



UNIVERSIDAD DE BELGRANO

# Las tesinas de Belgrano

Facultad de Ingeniería y Tecnología Informática  
Ingeniería en Telecomunicaciones

Modelos numéricos estocásticos para hidrología

Nº 39

Martín Darío Cernadas

Tutor: Víctor Rodríguez Carro

Departamento de Investigación  
Agosto 2002



## Índice de Contenidos

Resumen Ejecutivo .....	9
1. Introducción a la Hidrogeología y la Simulación .....	10
1.1 Introducción a la Hidrogeología .....	10
1.2 Introducción a la Simulación .....	10
2. El Ciclo del Agua y el Agua Subterránea .....	11
2.1 El Ciclo del Agua .....	11
2.2 El Agua Subterránea dentro del Ciclo del Agua .....	11
2.3 ¿Dónde se halla el Agua Subterránea? .....	11
2.4 Agua Subterránea y Agua Superficial .....	12
2.5 Contaminación y Sobreexplotación : Amenazas para el Agua Subterránea .....	12
2.6 Causas de la contaminación del Agua Subterránea .....	13
2.7 Protección de las Aguas Subterráneas .....	14
3. Fundamentos teóricos y aplicación en el modelo numérico .....	14
3.1 Ley de Darcy .....	14
3.2 Equilibrio Hidráulico en Pozos .....	15
3.3 Desequilibrio Hidráulico en Pozos .....	15
3.4 Análisis de Acuíferos .....	16
3.5 Distribuciones Fractales para valores iniciales del modelo .....	16
Método de Lorenz .....	16
Método de Henon .....	17
Método de Sierpinski .....	18
Método de Monte Carlo .....	18
4. Enfoque económico .....	19
4.1 Diferencia entre el mundo real y simulado .....	19
4.2 Políticas Científico – Económicas .....	19
5. Herramientas empleadas en la Teoría .....	20
5.1 MODFLOW .....	20
5.2 PRICKETT .....	20
6. Caso de Evaluación en Laboratorio .....	20
6.1 Prickett estándar, inicialización uniforme de estados .....	22
6.2 Prickett utilizando el método de Lorenz para iniciación .....	22
6.3 Prickett utilizando el método de Henon para iniciación .....	23
6.4 Prickett utilizando el método de Sierpinski para iniciación .....	24
6.5 Prickett utilizando el método de Monte Carlo para iniciación .....	24
6.6 Correcciones al modelo de evaluación en laboratorio .....	25
6.7 Conclusiones para el caso de evaluación en laboratorio .....	25
7. Presentación del caso práctico de explotación para riego .....	26
7.1 Información general .....	26
7.2 Implementación del modelo .....	26
7.3 Correcciones al modelo para explotación para riego .....	27
7.4 Caso de Máxima explotación estimada .....	28
7.5 Conclusiones para el caso de explotación para riego .....	28
8. Anexos .....	29
8.1 Anexo 1 : Desarrollo de la teoría para el Modelo Prickett-Lonnquist .....	29
8.2 Anexo 2: Modelo Prickett-Lonnquist en lenguaje C/C++ .....	31
Definición .....	31
Implementación .....	33
8.3 Anexo 3: Algoritmos de inicialización del Modelo Prickett-Lonnquist .....	43
Método de Lorenz .....	44
Método de Henon .....	45
Método de Sierpinski .....	45
Método MonteCarlo .....	46
8.4 Anexo 4: Diálogo Principal de la aplicación modelo Prickett-Lonnquist con iniciación estocástica / fractal .....	48
MENU DE SELECCIÓN PARA EJECUTAR EL MODELO PRICKETT-LONQUIST .....	48
NOMENCLATURA DE PARAMETROS para 2 DIMENSIONES .....	49

---

NOMENCLATURA DE PARAMETROS INCLUYENDO METODOS ESTOCASTICOS .....	50
8.5 Anexo 5: Aplicación Comercial "Visual ModFlow" .....	50
Característica principales de Visual ModFlow .....	51
Entrada de Datos .....	51
Capacidades de Simulación .....	51
Calibración del Modelo .....	51
Visualización del Modelo .....	51
Aplicaciones profesionales de Visual ModFlow .....	52
9. Bibliografía y Referencias Profesionales .....	52
PROFESIONALES .....	52
LIBROS, TESIS, INFORMES y PORCIONES DE SOFTWARE .....	52
SITIOS EN INTERNET .....	53
LICENCIAS DE SOFTWARE Y AVISOS LEGALES .....	53

## Glosario

- **Acuífero:** Formación geológica subterránea con capacidad de almacenamiento de agua. / Capa, zona o veta del terreno que contiene agua.
- **Base de Datos Relacional:** Conjunto de datos y métodos para soporte de aplicaciones informáticas, compuesto por un sistema administrador de datos que cumple con los enunciados del álgebra relacional de base de datos, proveyendo funciones de definición y manipulación sobre aquellos, y un conjunto de datos de tipos predefinidos, que se mantienen grabados en forma estructurada. Generalmente compuesto por un motor de base de datos que implementa estructuras arbóreas, y herramientas de definición, consulta y manipulación de datos y relaciones entre sí, cada una con un lenguaje de interpretación específico (DDL, DML, SQL).
- **Caudales de bombeo constante** =  $(m^3/día) = \text{metros cúbicos} \times \text{día}$
- **Densidad:** Relación entre masa y volumen de un cuerpo.
- **Encamisado de tramo:** Sección de una perforación para explotación de agua subterránea a la que se le coloca un tubo de material rígido para reforzarla.
- **Ensayo de bombeo:** Prueba de extracción de agua mediante bombas en un pozo de explotación de agua subterránea, que determina su viabilidad dentro de ciertos parámetros de utilización.
- **Factor de almacenamiento (confinado)** =  $(m^3/m) = \text{metros cúbicos} \times \text{metro lineal}$
- **Fractales:** Figuras geométricas virtuales, formadas por un número infinito de elementos, infinitamente pequeños, contenidos en una superficie finita.
- **Grava:** Guijas, guijarros, arena y arcilla que se halla en yacimientos geológicos.
- **Hidrogeología:** Rama de la geología que estudia las aguas dulces, en particular la subterránea, y su aprovechamiento.
- **Litología:** Parte de la geología que trata de la composición de las rocas.
- **Medio anisótropo:** Medio que no posee direcciones predeterminadas o privilegiadas con respecto a alguna de sus propiedades. Con respecto a un acuífero, medio que no presenta direcciones de flujo en el estudio del movimiento del agua.
- **Método de diferencias finitas:** Método algorítmico consistente en la determinación de valores finales aproximados a partir de valores iniciales ciertos, aplicados a una función no diferenciable, mediante el cálculo en etapas ciertas o finitas, hasta alcanzar un error máximo aceptable, discretizando un área en matrices de 2 dimensiones.
- **Modelo matemático:** Abstracciones de representaciones del mundo real aplicadas al tratamiento predictivo, que discretizan áreas o cuerpos en 2 o 3 dimensiones respectivamente, aplicando funciones aproximadas del comportamiento de las propiedades que se quieren estudiar.
- **Niveles piezométricos:** profundidad desde la superficie del terreno hacia el primer nivel de agua subterránea. Asimismo aplicado a los niveles subsiguientes.
- **Permeabilidad:** Propiedad de ser penetrado por el agua, o de dejar salir la misma, con respecto a un acuífero y sus límites con otras formaciones geológicas.
- **Porosidad:** Propiedad de poroso, con respecto al tamaño de las partículas de las rocas de un acuífero, que indica el espacio libre entre las mismas y la viabilidad para almacenar agua.
- **Prefiltro de perforación:** Filtro antepuesto al filtro principal, en un pozo de explotación de agua subterránea, generalmente constituido de grava y gravilla.
- **Regresión no lineal:** Es una técnica para resolver las relaciones entre variables independientes (o de regresión) en la definición de un modelo, mediante ecuaciones no lineales. Estas variables no son aleatorias y por ende no tienen propiedades distribucionales.
- **Método de Gauss Newton:** Consiste en minimizar una función objetivo de potencias cuadradas, tal que la suma de los cuadrados de los residuos (o errores) sea mínima.
- **Transmisividad:** Propiedad de dejar fluir el agua, con respecto a un acuífero, entre 2 puntos cualesquiera  $(m^3/día/m) = \text{metros cúbico} \times \text{día} \times \text{metro lineal}$ .



## Dedicatoria

A mi esposa Roxana, y a mi hijo Nicolás, que son lo mejor de la vida.

## Agradecimientos

Al estimado profesor Ing. Víctor Rodríguez Carro, tutor de la presente tesina, que dedicó muchas horas y reuniones para las sucesivas correcciones que parecían no tener fin.





## Resumen Ejecutivo

En este trabajo se trata la aplicación de técnicas estocásticas en el campo de la simulación en hidrogeología para predecir niveles de agua subterránea a través del tiempo.

Para ello, se abordarán los conceptos de modelos numéricos tradicionales en el área en cuestión, y se planteará la necesidad de modificar los mismos cuando se carece de información cierta o valedera para su ejecución satisfactoria, o ajuste posterior.

Una vez desarrollados los conceptos básicos del tema, se realiza una introducción a los modelos numéricos tradicionalmente empleados en hidrogeología. Aquí se planteará el caso híbrido: sobre un modelo analítico y certero, se desarrollará una inicialización simulada, estocástica, sobre las variables desconocidas al momento inicial del desarrollo, y también en aquellos casos puntuales donde no tengan solución las ecuaciones diferenciales tradicionales.

Finalmente, y contrastando contra los resultados de un modelo analítico original, se verificará el modificado, y se concluirá sobre si es aplicable o no y en que medida.

Técnicamente hablando, y sobre los detalles del trabajo, el mismo genera como resultado de la investigación y desarrollo, un software de simulación para explotación de aguas subterráneas, empleando métodos estocásticos para la inicialización o generalización de las propiedades físicas en el momento de la corrida o ejecución del mismo.

Para una usabilidad del software mas eficiente, se desarrolló una interfase visual en 3 dimensiones para el sistema operativo Ms-Windows95/98/NT, y una serie de funciones de exportación de resultados, en formato de grilla y DXF de AutoCad. La visualización 3D incorpora un componente ActiveX de un laboratorio de software, el ESRI, dedicado justamente a mapeo y extrapolación de objetos en 3 dimensiones.

## Antecedentes de la Tesina

La inquietud sobre el tema surge ya hace algunos años en mi experiencia como becario del Programa Nacional de Desarrollo Sustentable en Aguas Subterráneas (PNDSAS), perteneciente al Instituto Nacional de Ciencias y Técnicas Hídricas (INCYTH), entre los años 1994 y 1996. Allí, como personal de apoyo a los geólogos, me introduje en la teoría que sustenta los modelos numéricos del área, y desarrollamos en conjunto una actualización de las herramientas en uso desde la década del '70, básicamente un software de simulación escrito en un lenguaje de programación ya fuera de uso, Fortran 77, el cual fue migrado por razones de compilación, mejor abstracción y de mas actualidad técnica, hacia el lenguaje orientado a objetos C / C++ . Es sobre la base de este trabajo que se extiende ahora el campo teórico, recurriendo a métodos estocásticos.

Luego de finalizado en su momento el desarrollo de programación, se adicionó al mismo una base de datos de perforaciones hídricas de toda la República Argentina, sobre las cuales el Instituto tenía registros y eran de interés específico o de representatividad de alguna cuenca o región hídrica. Con los registros así obtenidos en diversas campañas de recolección, se desarrolló una interfase visual con la herramienta Microsoft Access, y posteriormente se publicó en la World Wide Web para mayor difusión. Fue una tarea pionera en su área en el país, que reunió mas de 1500 perforaciones e informaciones conexas.

Este desarrollo no hubiese sido posible sin los profesionales geólogos del Programa, el Dr. Jorge Santa Cruz y el Dr. Adrián Silva, ambos referentes obligados en su área. A pesar de los recursos económicos, tan escasos que luego llevarían a discontinuar el proceso de desarrollo y focalizar al Programa en tareas de prospección y trabajos científicos, vale reconocer un esfuerzo que dio resultados mas que positivos y concretos, que hoy día se siguen aplicando para determinar, por ejemplo, la mejor ubicación y niveles de producción de nuevas perforaciones hídricas.

Ese trabajo resultó en un aporte concreto a la economía nacional, tanto en el sector público como privado. Y es una prueba de que la investigación teórica y la aplicación práctica en la ciencia y técnica de un país en vías de desarrollo, son necesarias y útiles, produciendo resultados palpables.

Esto vale mas allá de la perspectiva financista de caja chica que pudiera tener algún Ministro de Economía circunstancial, que siempre recorta los presupuestos, cuando no elimina completamente la educación y la investigación aplicada, relegando al país a ser un mercado de consumo de productos tecnológicos de avanzada de países dominantes.

# 1. Introducción a la Hidrogeología y la Simulación

## 1.1 Introducción a la Hidrogeología

Para comprender el problema que atacaremos, debemos comprender primero de qué estamos hablando. Y para ello, introduciremos conceptos y definiremos términos.

La **hidrogeología** es la ciencia que estudia las aguas dulces del planeta, especialmente las aguas subterráneas, y en sus diversas dimensiones, sean éstas tanto geológicas y físicas, como económicas e industriales. Muy frecuentemente ese estudio incluye la **litología**, o lo que es lo mismo, el estudio de las rocas subyacentes.

Un **acuífero** es una formación geológica subterránea con capacidad de almacenamiento de agua. Los acuíferos pueden ser **confinados**, en cuyo caso están rodeados por una formación geológica generalmente impermeable, y se comportan como una botella cerrada, como también pueden ser **libres**, en cuyo caso no están limitados por formaciones geológicas, ni poseen un techo, sino que el flujo de agua interno es libre, y posee contacto con el exterior, produciéndose evaporaciones por calor o incidencia del sol directamente, conocidas como **evapotranspiraciones**. Asimismo, otra suerte de acuífero es el **semiconfinado**, que es una mezcla de los dos anteriores.

El acuífero posee diversas propiedades físicas. La **permeabilidad** es una de ellas, y significa el poder ser penetrado por el agua, o de dejar salir la misma, con respecto a sus límites con otras formaciones geológicas.

Otra de sus propiedades físicas es ser un **medio anisótropo**, o sea, es un medio que no presenta direcciones de flujo en el estudio del movimiento del agua.

También se tiene en cuenta la **porosidad** con respecto al tamaño de las partículas de las rocas del acuífero, que indica el espacio libre entre las mismas y la viabilidad para almacenar agua. De ella surge el **factor de almacenamiento**, que para acuíferos **confinados** se mide en metros cúbicos por metro lineal.

Muy vinculada a ésta última se halla otra propiedad, la **transmisividad**: es la capacidad de dejar fluir el agua entre dos (2) puntos cualesquiera, que generalmente se mide en metros cúbicos por día por metro lineal, usando el sistema métrico.

Para hacer esos estudios, la herramienta principal de acceso al agua subterránea es la **perforación**. La misma cuenta con diversas etapas, desde la exploratoria, hasta la productiva comercialmente, pasando por el desarrollo in situ.

Una perforación cuenta con diversos elementos constitutivos:

Un **encamisado de tramo**, que es la sección a la que se le coloca un tubo de material rígido para reforzarla y evitar desmoronamientos internos. Un **filtro principal** para dejar pasar solo el agua y retener los materiales gruesos y finos del acuífero, hecho de una malla o similar, de acero inoxidable usualmente, más un **prefiltro de perforación**, que es el filtro antepuesto al filtro principal, generalmente constituido de guijas, guijarros, arena y arcilla. Y la **válvula principal** en la boca del pozo para abrir o cerrar el paso del agua. Suele ir acompañada de una **bomba de extracción** que aspira el agua subterránea cuando ésta no es capaz de surgir libremente al exterior, o si lo es, no en los caudales necesarios.

En la perforación se practican **ensayos de bombeo**, que son las pruebas de extracción de agua mediante bombas, que determina su viabilidad dentro de ciertos parámetros de utilización (tiempos y caudales máximos). De ellos se desprende la capacidad de recuperar sus **niveles freáticos**, o sea los niveles de las napas de agua, que es capaz un acuífero en determinado momento.

De la necesidad de predecir **niveles freáticos futuros** dado un caso de estudio comercial o industrial, sobre la base de parámetros de explotación, es que intervienen los **modelos matemáticos**, ya que no se podrían realizar ensayos de bombeo en forma continua, sin perjudicar al acuífero o a la inversión económica, puesto que los ensayos cuestan tiempo y dinero.

## 1.2 Introducción a la Simulación

La simulación es una técnica muy poderosa y ampliamente utilizada para analizar y estudiar sistemas complejos, como lo es el hidrogeológico. Se puede definir como la técnica que imita el funcionamiento de un sistema del mundo real cuando evoluciona en el tiempo.

Un **modelo de simulación** comúnmente toma la forma de un conjunto de hipótesis acerca del funcionamiento del sistema, expresado como relaciones matemáticas o lógicas entre los objetos de interés del sistema. En contraste con las soluciones matemáticas exactas disponibles en la mayoría de los modelos analíticos, el proceso de simulación incluye la ejecución del modelo a través del tiempo para generar muestras representativas de las mediciones del desempeño o funcionamiento. En este aspecto, se puede considerar a la simulación como un experimento de muestreo acerca del sistema real, cuyos resultados son puntos de muestra.

Es evidente que cuanto más puntos de muestra generemos, mejor será nuestra estimación. Sin embargo, hay otros factores que tienen influencia sobre la bondad de nuestra estimación final, como las condiciones iniciales de la simulación, la longitud del intervalo que se simula y la exactitud del modelo mismo.

¿Qué es un **modelo matemático**? Es una abstracción de representación del mundo real aplicadas al tratamiento predictivo, que discretizan áreas o cuerpos en 2 o 3 dimensiones respectivamente, aplicando funciones aproximadas del comportamiento de las propiedades que se quieren estudiar.

Un método de aplicación de modelos matemáticos en hidrogeología es el **método de diferencias finitas**, que consiste en la determinación de valores finales aproximados a partir de valores iniciales ciertos, aplicados a una función no diferenciable, mediante el cálculo en etapas finitas.

Es sobre este método y la forma de aplicar soluciones estocásticas en donde se centrará el presente trabajo.

## 2. El Ciclo del Agua y el Agua Subterránea

### 2.1 El Ciclo del Agua

El agua subterránea es el agua que se encuentra entre las partículas del suelo y las rocas y entre las grietas del lecho rocoso. Debido a su disponibilidad y buena calidad, en general, el agua subterránea es usada ampliamente para uso doméstico y otros propósitos.

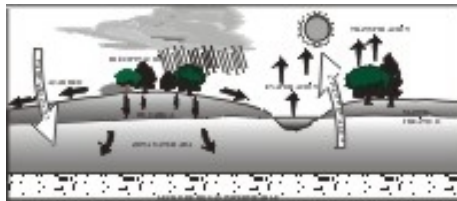


Figura 1. El ciclo del agua

### 2.2 El Agua Subterránea dentro del Ciclo del Agua

El agua subterránea es una parte integral del ciclo del agua (figura 1). El ciclo empieza con la precipitación sobre la superficie. Los *escurrimientos* de la lluvia van directamente a los lagos y arroyos. Algo del agua que se filtra en la tierra es usada por las plantas para la *transpiración*. El agua restante, llamada *agua de recarga*, es llevada a través del suelo a la *zona saturada*, donde el agua llena todos los espacios entre las partículas del suelo y las rocas.

Lo más alto de la zona saturada es la capa o manto freático que, si la geología local no es complicada, es el nivel al cual el agua se mantiene en un pozo. El agua se mueve dentro de la zona saturada desde áreas donde la capa de agua es alta hacia áreas donde ésta es baja, por lo que el agua subterránea se transforma en lagos, arroyos u océanos. Esta sale del subsuelo y forma el *agua superficial*, cuando esta agua se *evapora* a la atmósfera y se condensa, viene la *precipitación* completando el ciclo del agua.

### 2.3 ¿Dónde se halla el Agua Subterránea?

El agua puede ser encontrada debajo de la tierra casi en cualquier sitio. Cerca del 97 por ciento del agua dulce del mundo es agua subterránea. La calidad y cantidad del agua subterránea disponible varía de sitio a sitio. Las reservas mayores de agua subterránea son llamadas acuíferos.

Los acuíferos ocurren en dos tipos de formaciones geológicas. Las *formaciones consolidadas* son aquellas compuestas de rocas sólidas, donde el agua subterránea se encuentra en las grietas que estas poseen. La cantidad de agua en una formación consolidada depende de la cantidad de grietas que existen y del tamaño de éstas. Por ejemplo, las formaciones calizas frecuentemente contienen cavernas con mucha agua en su interior.

Las *formaciones no consolidadas* están compuestas de arena, grava, piedras, tierra suelta o material de suelos. La cantidad de agua subterránea en una formación consolidada varía en dependencia de la compactación del material sólido y la finura de sus granos. Las formaciones de arena, grava, y piedras generalmente producen acuíferos de gran capacidad, sin embargo, los suelos formados por partículas muy finas suelen tener bajas cantidades de agua.

El agua subterránea puede salir espontáneamente formando manantiales o puede ser extraída a través de un pozo. Un manantial ocurre cuando la capa de agua alcanza la superficie de la tierra (figura 2).

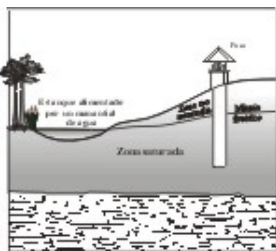


Figura 2. Relación de la capa freática.

## 2.4 Agua Subterránea y Agua Superficial

La mayoría de las personas está más familiarizada con el agua superficial que con el agua subterránea. Los depósitos de agua superficial como lagos, arroyos y mares pueden ser vistos en los alrededores, pero no así los depósitos de agua subterránea. Existen algunas diferencias entre el agua subterránea y el agua superficial que vale la pena aclarar.

El agua subterránea usualmente se mueve mucho más lentamente que el agua superficial. El agua en un arroyo puede moverse a varios metros por minuto, mientras que el agua subterránea puede moverse sólo a varios metros por mes. Esto es debido a que el agua subterránea encuentra una fricción o resistencia mayor al moverse a través de los espacios pequeños de las rocas y del suelo. Existen excepciones a esta regla, un ejemplo son los ríos subterráneos en cavernas de roca caliza donde el agua puede moverse relativamente más rápido.

Los intercambios de agua entre los depósitos de agua superficial y los acuíferos son importantes. Los ríos usualmente empiezan como pequeños arroyos y aumentan el caudal a medida que fluyen hacia el mar. El agua que ellos ganan frecuentemente viene del agua subterránea. Esta corriente se denomina *corriente ganante*. Es posible que las corrientes viertan agua al subsuelo en algunos puntos. En estos casos, los acuíferos son rellenados o *recargados* por agua de *corrientes de pérdida*. Un arroyo que fluye cerca de la superficie de un acuífero perderá agua hacia el acuífero si la superficie de agua del arroyo es más alta que la capa de agua del acuífero en la tierra adyacente. (figuras 3a y 3b)

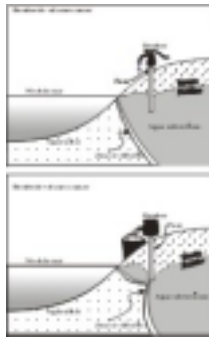
## 2.5 Contaminación y Sobreexplotación : Amenazas para el Agua Subterránea

El agua subterránea se *contamina* cuando algunas sustancias tóxicas se disuelven en el agua superficial y son acarreadas o lixiviadas a acuíferos con el *agua percolada*. Se deben considerar las propiedades y cantidades de las sustancias tóxicas y del suelo encima del acuífero para determinar si una sustancia en particular contaminará a un acuífero específico.

Algunas veces la contaminación del agua subterránea ocurre en forma natural, pero la contaminación aguda es usualmente el resultado de las actividades humanas en la superficie de la tierra. Un acuífero provee una gran cantidad de agua que frecuentemente atrae a una gran cantidad de personas en sus alrededores.

El agua es usada en actividades tales como beber, higiene personal, mantenimiento residencial y con propósitos industriales o agrícolas. Muchas de estas actividades involucran el uso y desecho de productos químicos que son potencialmente contaminantes. Cuando estos productos químicos son usados o desechados en forma incorrecta y en cantidades inaceptables, pueden llegar al agua subterránea y contaminarla.

Debido a que el agua subterránea se mueve lentamente, pueden pasar varios años antes de que un contaminante, liberado en la superficie de la tierra encima del acuífero, sea detectado en el agua del acuífero a cierta distancia del sitio de contaminación. Desafortunadamente, esto significa que la contaminación ocurre antes de que sea detectada. Aun si se ha detenido la liberación del contaminante, pueden pasar varios años antes de que el acuífero se purifique en forma natural.



Aunque el agua puede ser tratada para remover los contaminantes, esto puede ser muy costoso. La mejor protección contra la contaminación es la prevención.

El agua subterránea se *agota* en un área cuando se consume más agua de la que ingresa al acuífero. Esto causa que el manto freático disminuya, por lo que el agua subterránea se hace más difícil y cara de extraer. Puede suceder que se bombee una gran cantidad de agua de un acuífero pequeño y esto cause que el pozo del vecino se seque (el manto freático queda por debajo del nivel del pozo). La rápida expansión de las áreas urbanas resulta en el sobreuso de los suministros del agua subterránea y provoca su agotamiento y contaminación. En las áreas costeras, el sobrebombeo crónico puede causar la *intrusión de agua salada* ( figura 4) . La intrusión de agua salada ocurre en las zonas costeras donde la extracción excesiva de agua dulce permite que el agua salada del mar se introduzca dentro del acuífero.

## 2.6 Causas de la contaminación del Agua Subterránea

Muchas actividades humanas en la superficie de la tierra causan cambios en la calidad del acuífero. La importancia del efecto de una actividad en particular está relacionada con la habilidad del suelo y del sistema hídrico subterráneo de degradar o diluir los contaminantes así como del grado en que los contaminantes interfieran con el uso del agua. La contaminación es más aguda en las reservas de agua potable que en el agua para otros usos.

Todos los contaminantes del agua subterránea entran al acuífero, esencialmente, a través del agua de recarga de la superficie, con excepción de aquellos casos en que el agua contaminada es inyectada directamente en el acuífero. Algunos ejemplos de contaminantes son:

- Productos químicos orgánicos sintéticos (pesticidas y productos del petróleo)
- Metales pesados como el mercurio, arsénico, cadmio, cromo y plomo
- Nitratos
- Bacterias y virus
- Residuos del petróleo y productos de la combustión de motores

Estos son considerados dañinos si se ingieren con el agua potable, y pueden ser acarreados al agua superficial por el agua subterránea.

Cada actividad humana tiene un aspecto particular en el agua subterránea. Algunas actividades agrícolas añaden nitratos y pesticidas al agua subterránea.

Las áreas residenciales con tanques sépticos usualment adicionan nitratos, bacterias, virus y productos orgánicos sintéticos de limpieza usados en la limpieza casera y limpiadores de tanques sépticos.

Las actividades industriales tienden a añadir productos químicos orgánicos y metales en variadas cantidades. Las áreas de almacenamiento de gasolina (incluyendo las estaciones de servicio) pueden tener fugas y derramar los productos derivados del petróleo.

Las carreteras contribuyen a la contaminación debido al petróleo que es derramado por los vehículos y los metales que salen del escape. El impacto mayor viene de los basureros antiguos cubiertos cuyas filtraciones pueden contener diferentes sustancias químicas en una concentración relativamente alta ( Figura 5) .

## 2.7 Protección de las Aguas Subterráneas

La protección contra la contaminación de nuestra agua subterránea requerirá un manejo conciente y la cooperación por parte de los ciudadanos y de varias instancias gubernamentales. En varios casos, la planificación del uso del suelo es la mejor medida disponible para proteger los acuíferos que aún contienen agua de buena calidad. Si se planifica la localización de fuentes potenciales de contaminación y se les ubica lejos de las áreas críticas de recarga, el riesgo de contaminación se reducirá notablemente.

El uso cuidadoso y la eliminación apropiada de los productos químicos que causan contaminación son también necesarios. Las industrias, las granjas y los vecinos asentados encima de las reservas de aguas subterráneas necesitan practicar un buen manejo con respecto al uso y eliminación de productos químicos.

Los reglamentos gubernamentales para el uso y eliminación de materiales tóxicos tienen que cumplirse. Un paso igualmente importante es hacer que las personas estén concientes del impacto potencial que ellos pudieran tener en el agua subterránea.

## 3. Fundamentos teóricos y aplicación en el modelo numérico

### 3.1 Ley de Darcy

En el año 1856, Darcy confirmó la aplicabilidad de los principios de tubos capilares, desarrollados años antes por Hagen y Poiseuille, al flujo de agua a través de medios porosos. La llamada Ley de Darcy se expresa como :

$$v = k s$$

siendo  $v$  la velocidad del flujo,  $s$  la pendiente del gradiente hidráulico y  $k$  es un coeficiente que tiene unidades de velocidad, generalmente metros por día. El caudal,  $q$ , es el producto del área  $A$  de la sección transversal y la velocidad. El área efectiva es el área total multiplicado por la porosidad del medio,  $p$ . Por lo tanto,

$$q = kpAs = K(p)As$$

El coeficiente  $K(p)$  se denomina *coeficiente de permeabilidad* o conductividad hidráulica. Depende las propiedades del fluido y del medio poroso y se puede expresar como:

$$K(p) = k(w/u) = Cd^2(w/u)$$

donde  $k$  es la permeabilidad intrínseca,  $w$  es el peso específico del fluido,  $u$  es la viscosidad absoluta,  $C$  es un factor que describe la forma, porosidad y otras características del medio, y  $d$  es el tamaño promedio de los poros del material.

Es conveniente utilizar la transmisividad  $T$  para representar el flujo, en metros cúbicos por día, a través de una sección de 1 metro y una altura igual al espesor del acuífero bajo la acción de un gradiente unitario:

$$T = K(p)Y$$

donde  $Y$  es el espesor saturado del acuífero. Con este nuevo coeficiente, la ecuación pasa a ser

$$q = TBs$$

donde  $B$  es el ancho del acuífero.

Esta Ley de Darcy (Ley del Movimiento de Fluido) se aplica en modelos numéricos para abstraer y modelar el comportamiento de acuíferos.

### 3.2 Equilibrio Hidráulico en Pozos

Las perforaciones en el terreno, tubulares y con una bomba mecánica en la mayoría de los casos, y que penetran perpendicularmente al manto freático y son de utilidad hogareña, comercial o industrial, se conocen como pozos de extracción, o simplemente pozos.

Para que se establezca un flujo de agua hacia el pozo, es necesario que exista un gradiente hidráulico en la dirección del pozo. La forma inconfundida resultante se denomina *cono de depresión*. Si el abatimiento o descenso de la capa freática es pequeño con respecto al espesor del acuífero, se puede deducir una fórmula aproximada que relacione el caudal del pozo con las características del acuífero .

$$q = 2\pi xyK(p)(dy/dx)$$

donde  $2\pi xy$  es el área del cilindro y  $dy$  es la pendiente de la tabla de agua. Integrando con respecto a  $x$ , distancia a la perforación desde dos puntos de observación,  $r_1$  y  $r_2$ , y desde la base del acuífero hasta el nivel freático medido en esos puntos de observación,  $h_1$  y  $h_2$ , se obtiene

$$q = \frac{2K(p)(h_1^2 - h_2^2)}{\ln(r_1/r_2)}$$

donde  $h$  es la altura de la tabla de agua sobre la base del acuífero, a una distancia  $r$  del eje del pozo de bombeo y  $\ln$  es el logaritmo de base  $e$ . Ésta es la ecuación propuesta por Dupuit en el año 1863, justamente para medir el caudal (\*).

### 3.3 Desequilibrio Hidráulico en Pozos

Existe una sobreestimación del caudal hipotético del pozo aplicando la ecuación de Dupuit. Esto se debe a que durante el período inicial de bombeo de un pozo nuevo, la mayoría del caudal se obtiene del almacenamiento contenido en la parte del acuífero que se deseca a medida que se desarrolla el cono de depresión. Y el análisis en condiciones de equilibrio producen valores muy altos de la permeabilidad, cuando se calcula el caudal potencial.

Para corregir eso, en 1935, Theis, presentó una fórmula basada en una analogía de transmisión del calor, que tiene en cuenta el efecto del tiempo y las características de almacenamiento del acuífero.

$$W(u) = Z(r) = \frac{q}{4T} \int (e^{-u}/u) du$$

donde  $Z(r)$  es el abatimiento o depresión de un pozo de observación a una distancia  $r$  del pozo de bombeo,  $q$  el caudal en metros cúbicos por día,  $T$  es la transmisividad en metros cúbicos por día por metro, y  $u$  está dada por

$$u = \frac{r^2 S(c)}{4Tt}$$

(\*) LINSLEY, KOHLER y PAULUS, "Hidrología para Ingenieros", 2da. Edición, McGraw-Hill de México S.A., México, 1977.

donde  $t$  es el tiempo en días desde el inicio del bombeo, y  $S(c)$  es la constante de almacenamiento del acuífero o el volumen de agua desplazada en una columna del acuífero de 1 (un) metro cuadrado, cuando la superficie piezométrica desciende 1 metro. La integral de la función  $W(u)$  o *Función del Pozo*, puede resolverse por series (\*):

$$W(u) = -0.5772 - \ln(u) + u - u^2/2.2! + u^3/3.3! \dots$$

### 3.4 Análisis de Acuíferos

Los procedimientos de Dupuit, Theis y demás son valederos para pozos individuales. Cuando se quiere estudiar un acuífero, se debe recurrir a técnicas más sofisticadas.

Si el estudio requerido implica analizar el acuífero en forma bidimensional, se puede recurrir a un aparato de Hele-Shaw, que consta de dos láminas de vidrio y una separación muy pequeña, con un líquido viscoso entre ambas láminas. Las ecuaciones que gobiernan el flujo viscoso son las mismas que para agua subterránea.

Cuando se requiere el estudio tridimensional de los acuíferos, es solo posible con el uso de modelos analógicos o digitales (computadores).

Cuando se plantea un modelo analógico, se fabrica una red de resistores y capacitores en el cual el voltaje es análogo al potencial del flujo y el caudal es análogo a la corriente eléctrica; la permeabilidad se simula por medio de la resistividad.

Para el modelo en computadores digitales modernos, se recurre a la solución de la ecuación básica de flujo de aguas subterráneas:

$$\frac{d^2h}{dx^2} + \frac{d^2h}{dy^2} = \frac{S(c)dh}{T dt}$$

En términos de diferencias finitas, que es la forma de representar esta ecuación para un modelo numérico digital, la misma es:

$$\frac{h_2 + h_4 - 2h_1}{a^2} + \frac{h_3 + h_5 - 2h_1}{a^2} = \frac{h_2 + h_4 + h_3 + h_5 - 4h_1}{a^2} = \frac{S(c) h}{T dT}$$

donde  $a$  es la separación de las celdas de la red. Los tamaños de la red van desde los 30 a los 3.000 metros de acuerdo con la naturaleza del problema. Las propiedades del acuífero se deben suministrar en cada punto de la red (espesor, permeabilidad, y el coeficiente de almacenamiento). Si estas propiedades no se definen con precisión, la solución obtenida estará errada. (\*)

### 3.5 Distribuciones Fractales para valores iniciales del modelo

La Teoría del Caos hace su aporte a la simulación en el campo de la hidrogeología, al aportar patrones de distribución de aportes de agua (lluvias o también sequías), en forma no uniforme, a lo largo y ancho del área que se desea simular.

Es muy conveniente contar con una distribución de este tipo al iniciar un modelo de simulación, para no observar patrones constantes ni uniformes que luego jamás son verificados en la realidad del campo de trabajo.

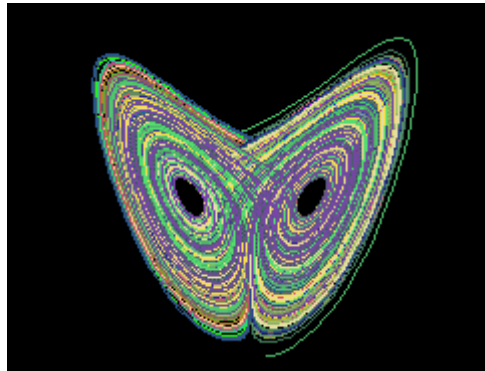
El aporte significativo se hace con tres tipos de métodos que figuran otros tantos tipos de superficies u obstáculos en el terreno: Lorenz, Henon y Sierpinski.

#### Método de Lorenz

El método de Lorenz inicializa un volumen y/o superficie sobre la base de tres ecuaciones con polinomios, siguiendo un patrón de generación de tipo fractal.

(\*) LINSLEY y otros, Op. Cit.





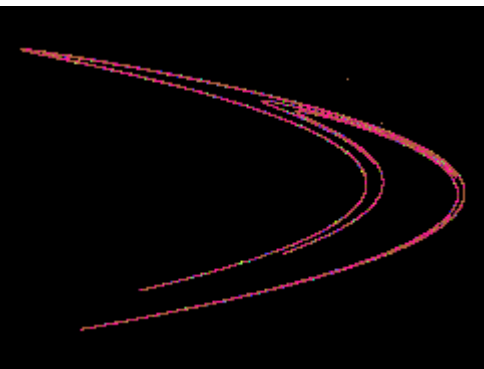
El "Atractor de Lorenz» es un conjunto simple de tres ecuaciones deterministas desarrolladas por Eduard Lorenz mientras estudiaba patrones climáticos no repetitivos. El problema básico en la predicción del clima es que ante cambios pequeñísimos en las condiciones iniciales, eventualmente reduce la mejor predicción a nada.

El atractor de Lorenz es la órbita de un sistema dinámico consistente en tres ecuaciones diferenciales de primer orden. La solución a la ecuación diferencial es una función de parámetro vectorial de una variable. Si se piensa en términos de tiempo, la solución traza una órbita, la cual está dibujada de dos espirales a cierto ángulo una de otra en tres dimensiones.

Las ecuaciones son :

$$\begin{aligned} dx/dt &= -a*x + a*y \\ dy/dt &= b*x - y -z*x \\ dz/dt &= -c*z + x*y \end{aligned}$$

Este método representa muy bien las distribuciones iniciales ( de lluvia por ejemplo) en un terreno donde existen por ejemplo lagunas o montes enfrentados.



...aliza una superficie sobre la base de dos ecuaciones de polinomios, siguiendo po fractal.

Michel Henon fue un astrónomo en el observatorio de Niza, en Francia. Él llegó al tema de los fractales a través de investigaciones en las órbitas de objetos astronómicos. El atractor extraño, nombrado en honor a Henon, no se inicia en ecuaciones diferenciales, sino en el mundo de las matemáticas discretas.

El mapa de Henon es un ejemplo de un sistema simple que exhibe un comportamiento extraño. La órbita se traza con la característica de contorno de una banana, pero si se inspecciona mas y más detalladamente, está hecho de partes gruesas y finas. A través de un detalle mayor, las partes gruesas se resuelven en otros componentes gruesos y finos. Y así indefinidamente.

$$\begin{aligned} dimX &= 1 + y - a*x*x \\ dimY &= b*x \end{aligned}$$

Los parámetros por omisión son  $a=1.4$  y  $b=0.3$ .

### Método de Sierpinski

El método de Sierpinski inicializa una superficie sobre la base de dos ecuaciones de polinomios, inicializados en forma aleatoria en arreglos vectoriales, y siguiendo un patrón de generación de tipo fractal.

Es otro modelo clásico previo al clásico Mandelbrot, y fue hallado por W. Sierpinski en los tiempos de la Primera Guerra Mundial. Es generado a través de la división de un triángulo en cuatro triángulos más pequeños y congruentes, repitiéndolo lo mismo una y otra vez, hasta el infinito.

No hay parámetros para ejecutar este patrón, ni ecuaciones complicadas.

### Método de Monte Carlo

La simulación de **Monte Carlo** fue creada para resolver integrales que no se pueden resolver por métodos analíticos: para resolver estas integrales se usaron números aleatorios. Posteriormente se utilizó para cualquier esquema que emplee números aleatorios, usando variables aleatorias con distribuciones de probabilidad conocidas, el cual es usado para resolver ciertos problemas estocásticos.

Por lo tanto es un proceso computacional que utiliza números aleatorios para derivar una salida, por lo que en vez de tener entradas con puntos dados, se asignan distribuciones de probabilidad a alguna o todas las variables de entrada. Esto generará una distribución de probabilidad para una salida después de una corrida de la simulación.

### Muestreo de Monte Carlo

El principio para hacer un muestreo en el caso de distribuciones discretas se basa en la representación de las frecuencias según su ley de probabilidad, que implica que los resultados se presentaran con las frecuencias especificadas por la distribución de probabilidades. Pero además de obtener las frecuencias correctas, el procedimiento de muestreo debe ser independiente; esto es, cada evento que se genera debe ser independiente de los eventos que le anteceden y que le siguen.

Una sucesión de números enteros aleatorios es tal que cada número en la secuencia tiene la misma probabilidad de ocurrir, y cada número aleatorio es independiente de los números anteriores y posteriores de él. Técnicamente, un número aleatorio,  $R_i$  se define como una muestra aleatoria independiente tomada de una distribución continua uniforme cuya función de densidad de probabilidad esta dada por

$$F(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Así, cada número aleatorio estará distribuido uniformemente sobre el intervalo entre 0 y 1. En consecuencia, es común referirse a estos números como números aleatorios  $U(0,1)$ , o simplemente como números aleatorios uniformes (equiprobables).

**Se pueden generar números aleatorios uniformes de muchos modos distintos. Como nuestro interés sobre los números aleatorios es para usarlos en simulación, necesitamos poder generarlos en una computadora. Esto se hace mediante funciones matemáticas llamadas generadores de números aleatorios. Algunos de esos generadores comprenden el generador congruente lineal, el generador multiplicativo y el generador mixto.**

Además de las propiedades estadísticas, los generadores de números aleatorios deben tener otras características importantes si se van a usar en forma eficaz en simulaciones con computadora. Algunas de estas características son:

- La rutina debe ser rápida
- La rutina no debe necesitar un gran espacio de almacenamiento.
- Los números aleatorios deben ser reproducibles
- La rutina debe tener un ciclo suficientemente largo; esto es, debemos poder generar una sucesión larga sin repetir los números aleatorios.  
El procedimiento para generar números aleatorios consta de dos pasos:
- Se elabora la distribución de probabilidad acumulada para la variable aleatoria dada.
- Se usa esa distribución para asignar los números aleatorios enteros en forma directa a los diversos valores de la variable aleatoria.

## 4. Enfoque económico

La hidrogeología contiene un componente económico fundamental, ya que realiza investigación y provee información sobre el agua del subsuelo, su composición, calidad, disponibilidad, magnitud de reserva, y algo fundamental, la viabilidad de explotación de esas aguas, que en términos económicos, significa evaluar si su extracción es posible, cómo, cuándo y dónde.

El cómo implica calcular las magnitudes de volúmenes posibles de agua a extraer, sin afectar en el futuro esa explotación. Se conoce como explotación sustentable. Es donde los modelos numéricos entran a jugar, y es donde también su eficacia (certeza) en la predicción se hace fundamental.

El cuándo implica determinar períodos de espera si se ha saturado la explotación, si se ha agotado, si el cono de depresión formado por la extracción exagerada es muy pronunciado, y demás. Está muy relacionado con el punto anterior, del cómo. Y depende también mucho, de cómo se planifica el modelo numérico a través del tiempo.

El dónde implica realizar perforaciones del suelo y analizar sus componentes. Se utilizan métodos analíticos y técnicas geológicas modernas.

### 4.1 Diferencia entre el mundo real y el simulado

Los modelos son abstracciones de la realidad, nunca son la realidad que representan, y por ello tienen errores al intentar simularla.

Siempre existen desviaciones entre lo simulado y lo real, que se corrigen a medida que las simulaciones avanzan y se contrastan contra lo real, ajustando a posteriori los datos, y volviendo a ejecutar las simulaciones. Es un proceso de retroalimentación, sin el cual, sería muy imprudente tomar como certeras las predicciones realizadas.

Para ello, el trabajo del hidrogeólogo en el campo o en el laboratorio, es disciplinado. Cada simulación se verifica y corrige. Para ello, dispone de diversos modelos numéricos, de los cuales, los más aplicados, Prickett-Lonnquist y ModFlow, se presentan más adelante. De ellos, el modelo Prickett-Lonnquist, es el más utilizado y difundido en nuestro país, y sobre el cual el Instituto de Aguas, a través del Programa Nacional de Desarrollo Sustentable en Aguas Subterráneas, ha desarrollado las políticas científicas y económicas tendientes a su mayor aplicación.

### 4.2 Políticas Científico – Económicas

Dentro de las políticas científico-económicas, se encuentran el relevamiento en el ámbito nacional de todas las perforaciones de explotación de aguas subterráneas, su compilación y publicación como banco de datos, como también el dictado de cátedras en la Universidad de Buenos Aires, y presentaciones en Ferias del Agro, donde se publicita y explica el uso de los modelos numéricos.

Como necesidad de contar con modelos numéricos más certeros y aproximados a lo real, y por ende, económicamente más ajustados, se plantea el ajuste de los tradicionales modelos numéricos de diferencias finitas, usando técnicas estocásticas.

¿Porqué técnicas estocásticas y fractales? La respuesta está en los datos y no en las ecuaciones de diferencias finitas de los modelos. No existen suficientes datos para correr los modelos numéricos en todas las zonas del país, y si existen, no son absolutamente confiables, o si lo son, no fueron recopilados en forma estadística en un largo período de tiempo.

Allí donde son necesarios, los datos son generados en forma estocástica, y donde las ecuaciones de diferencias finitas no tienen solución, también. Ese es el aporte que realiza este trabajo, que tienen una implicancia económica importante.

## 5. Herramientas empleadas en la Teoría

En el campo de la hidrogeología, se dispone de dos (2) modelos numéricos de predicción de niveles freáticos, Prickett-Lonnquist y ModFlow. Ambos desarrollan ecuaciones diferenciales, bien conocidas y con solución, razón por la cual no es necesario una modelación estocástica completa y pura.

El ModFlow es invención del United States Geological Survey, y data de 1984 con sucesivas adaptaciones y mejoras hasta 1990 inclusive. Es el modelo numérico en uso actualmente en ese país, el más ampliamente aceptado y documentado. Su uso es complejo y requiere mucho trabajo de campo para relevar los parámetros iniciales y ajustes posteriores.

El Prickett-Lonnquist es invención asimismo del Illinois State Water Survey, también de los EE.UU., y data de 1971. Si bien ya casi no posee difusión ni uso en los EE.UU., es el modelo matemático más ampliamente difundido y sobre el cual se han capacitado los especialistas de nuestro país. Su uso es mucho más sencillo que el ModFlow en cuanto al trabajo de campo requerido.

Debido a esto último, al uso en nuestro país y su capacitación y aceptación entre los profesionales del área, se ha decidido implementar la programación de los módulos estocásticos sobre este primer modelo, para luego de probar y aceptar su validación.

A continuación, un resumen funcional-matemático de los mismos.

### 5.1 MODFLOW

El modelo en uso por la United States Geological Survey (USGS) es conocido como MODFLOW, de medio de aplicación terrestre (\*), usado para hidrogeología, siendo un tipo de modelo de ecuaciones diferenciales parciales (diferencias finitas y tridimensional) junto a ecuaciones algebraicas estáticas, principalmente para investigación. Utiliza regresión no lineal para estimar los parámetros de entrada. Los parámetros son estimados por minimización de una función objetivo de potencias cuadradas mediante Gauss-Newton.

La regresión no lineal es una técnica para resolver las relaciones entre variables independientes (o de regresión) en la definición de un modelo, mediante ecuaciones no lineales.

El método de Gauss Newton consiste en minimizar una función objetivo de potencias cuadradas, tal que la suma de los cuadrados de los residuos (o errores) sea mínima.

### 5.2 PRICKETT

Otro modelo usado hoy día por los expertos en la materia es el PLASM, o Prickett Lonquist Aquifer Simulation Model, aunque el mismo es anterior al MODFLOW. El mismo también es de medio terrestre (\*), aplicado en hidrogeología, con una organización orientada al ecosistema, y al igual que el ModFlow, resuelve ecuaciones diferenciales parciales (diferencias finitas tridimensionales).

Las adaptaciones estocásticas a los modelos numéricos tradicionales se programarán sobre el modelo PRICKETT. Esas adaptaciones implica incorporar simulaciones al estado inicial del modelo (donde generalmente se desconocen los parámetros iniciales) usando los algoritmos de generación de fractales de Sierpinski, Lorenz y Henon para la distribución aleatoria de valores iniciales y generación de valores desconocidos al inicio de la corrida del modelo. También se usa Monte Carlo para los mismos fines.

## 6. Caso de Evaluación en Laboratorio

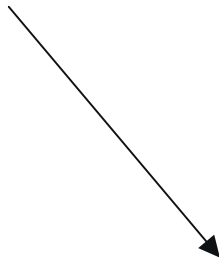
### 6.0 Interfase del software para modelizar: ¿Cómo ejecutar el modelo?

Para todos los casos de evaluación del modelo, se utilizó el software desarrollado. La interfase gráfica y su nomenclatura, se explican a continuación, en forma breve, y puede hallarse una ampliación de la misma en el anexo 8.2

Una vez ejecutado el programa "Prickett.Exe", se presenta el menú de selección para ejecutar el modelo en cuestión.

---

(\*) Clasificación adoptada por el Proyecto ECOBAS (base ecológica de modelos numéricos), Universidad de Kassel, Dinamarca ( <http://eco.wiz.uni-kassel.de/ecobas.html> ) Incluye también otros medios de aplicación, como ser acuáticos, y aéreos. Y no solo se clasifica el medio de aplicación, sino la teoría matemática utilizada para resolverlos. Es una clasificación realmente ejemplar.



Este botón cuadrado con una grilla de 5x5  
dispara el diálogo de  
parametrización del  
Modelo Prickett.

La nomenclatura que hallamos es la siguiente:

Niveles piezométricos = (m) metros

Factor de almacenamiento (confinado) (m<sup>3</sup> / m) = metros cúbicos x metro lineal

Caudales de bombeo constante (m<sup>3</sup> / día) = metros cúbicos x día

Transmisividad entre 2 nodos cualesquiera (m<sup>3</sup> / día / m) = metros cúbico x día x metro lineal

**Modelo Prickett-Lonnquist**

**Inicialización**

Cantidad de Filas: 10      Cantidad de Columnas: 10

Cantidad de Ciclos: 10      Mostrar ciclo...: 1

Delta del tiempo: 1      Incremento de Delta: 0.1

Error admisible: 3

Transmisividad inicial: 0.1

Nivel freático inicial: 0

Almacenamiento inicial: 0.1

Evaporación inicial: 0

**Punto de Bombeo**

Fila: 5      Columna: 5

Transmisividad inicial: 0.1

Nivel freático inicial: 0

Almacenamiento inicial: 0.1

Bombeo inicial: 1

Preparado

OK      Cancel      **Ejecutar**

La grilla solución es una abstracción bidimensional del terreno a ser modelizado. Cada celda representa el nivel de agua subterránea, o nivel freático.

El tiempo hace referencia a un período, y cada ciclo tiene un incremento de un delta de ese tiempo.

El punto de bombeo marca el pozo de extracción de agua en la grilla. Tiene coordenadas X e Y, comenzando en 0 desde el vértice izquierdo superior.

Esta lista detalla la cantidad de ciclos modelizados, disponibles para ser vistos en la grilla. Hay tantos ciclos como se especifiquen, luego de Ejecutar el modelo.

Este botón ejecuta el modelo según los parámetros, hasta una cantidad N de ciclos simulados. Al terminar, representa el primer ciclo en la grilla. Para cambiar de ciclo, elegirlo de la lista propia.

- Cantidad de filas / columnas: Número de nodos laterales que contendrá la superficie a modelizar.
- Cantidad de Ciclos: Iteraciones del modelo antes de la convergencia del error máximo admitido.
- Delta del tiempo: Tiempo referencial entre cada iteración.
- Incremento del delta: Incremento del delta del tiempo referencial.
- Error admisible: Error máximo acumulado admisible por las iteraciones del modelo.
- Transmisividad inicial: indicador de esta propiedad al inicio del modelo, en m<sup>3</sup>/día/m.
- Nivel freático inicial: valor por omisión de iniciación de la superficie, en cierto nivel freático, en m.
- Almacenamiento inicial: indicador de esta propiedad al inicio del modelo, en m<sup>3</sup>/m.
- Bombeo inicial: indicador de esta propiedad al inicio del modelo, generalmente representativo de una explotación artificial (planta de bombeo de agua), en m<sup>3</sup>/día.
- Evaporación inicial : indicador de esta propiedad al inicio del modelo, en m<sup>3</sup>/día.

### 6.1 Presentación del caso de laboratorio

Para el primer caso de evaluación y ajuste, se tomó como sistema real a una perforación de laboratorio, en la zona propiedad del Instituto Nacional de Aguas, AU Ezeiza-Cañuelas, Tramo J. Newbery, Km 1,620 (B1804) Ezeiza, Pcia. Buenos Aires, Argentina

La zona cuenta con una única bomba para extracción de agua, y es un terreno llano de la típica Pampa Húmeda bonaerense, sobre un área de campo de 100 x 100 m (1 hectárea), simulada en una grilla de 20 x 20 celdas (separación de 5 m entre celdas), al aire libre.

Para comparación de los métodos de iniciación, se ejecutó el modelo PRICKETT en las 4 variantes posibles de distribución fractal inicial.

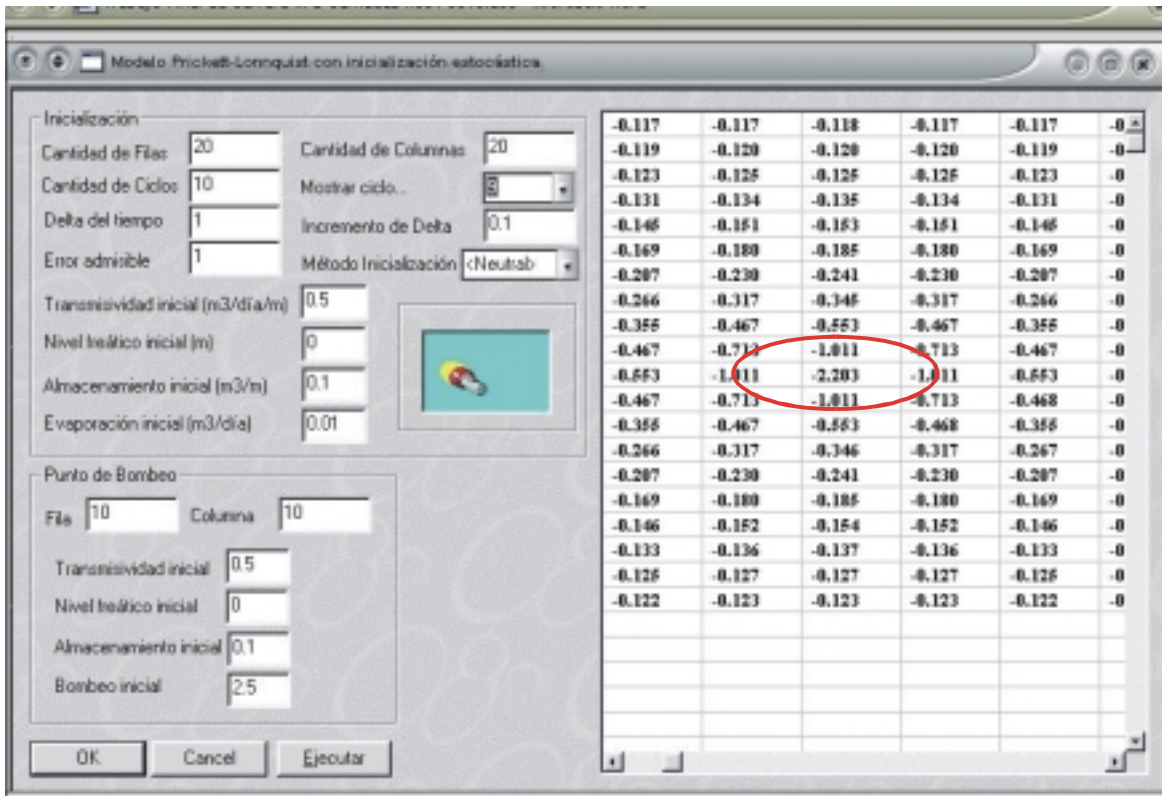
Los parámetros de ejecución del modelo se definieron en todos los casos como:

- Grilla de 20 x 20 celdas
- 10 ciclos de simulación representando 10 días
- Error admisible acumulado de 1 en cada ciclo
- Transmisividad de 0,5 m<sup>3</sup>/ día /m
- Niveles de agua originalmente en 0 metros
- Almacenamiento inicial 0,1 m<sup>3</sup>/m
- Evaporación inicial 0.01 m<sup>3</sup>/día, no uniforme, sino según distribución inicial
- Bomba de explotación en el centro del área, posición [10,10]
- Caudal de explotación de 2,5 m<sup>3</sup>/día

A continuación se presentan los resultados en diversas grillas, según muestra el software desarrollado, contando una grilla para cada caso de iniciación estocástica y por Monte Carlo.

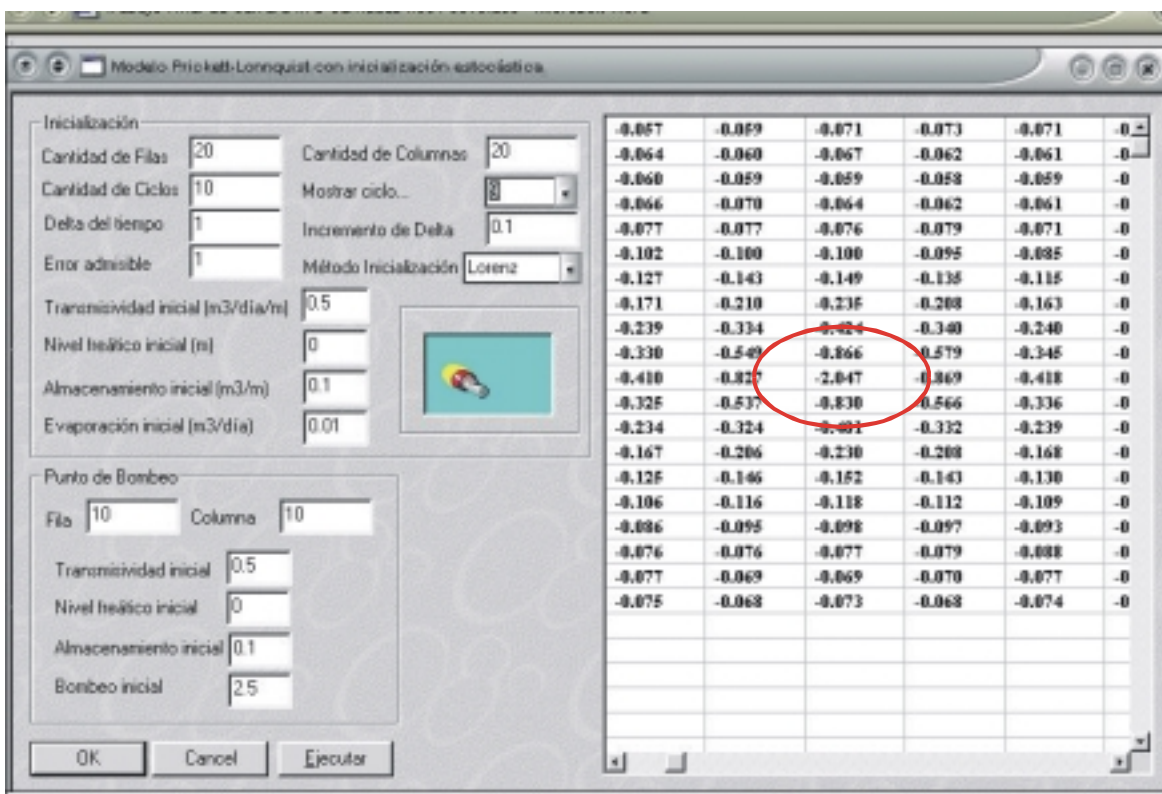
### 6.2 Prickett estándar, inicialización uniforme de estados

Para la iniciación estándar, en forma uniforme, se observa que al cabo de los 10 ciclos del modelo, en la posición de la bomba de extracción se halla una depresión del nivel freático de 2,203m con respecto al nivel inicial, y en un radio de 5m, la depresión es mayor a 1m, siendo aceptable entonces la incidencia del bombeo sobre el terreno.



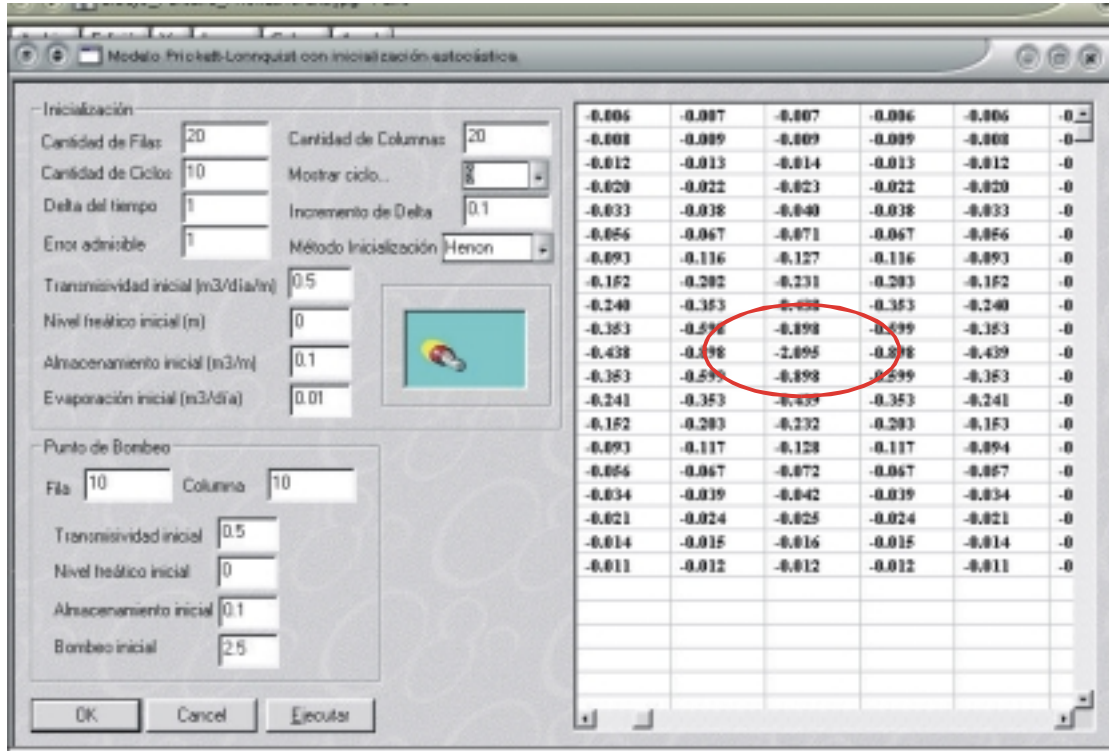
### 6.3 Prickett utilizando el método de Lorenz para iniciación

Para la iniciación con el método de Lorenz, se observa que al cabo de los 10 ciclos del modelo, en la posición de la bomba de extracción se halla una depresión del nivel freático de 2,047m con respecto al nivel inicial, y en un radio de 5m, la depresión es menor a 87cm, siendo aceptable entonces la incidencia del bombeo sobre el terreno.



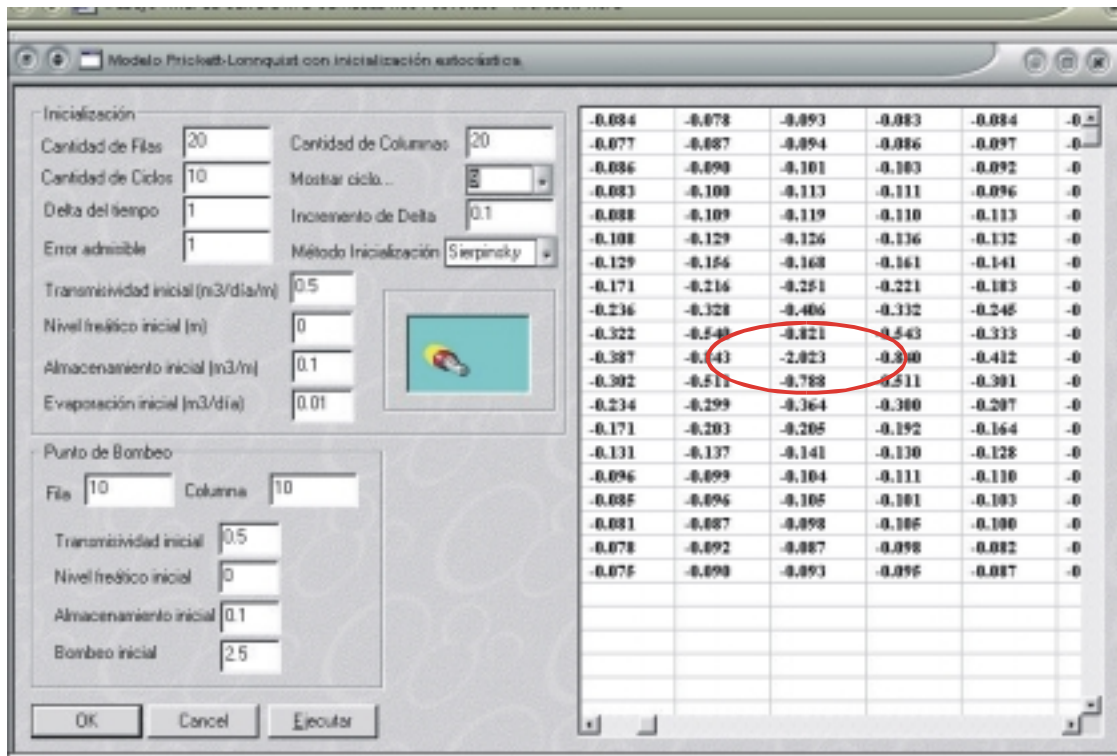
### 6.4 Prickett utilizando el método de Henon para iniciación

Para la iniciación con el método de Henon, se observa que al cabo de los 10 ciclos del modelo, en la posición de la bomba de extracción se halla una depresión del nivel freático de 2,095m con respecto al nivel inicial, y en un radio de 5m, la depresión es menor a 90cm, siendo aceptable entonces la incidencia del bombeo sobre el terreno.



### 6.5 Prickett utilizando el método de Sierpinski para iniciación

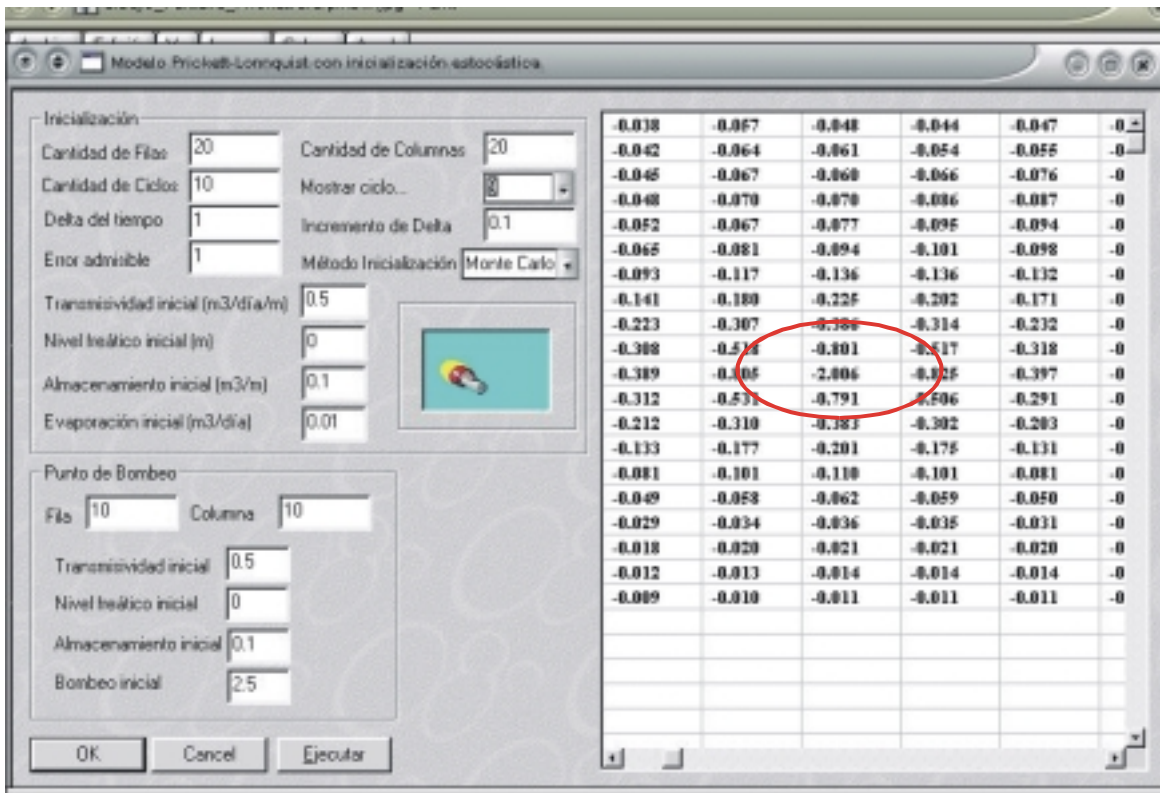
Para la iniciación con el método de Sierpinski, se observa que al cabo de los 10 ciclos del modelo, en la posición de la bomba de extracción se halla una depresión del nivel freático de 2,025m con respecto al nivel inicial, y en un radio de 5m, la depresión es menor a 85cm, siendo aceptable entonces la incidencia del bombeo sobre el terreno.





### 6.6 Prickett utilizando el método de Monte Carlo para iniciación

Para la iniciación con el método de Sierpinski, se observa que al cabo de los 10 ciclos del modelo, en la posición de la bomba de extracción se halla una depresión del nivel freático de 2,006m con respecto al nivel inicial, y en un radio de 5m, la depresión es menor a 82cm, siendo aceptable entonces la incidencia del bombeo sobre el terreno.



### 6.7 Correcciones al modelo de evaluación en laboratorio

Contrastando los resultados obtenidos contra las observaciones posteriores, al cabo de los 10 días, se pudo verificar que:

- 1) La transmisividad inicial de 0,5 m<sup>3</sup>/día/m es correcta . La misma fue obtenida con un ensayo de bombeo previo, mediante el método de Theis-Dupuit.
- 2) El nivel freático inicial de 0 m , relativo a nivel del mar, y relativo al primer nivel de agua encontrado en las napas, es correcto. Terreno nivelado , acuífero también.
- 3) La evaporación inicial de 0,01 m<sup>3</sup>/día debido a la incidencia de rayos solares y temperaturas por la estación climática del año, fué adecuada, ya que se registraron descensos uniformes en toda la zona demarcada
- 4) El almacenamiento inicial de 0,25 m<sup>3</sup>/día es en general correcto, aunque puede ser mayor según reflejó un segundo ensayo de bombeo, posterior a la corrida del modelo, donde se observaba una recuperación más rápida del nivel de agua.

Las correcciones al modelo del caso real, entonces, deben hacerse sobre el parámetro inicial de almacenamiento del acuífero.

En la nueva ejecución, se observan los mismos parámetros pero con un almacenamiento duplicado . Como resultado se obtuvo que la profundidad máxima luego de 10 días, es de 1,85m, y en un radio de 5m, alcanza a 71 cm., tomando como nivel inicial los cero metros (0m) , siempre en forma relativa a nivel del mar, para la primera napa de agua observada.

Este nivel de profundidad del cono de depresión refleja mejor lo realmente observado en la explotación, donde la medición indicó 1,80 m de profundidad, y luego, en el resto del acuífero, se ubicó una depresión uniforme por evapotranspiración, prácticamente nula.

### 6.8 Conclusiones para el caso de evaluación en laboratorio

Como comparación entre métodos de iniciación, siempre se ha observado que la distribución uniforme en el modelo PRICKETT generaba resultados que excedían lo registrado luego en la realidad. Con las distribuciones aplicadas, tenemos una corrección (*Ajustado*), que da como resultado, valores menores a los obtenidos en forma estándar uniforme y más ajustados a la contrastación posterior (*Realidad*).

Método	Máximo profundidad	Profundidad en radio 5 m	Variación porcentual
Estándar	2,203	1,01	N / A
Lorenz	2,047	0,87	-7,0
Henon	2,095	0,90	-4,9
Sierpinski	2,025	0,84	-8,0
MonteCarlo	2,006	0,82	-8,9
Ajustado	1,85	0,71	N/A
Realidad	1,80	0,71	N/A

## 7. Presentación del caso práctico de explotación para riego

### 7.1 Información general

Para el caso de aplicación práctica, se ejecutó sobre una perforación de explotación para riego en la filial del INTA en la zona de Tandil, Pcia. de Buenos Aires, Argentina, con una única bomba para extracción de agua del acuífero, y de terreno llano, sin observar depresiones ni lomadas, sobre un área de campo de 200 x 200 m (4 hectáreas), simulada en una grilla de 100 x 100 celdas (separación de 20 m entre celdas).

La bomba se utiliza para alimentar la red de riego del establecimiento agrícola, y el estudio en concreto se hace para evaluar la viabilidad de hacer trabajar la misma a un caudal de extracción importante, o si el impacto registrado hace necesario establecer una segunda bomba para no producir un cono de depresión muy pronunciado, sino más leve y más recuperable en menor tiempo.

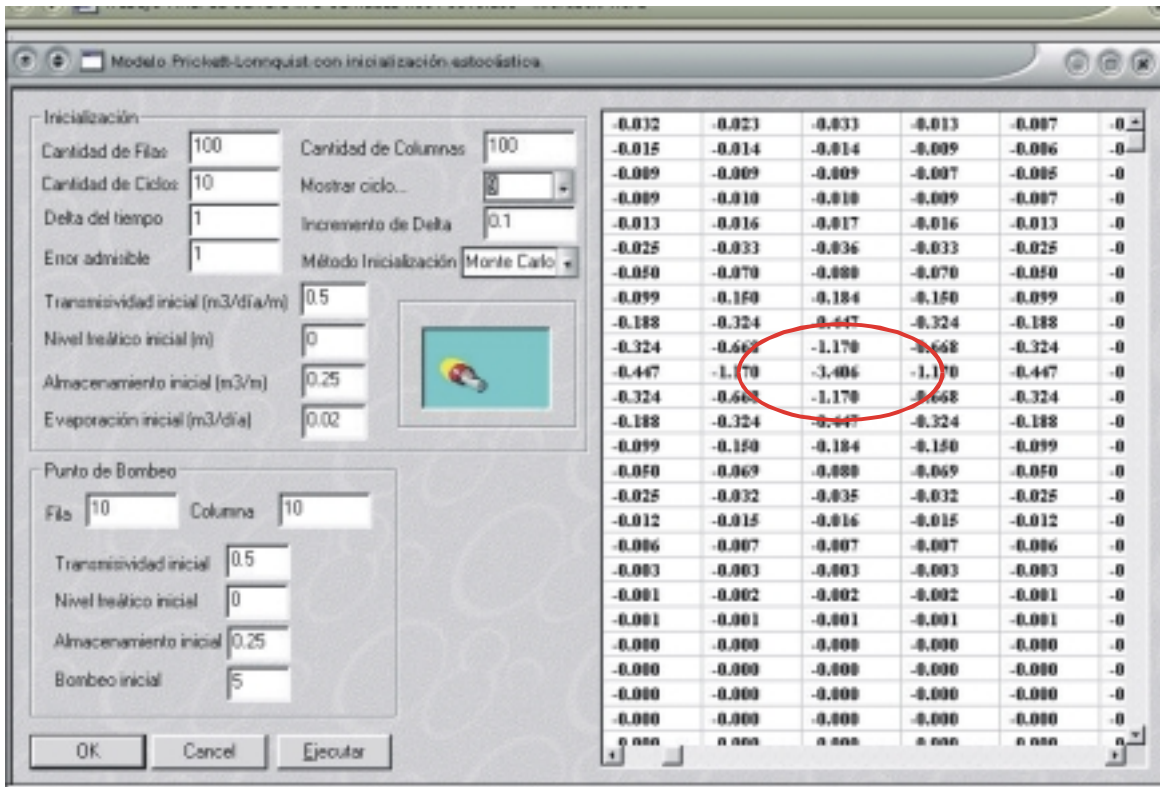
### 7.2 Implementación del modelo

Los parámetros de ejecución del modelo se definieron sobre la base de lo necesitado por el establecimiento dueño de la perforación, una proyección trimestral del estado de las napas freáticas.

- Grilla de 100 x 100
- 10 ciclos de simulación representando 10 días de extracción de agua
- Error admisible acumulado de 1 en cada ciclo
- Transmisividad de 0,5
- Niveles de agua originalmente en 0 m, según inicialización MonteCarlo
- Almacenamiento inicial 0,25, según inicialización MonteCarlo
- Evaporación inicial 0,02 según iniciación MonteCarlo
- Bomba de explotación en el cuadrante 1, posición [10,10]
- Caudal de explotación de 5 ,0

Como resultado se obtuvo que la profundidad máxima luego de 10 días, es de 3,48m, y en un radio de 20m, alcanza a 1,25m, tomando como nivel inicial los cero metros (0m) , siempre en forma relativa a nivel del mar, para la primera napa de agua observada.

A continuación, el resultado luego de la ejecución del modelo tal como lo refleja el software desarrollado al efecto.



### 7.3 Correcciones al modelo para explotación para riego

Contrastando los resultados obtenidos contra las observaciones posteriores, al cabo de los 10 días, se pudo verificar que:

- 1) La transmisividad inicial de 0,5 m<sup>3</sup>/día/m es correcta ( fue obtenida con un ensayo de bombeo )
- 2) El nivel freático inicial de 0 m , relativo a nivel del mar, y relativo al primer nivel de agua encontrado en las napas, es correcto, ya que se verifica un terreno nivelado , lo mismo que el acuífero subyacente.
- 3) La evaporación inicial de 0,02 m<sup>3</sup>/día debido a la incidencia de rayos solares y temperaturas por la estación climática del año, posiblemente haya resultado inferior a lo real, ya que los niveles de profundidad de la napa cayeron uniformemente en toda la zona demarcada
- 4) El almacenamiento inicial de 0,25 m<sup>3</sup>/día es en general correcto, aunque puede ser mayor según reflejó un segundo ensayo de bombeo, posterior a la corrida del modelo, donde se observaba una recuperación más rápida del nivel de agua.

Las correcciones al modelo del caso real, entonces, deben hacerse sobre el parámetro inicial de evapotranspiración (caída uniforme del nivel de agua en todo el acuífero , sin mediar bombeos), tentativamente triplicándolo, y asignar un 50% mas de capacidad de almacenamiento al acuífero en líneas generales.

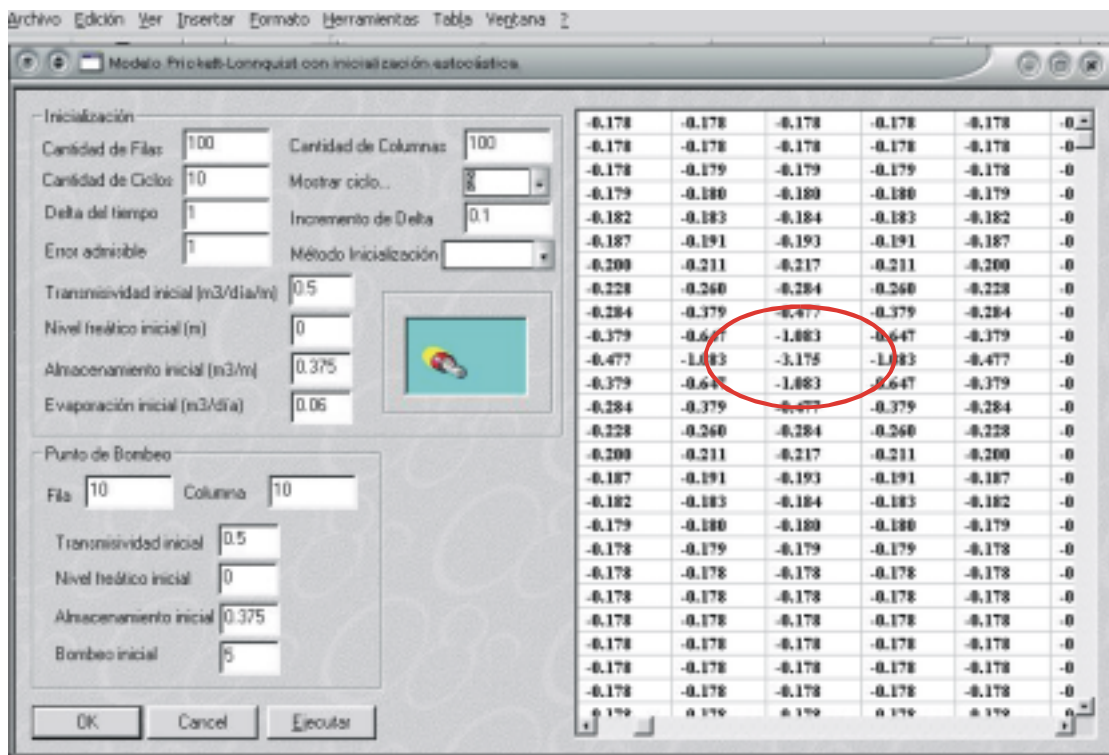
En la nueva ejecución, se tiene :

- Grilla de 100 x 100
- 10 ciclos de simulación representando 10 días de extracción de agua
- Error admisible acumulado de 1 en cada ciclo
- Transmisividad de 0,5
- Niveles de agua originalmente en 0 m, según inicialización MonteCarlo
- Almacenamiento inicial 0,375, según inicialización MonteCarlo
- Evaporación inicial 0,06 según iniciación MonteCarlo
- Bomba de explotación en el cuadrante 1, posición [10,10]
- Caudal de explotación de 5,0 m<sup>3</sup>/día

Como resultado se obtuvo que la profundidad máxima luego de 10 días, es de 3,033m, y en un radio de 20m, alcanza a 0,91 cm, tomando como nivel inicial los cero metros (0m) , siempre en forma relativa a nivel del mar, para la primera napa de agua observada.

Este nivel de profundidad del cono de depresión refleja mejor lo realmente observado en la explotación, donde la medición indicó 3.0 m de profundidad, y luego, en el resto del acuífero, se ubicó una depresión uniforme por evapotranspiración de 15cm.

A continuación, el resultado de la ejecución corregida.



### 7.4 Caso de Máxima explotación estimada

Para el caso de **máxima explotación posible** para este pozo de extracción de agua, se plantea el caso de conocer la profundidad máxima del cono de depresión. En función de ello, se calcula que diariamente se extraerán 50.000 litros de agua para riego, o lo que es lo mismo, 50 m<sup>3</sup>/día, y con los mismos parámetros corregidos del punto anterior.

- Grilla de 100 x 100
- 10 ciclos de simulación representando 10 días de extracción de agua
- Error admisible acumulado de 1 en cada ciclo
- Transmisividad de 0,5
- Niveles de agua originalmente en 0 m, según inicialización MonteCarlo
- Almacenamiento inicial 0,375, según inicialización MonteCarlo
- Evaporación inicial 0,06 según iniciación MonteCarlo
- Bomba de explotación en el centro geográfico, posición [50,50]
- Caudal de explotación de 50,0 m<sup>3</sup>/día

Como resultado se obtuvo que la profundidad máxima luego de 10 días, es de 30.1 m, y en un radio de 20m, alcanza a 9.30 m, tomando como nivel inicial los cero metros (0m) .

Este nivel de profundidad del cono de depresión refleja aproximadamente lo observado en la explotación, donde la medición indicó en realidad 28.0 m de profundidad. Esto es atribuible a que no se contrastó durante los 10 días de extracciones, sino solamente en la primera jornada, debido a no poder disponer del equipo y de lo costoso de la operación.

### 7.5 Conclusiones para el caso de explotación para riego

Al igual que para el caso del modelo de laboratorio, la comparación entre métodos de iniciación siempre arroja que la distribución uniforme en el modelo PRICKETT genera resultados que exceden lo registrado luego en la realidad. Aplicando la distribución de Monte Carlo, que es la mas conservadora para descensos de niveles, se obtiene una excelente aproximación a lo posteriormente contrastado desde los puntos de observación y control.

Método	Máximo de Profundidad	Profundidad en un radio de 5m
MonteCarlo (primer corrida)	3,48	1,25
MonteCarlo (corrida ajustada)	3,03	0,91
Realidad	3,00	0,90

## 8. Anexos

### 8.1 Anexo 1 : Desarrollo de la teoría para el Modelo Prickett-Lonnquist

El modelo Prickett-Lonnquist simula en medio anisótropo bidimensional el flujo subterráneo originado por la explotación en régimen transitorio de acuíferos libres o confinados. La resolución de la ecuación diferencial del flujo es aproximada en diferencias finitas. (\*)

Para poder definir un modelo así se necesitan :

1. Descripción de la geometría del terreno a aplicar
2. Descripción de las propiedades físicas del medio en cada punto del dominio. Permeabilidad, transmisividad, coeficiente de almacenamiento, etc.
3. Ley de Darcy que rige el movimiento del agua. Teorema de la Continuidad.
4. Condiciones de los límites del sistema
5. Condiciones en un tiempo inicial  $t_0$
6. Descripción de las acciones exteriores al sistema

Con todo esto se tiene la ecuación diferencial que describe el movimiento del agua para el caso bidimensional (\*) :

$$\frac{\delta}{\delta x} (T_x \delta h) + \frac{\delta}{\delta y} (T_y \delta h) + q' = - S \frac{\delta h}{\delta t}$$

siendo  $T$  y  $h$  funciones de  $x$  e  $y$  (coordenadas rectangulares en un punto) ,  $T$ , función transmisividad, y  $h$  el nivel del acuífero o napa de agua, en el instante  $t$ .

El balance instantáneo de volúmenes en un elemento infinitesimal del acuífero para un punto  $[x,y]$  en un momento dado (considerando a las propiedades físicas del cuerpo de agua como constantes verticalmente y que la velocidad es constante horizontalmente con un valor dado por la Ley de Dupuit) se representa por:

$$V = - K \text{ grad } h$$

siendo  $K$  la permeabilidad del suelo o medio.

Este es el problema general para caudales. Si se pretende resolver el movimiento y la concentración de los cuerpos disueltos habría que considerar además la ecuación del transporte de masa o la de la dispersión hidrodinámica, resolviéndolas conjuntamente con la anterior.

Si suponemos un esquema de acuífero constituido por una malla de prismas rectangulares (matriz) de base  $\Delta x$ ,  $\Delta y$ , entonces el desarrollo del esquema numérico se presenta a continuación.

a) Para representar las derivadas parciales, el esquema a emplear será:

$$\frac{\delta f}{\delta x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} = \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

con un error  $K^* \Delta x^2$ .

b) Las expresiones en diferencias finitas para el punto  $[x,y]$  son:

$$\frac{\delta T}{\delta x} = \frac{T(x + \Delta x, y) - T(x - \Delta x, y)}{2\Delta x}$$

$$\frac{\delta T}{\delta y} = \frac{T(x, y + \Delta y) - T(x, y - \Delta y)}{2\Delta y}$$

(\*) PRICKET & LONNQUIST, "Selected Digital Computer Techniques for Groundwater Resource Evaluation", Illinois State Water Supply, Bulletin 55, EE.UU., 1971.

lo mismo para la función **h**.

c) El modelo utiliza un **factor de almacenamiento** que se define como

$$SF1[i, j] = S * \Delta X * \Delta Y$$

donde **SF1[i, j]** es el factor de almacenamiento para la celda de la grilla [i, j], en m<sup>3</sup>/m, **S** es el coeficiente de almacenamiento de la napa de agua subterránea en condiciones de confinamiento (adimensional) y **DX** y **DY** son las dimensiones de la grilla o malla, en metros.

El primer paso es la discretización de las propiedades físicas del acuífero o napa de agua subterránea. A cada celda de la grilla se le asignan las propiedades o valores de transmisividad, factor de almacenamiento, nivel inicial y final del agua, y caudal de bombeo. Estas propiedades físicas se representan por medio de matrices, donde cada celda es la abstracción de un cuadrado en el terreno, de N x M metros, siendo las mismas:

1. La matriz **t0** representa la transmisividad inicial, en m<sup>3</sup>/día/m.
2. La matriz **t1** representa la transmisividad actual, en m<sup>3</sup>/día/m.
3. La matriz **h** representa los niveles de agua actuales, en m.
4. La matriz **h0** representa los niveles de agua iniciales, en m.
5. La matriz **sf1** representa el factor de almacenamiento actual, en m<sup>3</sup>/m.
6. La matriz **q** representa el bombeo o extracción actual, en m<sup>3</sup>/día.
7. La matriz **dl** representa las diferencias de incremento de niveles de agua, en m.

La resolución consta de tres partes principales: Entrada de datos, simulación propiamente dicha, y exportación de resultados en formato coordenado (i, j, valor).

Las dimensiones de filas y columnas están dadas dinámicamente, y limitan el área de cálculo. La variable **time** (tiempo) es puesta a cero e incrementada con su delta. No forma parte de la simulación sino de la salida o resultados por conjuntos de tiempos.

La predicción de niveles freáticos o de agua, predice el comportamiento del modelo en el futuro de acuerdo a sus comportamientos en períodos de tiempo anteriores.

La inclusión de la predicción aumenta la convergencia de la solución: se hacen hipótesis entre los niveles presentes y anteriores, siendo su ecuación (\*):

$$h[i, j] = h[i, j] + D * F \quad / \quad F = D / DL[i, j], \quad D = h[i, j] - h0[i, j]$$

Siendo **D** la diferencia de niveles entre la corrida anterior y la actual, y **F** un factor multiplicador según la capacidad de bombeo en ese punto [i, j]. Cuando **F** no es calculable o predecible, se la asume con el valor 1.0: También se contempla la división por cero y se acota **F** en límites razonables, entre 0.0 y 5.0.

Los últimos niveles freáticos calculados en el último intervalo de tiempo se hallan en la matriz **h[i, j]** y son trasladados a la matriz **h0[i, j]**, para preparación del siguiente intervalo de tiempo.

Luego, se realizan los cálculos por columnas, calculando el nivel como **h[i, j] + D \* F**, donde **D** es **h[i, j] - h0[i, j]**. Durante las iteraciones impares, se procesan las columnas en orden creciente, mientras que en las impares se lo hace en forma decreciente. Los términos **AA**, **BB**, **CC** y **DD** son variables de soporte auxiliar de resultados de ecuaciones.

En los vectores de soporte de cálculos **G** y **B**, se hallan los niveles de agua en cada celda para cada columna siguiendo un orden contrario al llevado para su cálculo. En el término **E**, que es el *error de cálculo en la iteración actual*, se añade la suma de los valores de dichos cambios. Al final del cálculo de todas las columnas, el programa calcula por filas los niveles en las celdas o nudos de la matriz. Este último paso es exactamente el mismo que el utilizado para las columnas, excepto por el cambio de índices necesario.

Concluido esto, se comparan **E** y **ERROR**, donde esta última es el *error acumulado* para todas las iteraciones. Si **E** es mayor a **ERROR**, la solución obtenida no es satisfactoriamente convergente, y se recomienza una nueva iteración.

Cuando se satisface la condición de convergencia, termina la simulación.

(\*) PRICKET & LONNQUIST, Op. Cit.

La condición de **ERROR** se elige en aquel punto donde su valor no hace variar significativamente la solución, y para esto se plantea una regla práctica :

$$ERROR=Q*DELTA/10*SF1$$

Donde **Q** es el caudal en m<sup>3</sup>/día, **DELTA** es el incremento de tiempo inicial, en días, **SF1** es el factor de almacenamiento medio de agua, en m<sup>3</sup>/m.

## 8.2 Anexo 2: Modelo Prickett-Lonnquist en lenguaje C/C++

La implementación programática del modelo original, escrito en Fortran77, fue migrada a lenguaje C/C++ en clases y módulos, e implementada en Sistema Operativo Ms.Windows y Linux.

### Definición

Lo que sigue a continuación, es la definición formal en lenguaje C++ de la clase que implementa al modelo Prickett-Lonnquist, junto a las clases bases de soporte :

```
// notación acordada previamente para designar elementos en el modelo:
// h0[i][j] : niveles piezométricos iniciales (m)
// h[i][j] : niveles piezométricos terminales (m)
// sf1[i][j] : factor de almacenamiento (confinado) (m3/m)
// q[i][j] : caudales de bombeo constante (m3/día)
// t0[i][j] : transmisividad entre i,j y i,j+1 (m3/día/m)
// t1[i][j] : transmisividad entre i,j y i+1,j (m3/día/m)

// Modelo Matemático con Teoría del Caos
class ChaosMModel
{
public:

    // Constructor
    ChaosMModel(int x_dim=0,int y_dim=0);

    // Lorenz method // Metodo de Lorenz
    void Initialize_Lorenz(float a,float b,float c,float dt, BaseReal **matrix, BaseReal value);
    // Henon method // Metodo de Henon
    void Initialize_Henon(float a,float b, BaseReal **matrix, BaseReal value);
    // Sierpinski method // Metodo de Sierpinski
    void Initialize_Sierpinski(BaseReal **matrix, BaseReal value);
    // MonteCarlo method // Metodo MonteCarlo
    void Initialize_MonteCarlo(BaseReal **matrix, BaseReal value, UInt ix,
    UInt jx);
    // Set Dimensions // Establecer dimensiones
    void SetDimensions(int x_dim,int y_dim);

private:

    // Matrix Dimensions // Dimensiones de matriz
    UInt dimx, dimy;

};

// clase prickett-lonnquist derivado de un modelo matricial base y unidad de
// punto flotante, y de otro de caos (estocástico)
```

```

class PricketLonquist : public BaseMModel, public ChaosMModel
{
public:

    PricketLonquist(void); // constructor
    ~PricketLonquist(void); // destructor

    // Modelization for 1 pump // modelizacion para un punto de bombeo
    USInt Modelization(Int dimrows, Int dimcols, USInt steps, BaseReal delta,
BaseReal deltainc, BaseReal error, Byte format, Int r, Int c, BaseReal vt0,
BaseReal vt1, BaseReal vsf1, BaseReal vh, BaseReal vq, ChaosType
ctType=Chaos_NoInit);

    // Modelization for 1 pump // modelizacion para un punto de bombeo
    USInt Modelization(Int dimrows, Int dimcols, USInt steps, BaseReal delta,
BaseReal deltainc, BaseReal error, Byte format, BaseReal ivt0, BaseReal ivsf1,
BaseReal ivh, BaseReal ivq, Int r, Int c, BaseReal vt0, BaseReal vt1, BaseReal
vsf1, BaseReal vh, BaseReal vq, ChaosType ctType=Chaos_NoInit);

    USInt Modelization(ChString nodesinfilename); // modelizacion
    inline USInt TotalIterations(void) { return totaliter; } // iteraciones
    inline Real TotalTime(void) { return totaltime; } // tiempo

protected:

    BaseReal **t0; // transmisivity m3/day/m // transmisividad m3/día/m
    BaseReal **t1; // transmisivity m3/day/m // transmisividad m3/día/m
    BaseReal **h; // ending piezometric level m // nivel piezometrico final
    BaseReal **h0; // piezometric level m // nivel piezometrico m
    BaseReal **sf1; // capacity m3/m // almacenamiento m3/m
    BaseReal **q; // pumping m3/day // bombeo m3/día
    BaseReal **dl; // ?
    BaseReal *g, *b; // equations processing hold vectors // vectores soporte
de procesamiento de ecuaciones
    USInt totaliter; // total iterations // total de iteraciones
    Real totaltime; // total time // tiempo total
    USInt ncount; // print-output ncount // cuenta de salidas impresas

    inline Dim Cols(void) { return nc; } // columns // columnas
    inline Dim Rows(void) { return nr; } // rows // filas
    inline void Cols(Dim ncols){ nc = ncols; } // columns // columnas
    inline void Rows(Dim nrows){ nr = nrows; } // rows // filas
    inline Dim MinCols(void) { return 0; } // min col // columna minima
    inline Dim MinRows(void) { return 0; } // min row // fila minima
    inline Dim MaxCols(void) { return nc-1; } // max col // columna maxima
    inline Dim MaxRows(void) { return nr-1; } // max row // fila maxima
    inline Boolean Precondition(void)
    {
        return ((h != NULL) && (h0 != NULL) && (sf1 != NULL) && (q != NULL) && dl
!= NULL) && (g != NULL) && (b != NULL) && (t0 != NULL) && t0 != NULL));
    }
    virtual void MatricesAllocation(void); // matrices allocation // alojamiento
de matrices
    virtual void OutPut(Byte format); // external nodes // nodos externos
    virtual void Initialization(BaseReal tt, BaseReal s1, BaseReal hh, BaseReal
qq, ChaosType ctType=Chaos_NoInit); // initialization // inicializacion
    virtual void Prediction(USInt istep); // prediction // prediccion
    virtual void RowsProcess(USInt istep, USInt liter, BaseReal ldelta, BaseReal

```



```

*lerror);
    virtual void ColumnsProcess(USInt istep, USInt liter, BaseReal ldelta,
BaseReal *lerror);
    virtual void Simulation(USInt nsteps, BaseReal delta, BaseReal deltainc,
BaseReal error, Byte format); // simulation // simulacion

private:

    Dim nc; // number of columns // numero de columnas
    Dim nr; // number of rows // numero de filas

};

```

## Implementación

```

////////////////////////////////////
// Constructor de clase
////////////////////////////////////
PricketLonnquist::PricketLonnquist(void)
{
    t0 = NULL; // no matrices or vectors // sin matrices ni vectores
    t1 = NULL;
    h = NULL;
    h0 = NULL;
    sf1 = NULL;
    q = NULL;
    dl = NULL;
    g = NULL;
    b = NULL;
    totaltime = 0.0; // no statics // sin estadísticas
    totaliter = 0;
    Cols(0); // assign the columns // asignar las columnas
    Rows(0); // assign the rows // asignar las filas
    ncount = 0; // output count to zero // salidas impresas en cero
}

////////////////////////////////////
// Destructor de clase
////////////////////////////////////
PricketLonnquist::~PricketLonnquist(void)
{
    if ( Precondition() )
    {
        Deallocation(t0 , Cols(), Rows()); // deallocate // desalojar
        Deallocation(t1 , Cols(), Rows());
        Deallocation(h , Cols(), Rows());
        Deallocation(h0 , Cols(), Rows());
        Deallocation(sf1, Cols(), Rows());
        Deallocation(q , Cols(), Rows());
        Deallocation(dl , Cols(), Rows());
        if (Cols() >= Rows()) // verificar el maximo requerido
        {
            Deallocation(g, Cols());
            Deallocation(b, Cols());
        }
    }
    else
    {

```

```

        Deallocation(g, Rows());
        Deallocation(b, Rows());
    }
    t0 = NULL; // no matrices or vectors // sin matrices/vectores
    t1 = NULL;
    h = NULL;
    h0 = NULL;
    sf1 = NULL;
    q = NULL;
    dl = NULL;
    g = NULL;
    b = NULL;
}
ncount = 0; // output count to zero // salidas impresas en cero
}

////////////////////////////////////
// Alojamiento de la matriz
////////////////////////////////////
void PricketLonquist::MatricesAllocation(void)
{
    t0 = Allocation(Cols(), Rows()); // alojar matrices y vectores
    t1 = Allocation(Cols(), Rows());
    h = Allocation(Cols(), Rows());
    h0 = Allocation(Cols(), Rows());
    sf1 = Allocation(Cols(), Rows());
    q = Allocation(Cols(), Rows());
    dl = Allocation(Cols(), Rows());
    if (Cols() >= Rows())// verificar el maximo requerido
    {
        g = Allocation(Cols());
        b = Allocation(Cols());
    }
    else
    {
        g = Allocation(Rows());
        b = Allocation(Rows());
    }
    if ( !Precondition() )
        throw (-1); // verify // verificar
}

////////////////////////////////////
Inicializacion de toda la matriz con los parametros de transmisividad,
// almacenamiento, niveles iniciales y bombeo
////////////////////////////////////void
PricketLonquist::Initialization(BaseReal tt, BaseReal s1, BaseReal hh, BaseReal
qq)
{
    Int i, j;
    // matrix initialization // inicializacion de la matriz
    for (i = MinCols(); i <= MaxCols(); i++) // para cada columna
    {
        for (j = MinRows(); j <= MaxRows(); j++) // para cada fila
        {
            t0[i][j] = tt; // asignar valores por omision
            t1[i][j] = tt;
            sf1[i][j] = s1;
        }
    }
}

```

```

                h[i][j] = hh;
                h0[i][j] = hh;
                q[i][j] = qq;
            }
        }
    }

// Prediccion para el paso actual
// istep=paso actual de procesamiento
// Prediccion para el paso actual
void PricketLonnquist::Prediction(USInt istep)
{
    // aux vars/ // variables auxiliares
    Int i = 0.0, j = 0.0;
    BaseReal d = 0.0, f = 0.0;

    // prediction // prediccion
    for (i = MinCols(); i <= MaxCols(); i++) // para cada columna
    {
        for (j = MinRows(); j <= MaxRows(); j++) // para cada fila
        {
            d = h[i][j] - h0[i][j];
            h0[i][j] = h[i][j];
            f = 1.0;
            if (istep > 2)
            {
                if (dl[i][j] != 0.0) f = d/dl[i][j];
                else f = d/ZERODIVIDE_REPLACE;
            }
            if (f > 5.0) f = 5.0;
            if (f < 0.0) f = 0.0;
            dl[i][j] = d;
            h[i][j] += d*f;
        } // endfor-byrow
    } // endfor-byrow-prediction
}

// Procesamiento por Columnas
// istep=indicador de paso o ciclo (input)
// liter=iteraccion actual (input)
// ldelta=tiempo acumulado (input)
// *lerror=error acumulado (output)
// ColumnasProcess
void PricketLonnquist::ColumnsProcess(USInt istep, USInt liter, \
                                        BaseReal ldelta, BaseReal *lerror)
{
    // aux vars/ // variables auxiliares
    Int ii = 0, i = 0, j = 0, n = 0;
    BaseReal aa = 0.0, bb = 0.0, cc = 0.0,
            dd = 0.0, ha = 0.0, w = 0.0;

    // column calculation process // proceso de calculo de columnas
    for (ii = MinCols(); ii <= MaxCols(); ii++)
    {
        if ((istep+liter)%2) == 1) i = MaxCols()-ii+MinCols();
        else i = ii;
    }
}

```

```

for (j = MinRows(); j <= MaxRows(); j++)
{
    // row initialization // inicializacion para fila
    aa = 0.0;
    bb = 0.0;
    cc = 0.0;
    dd = 0.0;
    if (ldelta != 0.0) // si hay un error acumulado (delta)
    {
        bb = sf1[i][j]/ldelta;
        dd = h0[i][j]*sf1[i][j]/ldelta-q[i][j];
    }
    else
    {
        bb = sf1[i][j]/ZERODIVIDE_REPLACE;
        dd = h0[i][j]*sf1[i][j]/ZERODIVIDE_REPLACE-q[i][j];
    }
    if ((j-MinRows()) != 0) // si no esta en limite
    {
        aa = -t0[i][j-1];
        bb += t0[i][j-1];
    }
    if ((j-MaxRows()) != 0) // si no esta en limite
    {
        bb += t0[i][j];
        cc = -t0[i][j];
    }
    if ((i-MinCols()) != 0) // si no esta en limite
    {
        bb += t1[i-1][j];
        dd += h[i-1][j]*t1[i-1][j];
    }
    if ((i-MaxCols()) != 0) // si no esta en limite
    {
        bb += t1[i][j];
        dd += h[i+1][j]*t1[i][j];
    }
    if ((j-MinRows()) != 0) // si no esta en limite
    {
        w = bb-aa*b[j-1];
        if (w != 0.0)
        {
            b[j] = cc/w;
            g[j] = (dd-aa*g[j-1])/w;
        }
        else
        {
            b[j] = cc/ZERODIVIDE_REPLACE;
            g[j] = (dd-aa*g[j-1])/ZERODIVIDE_REPLACE;
        }
    }
    else
    {
        w = bb;
        if (w != 0.0)
        {
            b[j] = cc/w;
            g[j] = dd/w;
        }
    }
}

```

```

    }
    else
    {
        b[j] = cc/ZERODIVIDE_REPLACE;
        g[j] = dd/ZERODIVIDE_REPLACE;
    }
} // endfor-column-calculations

// error reestimation // reestimación del error acumulado
*lerror      += Absolute(h[i][MaxRows()-g[MaxRows()]]);
h[i][MaxRows()] = g[MaxRows()];
for (n = MaxRows()-1; n >= MinRows(); n-)
{
    ha      = g[n]-b[n]*h[i][n+1];
    *lerror += Absolute(ha-h[i][n]);
    h[i][n] = ha;
} // endfor-reestimation

} // endfor-column-process
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Procesamiento por Filas
// istep=indicador de paso o ciclo (input)
// liter=iteracion actual (input)
// ldelta=desvio acumulado (input)
// *lerror=error acumulado (output)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void PricketLonnquist::RowsProcess(USInt istep, USInt liter, \
                                   BaseReal ldelta, BaseReal *lerror)
{
    // aux vars/ // variables auxiliares
    Int      i = 0 , j = 0 , jj = 0 , n = 0;
    BaseReal aa = 0.0, bb = 0.0, cc = 0.0,
             dd = 0.0, ha = 0.0, w = 0.0;

    // rows calculations process // proceso de calculos de filas
    for (jj = MinRows(); jj <= MaxRows(); jj++)
    {
        if (((istep+liter)%2) == 1) j = MaxRows()-jj+MinRows();
        else                          j = jj;
        for (i = MinCols(); i <= MaxCols(); i++)
        {
            // column initialization // inicializacion para columna
            aa = 0.0;
            bb = 0.0;
            cc = 0.0;
            dd = 0.0;
            if (ldelta != 0.0) // si hay un error acumulado (delta)
            {
                bb = sf1[i][j]/ldelta;
                dd = h0[i][j]*sf1[i][j]/ldelta-q[i][j];
            }
            else
            {
                bb = sf1[i][j]/ZERODIVIDE_REPLACE;
                dd = h0[i][j]*sf1[i][j]/ZERODIVIDE_REPLACE-q[i][j];
            }
        }
    }
}

```

```

    }
    if ((j-MinRows()) != 0) // si no esta en limite
    {
        bb += t0[i][j-1];
        dd += h[i][j-1]*t0[i][j-1];
    }
    if ((j-MaxRows()) != 0) // si no esta en limite
    {
        bb += t0[i][j];
        dd += h[i][j+1]*t0[i][j];
    }
    if ((i-MinCols()) != 0) // si no esta en limite
    {
        aa += -t1[i-1][j];
        bb += t1[i-1][j];
    }
    if ((i-MaxCols()) != 0) // si no esta en limite
    {
        bb += t1[i][j];
        cc += -t1[i][j];
    }
    if ((i-MinCols()) != 0) // si no esta en limite
    {
        w = bb-aa*b[i-1];
        if (w != 0.0)
        {
            b[i] = cc/w;
            g[i] = (dd-aa*g[i-1])/w;
        }
        else
        {
            b[i] = cc/ZERODIVIDE_REPLACE;
            g[i] = (dd-aa*g[i-1])/ZERODIVIDE_REPLACE;
        }
    }
}
else
{
    w = bb;
    if (w != 0.0)
    {
        b[i] = cc/w;
        g[i] = dd/w;
    }
    else
    {
        b[i] = cc/ZERODIVIDE_REPLACE;
        g[i] = dd/ZERODIVIDE_REPLACE;
    }
}

} // endfor-row-calculations

// reestimation // reestimacion
*lerror      += Absolute(h[MaxCols()][j]-g[MaxCols()]);
h[MaxCols()][j] = g[MaxCols()];
for (n = MaxCols()-1; n >= MinCols(); n-)
{
    ha      = g[n]-b[n]*h[n+1][j];

```

```

        *lerror += Absolute(h[n][j]-ha);
        h[n][j] = ha;
    } // endfor-reestimation

} // endfor-row-process
}

////////////////////////////////////
// Proceso de Simulacion parametrizado
// nsteps=cantidad de pasos minimos y necesarios (input)
// delta=tiempo acumulado maximo aceptable (input)
// deltainc=incremento del delta (input)
// error=error acumulado maximo aceptable (input)
// format=tipo de formato de salida {matricial,o por puntos} (input)
////////////////////////////////////
void PricketLonquist::Simulation(USInt nsteps, BaseReal delta, \
                                BaseReal deltainc, BaseReal error, \
                                Byte format)
{
    USInt    istep = 0, liter = 0;
    BaseReal lerror = 0.0, perror = 0.0, ltime = 0.0, ldelta = 0.0;

    // prediction cicle up to N steps // ciclo de prediccion hasta N pasos
    for (ldelta = delta, ltime = ldelta, istep = 1;
        istep <= nsteps;
        istep++, ldelta *= deltainc, ltime += ldelta) // simulation
    {
        Prediction(istep); // prediccion
        liter = 0; // contar iteraciones
        lerror = MAX_REAL; // limite del ciclo de refinado
        do // refining // refinamiento
        {
            liter++;
            perror = lerror; // keep previous error
            lerror = 0.0; // null error
            ColumnsProcess(istep, liter, ldelta, &lerror);
            RowsProcess(istep, liter, ldelta, &lerror);
        } while ((lerror > error) && (lerror < perror));
    } // endwhile-refining

    if (lerror < perror) // verificar error menor al previo
    {
        OutPut(format); // print output // imprimir salida
    }
    else
    {
        throw ("No converge, incrementar error");
        break; // exit loop // salir del ciclo
    }
} // endfor-simulation

totaltime = ltime; // statics // estadisticas
totaliter = istep-1; // statics // estadisticas
}

```

```

////////////////////////////////////
// Proceso de modelizacion con archivo de entrada de nodos inicializados
// nodesfilename=archivo de nodos inicializados
////////////////////////////////////
USInt PricketLonquist::Modelization(ChString nodesfilename)
{
    ifstream nodestream;
    Char inbuffer[256+1];
    Char format[256+1];
    USInt steps;
    Int dimrows, dimcols, r, c;
    BaseReal delta, deltainc, error, vt0, vt1, vsf1, vh, vq;

    nodestream.open(nodesfilename); // abrir flujo entrada
    for (nodestream.getline(inbuffer, sizeof(inbuffer)-1);
         (inbuffer[0] == '/') && (nodestream.fail() == 0);
         nodestream.getline(inbuffer, sizeof(inbuffer)-1));
    // tomar primer linea valida

    // buscar, convertir y guardar las 11 variables
    if (sscanf(inbuffer, COMMON_SCANNING, &steps, &delta, &deltainc,
               &error, &dimcols, &dimrows, &vt0, &vsf1, &vh, &vq, &format)
    != 11)
    {
        throw ("Error COMMON_SCANNING | Argumentos iniciales ");
        return (-1);
    }

    if ( (dimcols != 0) && (dimrows != 0) )
    {
        Cols(dimcols);
        Rows(dimrows);

        MatricesAllocation(); // allocate matrices // alojar matrices

        Initialization(vt0, vsf1, vh, vq); // inicializarlas

        for (nodestream.getline(inbuffer, sizeof(inbuffer)-1);
             (nodestream.fail() == 0) && (nodestream.gcount() > 1);
             nodestream.getline(inbuffer, sizeof(inbuffer)-1))
        {
            // check validity // verificar validéz
            if (inbuffer[0] == '/')
            {
            }
            // buscar, convertir y guardar las 7 variables
            else if (sscanf(inbuffer, SPECIFIC_SCANNING,
                           &c, &r, &vt0, &vt1, &vsf1, &vh, &vq) == 7)
            {
                if ( (c >= MinCols() && c <= MaxCols()) && (r >= MinRows() &&
                r <= MaxRows()) )
                {
                    t0[c][r] = vt0;
                    t1[c][r] = vt1;
                    sf1[c][r] = vsf1;
                    h[c][r] = vh;
                    q[c][r] = vq;
                }
            }
        }
    }
}

```



```

        else
        {
            throw("Error SPECIFIC_SCANNING");
        }
    }
    else
    {
        throw("Error SPECIFIC_SCANNING y scanf() != 7");
    }
}
nodestream.close(); // close input stream // cerrar flujo entrada
Simulation(steps, delta, deltainc, error, (Byte)format[0]); // simu-
lation // simulación
}
// Total number of iterations // Numero total de iteraciones
return (totaliter);
}

////////////////////////////////////
// Proceso de modelizacion con parametros indicativos de cantidad de pasos,
// dimensiones fila-columna de la matriz, delta de tiempo acumulado,
// acumulado de error maximo, etc.
// dimrows=filas
// dimcols=columns
// steps=pasos de ciclado
// delta=delta de tiempo inicial
// deltainc=incremento del tiempo en cada paso
// error=error maximo acumulado admisible
// format=formato matriz o encolumnado x-y-value
// ivt0=transmisividad inicial en la matriz
// ivsf1=almacenamiento inicial en la matriz
// ivh=nivel inicial uniforme en la matriz
// ivq=bombeo inicial uniforme en la matriz
// r=fila del punto de bombeo
// c=columa del punto de bombeo
// vt0=transmisividad en el punto
// vt1=transmisividad en el punto
// vsf1=almcacenamiento en el punto
// vh=nivel inicial en el punto
// vq=bombeo en el punto
////////////////////////////////////
USInt PricketLonquist::Modelization(Int dimrows, Int dimcols, USInt steps,
BaseReal delta,
BaseReal deltainc,
BaseReal error, Byte format,
BaseReal ivt0, BaseReal
ivsf1, BaseReal ivh, BaseReal ivq,
Int r, Int c,
BaseReal vt0, BaseReal
vt1, BaseReal vsf1,
BaseReal vh, BaseReal
vq)
{
    // Precondition // Precondicion
    if ( (dimcols != 0) && (dimrows != 0) )
    {
        Cols(dimcols);
    }
}

```

```

    Rows(dimrows);

    MatricesAllocation(); // allocate matrices // alojar matrices

    Initialization(ivt0, ivsfl, ivh, ivq); // inicializarlas

    // pumpin point // punto de bombeo
    if ( (c >= MinCols() && c <= MaxCols()) && (r >= MinRows() && r <=
MaxRows()) )
    {
        t0[c][r] = vt0;
        t1[c][r] = vt1;
        sfl[c][r] = vsfl;
        h[c][r] = vh;
        q[c][r] = vq;
    }
    else
    {
        cout<<"Error en el Punto de Bombeo ";
    }
    // Simulation itself // Simalacion en si misma
    Simulation(steps, delta, deltainc, error, format); // simulación
}

// Total number of iterations // Total de iteraciones
return (totaliter);
}

////////////////////////////////////
// Proceso de modelizacion con parametros indicativos de cantidad de pasos,
// dimensiones fila-columna de la matriz, delta de tiempo acumulado,
// acumulado de error maximo, etc.
// dimrows=filas
// dimcols=columnas
// steps=pasos de ciclado
// delta=delta de tiempo inicial
// deltainc=incremento del tiempo en cada paso
// error=error maximo acumulado admisible
// format=formato matriz o encolumnado x-y-value
// r=fila del punto de bombeo
// c=columna del punto de bombeo
// vt0=transmisividad en el punto
// vt1=transmisividad en el punto
// vsfl=almacenamiento en el punto
// vh=nivel inicial en el punto
// vq=bombeo en el punto
////////////////////////////////////
USInt PricketLonquist::Modelization(Int dimrows, Int dimcols, USInt steps,
BaseReal delta,
BaseReal deltainc,
BaseReal error, Byte format,
Int r, Int c,
BaseReal vt0, BaseReal
vt1, BaseReal vsfl,
BaseReal vh, BaseReal
vq)
{

```

```

// Precondition // Precondicion
if ( (dimcols != 0) && (dimrows != 0) )
{
    Cols(dimcols);
    Rows(dimrows);

    MatricesAllocation(); // allocate matrices // alojar matrices

    Initialization(vt0, vsf1, vh, 0.0); // inicializarlas

    // pumpin point // punto de bombeo
    if ( (c >= MinCols() && c <= MaxCols()) && (r >= MinRows() && r <=
MaxRows()) )
    {
        t0[c][r] = vt0;
        t1[c][r] = vt1;
        sf1[c][r] = vsf1;
        h[c][r] = vh;
        q[c][r] = vq;
    }
    else
    {
        cout<<"Error en el Punto de Bombeo";
    }
    // Simulation itself // Simalacion en si misma
    Simulation(steps, delta, deltainc, error, format); // simulation
}

// Total number of iterations // Total de iteraciones
return (totaliter);
}

```

### 8.3 Anexo 3: Algoritmos de inicialización del Modelo Prickett-Lonnquist

Lo que sigue a continuación, es la implementación completa en lenguaje C++ de la clase modelo Chaos, con los algoritmos de iniciación implementados, estocásticos y de fractales.

Primeramente, se describe el generador de números aleatorios utilizado en esta implementación, el cual, parte de un numero base o semilla, y luego arma vectores de acumulación .

```

////////////////////////////////////
// Numero Base de generacion aleatoria
#define __SEED 161803398
////////////////////////////////////

////////////////////////////////////
// Random Generator class // Clase Generador Random
class __random_generator {
protected:
    unsigned long table[55];
    size_t index1;
    size_t index2;
public:
    unsigned long operator()(unsigned long limit) {
        index1 = (index1 + 1) % 55;
        index2 = (index2 + 1) % 55;
        table[index1] = table[index1] - table[index2];
        return table[index1] % limit;
    }
}

```

```

    void seed(unsigned long j);
    __random_generator(unsigned long j) { seed(j); }
};

// Seed assign method // Metodo de asignacion de base aleatoria
void __random_generator::seed(unsigned long j) {
    unsigned long k = 1;
    table[54] = j;
    for (size_t i = 0; i < 55; i++) {
        size_t ii = 21 * i % 55;
        table[ii] = k;
        k = j - k;
        j = table[ii];
    }
    for (int loop = 0; loop < 4; loop++) {
        for (i = 0; i < 55; i++)
            table[i] = table[i] - table[1 + (i + 30) % 55];
    }
    index1 = 0;
    index2 = 31;
};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Random generator instance // Instancia de Generador Aleatorio
__random_generator rd(__SEED);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Proprietary random number function // Funcion propietaria de numero aleatorio
unsigned long __long_random(unsigned long limit) {
    return rd(limit);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Constructot
ChaosMModel::ChaosMModel(int x_dim,int y_dim)
{
    // Matrix dimension // Dimension de la matriz
    dimx = x_dim;
    dimy = y_dim;
};

// Set Dimensiones // Establecer dimensiones de Matriz
void ChaosMModel::SetDimensions(int x_dim,int y_dim)
{
    // Matrix dimension // Dimension de la matriz
    dimx = x_dim;
    dimy = y_dim;
};

```

### Método de Lorenz

El método de Lorenz inicializa un volumen y/o superficie en base a tres ecuaciones con polinomios, siguiendo un patrón de generación de tipo fractal.

```

void ChaosMModel::Initialize_Lorenz(float a,float b,float c,float dt,BaseReal
**matrix, BaseReal value)
{
double x = 0.0, y = 0.0, z = 0.0,
dx = 0.0, dy = 0.0, dz = 0.0;
long cont = 0;
unsigned int xx = 0, yy = 0;
const long limit = min(10000,dimx*dimy);

for(x = y = z = 1 ; cont < limit ; cont++)
{
dx = a*(y-x);
dy = x*(b-z)-y;
dz = x*y-c*z;
x += dt*dx;
y += dt*dy;
z += dt*dz;
xx = (long)(x*16) % dimx;
yy = (long)(z*16) % dimy;
if(xx < dimx && yy < dimy && xx >= 0 && yy >= 0)
matrix[xx][yy] = value;
}
};

```

### Método de Henon

El método de Henon inicializa una superficie en base a dos ecuaciones de polinomios, siguiendo un patrón de generación de tipo fractal.

```

void ChaosMModel::Initialize_Henon(float a,float b,BaseReal **matrix, BaseReal
value)
{
long i = 0;
unsigned int xp = 0, yp = 0;
double x = 0.0, y = 0.0, xx = 0.0, yy = 0.0;
const long limit = min(10000,dimx*dimy);

for(i = 0 ; i < limit ; i++)
{
xx = 1+y-a*x*x;
yy = b*x;
x = xx;
y = yy;
xp = (long)x % dimx;
yp = (long)y % dimy;
if(xp < dimx && yp < dimy && xp >= 0 && yp >= 0)
matrix[xp][yp] = value;
}
};

```

### Método de Sierpinski

El método de Sierpinski inicializa una superficie en base a dos ecuaciones de polinomios, inicializados en forma aleatoria en arreglos vectoriales, y siguiendo un patrón de generación de tipo fractal.

```

void ChaosMModel::Initialize_Sierpinski(BaseReal **matrix, BaseReal value)
{
float a[3]={0.5,0.5,0.5};
float b[3]={0,0,0};

```

```

float c[3]={0,0,0};
float d[3]={0.5,0.5,0.5};
float e[3]={1,50,50};
float f[3]={1,1,50};
float xx = 0.0, x = 0.0,
      yy = 0.0, y = 0.0;
unsigned long i = 0;
long k = 0;
unsigned int xp = 0, yp = 0;
const unsigned long limit = min(10000,dimx*dimy);

for(i = 0 ; i < limit ; i++)
{
    k = __long_random( sizeof(a)/sizeof(a[0]) );
    xx = a[k]*x+b[k]*y+e[k];
    yy = c[k]*x+d[k]*y+f[k];
    x = xx;
    y = yy;
    xp = (long)(x * 4.5 + 0.5) % dimx;
    yp = (long)(y * 4.5 + 0.5) % dimy;
    if(xp < dimx && yp < dimy && xp >= 0 && yp >= 0)
        matrix[xp][yp] = value;
}

};

```

### Método MonteCarlo

El método de Monte Carlo inicializa una superficie en base a una iteración aleatoria de posicionamiento dentro de la misma, con valores predeterminados, siguiendo un patrón estocástico.

```

void ChaosMModel::Initialize_MonteCarlo(BaseReal **matrix, BaseReal value,
UInt ix, UInt jx)
{
    UInt          i, j, in, jn,      // indices de matrices
              n,      // indice de ciclo
              middlex, middley, // puntos medios de los ejes
              quadrx, quadry;      // cuadrante actual
    Boolean      end;               // indicador para terminar ciclo
    const unsigned long limit = min(10000,dimx*dimy);

    // ciclo principal de generación de base aleatoria y posicionamiento
    for (middlex = dimx/2, middley = dimy/2,
        n = 0;
        n < limit;
        n++)
    {
        // posiciones resguardadas en la matriz (i,j)
        i = in = ix;
        j = jn = jx;
        // señal de fin de ciclado a FALSO
        end = FALSE;
        // ciclado SET de nodos según posiciones aleatorias generadas
        while (end == FALSE)
        {
            quadrx = (UInt)__long_random(dimx);
            quadry = (UInt)__long_random(dimy);
            if ((quadrx > middlex) && (quadry > middley)) // cuadrante 1
            {

```

```

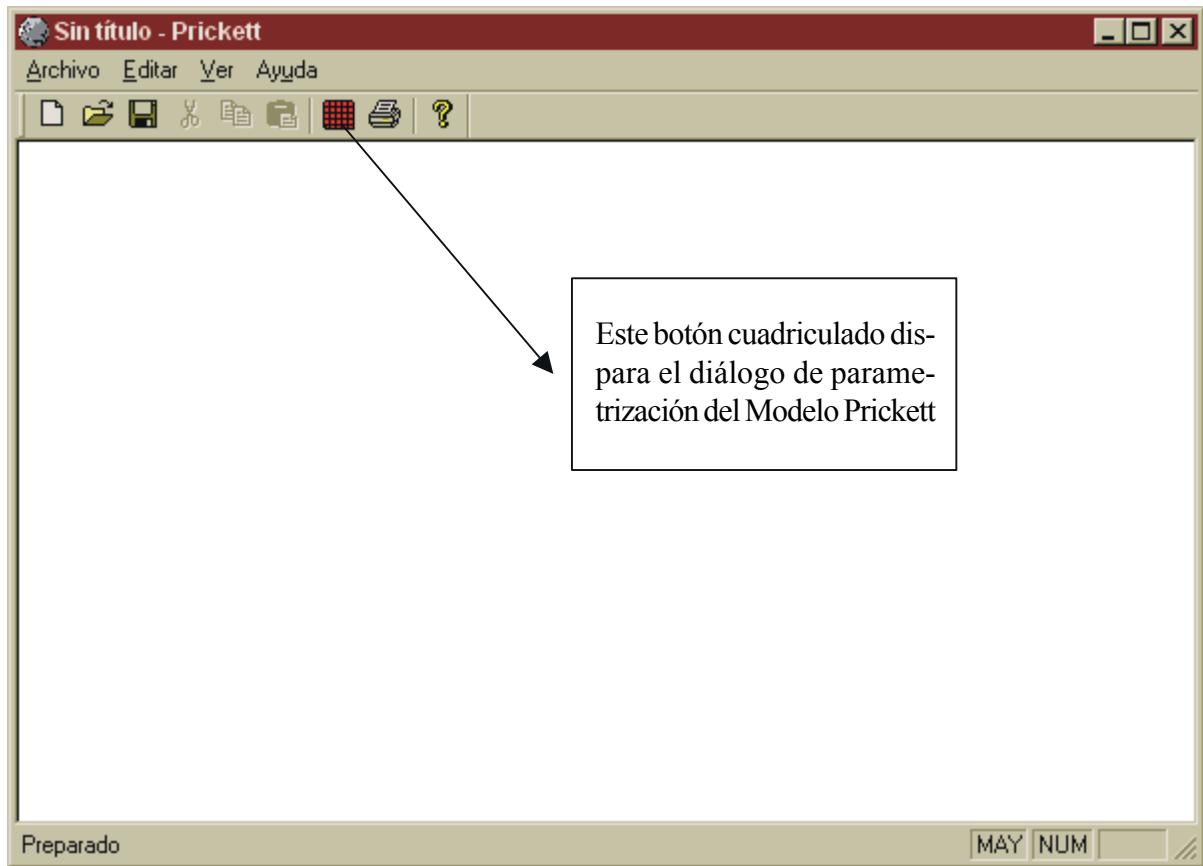
        if ((i+1) < dimx && matrix[i+1][j] != value)
            in = i + 1;
        else
        {
            matrix[i][j] = value;
            end          = TRUE;
        }
    }
else if ((quadrx > middlex) && (quadry < middley)) // cuad. 2
{
    if ((j-1) < dimy && (j-1) > 0 && matrix[i][j-1] != value)
        jn = j - 1;
    else
    {
        matrix[i][j] = value;
        end          = TRUE;
    }
}
else if ((quadrx < middlex) && (quadry < middley)) // cuad. 3
{
    if ((i-1) < dimx && (i-1) >= 0 && matrix[i-1][j] != value)
        in = i - 1;
    else
    {
        matrix[i][j] = value;
        end          = TRUE;
    }
}
else // cuadrante 4
{
    if ((j+1) < dimy && matrix[i][j+1] != value)
        jn = j + 1;
    else
    {
        matrix[i][j] = value;
        end          = TRUE;
    }
}
i = in;
j = jn;
}
};

```

////////////////////////////////////

### 8.4 Anexo 4: Diálogo Principal de la aplicación modelo Prickett-Lonnquist con iniciación estocástica / fractal

MENU DE SELECCIÓN PARA EJECUTAR EL MODELO PRICKETT-LONQUIST





**NOMENCLATURA DE PARAMETROS para 2 DIMENSIONES**

- Niveles piezométricos = (m) metros
- Factor de almacenamiento (confinado) ( $m^3 / m$ ) = metros cúbicos x metro lineal
- Caudales de bombeo constante ( $m^3 / día$ ) = metros cúbicos x día
- Transmisividad entre 2 nodos cualesquiera ( $m^3 / día / m$ ) = metros cúbico x día x metro lineal

**Modelo Prickett-Lonnquist**

**Inicialización**

Cantidad de Filas: 10    Cantidad de Columnas: 10

Cantidad de Ciclos: 10    Mostrar ciclo...: 1

Delta del tiempo: 1    Incremento de Delta: 0.1

Error admisible: 3

Transmisividad inicial: 0.1

Nivel freático inicial: 0

Almacenamiento inicial: 0.1

Evaporación inicial: 0

**Punto de Bombeo**

Fila: 5    Columna: 5

Transmisividad inicial: 0.1

Nivel freático inicial: 0

Almacenamiento inicial: 0.1

Bombeo inicial: 1

OK    Cancel    **Ejecutar**

El tiempo hace referencia a un período, y cada ciclo tiene un incremento de un delta de ese tiempo.

El punto de bombeo marca el pozo de extracción de agua en la grilla. Tiene coordenadas X e Y.

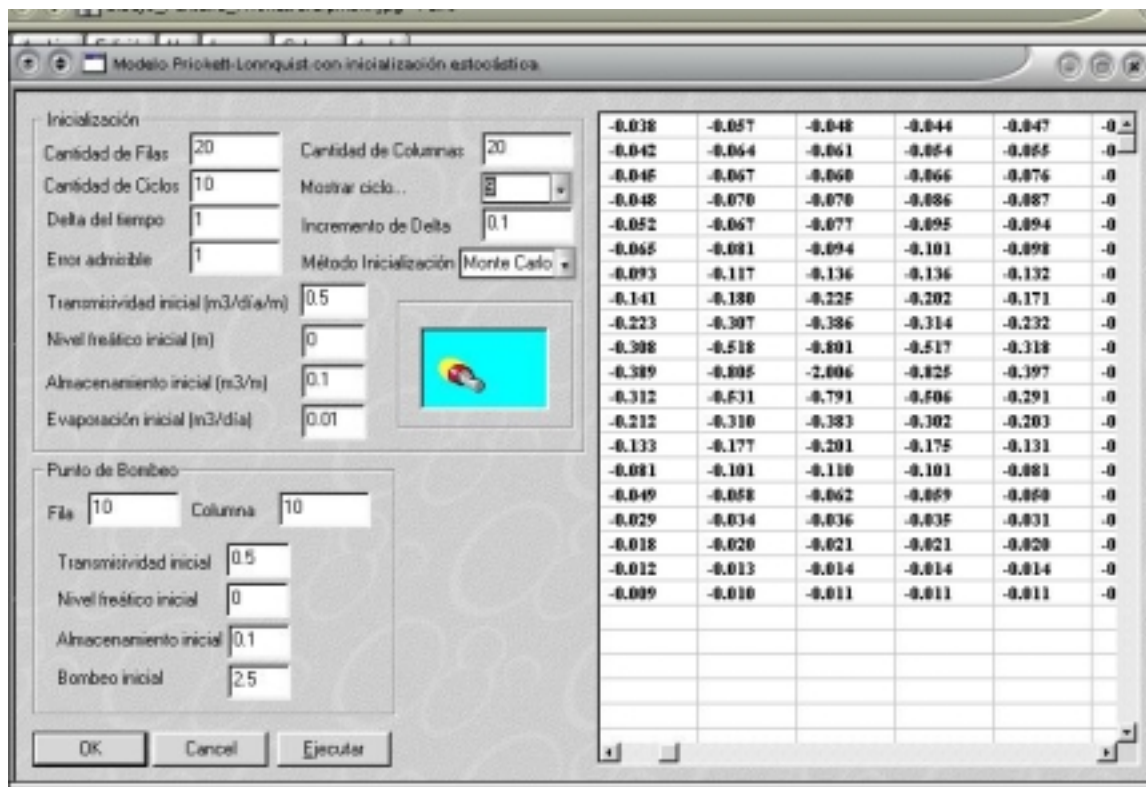
La grilla es una abstracción bidimensional del terreno a ser modelizado. Cada celda representa el nivel de agua subterránea, o nivel freático.

Esta lista detalla la cantidad de ciclos modelizados, disponibles para ser vistos en la grilla. Hay tantos ciclos como se especificuen, luego de Ejecutar el modelo.

Este botón ejecuta el modelo según los parámetros, hasta una cantidad N de ciclos simulados. Al terminar, representa el primer ciclo en la grilla. Para cambiar de ciclo, elegirlo de la lista propia.

-0.001					
-0.001					
-0.002					
-0.004					
-0.006					
-0.007					
0.006					
0.004	-0.008	-0.022	-0.053	-0.114	-0
0.003	-0.005	-0.011	-0.022	-0.041	-0
0.002	-0.003				-0

## NOMENCLATURA DE PARAMETROS INCLUYENDO METODOS ESTOCASTICOS



- Cantidad de filas / columnas: Número de nodos laterales que contendrá la superficie a modelizar.
- Cantidad de Ciclos: Iteraciones del modelo antes de la convergencia del error máximo admitido.
- Delta del tiempo: Tiempo referencial entre cada iteración. Representativo.
- Incremento del delta: Incremento del delta del tiempo referencial. Representativo.
- Error admisible: Error máximo acumulado admisible por las iteraciones del modelo.
- Transmisividad inicial: indicador de esta propiedad al inicio del modelo, en m<sup>3</sup>/día/m.
- Nivel freático inicial: valor por omisión de iniciación de la superficie, en cierto nivel freático. En metros.
- Almacenamiento inicial: indicador de esta propiedad al inicio del modelo. En m<sup>3</sup>/m.
- Bombeo inicial: indicador de esta propiedad al inicio del modelo, generalmente representativo de una explotación artificial (planta de bombeo de agua). En m<sup>3</sup>/día.
- Evaporación inicial : indicador de esta propiedad al inicio del modelo. En m<sup>3</sup>/día.

### 8.5 Anexo 5: Aplicación Comercial “Visual ModFlow”

**Visual MODFLOW** es el estándar americano en el área de modelización para transporte de contaminantes y flujo de agua subterránea en 3-D. Combina una interfase poderosa e intuitiva, soportando calibración automática. Permite al usuario:

1. Diseñar gráficamente la grilla de modelación.
2. Digitalizar y visualizar los parámetros del modelo.
3. Ejecutar el modelo.
4. Calibrar el modelo manual o automáticamente.  
Mostrar e interpretar los resultados.

[A] Vistas superficiales y por cortes longitudinales en la grilla

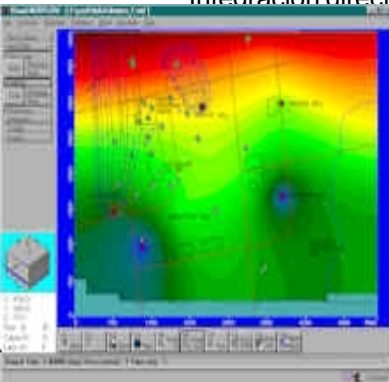
### Característica principales de Visual ModFlow

#### Entrada de Datos

- Interfase gráfica poderosa y facil de usar
- Vista interactiva superficial y por cortes
- Importación de mapas bases en formato AutoCAD
- Rotación y alineación de modelos con respecto a los mapas bases
- Diseño automático de grillas y correcciones
- Niveles del terreno importados directamente desde archivos

#### Capacidades de Simulación

- Soporta los modelos de contaminantes denominados MT3DMS, RT3D, MT3D99 y WinPEST
- Integración directa con Win32 MODFLOW del USGS



· Interfaz interactiva para convergencia de la solución del modelo  
 · Motor de ecuaciones WHS Solver  
 · Optimización de lotes para simulaciones pesadas y de larga ejecución

[B] Contornos y depresiones con matices de colores

#### Calibración del Modelo

- Selección de puntos de calibración y estadísticas
- Calibración automática mediante WinPEST

#### Visualización del Modelo

- Contorneado mediante matices de colores, de resultados y puntos de referencia
- Detalles gráficos detallados
- Impresión en alta resolución en cualquier impresora o plotter
- Exportación de visualizaciones a formatos gráficos AutoCad, ESRI y Windows
- Animación 3-D de avanzada

*[C] Visualización y animación 3D**Aplicaciones profesionales de Visual ModFlow*

- Evaluación de Sistemas de Aguas Subterráneas
  - Delimitar zonas de captura para perforaciones de explotación de aguas subterráneas
  - Simular la atenuación natural en la contaminación del agua subterránea
  - Diseñar y optimizar las localizaciones de perforaciones para proyectos comerciales
  - Determinar los contaminantes y su exposición para evaluación de riesgos
- Visual MODFLOW fue lanzado por primera vez al mercado en Agosto de 1994 y actualmente está siendo usado por mas de 5.000 consultores, reguladores y educadores en todo el mundo.

**9. Bibliografía y Referencias Profesionales****PROFESIONALES**

- Dr. Jorge Santa Cruz, Geólogo de la Universidad de Buenos Aires, Argentina. Doctor en Gestión de Aguas Subterráneas de la Universidad Complutense de Madrid, España. Director del Programa Nacional de Tecnología Sustentable en Aguas Subterráneas (PNTSAS) del Instituto Nacional del Agua, Argentina. (\*)
- Dr. Adrián A. Silva Busso, Geólogo de la Universidad de Buenos Aires, Argentina. Especialista en hidrogeología de la Universidad Complutense de Madrid, España. Miembro del Programa Nacional de Tecnología Sustentable en Aguas Subterráneas (PNTSAS) del Instituto Nacional del Agua, Argentina. (\*)

**LIBROS, TESIS, INFORMES y PORCIONES DE SOFTWARE**

1. ANGUITA, CUSTODIO y otros, "Teoría Básica de Modelos Analógicos y Digitales de Acuíferos", Servicio Geológico de Obras Públicas y Gabinete de Cálculo del Ministerio de Obras Públicas y Transportes, España, 1976.
2. BIERMAN, BONNINI y otros. «Análisis Cuantitativo para la Toma de Decisiones», IRWIN, 8va. Edición, México, 1999.
3. CERNADAS, Martín. "Sistema Nacional de Aguas Subterráneas (SIAS)", Informe Final de Beca del PNTSAS, INA. Librería del Instituto Nacional del Agua, Argentina, 1996.
4. LINSLEY, KOHLER y PAULUS, "Hidrología para Ingenieros", 2da. Edición, traducida al castellano, McGraw-Hill de México S.A., México, 1977.
5. PRICKET y LONNQUIST, "Selected Digital Computer Techniques for Groundwater Resource Evaluation", Illinois State Water Supply, Bulletin 55, EE.UU., 1971.
6. SANTA CRUZ, Jorge. "Estudio del Acuífero Puelches", Librería del Instituto Nacional de Agua, Argentina, 1975.
7. SILVA BUSSO, Adrián. "Estudio del Acuífero Puelches, Argentina", Librería del Instituto Nacional de Agua, Argentina, 1995.
8. STEVENS, "Object-Oriented Graphics Programming in C++", Academic Press Inc., EE.UU., 1994.
9. WALPONE Y MYERS, "Probabilidad y Estadística", 4ta Edición, Editorial McGraw-Hill, México-USA, 1992.

10. WALTON, "Selected Analytical Methods for Well and Aquifer Evaluation", Illinois State Water Supply, Bulletin 49, EE.UU., 1962.
11. HORNSBY, "Agua Subterránea : El Recurso Oculto", documento SL48S, Instituto de Ciencias Alimenticias y Agrícolas, Universidad de Florida. Primera Impresión, Enero de 1986.
12. ESRI, "Map Objects LT 2.0", Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100 USA.

### SITIOS EN INTERNET

1. Banco de datos de modelos numéricos. Universidad de Hannover. <http://www.fi.uni-hannover.de/~am/isess97/isess1997.html>
2. CSEP (Computational Science Education Project), email verena@csep1.phy.ornl.gov, referencias sobre método Monte Carlo y modelización en aguas subterráneas [http://www.cped.ornl.gov/cad\\_cp/text/imy.html](http://www.cped.ornl.gov/cad_cp/text/imy.html)
3. Interfase visual para el modelo ModFlow. <http://www.visual-modflow.com>, por la Waterloo Hydrogeologic.
4. InquiMap. Software modular para la Modelación y análisis de acuíferos. [http://www.geoandsoft.com/espanol/ground\\_water.htm](http://www.geoandsoft.com/espanol/ground_water.htm)
5. Instituto Nacional de Agua : <http://www.ina.gov.ar>
6. Modelo de Simulación de Acuíferos. Universidad de Kassel, Dinamarca. [http://www.wiz.uni-kassel.de/model\\_db/mdb/asm.htmls\\_y Proyecto ECOBAS \(base ecológica de modelos numéricos\) :](http://www.wiz.uni-kassel.de/model_db/mdb/asm.htmls_y Proyecto ECOBAS (base ecológica de modelos numéricos) : http://eco.wiz.uni-kassel.de/ecobas.html) <http://eco.wiz.uni-kassel.de/ecobas.html>
7. Paradigm. Componentes estocásticos del modelo Paradigm para modelado de sistemas de acuíferos. <http://ncl.ac.uk/wrgi/wrsl/gwms/gwms.pdf>
8. Prickett – Lonquist Acuífer Simulation Model, en el banco de datos de modelos numéricos de la Universidad de Kassel. [http://www.wiz.uni-kassel.de/model\\_db/mdb/plasm.html](http://www.wiz.uni-kassel.de/model_db/mdb/plasm.html)
9. Programa Nacional de Tecnología Sustentable en Aguas Subterráneas (PNTSAS), pntsas@ina.gov.ar, (Pertenece al Instituto Nacional de Agua) AU Ezeiza-Cañuelas, Tramo J. Newbery Km 1,620 (1804) Ezeiza, Pcia. Buenos Aires, Argentina. TE: (54 11) 4480-0295. <http://www.ina.gov.ar>
10. Revistas Técnicas CPRM, Brasil. URL: <http://www.cprm.gov.br/rev/agua0401.html>
11. Sistema Nacional de Aguas Subterráneas (SIAS) : <http://www.sias.freesevers.com>. Sobre banco de registros hidrogeológicos en el ámbito de la República Argentina, actualizados hasta el año 1995. Producto del PNTSAS, del Inst. Nacional del Agua, de la RR. AA.
12. Departamento de Ciencias de Suelos y Agua. Servicio Cooperativo de Extensión del Instituto de Ciencias Alimenticias y Agrícolas, Universidad de Florida. <http://edis.ifas.ufl.edu>

### LICENCIAS DE SOFTWARE Y AVISOS LEGALES

**Sobre MAP OBJECTS LT 2.0 , "U.S. Government Restricted/Limited Rights: Use, duplication, and disclosure by the U.S. Government are subject to restrictions as set forth in FAR section 52.227-14 Alternate III (g)(3) (JUN 1987), FAR section 52.227-19 (JUN 1987), DFARS section 252.227-7015 (NOV 1995) [Technical Data], and/or DFARS section 227.7202 [Computer Software], as applicable. Contractor/Manufacturer is Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100 USA."**

Sobre este documento, "Queda prohibida la reproducción total o parcial por cualquier medio y para cualquier destino, sin el permiso del autor. Copyright ©2002 , Martín D. Cernadas. Hecho el depósito que marca la Ley de Derecho de Autor, 11.723"

Sobre el software adjunto, Modelo Prickett Estocástico, versión 2.0, "Queda autorizada la distribución en formato binario ejecutable, sin restricciones. Copyright ©2002 , Martín D. Cernadas. Hecho el depósito que marca la Ley de Derecho de Autor, 11.723, y sucesivos decretos y leyes modificatorios."