P(x = 0, t > 0) &= 0 \\ P(x = L, t > 0) &= 1200 \end{aligned}$$

La solución analítica para la ecuación de flujo de fluidos en medios porosos sin considerar pozos (fuentes o sumideros) es la siguiente:

$$P(x, t) = P_L + (P_R - P_L) \left[\frac{x}{L} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \exp\left(-\frac{n^2 \pi^2}{L^2} \frac{k}{\phi \mu c} t\right) \sin\left(\frac{n\pi x}{L}\right) \right] \quad (5.2)$$

Si aplicamos las ecuaciones anteriores y comparamos los resultados contra la solución numérica obtenida por diferencias finitas bajo determinadas condiciones para cada caso respectivamente, se obtienen las siguientes curvas que nos permiten validar con aproximación nuestro modelo numérico, tal como se puede observar.

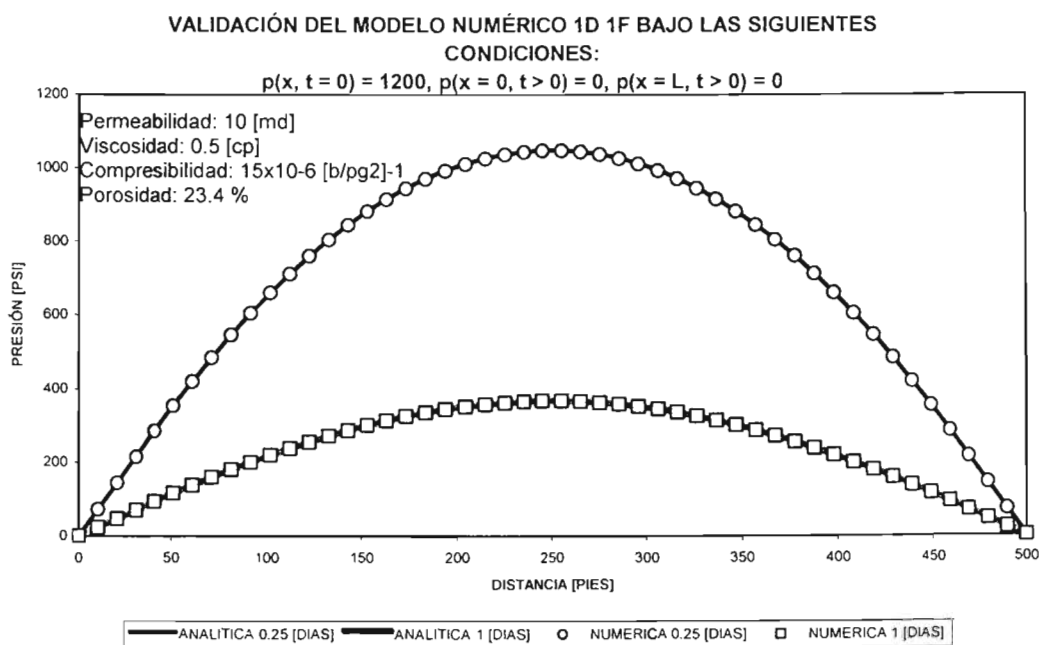


Fig. 5.1 Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en una sola dimensión (caso I).

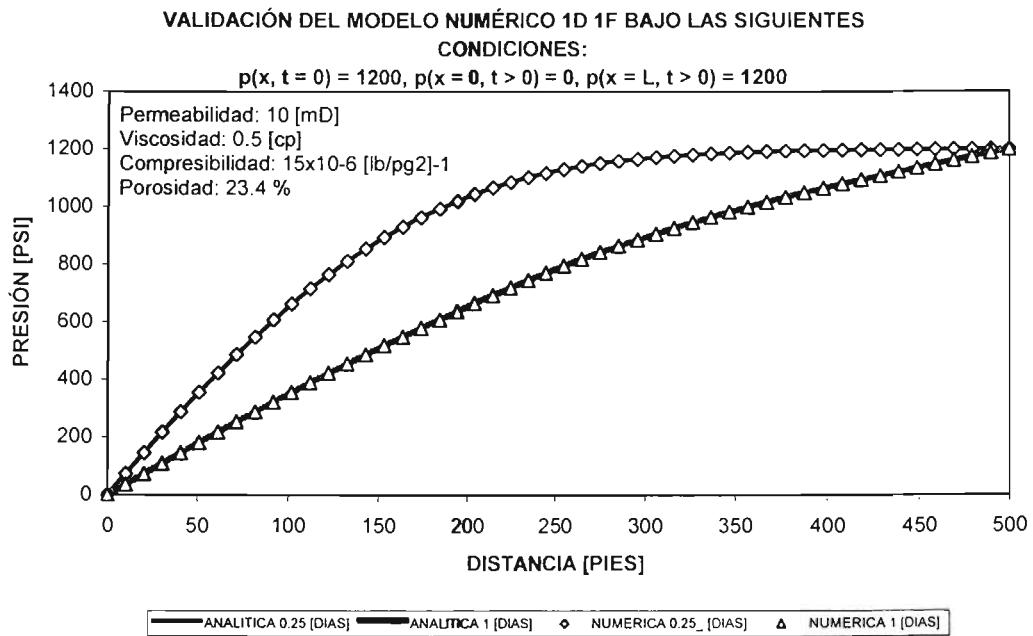


Fig. 5.2 Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en una sola dimensión (caso II).

V.1.2. Coordenadas Radiales.

La ecuación que gobierna la transmisión de presión en un medio poroso conteniendo un fluido ligeramente compresible esta dada por:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial \theta^2} + \frac{\partial^2 p}{\partial z^2} = \frac{\phi \mu c}{k_r} \frac{\partial p}{\partial t} \quad (5.3)$$

La ec. 5.3 es obtenida bajo las siguientes suposiciones:

- a. La ecuación de Darcy es válida
- b. Difusividad hidráulica constante
- c. Compresibilidad del fluido pequeña
- d. Gradientes en el yacimiento pequeños
- e. Flujo monofásico
- f. Efectos de gravedad y térmicos no considerados.
- g. Permeabilidad es isotrópica

Si se considera únicamente flujo radial, la ecuación 5.3 se reduce a:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} = \frac{\phi \mu c}{k_r} \frac{\partial p}{\partial t} \quad (5.4)$$

Esta ecuación es conocida como la ecuación de Difusividad, la cual aparece en varios campos de ingeniería y ciencias. Esta ecuación es de suma importancia en la teoría de pruebas de presión.

Después del periodo de almacenamiento, el transiente de presión que se podría medir en la cara del pozo refleja la transmisión de la presión a través del yacimiento. A tiempos largos, la respuesta esta influenciada por las fronteras del yacimiento, pero antes de estos efectos, la respuesta de presión no se entera de las fronteras y el yacimiento actúa si fuera infinito en extensión. Este tiempo intermedio, entre tiempos cortos donde la respuesta esta influenciada por el almacenamiento y tiempos largos donde dominan los efectos de frontera es conocido como periodo de yacimiento infinito. Aunque existen varios tipos de flujos identificados durante este periodo de yacimiento infinito, uno de los más fácilmente identificados es el flujo radial. El flujo radial para yacimientos comportándose infinitamente es la base de la interpretación de las técnicas de pruebas de presión.

En ausencia de almacenamiento y daño, el transiente de presión (respuesta de presión que se podría medir en la cara del pozo) bajo un comportamiento infinito de un flujo radial considerando al pozo como una línea fuente (source line) bajo la condición de gasto constante y en forma adimensional esta dada por (Da Prat, G., 1990):

$$p_D = -\frac{1}{2} E_i \left[-\frac{r_D^2}{4t_D} \right] \quad (5.5)$$

Donde:

$$p_D = \frac{2\pi kh}{qB\mu} (p_i - p_{(r,t)}) \quad (5.6)$$

$$t_D = \frac{kt}{\phi\mu cr_w^2} \quad (5.7)$$

$$r_D = \frac{r}{r_w} \quad (5.8)$$

Sustituyendo ecs. 5.6 a 5.8 en la ec. 5.5, se tiene lo siguiente:

$$p_{(r,t)} = p_i - \frac{q\mu B}{4\pi kh} E_i \left[\frac{r^2}{4\eta t} \right] \quad (5.9)$$

Donde:

$$\frac{1}{\eta} = \frac{\phi\mu c}{k} \text{ es la Difusividad hidráulica.} \quad (5.10)$$

$p_{(r,t)}$ es la presión en un punto r al tiempo t .

La integral exponencial, E_i , en la ec. 5.9 puede ser aproximada como:

$$E_i(-x) = \ln x + 0.5772 \text{ cuando } x < 0.01 \quad (5.11)$$

$$\text{Donde: } x = \frac{r_D^2}{4t_D} = \frac{r^2}{4\eta t} = \frac{r^2}{\frac{4kt}{\phi\mu c}} = \frac{r^2\phi\mu c}{4kt}$$

Cuando la ec. 5.5 es evaluada en la cara del pozo (disparos) $r = r_w$, la misma aproximación logarítmica puede ser usada:

$$\text{En la cara del pozo se tiene que: } r_D = \frac{r_w}{r_w} = 1 \text{ y } x = \frac{1}{4t_D}$$

$$\text{Haciendo } \ln x = \ln \frac{1}{4t_D} = \ln(1) - \ln(4t_D) = 0 - [\ln(4) + \ln(t_D)]$$

Entonces la ec. 5.5 se reduce a (note el cambio de signos):

$$p_{wD} = \frac{1}{2} \left[\ln t_D + \frac{0.80907}{-\ln(4)+0.5572} \right] + s \quad (5.12)$$

donde s es el daño de la formación y en la ec. 5.12 es representado como una caída de presión adimensional extra. En unidades consistentes la ec. 5.12 es, (nótese la incorporación de factor de daño s dentro del paréntesis) :

$$p_{wf} = p_i - \frac{\ln(10)qB\mu}{4\pi kh} \left[\log \frac{kt}{\phi\mu cr_w^2} + \frac{0.35137}{\frac{0.80907}{\ln(10)}} + \frac{0.8686s}{\frac{2}{\ln(10)}} \right] \quad (5.13)$$

para cambiar de \ln a \log se uso : $\ln x = \ln(10)\log x = 2.3025 \log x$

La ec. 5.13 en sistema de unidades de campo es:

$$p_{wf} = p_i - 162.6 \frac{qB\mu}{kh} \left[\log t + \log \frac{k}{\phi\mu cr_w^2} - 3.2274 + 0.8686s \right] \quad (5.14)$$

Donde:

p : lb/pg²

q : BPD

k : mD

h : pie

μ : cp

t : horas

c : (lb(pg²))⁻¹

r_w : pie

Si ahora aplicamos la ecuación 5.9 que gobierna el flujo de fluidos en medios porosos en una sola fase y en coordenadas radiales, y comparamos los resultados contra la solución numérica obtenida por diferencias finitas bajo determinadas condiciones para cada caso respectivamente, se obtienen las siguientes curvas que de igual manera que en caso de coordenadas cartesianas nos permiten validar con aproximación nuestro modelo numérico.

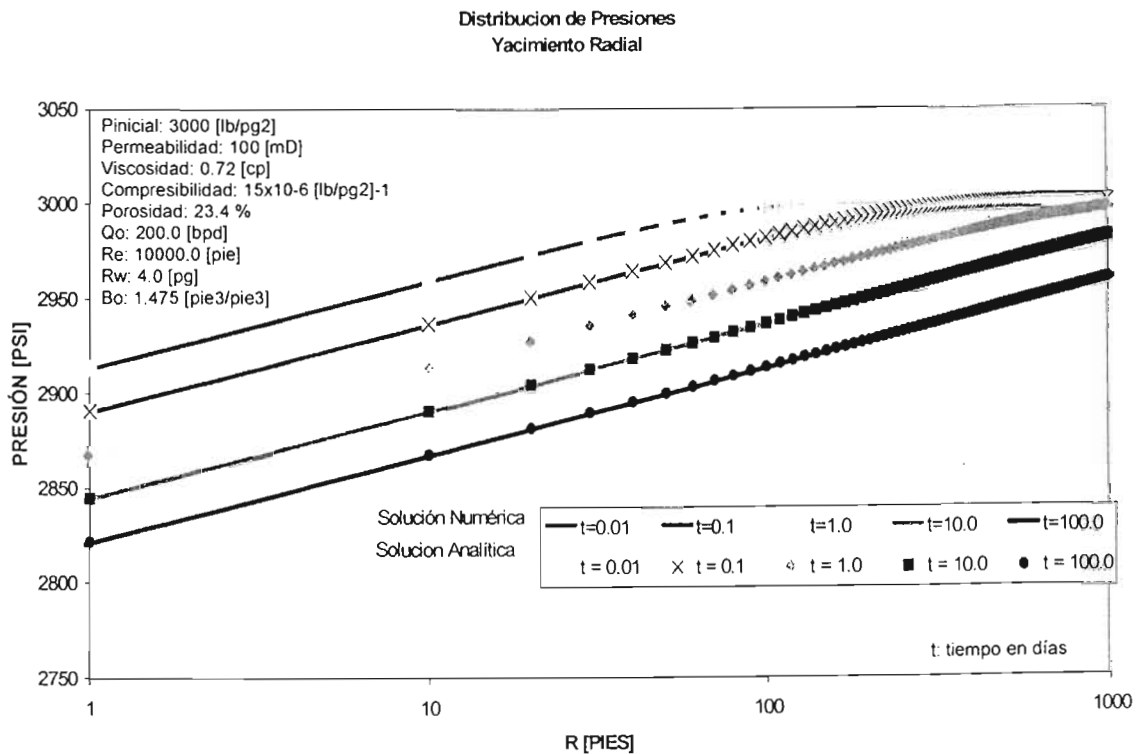


Fig. 5.3 Comparación entre la solución analítica y la solución numérica para la ecuación de flujo de fluidos en medios porosos en una sola fase y en coordenadas radiales.

V.2. Comportamiento de un solo pozo productor.

Como primer caso de aplicación se corrió la simulación de un pozo productor vertical bajo condiciones de gasto de 200 bbl/día con una malla de 21 X 21 nodos en el plano cartesiano XY, con una longitud total de 1100 pies en cada lado del plano respectivamente y considerando una sola capa en la dirección Z de 550 pies de espesor y coordenadas (11, 11,1) en las direcciones x, y y z respectivamente y con las siguientes propiedades:

Longitud en X (pie)	: 1100.0
Longitud en Y (pie)	: 1100.0
Longitud en Z (pie)	: 550.0
Numero de nodos en x	: 21
Numero de nodos en y	: 21
Numero de nodos en Z	: 1
Numero de pozos en prod	: 1
Radio del pozo (pies)	: 0.45
Tiempo Total de Simulación (dias)	: 20.0
Permeabilidad en X (mD)	: 2.0
Permeabilidad en Y (mD)	: 2.0
Permeabilidad en Z (mD)	: 2.0
Viscosidad (cp)	: 0.5
Gravedad API	: 40.0
Compresibilidad fluido (psi-1)	: 6.9e-06
Compresibilidad roca (psi-1)	: 1.0e-06
Porosidad (-)	: 0.15
FVF Base (Pi)	: 1.1
Presión base (psi)	: 500.0
Presión inicial (psi)	: 3000.0
Gasto de Producción (bbl/día)	: 200

Los resultados obtenidos muestran el siguiente comportamiento:

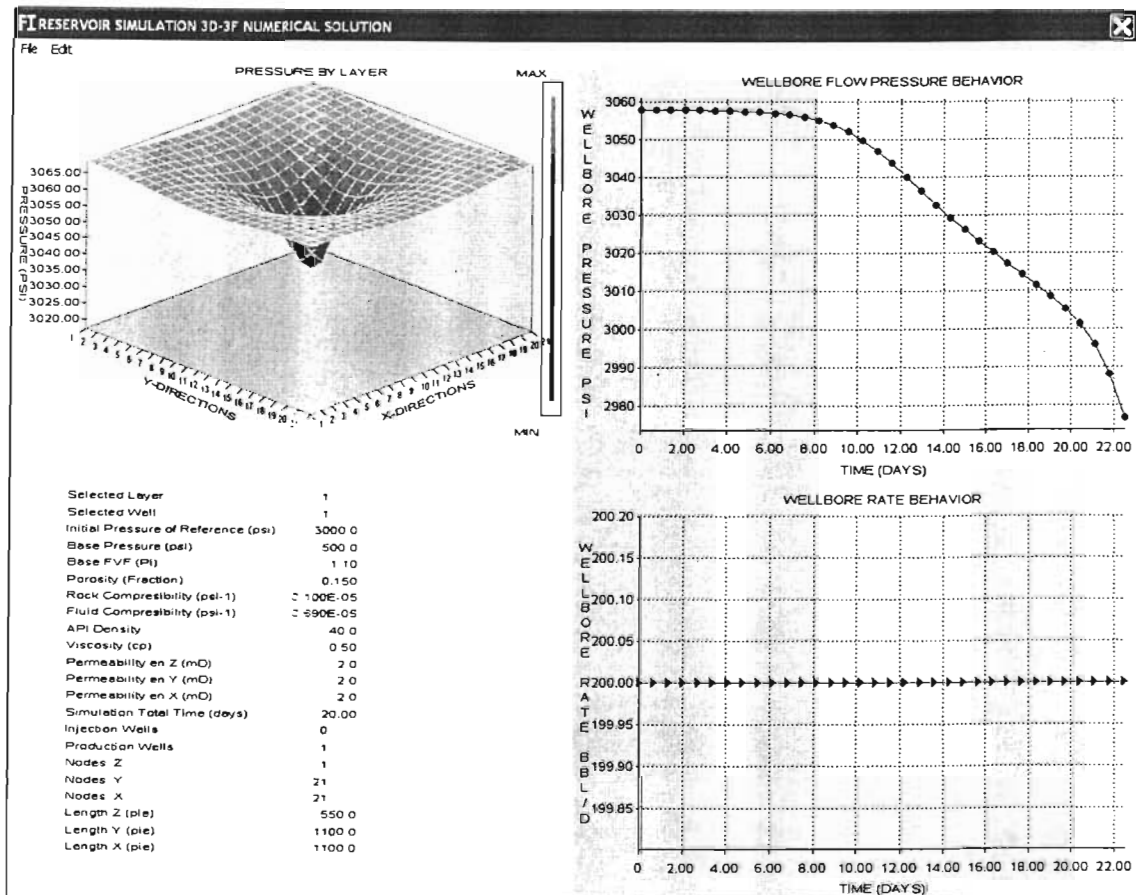


Fig. 5.4 Simulación Pozo Productor Vertical al tiempo total de simulación.

Wells Position

Production well data

Input well position (X, Y, Z) rate and radio

	Xp	Yp	Zp	Rp	Qop	Tin_p(D)	Tf_p(D)
1	11	11	1	0.45000	-200.00		

Buttons: Simulate, Exit

Fig. 5.5 Coordenadas. Caso de Simulación Pozo Productor Vertical al tiempo total de simulación.

V.3. Comportamiento de cuatro pozos productores.

Como segundo caso de aplicación se corrió la simulación de cuatro pozos productores verticales bajo condiciones de gasto de 100, 150, 200 y 250 bbl/día respectivamente para cada pozo, con una malla de 21 X 21 nodos en el plano cartesiano XY, con una longitud total de 1100 pies en cada lado del plano respectivamente y considerando una sola capa en la dirección Z de 550 pies de espesor y coordenadas (21,1,1), (1,21,1), (21,21,1), (1,1,1) en las direcciones x, y y z respectivamente para cada pozo y con las mismas propiedades que el caso anterior.

Los resultados obtenidos muestran el siguiente comportamiento:

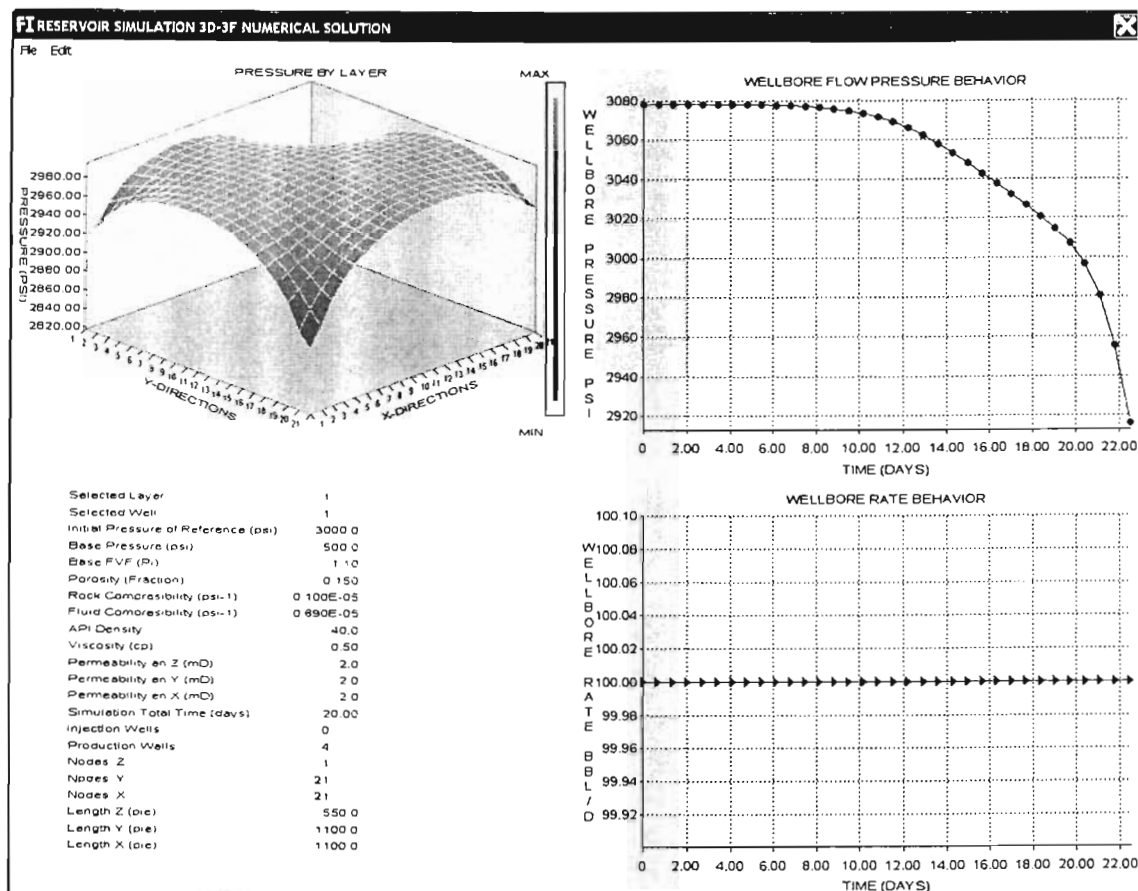


Fig. 5.6 Simulación Cuatro Pozos Productores Verticales al tiempo total de simulación.

Wells Position

Production well data

Input well position (X, Y, Z) rate and radi

	Xp	Yp	Zp	Rp	Qop	Tm_p(D)	Tf_p(D)
1	21	21	1	0.45000	-100.00		
2	21	1	1	0.45000	-250.00		
3	1	21	1	0.45000	-200.00		
4	1	1	1	0.45000	-150.00		

Simulate Exit

Fig. 5.7 Coordenadas. Caso de Simulación Cuatro Pozos Productores Verticales al tiempo total de simulación.

V.4. Comportamiento de un solo pozo productor horizontal.

Como tercer caso de aplicación se corrió la simulación de un pozo productor horizontal bajo condiciones de gasto de 40 bbl/día con una malla de 21 X 21 nodos en el plano cartesiano XY, con una longitud total de 1100 pies en cada lado del plano respectivamente y considerando una sola capa en la dirección Z de 550 pies de espesor, excepto que el pozo horizontal lo forman seis pozos verticales, uno en cada celda horizontal en las direcciones x y y respectivamente para cada pozo y con las mismas propiedades que el primer caso.

Los resultados obtenidos muestran el siguiente comportamiento:

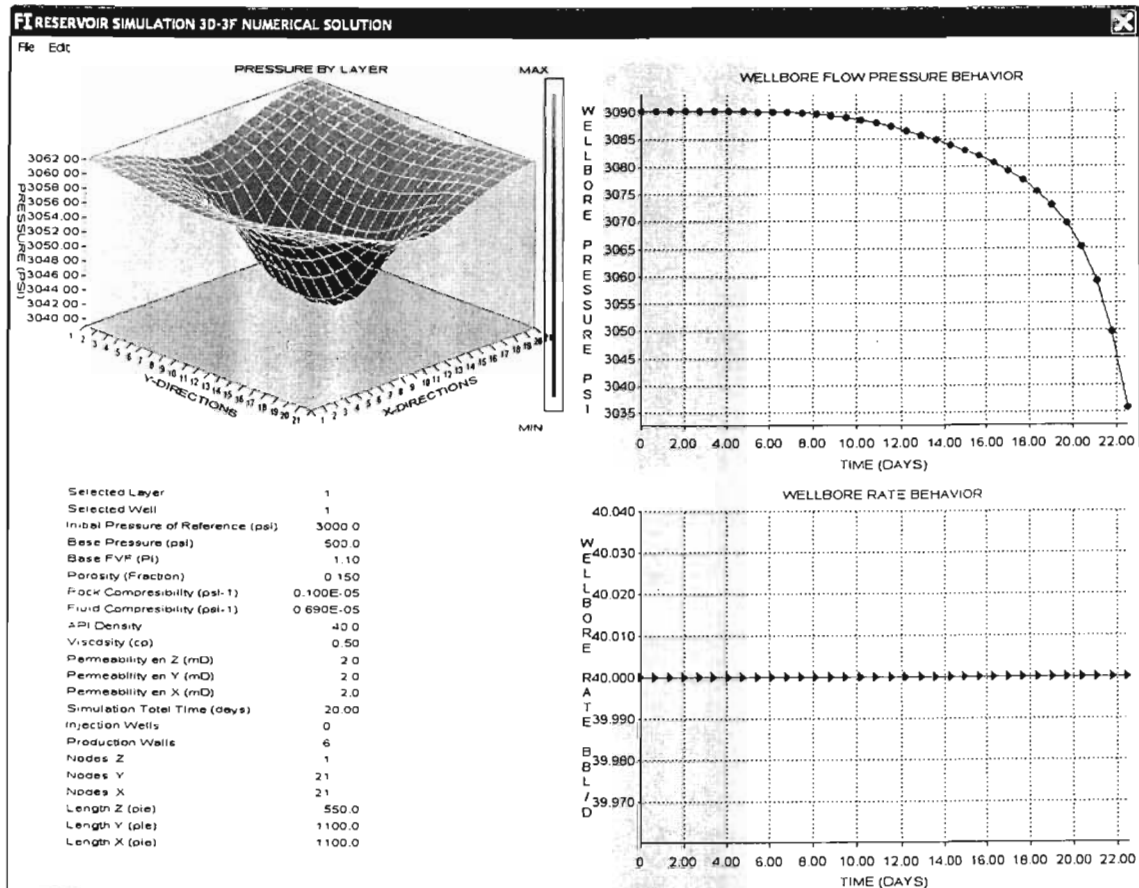


Fig. 5.8 Simulación Pozo Productor Horizontal al tiempo total de simulación.

Wells Position

Production well data

Input well position [X, Y, Z] rate and radio

	Xp	Yp	Zp	Rp	Qop	Tin_p(D)	Tf_p(D)
1	11	6	1	0.45000	-40.000		
2	11	8	1	0.45000	-40.000		
3	11	10	1	0.45000	-40.000		
4	11	12	1	0.45000	-40.000		
5	11	14	1	0.45000	-40.000		
6	11	16	1	0.45000	-40.000		

Simulate Exit

Fig. 5.9 Coordenadas. Caso de Simulación Pozo Productor Horizontal al tiempo total de simulación.

CAPITULO VI. DISCUSIÓN DE RESULTADOS.

VI.1. Comportamiento de un solo pozo productor.

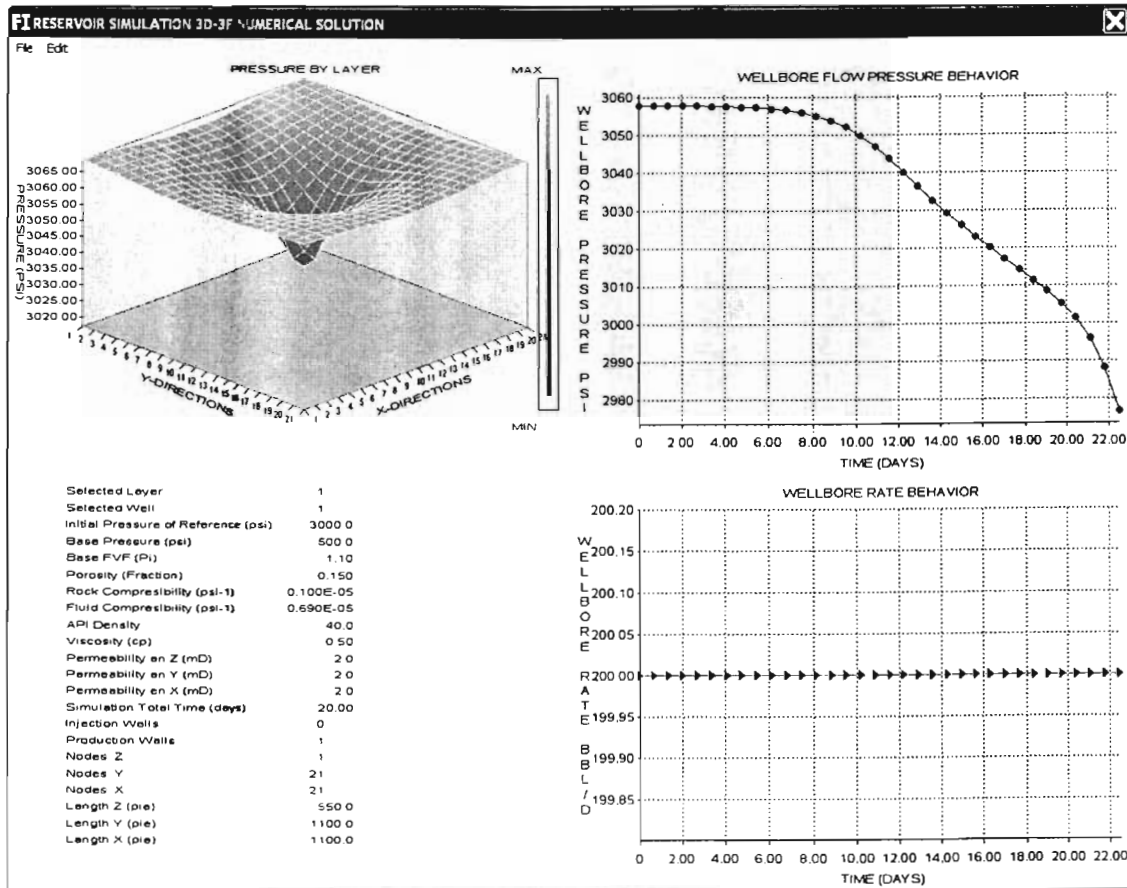


Fig. 6.1 Primer caso. Comportamiento de un pozo productor vertical.

Se hizo una corrida de simulación para un solo pozo vertical con gasto de 200 bbl/día que de acuerdo al modelo numérico del simulador se mantiene constante durante el tiempo total de simulación tal como se puede observar en la gráfica inferior derecha de la figura. En la gráfica superior derecha podemos apreciar la variación de la presión de fondo fluyendo en el pozo productor con una tendencia de declinación que muestra dos concavidades como resultado de los efectos de las fronteras en los extremos del yacimiento representado, y que

se aprecian a partir de un tiempo de simulación de 5 días y hasta el tiempo total de simulación de 20 días, aunque cabe aclarar que debido al incremento en tiempo utilizado en el método iterativo para el cálculo de las presiones, esta gráfica y la anterior mencionada para el gasto muestran un tiempo total de simulación de 22 días que no se extrapolan, si no que durante la ejecución del simulador se calculan antes de cumplir con una condición necesaria para que no se sigan realizando cálculos. Por otra parte, en la gráfica superior izquierda de la figura se tiene la distribución simétrica de presiones que en este caso se presentan en el espesor total del yacimiento representado debido a que se tiene una sola capa en la dirección Z de la malla numérica utilizada para este caso en específico. Aquí podemos apreciar tanto en escala numérica como en gama de color la declinación de las presiones desde las fronteras del yacimiento hasta el pozo productor representado en el centro de la malla numérica donde por consecuencia de la producción se tiene una mayor caída de presión. Por último en la parte inferior derecha de lo que es la ventana o consola de visualización gráfica, se muestran al final de la simulación tanto las propiedades empleadas para la simulación como los datos que representan las condiciones básicas para que se lleve a cabo esta, como la presión inicial, longitud de la malla en las direcciones x,y,z , número de nodos en cada dirección respectivamente, y número de pozos productores.

VI.2. Comportamiento de cuatro pozos productores.

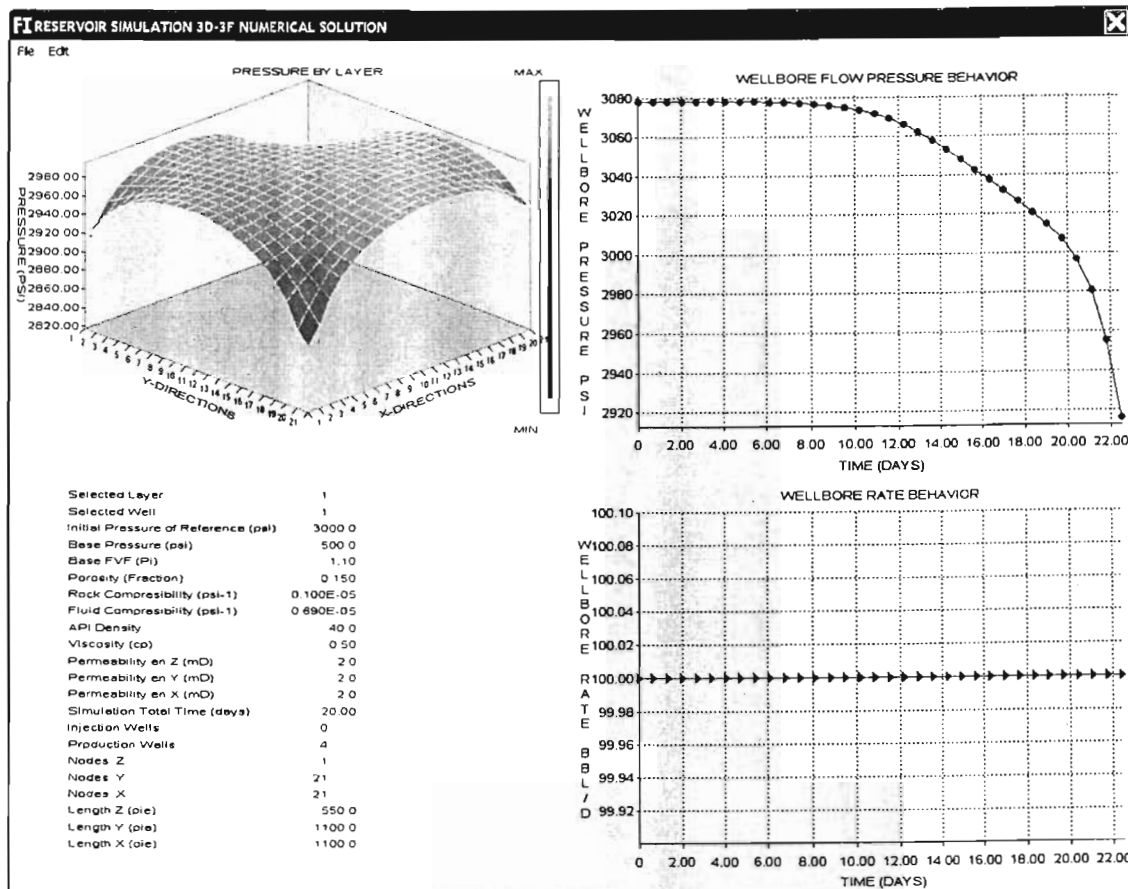


Fig. 6.2 Segundo caso. Comportamiento de cuatro pozos productores verticales.

Como segundo caso se hizo una corrida de simulación para cuatro pozos productores verticales, con diferentes gastos producción cada uno, aunque con tiempos de inicio de producción iguales y con el mismo tiempo total de simulación. Cada pozo se ubicó con coordenadas en cada esquina de la malla numérica utilizada, de tal forma que podemos observar una mayor uniformidad en la gráfica que representa la presión de fondo fluyendo de uno de los cuatro pozos productores elegido y que debido a su posición dentro de la malla no manifiesta concavidades como en el caso anterior por lo que la curva de presión de fondo fluyendo es mas parecida a las gráficas que comúnmente se pueden apreciar de índices de productividad con una parte lineal en los primeros tiempos de producción y con

una declinación más severa de la presión en los tiempos intermedios y finales de producción. De igual forma que el caso anterior la gráfica también muestra un gasto constante que aunque solo presenta el de un solo pozo seleccionado para la simulación también se consideran constantes los gastos de cada uno de los pozos que se incluyen en la corrida. Y así como en el primer caso, en la gráfica tridimensional, también podemos apreciar para este caso la variación de las presiones a lo largo del yacimiento representado, aunque sin embargo la distribución de presiones en esta ocasión no es uniforme principalmente debido a que se tiene que cada uno de los pozos considerados produce con distinto gasto, e independientemente de que el tiempo de inicio de producción sea el mismo para cada pozo.

VI.3. Comportamiento de un solo pozo productor horizontal.

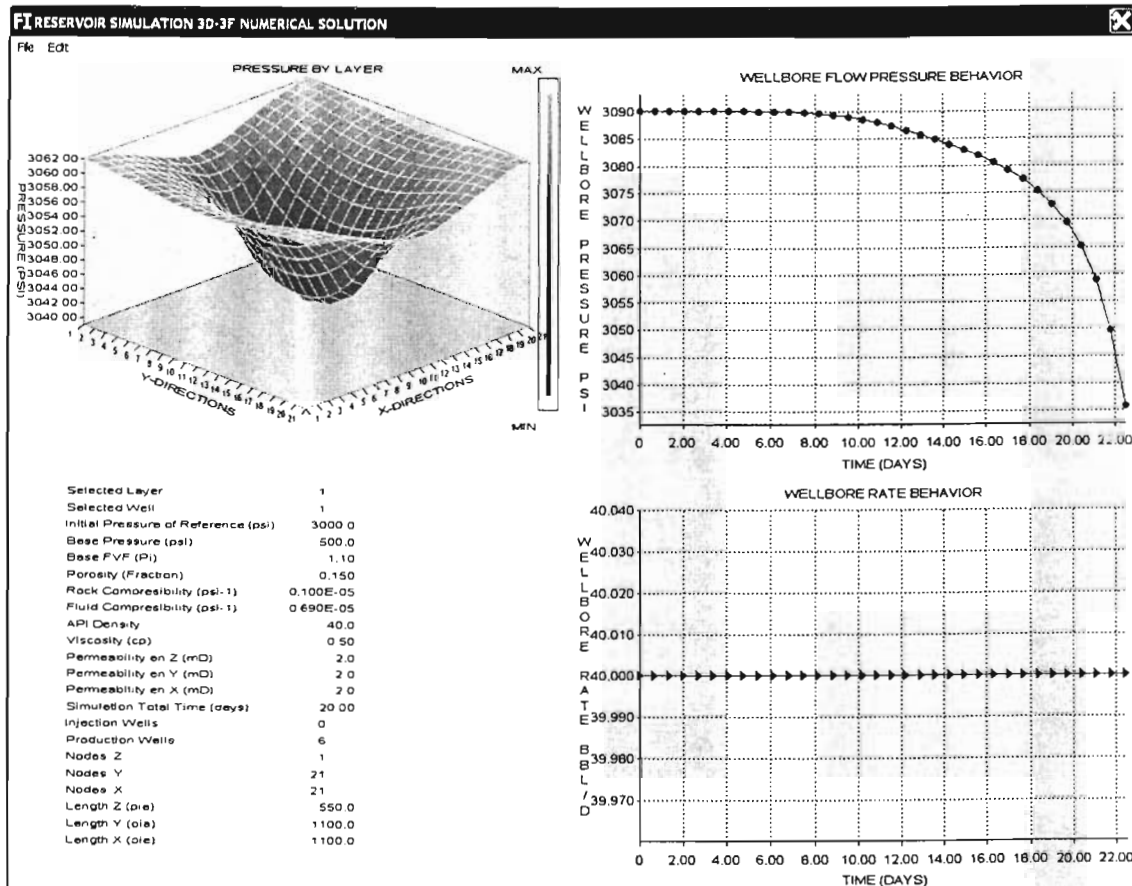


Fig. 6.3 Tercer caso. Comportamiento de un pozo productor horizontal.

Como ultimo caso, se hizo una corrida de simulación para un pozo horizontal que a diferencia los anteriores, este pozo esta formado por seis pozos verticales uno en cada celda diferente de la malla numérica utilizada y con coordenadas horizontales simétricas en el plano XY. Como se trata de un caso especial, dado que el modelo numérico no esta diseñado para simular pozos horizontales, es importante mencionar que para esta simulación se consideró igual gasto para cada uno de los seis pozos que se emplearon en la simulación del modelo horizontal de pozo. Además puesto que el modelo numérico de simulación que se propone emplea el Modelo de Peaceman que se utiliza en el cálculo de la presión de fondo fluyendo únicamente para pozos verticales, se asume como una

aproximación para el cálculo de la presión de fondo fluyendo del pozo horizontal que se trata de representar en este caso. Por lo tanto como una aproximación para la declinación de la presión de fondo fluyendo de acuerdo a la Teoría de Peaceman, la gráfica que representa la declinación de la presión de fondo fluyendo para el pozo horizontal muestra únicamente la variación de la presión de fondo fluyendo de un pozo seleccionado de los seis pozos verticales que conforman el pozo horizontal de este caso. Y al igual que el caso anterior de los cuatro pozos verticales e igualmente a diferencia del primer caso, la curva de variación de la presión de fondo fluyendo, es mayormente uniforme y las concavidades propiciadas por los efectos de frontera no se presentan de la misma manera, aunque sin embargo, se observa una declinación menos drástica a lo largo de la mayor parte del tiempo de simulación que va del inicio hasta el día 18, no siendo así a partir de este día hasta el tiempo total de simulación de 20 días en que el abatimiento de presión comienza a ser drástico. En el caso de la representación del gasto mostrado en la gráfica inferior derecha de la consola de visualización, para el pozo horizontal, se consideró únicamente el gasto del mismo pozo vertical seleccionado para la representación de la presión de fondo fluyendo, dado que, como ya se había mencionado se consideró igual gasto para cada uno de los pozos que conforman el modelo horizontal de pozo.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Después de realizar varios análisis con diferente información el presente trabajo arroja las siguientes conclusiones y recomendaciones:

- El simulador numérico con la interfaz de visualización es una herramienta útil para el análisis del comportamiento de yacimientos de aceite bajosaturados.
- Desde el punto de vista académico la herramienta desarrollada podrá ser de utilidad en las materias de Ingeniería de Yacimientos, donde los usuarios podrán implantar sus propios desarrollos para observar diferentes procesos en la dinámica del yacimiento.
- La interfaz de visualización podrá ser usada con otros simuladores, debido a su estructura de módulos.

RECOMENDACIONES

- Extender su análisis a un sistema trifásico (tres dimensiones; x,y,z y tres fases; o,g,w).
- Importar a otro tipo y geometría de malla, (geometría cilíndrica, simulación de un pozo, en las direcciones r,Θ y z).
- Realizar el modelado para pozos horizontales, con el empleo de una formulación adecuada y específica para este tipo de simulación.

NOMENCLATURA

$$\tilde{m} = \text{gasto másico por unidad de área.} \left[\frac{M}{tL^2} \right]$$

$$\rho = \text{densidad del fluido en el medio poroso.} \left[\frac{M}{L^3} \right]$$

$$u = \text{velocidad de flujo de fluido en el medio poroso.} \left[\frac{L}{t} \right]$$

$$m = \text{masa en el volumen de control} [M]$$

$$S = \text{saturación del fluido en el medio poroso.} \left[\frac{L^3}{L^3} \right]$$

$$\phi = \text{porosidad de la roca. (\% o fracción)}$$

$$V_v = \text{volumen de roca (Volumen de Control VC).} [L^3]$$

$$\phi V_v = \text{volumen poroso en el volumen de control.} [L^3]$$

$$\Delta t = \text{incremento en tiempo.} [t]$$

$$A = \text{área en el volumen de control.} [L^2]$$

$$\Delta x = \text{incremento de longitud o distancia en dirección } x. [L]$$

$$q_m = \text{masa que se inyecta o se produce en el volumen de control.} \left[\frac{M}{t} \right]$$

$$\tilde{q}_m = \frac{q_m}{V_v} = \text{representa el gasto másico de fluido extraído por unidad de volumen de roca.} \tilde{q}_m \left[\frac{M}{tL^3} \right] = \frac{q_m}{V_v} \left[\frac{M}{t} \right] \left[\frac{1}{L^3} \right]$$

$$\gamma_p = \text{peso específico de la fase } p \left[\frac{lb}{pie^3} \right]$$

$$\vec{\nabla} = \text{operador nabla, divergencia del gasto másico en sus diferentes direcciones o geometrías.}$$

$$\vec{u}_p = \text{velocidad de flujo en el medio poroso (Ley de Darcy).}$$

$$c_f = \text{compresibilidad del fluido.} \left[\frac{lb}{pg^2} \right]^{-1}$$

$$c_r = \text{compresibilidad de la roca.} \left[\frac{lb}{pg^2} \right]^{-1}$$

$$c_t = c_r + c_f \text{ compresibilidad total.} \left[\frac{lb}{pg^2} \right]^{-1}$$

$$PV = ZnRT : \text{Ley de los Gases Reales.}$$

$$B = \text{factor de volumen del fluido, Vol @ c. y. / Vol @ c. s.}$$

$$\tilde{q}_v = \text{gasto volumétrico @ c.s. por unidad de volumen de roca.} \tilde{q}_v \left[\frac{L^3}{tL^3} \right] = \frac{\tilde{q}_m \left[\frac{m}{tL^3} \right]}{\rho \left[\frac{m}{L^3} \right]}$$

$$K = \text{permeabilidad del medio poroso. (md)}$$

$$\mu = \text{viscosidad del fluido. (cp)}$$

u_x = componente de la velocidad en dirección x
 u_y = componente de la velocidad en dirección y
 u_z = componente de la velocidad en dirección z
 η = coeficiente de difusividad hidráulica
 E_i = integral exponencial; $E_i(-x) = \ln x + 0.5772$ cuando $x < 0.01$
 P_o = presión a dim ensional
 t_o = tiempo a dim ensional
 r_o = dis tan cia radial a dim ensional
 $p_{r,t}$ = presión en un punto r al tiempo t
 p_{-o} = presión a dim ensional en la cara del pozo
 p_i = presión inicial, (lb/pg³)
 s = factor de daño a la formación. a dim ensional
 P_{-f} = presión de fondo fluyendo, (lb/pg³)

$$\Psi = \frac{1}{IMAX} \ln \left(\frac{r_i}{r_w} \right)$$
 Vr_i = volumen de roca en la celda i : (AΔA)

REFERENCIAS.

- Arana Ortiz, V. H.** “Desarrollo de un simulador semi-implícito del flujo multifásico composicional en yacimientos naturales fracturados”, Tesis Maestría (Maestría en Ingeniería Petrolera)-UNAM, Facultad de Ingeniería, 107 pp., México: El autor, 1996.
- Arreola, T., M.A. y Rodríguez de la G., F.** ”Efecto de la geometría de la Malla Sobre la Estabilidad de un Simulador Numérico”, Instituto Mexicano del Petróleo, Subdirección de Tecnología de Explotación, División de Yacimientos, Diciembre 1986.
- Aziz, Khalid. y Settari, Antonin.** “Petroleum Reservoir Simulation”, London: Elsevier Applied Science Publishers, [s.f.], 1979.
- Calahan, D.A. (1968).** “A stable accurate method of numerical integration for non linear systems”, IEEE Proc. 56 pp.
- Chapman Stephen J.** “Fortran 90/95 for Scientists and Engineers”, McGraw-Hill Education (ISE Editions). 1998.
- Chierici, G.L.** “Principles of Petroleum Reservoir Engineering”, Vol. 1 y 2, Springer-Verlag Berlin and Heidelberg GmbH & Co. K. 1994.
- Chilingarian, G.V., Mazzullo S.J. y Rieke, H.H. et al.** “Carbonate Reservoir Characterization: A Geologic-Engineering Analysis, Part I”, Ámsterdam: Elsevier Science Publishing Co. N.Y. 640 pp., 1992.
- Chivers I.D., Sleightholme J.** “Introducing Fortran 90”, Springer-Verlag Berlin and Heidelberg GmbH & Co. K. 1995.
- Chrichlow, H.B.** “Modern Reservoir Engineering. A Simulation Approach”, Prentice-Hall, Inc. New Jersey 1977.
- Coats, K.H., Dempsey, J.R. y Henderson, J.H. (1971a).** “The use of vertical equilibrium in two-dimensional simulation of three-dimensional reservoir performance, Trans”, SPE of AIME, 251 pp. (SPEJ).
- Craft, B. C. y Hawkins, M. F. (1959).** “Applied Petroleum Reservoir Engineering”, Prentice-Hall, New Jersey.

- Crank, J.** “The Mathematics of Diffusion”, Oxford University Press, 2^a. Ed., New York, 1975.
- Da Prat, G.** “Well Test Analysis for Fracture Reservoir Evaluation”, Ámsterdam: Elsevier Science Publishing Co. N.Y. Vol. 27, 210 pp., 1990.
- Ellis, Philips, Lahey.** “Fortran 90 Programming”, Addison-Wesley. 1994.
- Ertekin Turgay, Abou-Kassem Jamal H. y King Gregory R.** “Basic Applied Reservoir Simulation”, SPE, Textbook Series, Vol.7, Richardson, Texas 2001.
- Forsythe, G.E. y Wasow, W. (1960).** “Finite-Difference Methods for Partial Differential Equations”, J. Wiley&Sons, New York.
- Galindo Nava, A. P.** “Un simulador numérico de flujo multifásico tridimensional hacia un pozo naturalmente fracturado”, Tesis Maestría (Maestría en Ingeniería Petrolera)-UNAM, Facultad de Ingeniería, 425 pp., México: El autor, 1998.
- Interactive Software Services Ltd. 1997-2004.** “Winteracter. Programming Guide”, United Kingdom. Winteracter Version 6.0.
- Jeanne Adams, Walter Brainerd, Jerry Wagener, Brian Smith.** “Fortran 90 handbook: Complete ANSI Reference”, McGraw-Hill Education. 1991.
- Jeanne C. Adams, Walter S. Brainerd, Jeanne T. Martin, Brian Smith, Jerrold L. Wagener.** “Fortran 95 Handbook (Scientific & Engineering Computation S.)”, The MIT Press. 1997.
- Loren P. Meissner.** “Fortran 90”, PWS. 1995.
- Mattax Calvin C. y Robert Dalton.** “Reservoir Simulation”, SPE, Monograph, Vol. 13, Richardson, Texas 1990.
- Metcalf, Reid.:** “Fortran 90/95 explained”, Oxford Univ. Press 1999.
- Michael Metcalf (Foreword), William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery.** “Numerical Recipes in Fortran 90”, Vol. 2, 1996.
- Muskat, M. (1949).** “Physical Principles of Oil Production”, Mc Graw-Hill, New York.
- Peaceman, Donald W. (1967).** “Numerical solution of the non linear equations for two-phase flow through porous media, in: Non linear Partial Differential Equations – Methods of Solutions”, ed. W.F. Ames, Academic Press, New York.

- Peaceman, Donald W. (1977).** "A non linear stability for difference equations using semi-implicit Mobility", Soc. Petrol. Eng. Journal, 17 No.1 pp. 79-91.
- Peaceman, Donald W.** "Fundamentals of numerical reservoir simulation", Ámsterdam: Elsevier Science Publishing Co., 176 pp. 1977
- Price, H.S. y Coats, K.H.** "Direct Methods in Reservoir Simulation", SPEJ, (Junio 1974).
- Rodríguez de la G. F.** "Un Simulador Numérico del Fenómeno de Conificación", Instituto Mexicano del Petróleo, Subdirección de Tecnología de Explotación, División de Yacimientos, Publicación interna, Diciembre 1986.
- Sherman, Andrew H.** "Algorithm 533: NSPIV, A Fortran Subroutine for Sparse Gaussian Elimination with Partial Pivoting [F4]", ACM Transactions on Mathematical Software (TOMS) Vol. 4, 4ª. Ed., Páginas 391-398, Diciembre 1978.
- W.S. Brainerd, J.C. Adams, C.H. Goldberg.** "Programmer's Guide to Fortran 90", Springer-Verlag New York Inc. 1996.

APÉNDICE A. Principio de conservación de la materia. Balance de masa.

Para obtener las ecuaciones de flujo multifásico en medios porosos, considérese un volumen elemental representativo del medio poroso, en donde existe flujo unidimensional. Con base en el principio de conservación de materia, se puede establecer el siguiente balance de masa en el elemento.

Existen tres fases móviles fluyendo en el elemento de volumen. La fase agua no intercambia su masa con otras fases.

Cada una de las masas de las fases deberá conservarse a través de todo el sistema.

Considérese la conservación de masa para cada fase en coordenadas rectangulares y sólo en la dirección x :

Ritmo de entrada de masa de la fase de aceite (o) al volumen de control =

$$(A\rho_o v_{ox})|_x \quad (A.1)$$

Ritmo de salida de masa de la fase de aceite (o) al volumen de control =

$$(A\rho_o v_{ox})|_{x+\Delta x} + \quad (A.2)$$

Donde A , es el área transversal del elemento expuesto al flujo, ρ_p es la densidad volumétrica de la fase p y v_{px} son las velocidades macroscópicas de la fase p en la dirección x .

El ritmo de acumulación de masa en el volumen de control, está dado por el ritmo de cambio de la masa de la fase aceite.

Ritmo de acumulación de masa en el volumen de control =

$$A\Delta x \frac{\partial}{\partial t} (\phi S_o \rho_o) \quad (A.3)$$

El ritmo de producción / inyección de masa de aceite del volumen de control, definiendo \hat{q}_p = gasto volumétrico del la fase $p = o, g, w$ a condiciones de yacimiento por unidad de volumen de roca, es :

Ritmo de producción/ inyección del aceite =

$$A\Delta x (\rho_o \hat{q}_o) \quad (A.4)$$

Substituyendo las ecuaciones (A.1) a (A.4) en el postulado de conservación de materia, dividiendo la expresión resultante entre el volumen de control, $A\Delta x$, reorganizando y tomando límites cuando $\Delta x \rightarrow 0$, se llega a:

$$-\frac{\partial}{\partial x} (\rho_o v_{ox}) + \rho_o \hat{q}_o = \frac{\partial}{\partial t} [\phi \rho_o S_o] \quad (A.5)$$

La ecuación (A.5) puede extenderse al caso de flujo multidimensional. En notación del operador diferencial nabra, ∇ , estas ecuaciones pueden escribirse en forma más general como :

$$-\nabla \cdot [(\hat{\rho}_o v_o)] + \rho_o \hat{q}_o = \frac{\partial}{\partial t} [\phi \rho_o S_o] \quad (A.6)$$

Para flujo laminar en el yacimiento, podemos expresar la velocidad de cada fase mediante la ecuación de Darcy:

$$v_p = -\frac{kk_{rp}}{\mu_p}(\nabla p_p - \gamma_p \nabla D) \quad (\text{A.7})$$

Substituyendo la ecuación (A.7) en la ecuación (A.6), se tiene lo siguiente:

$$\nabla \cdot \left(\frac{kk_{ro}}{B_o \mu_o} (\nabla p_o - \gamma_o \nabla D) \right) + \hat{q}_o = \frac{\partial}{\partial t} \left[\frac{\phi S_o}{B_o} \right] \quad (\text{A.8})$$

Nótese que el operador diferencial nabla, ∇ , se define en el sistema de coordenadas cartesianas como :

$$\nabla = \frac{\partial}{\partial x} i + \frac{\partial}{\partial y} j + \frac{\partial}{\partial z} k \quad (\text{A.9})$$

y el vector de velocidades de la fase $p = o, g, w$, v_p , es :

$$v_p = v_{px} i + v_{py} j + v_{pz} k \quad (\text{A.10})$$

Un tratamiento similar para la fase gas y agua conduce a lo siguiente:

$$\nabla \cdot \left[\frac{kk_{rg}}{B_g \mu_g} (\nabla p_g - \gamma_g \nabla D) \right] + \nabla \cdot \left[R_s \frac{kk_{ro}}{B_o \mu_o} (\nabla p_o - \gamma_o \nabla D) \right] + \hat{q}_g = \frac{\partial}{\partial t} \left(\frac{\phi S_g}{B_g} + R_s \frac{\phi S_o}{B_o} \right) \quad (\text{A.11})$$

$$\nabla \cdot \left(\frac{kk_{rw}}{B_w \mu_w} (\nabla p_w - \gamma_w \nabla D) \right) + \hat{q}_w = \frac{\partial}{\partial t} \left[\frac{\phi S_w}{B_w} \right] \quad (\text{A.12})$$

Las ecuaciones (A.8) y (A.12) describen el flujo multifásico en un yacimiento bajo condiciones isotérmicas y laminares.

APÉNDICE B. Subrutina de visualización gráfica (Código).

```

MODULE IDENTIFIER_MOD
IMPLICIT NONE
INCLUDE 'Arrays.inc'
INTEGER      :: IVIEW,IShade,NContour,IGrid,m,IEDITWIN,ALLO
INTEGER, DIMENSION(2) :: NVAL
INTEGER, DIMENSION(2) :: NVAL2
REAL        :: ELEV,ROTA
CHARACTER(LEN=255)  :: FNAME1
CHARACTER(LEN=255)  :: FNAME2
END MODULE IDENTIFIER_MOD
!
PROGRAM SIMBOIL
USE WINTERACTER
USE VARIABLES
USE IDENTIFIER_MOD
USE constants
USE my_library
USE input_data
IMPLICIT NONE
TYPE(WIN_MESSAGE)::MESS
INTEGER::ITYPE,I,status
INTEGER,DIMENSION(2) :: ISUB
INTEGER,DIMENSION(3) :: IHANDLE
CALL WInitialise()
CALL WindowOpen(FLAGS=SysMenuOn+MinButton+FixedSizeWin,MENUID=IDM_MENU1, &
                TITLE='Black Oil Simulator 3D',DIALOGID=DIALOG001)
CALL WDialogLoad(DIALOG002)
CALL WDialogLoad(AYUDA)
DO
  CALL WMessage(ITYPE,MESS)
  SELECT CASE(ITYPE)
    CASE(MenuSelect)
      SELECT CASE(MESS%VALUE1)
        CASE(MENNEW)
          CALL LIMPIAR()
        CASE(MENOPEN)
          CALL LIMPIAR()
        CASE(MENSAVE)
          CALL WMessageBox(3,3,1,'SAVE','GRAFICA')
        CASE(MENPRINT)
          CALL WMessageBox(3,3,1,'PRINT','GRAFICA')
        CASE(MENSALIR)
          EXIT
        CASE(MENSIM)
          CALL WMessageBox(3,3,1,'SIMULACION','GRAFICA')
        CASE(MENHELP)
          CALL WdialogSelect(AYUDA)
          CALL WDialogShow(ITYPE=Modal)
          CALL WDialogSelect(DIALOG001)
      END SELECT
    CASE (PushButton)
      SELECT CASE (MESS%VALUE1)
        CASE (IDCLOSE)
          EXIT
      CASE (BUTTON05)
        CALL WDialogSelect(DIALOG001)
        CALL WDialogGetInteger(ENTERO001,wpro)
        CALL WDialogGetInteger(ENTERO002,winj)
        CALL WDialogGetInteger(ENTERO003,nx)
        CALL WDialogGetInteger(ENTERO004,ny)
        CALL WDialogGetInteger(ENTERO005,nz)
        CALL WDialogGetInteger(ENTERO006,LAYERSLCTD)
        CALL WDialogGetInteger(ENTERO007,BLCKSLCTD)
        CALL WDialogGetDouble(REAL001,Lx)
        CALL WDialogGetDouble(REAL002,Ly)
        CALL WDialogGetDouble(REAL003,Lz)
        CALL WDialogGetDouble(REAL004,permx)
        CALL WDialogGetDouble(REAL005,permy)
        CALL WDialogGetDouble(REAL006,permz)
        CALL WDialogGetDouble(REAL007,vis)
        CALL WDialogGetDouble(REAL008,poro)
        CALL WDialogGetDouble(REAL009,Cf)
        CALL WDialogGetDouble(REAL010,Cr)
      END CASE
    END CASE
  END DO

```

```

CALL WDialogGetDouble(REAL011,Tiempo_t)
CALL WDialogGetDouble(REAL012,Prs_ref)
CALL WDialogGetDouble(REAL013,Prs_ini)
CALL WDialogGetDouble(REAL014,FVF)
CALL WDialogGetDouble(REAL015,API)
CALL WDialogSelect(DIALOG002)
CALL WGridRows(GRID01,wpro)
CALL WDialogShow(ITYPE=Modal)
IF (WInfoDialog(ExitButton)==BUTTON08)THEN
CALL WDialogSelect(DIALOG001)
IF (ALLO==1)DEALLOCATE(xp,yp,zp,rwp,qop,Tin_p,Tfi_p)
ALLO=0
ELSE
ALLOCATE(xp(wpro),yp(wpro),zp(wpro),rwp(wpro),qop(wpro),Tin_p(wpro),Tfi_p(wpro))
DO I=1,wpro
CALL WGridGetCellInteger(GRID01,1,I,yp(I))
CALL WGridGetCellInteger(GRID01,2,I,yp(I))
CALL WGridGetCellInteger(GRID01,3,I,zp(I))
CALL WGridGetCellDouble(GRID01,4,I,rwp(I))
CALL WGridGetCellDouble(GRID01,5,I,qop(I))
CALL WGridGetCellDouble(GRID01,6,I,Tin_p(I))
CALL WGridGetCellDouble(GRID01,7,I,Tfi_p(I))
ALLO=1
END DO
CALL SIM3D3F()
END IF
END SELECT
CASE(CloseRequest)
EXIT
END SELECT
END DO
CALL Windowclose()
STOP
CONTAINS!*****SUBROUTINAS EVENTOS*****
!
SUBROUTINE SIM3D3F()
INTEGER::ISTATE,IX,IY
!
! Open main (root) window, with:
! - Status bar non_enabled
! - Width & height 90% of full screen
! - Default position (i.e. centred)
! - menu
!
CALL WindowOpenChild(IEDITWIN,FLAGS =SysMenuOn+FixedSizeWin+StatusBar+OwnedByParent, &
WIDTH =INT(WInfoScreen(ScreenWidth) *0.99), &
HEIGHT=INT(WInfoScreen(ScreenHeight)*0.90), &
MENUID=IDR_MENU1,TITLE='RESERVOIR SIMULATION 3D-3F NUMERICAL SOLUTION')
CALL WindowOutStatusBar(1,"Click and drag the separator between sub-windows to resize")
CALL WDialogLoad(IDD_OPTIONS)
CALL WDialogLoad(IDD_TIME)
!
! Split vertically into sub-windows. (50/50 split)
! Split second sub-window horizontally into two further sub-windows. (Also 50/50 split)
! Use proportional resizing of sub-windows when main window is resized.
! Suppress Titles on sub-windows.
!
CALL WindowSplit(ISUB,ProportionalResize,(/NoTitle,NoTitle))
IHANDLE(1) = ISUB(1)
IHANDLE(2) = ISUB(2)
CALL WindowSelect(IHANDLE(2))
CALL WindowSplit(ISUB,SplitHoriz+ProportionalResize,(/NoTitle,NoTitle))
IHANDLE(2) = ISUB(1)
IHANDLE(3) = ISUB(2)
!
! Retrieve default options from dialog.
!
CALL SurfGetOptions()
CALL WindowSelect(IHANDLE(1))
CALL IGrAreaClear()
CALL WindowSelect(IHANDLE(2))
CALL IGrAreaClear()

CALL DRAW_BACKGROUND()

CALL WindowSelect(IHANDLE(3))
CALL IGrAreaClear()
!
! Enable field changed messages.
!

```

```

CALL WMessageEnable(FieldChanged,Enabled)
FNAME1 = ''
FNAME2 = ''
CALL SurfGetOptions()
CALL SurfPlot()
DO
  CALL WMessage(ITYPE,MESS)
  SELECT CASE(ITYPE)
    CASE(MenuSelect)
      SELECT CASE (MESS%VALUE1)
        CASE (ID_OPTIONS)
          CALL WDialogSelect(IDD_OPTIONS)
          IX = WInfoWindow(7) + 295
          IY = WInfoWindow(8) + 350
          CALL WDialogShow(IX,IY,0,Modal)
        CASE(ID_OPEN)
          DEALLOCATE(prs_old,prs_new,depth,Qo,bo,Vo,kx,ky,kz,fi,dfi,dbo,Coef_n,Coef_s,Coef_c, &
                    Coef_e,Coef_o,Coef_L,Coef_U,Coef_R,B,X,ia,ja,AREd,xi,yi,zi,rwi,qoi, &
                    Tin_i,TFi_i,pwf)
          ALLO=2
          CALL SurfGetOptions()
          CALL SurfPlot()
          ALLO=1
        CASE(ID_PRINT)
          CALL WindowSelect(IHANDLE(1))
          CALL IGrPrintImageOptions(ImageWidth,6*72)
          CALL IGrPrintImageOptions(ImageHeight,10*72)
          CALL IGrPrintImageOptions(Orientation,IntPortrait)
          CALL IGrPrintImage() ! Print 6"x10" portrait image
          CALL WindowSelect(IHANDLE(2))
          CALL IGrPrintImageOptions(ImageWidth,6*72)
          CALL IGrPrintImageOptions(ImageHeight,10*72)
          CALL IGrPrintImageOptions(Orientation,IntPortrait)
          CALL IGrPrintImage() ! Print 6"x10" portrait image
          CALL WindowSelect(IHANDLE(3))
          CALL IGrPrintImageOptions(ImageWidth,6*72)
          CALL IGrPrintImageOptions(ImageHeight,10*72)
          CALL IGrPrintImageOptions(Orientation,IntPortrait)
          CALL IGrPrintImage() ! Print 6"x4" portrait image
        CASE (PushButton)
          IF (MESS%VALUE1.EQ.IDCANCEL) THEN
            !
            ! If OK or Cancel then remove the dialog from the screen
            !
            CALL WDialogSelect(IDD_OPTIONS)
            CALL WDialogHide()
            CALL SurfPutOptions()
            END IF
            IF (MESS%VALUE1.EQ.IDOK) THEN
            !
            ! If OK or Cancel then remove the dialog from the screen
            !
            CALL WDialogSelect(IDD_OPTIONS)
            CALL WDialogHide()
            CALL SurfGetOptions()
            CALL SurfPlot()
            END IF
            IF (MESS%VALUE1.EQ.IDAPPLY) THEN
            !
            ! If OK or Apply then retrieve the entered options and
            ! re-draw the plot.
            IF (FNAME1.NE.' ' .AND.FNAME2.NE.' ') THEN
              CALL SurfGetOptions()
              CALL SurfPlot()
            END IF
            ELSE
            !
            ! If Cancel then restore the options to their previous values.
            !
            CALL SurfPutOptions()
            END IF
            !
            ! Expose or Resize - re-draw the plot.
            !
            ! CASE (Expose,Resize)
            ! CALL SurfPlot()
            !
            ! Field change - if ZSHADING checkbox toggles then enable/disable NCONTOURS
            ! field as appropriate
            !

```

```

CASE (FieldChanged)
  IF (MESS%VALUE2.EQ.IDF_SHADE3) THEN
    CALL WDialogSelect(IDD_OPTIONS_SHADING)
    CALL WDialogGetCheckBox(IDF_SHADE3,ISTATE)
    CALL WDialogFieldState(IDF_NCONTOURS,ISTATE)
  END IF
  END SELECT
  CASE (CloseRequest)
    EXIT
  END SELECT
END DO
DEALLOCATE(prs_old,prs_new,depth,Qo,bo,Vo,kx,ky,kz,fi,dfi,dbo,Coef_n,Coef_s,Coef_c, &
  Coef_e,Coef_o,Coef_L,Coef_U,Coef_R,B,X,ia,ja,ARED,xi,yi,zi,rwi,qoi, &
  Tin_i,TFi_i,pwf,yp,yp,zp,rwp,qop,Tin_p,Tfi_p)
CALL WindowCloseChild(MESS%WIN)
RETURN
END SUBROUTINE SIM3D3F
!
! SUBROUTINE DRAW_BACKGROUND()
!
!
! Set background colour
!
  CALL IGrInit(' ')
  CALL IGrFillPattern(Solid)
  CALL IGrColourN(8)
  CALL IGrRectangle(0.0,0.0,1.0,1.0)
  RETURN
END SUBROUTINE DRAW_BACKGROUND
!
SUBROUTINE LIMPIAR()
CALL WDialogSelect(DIALOG001)
CALL WDialogPutString(REAL001,"")
CALL WDialogPutString(REAL002,"")
CALL WDialogPutString(REAL003,"")
CALL WDialogPutString(REAL004,"")
CALL WDialogPutString(REAL005,"")
CALL WDialogPutString(REAL006,"")
CALL WDialogPutString(REAL007,"")
CALL WDialogPutString(REAL008,"")
CALL WDialogPutString(REAL009,"")
CALL WDialogPutString(REAL010,"")
CALL WDialogPutString(REAL011,"")
CALL WDialogPutString(REAL012,"")
CALL WDialogPutString(REAL013,"")
CALL WDialogPutString(REAL014,"")
CALL WDialogPutString(REAL015,"")
CALL WDialogPutString(ENTERO001,"")
CALL WDialogPutString(ENTERO002,"")
CALL WDialogPutString(ENTERO003,"")
CALL WDialogPutString(ENTERO004,"")
CALL WDialogPutString(ENTERO005,"")
CALL WDialogPutString(ENTERO006,"")
CALL WDialogPutString(ENTERO007,"")
RETURN
END SUBROUTINE LIMPIAR
!
SUBROUTINE SurfPutOptions()
!
! Store settings in Options dialog.
! SIZE sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_SIZE)
CALL WDialogPutOption(IDF_GRIDTYPE,IVIEW)
!
! VIEW sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_VIEW)
CALL WDialogPutReal(IDF_ELEVATION,ELEV)
CALL WDialogPutReal(IDF_ROTATION,ROTA)
CALL WDialogPutOption(IDF_GRID,IGrid)
!
! SHADING sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_SHADING)
!
CALL WDialogPutInteger (IDF_NCONTOURS,NContour)
CALL WDialogPutRadioButton(IDF_SHADE1+IShade-1)
!
RETURN

```

```

END SUBROUTINE SurfPutOptions
!
SUBROUTINE SurfGetOptions()
!
! Retrieve settings from Options Dialog
! SIZE sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_SIZE)
CALL WDialogGetMenu(IDF_GRIDTYPE,IVIEW)
!
! VIEW sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_VIEW)
!
CALL WDialogGetReal(IDF_ELEVATION,ELEV)
CALL WDialogGetReal(IDF_ROTATION ,ROTA)
CALL WDialogGetMenu(IDF_GRID ,IGrid)
!
! SHADING sub-dialog
!
CALL WDialogSelect(IDD_OPTIONS_SHADING)
CALL WDialogGetInteger(IDF_NCONTOURS,NContour)
CALL WDialogGetRadioButton(IDF_SHADE1, IShade)
RETURN
END SUBROUTINE SurfGetOptions
!
!
SUBROUTINE SurfPlot()
!
! Call Presentation Graphics routines to plot the surface
!
INTEGER :: NXDIM,NYDIM,IVALUE,IVALUE2,NSET,IC,ICOLSTEP,IX,IY,IXX,NPTS,IPT,LAYOUT,i,j,k,increment, &
STATUS,NGRIDX,NGRIDY
REAL :: NVALUE, NVALUE2,XMIN,YMIN,XMAX,YMAX,BOXSIZEX,BOXSIZEY
REAL, DIMENSION(10000) :: VALUES, VALUES2
REAL, ALLOCATABLE, DIMENSION(:,:) :: ZDATA
REAL, ALLOCATABLE, DIMENSION(:) :: XIPT,YIPT,ZIPT,XSCALE,YSCALE
REAL, ALLOCATABLE, DIMENSION(:,:) :: ZValue
REAL , DIMENSION(NContour-1) :: ZHeight
INTEGER, DIMENSION(3) :: ICOLI = (/8,10,12/)
REAL :: XTICKANGLE,YTICKANGLE,ZMIN,ZMAX,X0,Y0
REAL(KIND=8) :: TIMEACUM,wfp,wrate
CHARACTER(LEN=25) :: TIMEACUMRESULT
CHARACTER(LEN=25) :: wfps,wrates,BLCKSLCTDS,LAYERSLCTDS,CHR,LxS,LyS,LzS,nxS,nyS,nzS, &
wproS,winjS,Tiempo_tS,permxS,permyS,permzS,visS,APIS,CfS,CrS,poroS, &
FVFS,Prs_refS,Prs_iniS
!
! Generate surface data
!
!Setting dimensions
include "all_arrays.inc"
!Change of units to SI
IF (ALLO.NE.2)then
call Cambio_unidades
end if
NVAL(1)=nx
NVAL(2)=ny
NPTS = NVAL(1) * NVAL(2)
NXDIM = NPTS
NYDIM = NPTS
IVALUE = 0
IVALUE2 = 0
ALLOCATE(XIPT(NPTS),YIPT(NPTS),ZIPT(NPTS),ZDATA(NXDIM,NYDIM),ZValue(NXDIM,NYDIM),XSCALE(nx),YSCALE(ny))
DO I = 1,nx
XSCALE(I) = 1*REAL(I)
END DO
DO J = 1,ny
YSCALE(J) = 1*REAL(J)
END DO
!Some variables
ne=1

```



```

Nc=ne*nx*ny*nz

!This vector is used by NSPIV

max=Nc*Nc !I need to figure it out

! Dx y Dy are constants. The distans between node 1 and node Nx,Ny
! is Lx and Ly, respectivaly.
!
!<----- Lx ----->
!
!| | | | | | | | | |
!o | o | o | o | o | o | o | Dy |
!| | | | | | | | | |
!----- |
!| | | | | | | | | |
!o | o | o | o | o | o | o | Dy |
!| | | | | | | | | |
!----- Ly
!| | | | | | | | | |
!o | o | o | o | o | o | o | o |
!| | | | | | | | | |
!----- |
!| | | | | | | | | |
!o | o | o | o | o | o | o | o |
!| | | | | | | | | |
!----- |
!<--> <-->
! Dx Dx

! Setting Dx,Dy
CALL nodos

! Initial DT
DT=1.0 !Seconds

!Initial Conditions
CALL CIniciales

!Boundary Conditions
CALL CFrontera

! Accumulative Time
T_acum=0.0D0
CALL WindowSelect(IHANDLE(1))
CALL WDialogSelect(IDD_TIME)
IX = WInfoWindow(7) + 5
IY = WInfoWindow(8) + 350
CALL WDialogShow(IX,IY,0,SEMIMODELESS)

DO WHILE (T_acum <= Tiempo_t)
!
! Draw initial window contents
!
CALL WindowSelect(IHANDLE(1)) ! select window 1
!
! Clear graphics area
!
IF(T_acum == 0.0D0) CALL IGrAreaClear() ! clear window 1
!

call Coeficients

!call LU(A,B,nx*ny,L,U)
!call Forward(L,B,Y,nx*ny)
!call Backward(U,Y,X,nx*ny)
!call GAUSS(A,B,nx*ny,X)

call NSPIV(Nc,IA,JA,ARED,B,MAX,X,IERR,ne)

m=0
DO k=1,nz
DO j=1,ny
DO i=1,nx

```

```

m=m+1
  Prs_new(i,j,k)=X(m)
END DO
END DO
END DO

! Accumulative Time
T_acum=T_acum + DT ! En segundos
TIMEACUM = T_acum/86400.0D0
call results
!
CALL IDoubleToString(TIMEACUM,TIMEACUMRESULT,"(F12.5)")
CALL WDialogPutString(IDF_LABEL4,TIMEACUMRESULT)
wfp = pwf(BLCKSLCTD)/ pa_psi
CALL IDoubleToString(wfp,wfps,"(F12.5)")
CALL WDialogPutString(IDF_LABEL8,wfps)
wrate = Qop(BLCKSLCTD)*(sec_day*bl_m3)
CALL IDoubleToString(wrate,wrates,"(F12.5)")
CALL WDialogPutString(IDF_LABEL9,wrates)
CALL IntegerToString(BLCKSLCTD,BLCKSLCTDS,"(I4)")
CALL WDialogPutString(IDF_LABEL11,BLCKSLCTDS)

DO k=1,nz

  IF (k == LAYERSLCTD) then

    CALL IntegerToString(LAYERSLCTD,LAYERSLCTDS,"(I4)")
    CALL WDialogPutString(IDF_LABEL5,LAYERSLCTDS)

    DO IX = 1,NVAL(1)
      i=IX
      DO IY = 1,NVAL(2)
        j=IY
        ZVALUE(IX,IY) = prs_new(i,j,k)/pa_psi
      END DO
    END DO

  END IF

END DO
!
! Read data from a file assuming that the first line
! contains the number of points as an integer, with
! each subsequent line containing a real (x,y,z) triplet
!
DO k=1,nz

  IF (k == LAYERSLCTD) THEN

    increment = 0

    DO i = 1,nx

      DO j = 1,ny

        increment = increment + 1

        XIPT(increment) = i

        YIPT(increment) = j

        ZIPT(increment) = prs_new(i,j,k)/pa_psi

      END DO

    END DO

  END IF

END DO

!
! Determine min/max X/Y/Z values
!
CALL IRealMaxMin1(XIPT,NPTS,XMIN,XMAX)
CALL IRealMaxMin1(YIPT,NPTS,YMIN,YMAX)
CALL IRealMaxMin1(ZIPT,NPTS,ZMIN,ZMAX)
CALL IPgUnits(XMIN,YMIN,XMAX,YMAX)

```

```

CALL IPgUnitsZ(ZMIN,ZMAX)
!
! select plot type and style
!
LAYOUT = PgLay3D
IF (IVIEW=2) LAYOUT = LAYOUT + PgLayVariable
IF (IShade=1) THEN
!
! Uniform surface colour
!
NSET = 1
CALL IPgStyle(1,4,0,0,159,0)
CALL IPgStyle(2,4,0,0,31,0)
ELSE IF (IShade=2) THEN
!
! Height dependent, program-supplied colour contours
!
LAYOUT = LAYOUT + PgLayCont
NSET = NContour
DO IC = 1,NContour - 1
ZHeight(IC) = ZMIN + (ZMAX-ZMIN)*(REAL(IC)-0.5)/REAL(NContour)
END DO
ICOLSTEP = 255/NContour
DO IC = 1,NContour
CALL IPgStyle(IC,4,0,0,IC*ICOLSTEP,0)
END DO
ELSE
!
! Graduated shading
! (Note : For best results, this option requires a display
! which supports at least 16-bit colour)
!
LAYOUT = LAYOUT + PgLayContGrad
NSET = 1
CALL IPgStyle(1,4,0,0,20,0)
CALL IPgStyle(2,4,0,0,52,0)
END IF
!
! Request a 3D surface plot with height-dependent colours.
! Here we plot a 50x50 grid stored in a 100x100 array to illustrate
! that the whole of the array need not necessarily be used.
!
NGRIDX = nx
NGRIDY = ny
NVALZ(1) = NGRIDX
NVALZ(2) = NGRIDY
CALL IPgNewPlot(PgZarrayPlot,NSET,NVALZ,LAYOUT)
!
! Convert the (x,y,z) triplets to a z(m,n) array
! The 'search box' size is limited to a rectangle which
! is one tenth the width/height of the overall area to be
! plotted. Increasing this vale may give better interpolation,
! but at the expense of increased CPU time.
!
BOXSIZEX = (XMAX-XMIN)/10.0
BOXSIZEY = (YMAX-YMIN)/10.0
CALL IPgXYZSearchBox(BOXSIZEX,BOXSIZEY)
CALL IPgXYZToGrid(XIPT,YIPT,ZIPT,NPTS,ZDATA,NGRIDX,NGRIDY,NXDIM,NYDIM)
!
! Now plot the surface
!
! Define PG area and units
!
CALL IPgArea(.13,.58,.93,.98)
!
! define angle of elevation/rotation
!
CALL IPgElevation(ELEV)
CALL IPgRotation(ROTA)
!
! Modify x/y scale tick mark angles & lengths
!
SELECT CASE (MOD(INT((ROTA+45.)/90.0),4))
CASE (0) ! Angle = 315-44 degrees
XTICKANGLE = 315.0 - ROTA
YTICKANGLE = 45.0 - ROTA
CASE (1) ! Angle = 45-134 degrees
XTICKANGLE = ROTA - 135.
YTICKANGLE = ROTA - 45.
CASE (2) ! Angle = 135-224 degrees

```

```

XTICKANGLE = 135.0 - ROTA
YTICKANGLE = 225.0 - ROTA
CASE (3) ! Angle = 225-314 degrees
XTICKANGLE = ROTA - 315.
YTICKANGLE = ROTA - 225.
END SELECT
CALL IPgXScaleAngle(XTICKANGLE,0.0)
CALL IPgYScaleAngle(YTICKANGLE,0.0)
!
! add axes & scales and graticules
!
CALL IGrColourN(255)
CALL IPgStyle3DAxes(Solid,ICOL1)
CALL IPgAxes()
CALL IGrColourN(223)
IF (nx+1>10.OR.ny+1>10) THEN
CALL WGrTextFont(FFUser,0,0.0110,0.016)
ELSE
CALL WGrTextFont(FFUser,0,0.0147,0.016)
END IF
CALL WGrTextSubstitution(Disabled)
CALL IPgDecimalPlaces(-1)
CALL IPgXTickLength(RELLEN=0.4)
CALL IPgXUserScale(XSCALE,nx)
CALL IPgXScale(' ')
CALL IPgDecimalPlaces(-1)
CALL IPgYTickLength(RELLEN=0.4)
CALL IPgYUserScale(YSCALE,ny)
CALL IPgYScaleLeft(' ')
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextSubstitution(Disabled)
CALL IPgDecimalPlaces(2)
CALL IPgZTickLength(RELLEN=0.5)
CALL IPgZScale(' ')
!
CALL IPgGridDirection(IGrid-1)
CALL IPgSurf3DataCont(ZDATA,NXDIM,NYDIM,ZHeight)
!
CALL IOsWait(20)
!
IF (T_acum<Tiempo_t) THEN
DO k=1,nz
IF (k = LAYERSLCTD) THEN
CALL IGrColourN(0)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextSubstitution(Disabled)
CALL IPgDecimalPlaces(2)
CALL IPgZTickLength(RELLEN=0.5)
CALL IPgZScale(' ')
END IF
END DO
END IF
!
IF (IShade==1) THEN
!
! And a key by uniform surface colour
!
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0947,0.016)
CALL IPgKeyArea(0.0,0.0,0.0,0.0)
CALL IPgKeyAll(DESCR=('/',"/"),LAYOUT='B')!,TITLE=' ')
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextOrientation(AlignLeft,ANGLE=0.0)
CALL WGrTextString(0.94,0.80,'PRS')
!
ELSE IF (IShade==2) THEN
!
! And a key by height dependent, program-supplied colour contours
!
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0947,0.016)
CALL IPgKeyArea(0.0,0.0,0.0,0.0)

```

```

CALL IPgKeyAll(DESCR=(/','/),LAYOUT='BR'),TITLE=' ')
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextOrientation(AlignLeft,ANGLE=0.0)
CALL WGrTextString(0.86,0.99,' MAX')
CALL WGrTextString(0.86,0.56,' MIN')
!
ELSE
!
! And a key by graduated shading
!
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0947,0.016)
CALL IPgKeyArea(0,0,0,0,0,0)
CALL IPgKeyAll(DESCR=(/','/),LAYOUT='B'),TITLE=' ')
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextOrientation(AlignLeft,ANGLE=0.0)
CALL WGrTextString(0.86,0.99,' MAX')
CALL WGrTextString(0.86,0.56,' MIN')
!
END IF

! Draw label axes and main title

CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL IPgTitlePos(0.5)
CALL IPgTitle('PRESSURE BY LAYER','C')
CALL IPgZLabelPos(0.85)
CALL IPgZLabelLeft('PRESSURE (PSI)','RC')
CALL IPgXLabelPos(0.06)
CALL IPgYLabelPos(0.06)
CALL IPgXLabel('X-DIRECTIONS')!('X NODES')
CALL IPgYLabelLeft('Y-DIRECTIONS')!('Y NODES')
!
! Draw initial window contents
!
CALL WindowSelect(IHANDLE(2)) ! select window 2
!
call DRAW_BACKGROUND()
!
! Call IPgNewPlot to start a new plot
!
NVALUE = T_acum/sec_day
IVALUE = IVALUE + 1
VALUES(IVALUE) = pwf(BLCKSLCTD)/pa_psi
IF (Qop(BLCKSLCTD)<=0.0) THEN
YMIN = 0.999*pwf(BLCKSLCTD)/pa_psi
YMAX = 1.001*VALUES(1)
ELSE
YMAX = 1.001*pwf(BLCKSLCTD)/pa_psi
YMIN = 0.999*VALUES(1)
END IF
CALL IPgNewPlot(PgLinePlot,1,IVALUE,LAYOUT=PgLayStacked)
!
! Set style options
!
CALL IPgStyle(1,SolidLine,PgMarker,Outline,159,31)
CALL IPgMarker(1,14)
!
! Define PG area and units
!
CALL IPgArea(.12,.12,.98,.92)
!
! Set appropriate Presentation Graphics units for the plot
!
CALL IPgUnits(0.0,YMIN,NVALUE,YMAX)
!
! Draw a border around the Presentation Graphics area
!
CALL IPgBorder()
!
CALL IGrColourN(223)
CALL IPgAxes()
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.036)
CALL WGrTextSubstitution(Disabled)
CALL IPgDecimalPlaces(2)

```

```

CALL IPgXTickLength(RELLEN=0.9)
CALL IPgXUserScale(XSCALE,-1)
CALL IPgXGraticules(Dotted)
CALL IPgXScale(' ')
CALL IPgDecimalPlaces(-1)
CALL IPgYTickLength(RELLEN=0.5)
CALL IPgYUserScale(YSCALE,-1)
CALL IPgYGraticules(Dotted)
CALL IPgYScaleLeft(' ')
CALL IPgLinePlot(VALUES) ! draw line plot
CALL IPgTitlePos(0.5)
CALL IPgTitle('WELLBORE FLOW PRESSURE BEHAVIOR','C')
CALL IPgXLabelPos(0.80)
CALL IPgYLabelPos(0.80)
CALL IPgXLabel('TIME (DAYS)')
CALL IPgYLabelLeft('WELLBORE PRESSURE PSI')
CALL IPgKeyArea(0.0,0.0,0.0,0.0)
CALL IPgKeyAll(DESCR=('/',','),LAYOUT='E')

!
! Draw initial window contents
!
CALL WindowSelect(IHANDLE(3)) ! select window 3
!
CALL DRAW_BACKGROUND()
!
! Call IPgNewPlot to start a new plot
!
NVALUE2 = T_acum/sec_day
IVALUE2 = IVALUE2 + 1
VALUES2(IVALUE2) = ABS(Qop(BLCKSLCTD))*(sec_day*bl_m3)
YMIN = 0.999*ABS(Qop(BLCKSLCTD))*(sec_day*bl_m3)
YMAX = 1.001*VALUES2(1)
CALL IPgNewPlot(PgLinePlot,1,IVALUE2,LAYOUT=PgLayStacked)
!
! Set style options
!
CALL IPgStyle(1,SolidLine,PgMarker,Outline,31,159)
CALL IPgMarker(1,17)
!
! Define PG area and units
!
CALL IPgArea(.12,.12,.98,.92)
!
! Set appropriate Presentation Graphics units for the plot
!
CALL IPgUnits(0.0,YMIN,NVALUE2,YMAX)
!
! Draw a border around the Presentation Graphics area
!
CALL IPgBorder()
!
CALL IGrColourN(223)
CALL IPgAxes()
CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.036)
CALL WGrTextSubstitution(Disabled)
CALL IPgDecimalPlaces(2)
CALL IPgXTickLength(RELLEN=0.9)
CALL IPgXUserScale(XSCALE,-1)
CALL IPgXGraticules(Dotted)
CALL IPgXScale(' ')
CALL IPgDecimalPlaces(-1)
CALL IPgYTickLength(RELLEN=0.5)
CALL IPgYUserScale(YSCALE,-1)
CALL IPgYGraticules(Dotted)
CALL IPgYScaleLeft(' ')
CALL IPgLinePlot(VALUES2) ! draw line plot
CALL IPgTitlePos(0.5)
CALL IPgTitle('WELLBORE RATE BEHAVIOR','C')
CALL IPgXLabelPos(0.80)
CALL IPgYLabelPos(0.80)
CALL IPgXLabel('TIME (DAYS)')
CALL IPgYLabelLeft('WELLBORE RATE BBL/D')
CALL IPgKeyArea(0.0,0.0,0.0,0.0)
CALL IPgKeyAll(DESCR=('/',','),LAYOUT='E')

!New DT
DT = DT*1.5 ! En segundos

```

```

!New Dt, Now P @ n+1 becomes P @ n
do k=1,nz
do j=1,ny
do i=1,nx
prc_old(i,j,k)=prc_new(i,j,k)
end do
end do
end do

!Next DT y T_acum

END DO !Do while
!

CALL IDoubleToString(Lx/ft_m,LxS,"(F12.1)")
CALL IDoubleToString(Ly/ft_m,LyS,"(F12.1)")
CALL IDoubleToString(Lz/ft_m,LzS,"(F12.1)")
CALL IntegerToString(nx,nxS,"(I4)")
CALL IntegerToString(ny,nyS,"(I4)")
CALL IntegerToString(nz,nzS,"(I4)")
CALL IntegerToString(wpro,wproS,"(I4)")
CALL IntegerToString(winj,winjS,"(I4)")
CALL IDoubleToString(Tiempo_t/sec_day,Tiempo_tS,"(F12.2)")
CALL IDoubleToString(permx/mD_m2,permxS,"(F12.1)")
CALL IDoubleToString(permy/mD_m2,permyS,"(F12.1)")
CALL IDoubleToString(permz/mD_m2,permzS,"(F12.1)")
CALL IDoubleToString(vis/cp_pa_s,visS,"(F12.2)")
CALL IDoubleToString(API,APIS,"(F12.1)")
CALL IDoubleToString(Cr*pa_psi,CrS,"(E12.3)")
CALL IDoubleToString(Cr*pa_psi,CrS,"(E12.3)")
CALL IDoubleToString(poro,poroS,"(F12.3)")
CALL IDoubleToString(FVF,FVFS,"(F12.2)")
CALL IDoubleToString(Prs_ref/pa_psi,Prs_refS,"(F12.1)")
CALL IDoubleToString(Prs_ini/pa_psi,Prs_iniS,"(F12.1)")
CALL WDialogHide()

!
! Draw initial window contents
!
CALL WindowSelect(IHANDLE(1)) ! select window 1
!

CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextOrientation(AlignLeft,ANGLE=0.0)
CALL WGrTextString(0.1,0.06,'Length X (pie)')
CALL WGrTextString(0.1,0.08,'Length Y (pie)')
CALL WGrTextString(0.1,0.10,'Length Z (pie)')
CALL WGrTextString(0.1,0.12,'Nodes X')
CALL WGrTextString(0.1,0.14,'Nodes Y')
CALL WGrTextString(0.1,0.16,'Nodes Z')
CALL WGrTextString(0.1,0.18,'Production Wells')
CALL WGrTextString(0.1,0.20,'Injection Wells')
CALL WGrTextString(0.1,0.22,'Simulation Total Time (days)')
CALL WGrTextString(0.1,0.24,'Permeability en X (mD)')
CALL WGrTextString(0.1,0.26,'Permeability en Y (mD)')
CALL WGrTextString(0.1,0.28,'Permeability en Z (mD)')
CALL WGrTextString(0.1,0.30,'Viscosity (cp)')
CALL WGrTextString(0.1,0.32,'API Density')
CALL WGrTextString(0.1,0.34,'Fluid Compresibility (psi-1)')
CALL WGrTextString(0.1,0.36,'Rock Compresibility (psi-1)')
CALL WGrTextString(0.1,0.38,'Porosity (Fraction)')
CALL WGrTextString(0.1,0.40,'Base FVF (Pi)')
CALL WGrTextString(0.1,0.42,'Base Pressure (psi)')
CALL WGrTextString(0.1,0.44,'Initial Pressure of Reference (psi)')
CALL WGrTextString(0.1,0.46,'Selected Well')
CALL WGrTextString(0.1,0.48,'Selected Layer')


CALL IGrColourN(223)
CALL WGrTextFont(FFUser,0,0.0147,0.016)
CALL WGrTextOrientation(AlignRight,ANGLE=0.0)
CALL WGrTextString(0.7,0.06,LxS)
CALL WGrTextString(0.7,0.08,LyS)
CALL WGrTextString(0.7,0.10,LzS)
CALL WGrTextString(0.7,0.12,nxS)
CALL WGrTextString(0.7,0.14,nyS)
CALL WGrTextString(0.7,0.16,nzS)
CALL WGrTextString(0.7,0.18,wproS)
CALL WGrTextString(0.7,0.20,winjS)
CALL WGrTextString(0.7,0.22,Tiempo_tS)
CALL WGrTextString(0.7,0.24,permxS)


```


```
CALL WGrTextString(0.7,0.26,permyS)
CALL WGrTextString(0.7,0.28,permzS)
CALL WGrTextString(0.7,0.30,visS)
CALL WGrTextString(0.7,0.32,APIS)
CALL WGrTextString(0.7,0.34,CfS)
CALL WGrTextString(0.7,0.36,CrS)
CALL WGrTextString(0.7,0.38,poroS)
CALL WGrTextString(0.7,0.40,FVFS)
CALL WGrTextString(0.7,0.42,Prs_refS)
CALL WGrTextString(0.7,0.44,Prs_iniS)
CALL WGrTextString(0.7,0.46,BLCKSLCTDS)
CALL WGrTextString(0.7,0.48,LAYERSLCTDS)
RETURN
END SUBROUTINE SurfPlot
END PROGRAM SIMBOIL
```


APÉNDICE C. CD de aplicación.

El CD cuenta con:

- Software de aplicación (Simulador con Interfaz Gráfica). 

- Ejemplos de aplicación (3 casos). 

- Ejemplar de Tesis en formato PDF. 

- Acerca de dudas, sugerencias o comentarios, dirigirse a:



orlando_hernandez_robles@yahoo.com.mx