

# IMPLEMENTACIÓN EFICIENTE DE UN CÓDIGO EN PARALELO PARA RESOLVER LA ECUACIÓN DE FLUJO SUBTERRÁNEO EN MEDIOS HETEROGÉNEOS Y ANISÓTROPOS

## EFFICIENT PARALLEL COMPUTING IMPLEMENTATION OF A CODE FOR SOLVING THE GROUNDWATER FLOW EQUATION IN HETEROGENEOUS AND ANISOTROPIC MEDIA

Álvarez-Villa, Oscar D.<sup>1</sup>; Ospina, Santiago<sup>2</sup>; Pérez, Kevin; Monsalve, Gaspar<sup>2</sup>; Blessent, Daniela<sup>3</sup>; Vélez, Jaime Ignacio<sup>2</sup>

### Resumen

En este trabajo se presenta una implementación altamente eficiente que resuelve en paralelo la Ecuación en Derivadas Parciales de Flujo Subterráneo (EDPFS) en régimen transitorio y en tres dimensiones, linealizada aplicando diferentes esquemas de diferencias finitas. El código desarrollado considera los siguientes aspectos: (i) medios heterogéneos donde la conductividad hidráulica puede variar abruptamente en el espacio, (ii) medios anisótropos mediante el tensor no diagonal de conductividades hidráulicas, (iii) geometrías irregulares del acuífero y parametrizaciones variables en tiempo y espacio de sus condiciones de contorno, (iv) la variabilidad espacio-temporal de las acciones exteriores actuando sobre dicho acuífero. Para efectos de ejecución computacional, se han adoptado las herramientas ofrecidas por la interfaz de altas prestaciones PETSc, la cual permite la ejecución en paralelo del planteamiento de los sistemas de ecuaciones del modelo, la resolución en paralelo de dichos sistemas mediante el método del gradiente conjugado y el acceso a diferentes esquemas de pre condicionamiento. Para probar la pertinencia y eficiencia de la implementación creada se realizaron tres experimentos computacionales. Primero, la modelación del flujo en un acuífero bidimensional homogéneo anisótropo, con el fin de evaluar la representación matemática de la anisotropía del medio. Segundo, la solución de la EDPFS para medio fracturado, considerando propiedades hidráulicas generadas a partir de un modelo estocástico equivalente de medio poroso. En este experimento se realizó la comparación de las soluciones obtenidas con otras de referencia simuladas en MODFLOW, con lo cual se comprobó la pertinencia y precisión de la implementación. Tercero, un acuífero sintético en forma de cubo, heterogéneo y anisótropo con diferentes niveles de discretización espacial. El desempeño de la solución numérica de la EDPFS se evalúa utilizando varios preconditionamientos para los diferentes escenarios de discretización propuestos. En este caso se concluye que, a pesar de que la selección del preconditionamiento influye en el aumento o disminución de la eficiencia de ejecución del modelo, la principal limitación computacional radica en la adecuada selección de un hardware que pueda soportar adecuadamente las ejecuciones en paralelo.

**Palabras claves:** Flujo subterráneo, anisotropía, heterogeneidad, implementación en paralelo, escalabilidad.

### Abstract

In this paper a highly efficient parallel computing implementation for solving the transient 3D Groundwater Flow Equation (GFE) via finite differences is presented. The code was developed considering the following aspects: (i) heterogeneous porous media with a spatially varying hydraulic conductivity, (ii) anisotropic media, including the non-diagonal hydraulic conductivity tensor, (iii) irregular aquifer geometry, with time and space varying boundary conditions, (iv) time and space varying external actions on the aquifer. To enhance the computational performance of the models, the tools offered by the PETSc interface have been used. They allow the parallel building of the model's systems of equations, the parallel resolution of such equations via the conjugate gradient method, and the access to different techniques of preconditioning. To test the accuracy and efficiency of the implementation, three computational experiments were carried out. First, a groundwater flow modelling on a 2D homogeneous anisotropic aquifer was considered for evaluating the accurate mathematical representation of the anisotropy of the hydraulic conductivity. Second, the groundwater flow in fractured rocks was evaluated considering hydraulic properties derived from a stochastic model of equivalent porous media. This particular experiment was made to test the equivalency of the obtained simulations, and others obtained via MODFLOW. Third, a cubical synthetic, heterogeneous and anisotropic aquifer was considered to test the efficiency of the finite differences resolution of the GFE using different spatial discretizations. The performances of these solutions were evaluated by incorporating different kinds of preconditionings. Despite the fact that the selection of the preconditioning increases the speed of the computation, the main aspect that enhances the performance of the models is the appropriate selection of a hardware that supports the parallel computations.

**Keywords:** Groundwater flow, anisotropy, heterogeneity, parallel computing, scalability.

<sup>1</sup> Universidad de Antioquia. oscar.alvarez@gottaingenieria.com; kevin.perez@gottaingenieria.com; jivelezu@unal.edu.co

<sup>2</sup> Universidad Nacional de Colombia. sospinar@gmail.com; gmonsalvem@unal.edu.co

<sup>3</sup> Universidad de Medellín. dblessent@udem.edu.co

## 1. INTRODUCCIÓN

Durante los últimos años, la disponibilidad de fuentes hídricas superficiales para el abastecimiento de diferentes tipos de uso se ha reducido debido a la fuerte presión que se ha hecho sobre ellas, ocasionada tanto por el constante aumento de las demandas, como por efecto de la degradación continua de su calidad. Por su parte, las aguas subterráneas se consideran como una excelente opción para mitigar los desabastecimientos de recurso hídrico ya que, primero, la lenta respuesta de los acuíferos regula naturalmente su disponibilidad y, segundo, presentan una mejor calidad que las aguas superficiales, y por lo tanto los tratamientos necesarios para su potabilización son menos costosos (Sahuquillo *et al.*, 2009). En consecuencia, se hace necesario desarrollar herramientas potentes para gestionar adecuadamente el recurso hídrico subterráneo. Al respecto, una de las formas más aceptadas actualmente para evaluar la disponibilidad y el estado de las aguas subterráneas es a través de modelos computacionales de flujo y transporte, herramientas que permiten entender y simular las direcciones y velocidades del flujo en el acuífero, al tiempo que representan la propagación de contaminantes en él. El éxito en la aplicación de estos modelos se fundamenta, entre otras cosas, en lograr la mejor representación posible de la variabilidad espacio-temporal de los parámetros en juego, puesto que el comportamiento de un soluto está fuertemente influenciado por la heterogeneidad y la anisotropía del medio geológico (Sánchez Barrero, 2011), lo cual se podría ver reflejado en la propagación anómala de la pluma y en cambios de su velocidad de dilución (Salamon, 2006; Aggelopoulos y Tsakiroglou, 2012).

No sólo la heterogeneidad de los parámetros hidráulicos del acuífero afecta la dinámica de los procesos relacionados con el transporte, sino que la anisotropía de la conductividad hidráulica puede modificar el patrón espacio-temporal de las líneas de corriente, con lo cual se alterarían las direcciones preferenciales de avance de una pluma de soluto por efecto de la advección. De ello sigue que es crucial lograr una representación detallada de la dinámica de flujo subterráneo al momento de modelar el transporte. Sin embargo, tradicionalmente se ha utilizado el tensor diagonal de conductividades como una manera simplificada para considerar la anisotropía del medio (Harbaugh, 2005). Ahora bien, diferentes autores han desarrollado trabajos donde plantean una representación parcial o total de la anisotropía de la conductividad hidráulica. Anderman *et al.* (2002) desarrollan el paquete MODFLOW Layer Variable Direction Anisotropy (LDVA); esquema conceptual de modelación que considera únicamente la anisotropía por capas del modelo, pero sin incluir las componentes cruzadas del tensor de conductividades hidráulicas. Por su parte, Li *et al.* (2010) proponen la modelación del flujo subterráneo tridimensional para régimen permanente usando el

tensor anisótropo de conductividades hidráulicas y un esquema de diferencias finitas de 18 bloques adyacentes.

No solo los efectos de la heterogeneidad del medio se ven reflejados en la solución numérica, sino que también el modelo matemático y su eficiencia computacional de resolución inciden en los resultados obtenidos (Anderson y Woessner, 1992). En general, la solución de un modelo hidrogeológico detallado utilizando técnicas numéricas tradicionales puede ser altamente costosa en términos de recursos informáticos. Por un lado, la memoria virtual del ordenador puede ser insuficiente para almacenar las matrices que surgen al resolver las ecuaciones dinámicas durante el proceso de modelación. Por otro lado, si la discretización espacio-temporal es muy fina, su ejecución computacional puede ser muy lenta, imposibilitando su utilización en aplicaciones prácticas (Álvarez-Villa, 2014). Considerando lo anterior, el presente trabajo introduce una implementación en paralelo de un código eficiente que resuelve tridimensionalmente la ecuación de flujo subterráneo para medios heterogéneos y anisótropos mediante las diferencias finitas. La solución de los sistemas de ecuaciones lineales resultantes se realizó mediante métodos iterativos para matrices dispersas (Saad, 2003), para los cuales se analizó la influencia de diferentes tipos de preconditionamiento en la aceleración de la ejecución de los modelos y en la optimización del recurso computacional disponible.

## 2. MARCO CONCEPTUAL

### 2.1 La ecuación en derivadas parciales de flujo subterráneo

La Ecuación en Derivadas Parciales de Flujo Subterráneo (EDPFS) en régimen transitorio de un medio heterogéneo saturado, considerando un sistema de coordenadas cartesianas no alineado con los ejes de anisotropía del medio, puede ser expresada como (Bear y Cheng, 2010):

$$\nabla \cdot (\mathbf{K} \nabla h) + q = S \frac{\partial h}{\partial t} \quad (1)$$

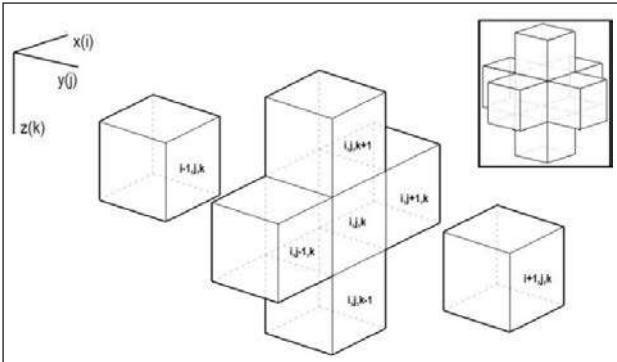
donde  $h=h(x,y,z,t)$  [L] es la altura piezométrica;  $q=q(x,y,z,t)$  [L<sup>3</sup>/T] es el término fuente-sumidero de flujos,  $S=S(x,y,z)$  [L<sup>2</sup>] es el coeficiente de almacenamiento específico y, en la formulación más general, el tensor de conductividades hidráulicas es  $\mathbf{K}$ , el cual tiene la forma:

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \quad (2)$$

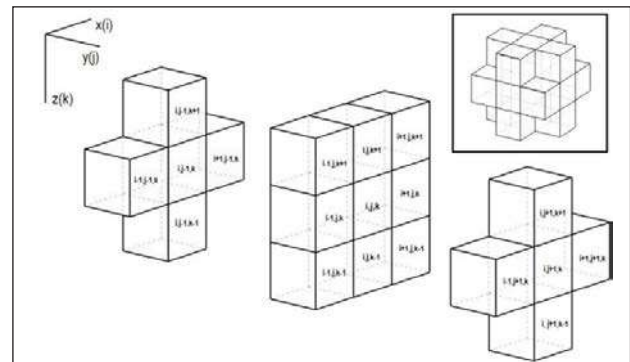
y representa una entidad matricial que contiene información sobre tres vectores de flujo mutuamente perpendiculares en el espacio. Si dichos vectores son paralelos a los ejes de un sistema de referencia dado,

se dice que éstos definen las direcciones principales y el tensor es una matriz diagonal. Por el contrario, si el sistema de referencia se encuentra rotado, los vectores asociados a las direcciones principales

del tensor se deben descomponer en cada eje del nuevo sistema, obteniendo proyecciones paralelas y perpendiculares que se manifiestan en la matriz tensorial con nueve componentes presentada en la ecuación (2).



**Figura 1.** Organización de los seis bloques del grafo de DF para el esquema de resolución de la EDPFS propuesto por Harbaugh (2005)



**Figura 2.** Organización de los diecinueve bloques del grafo de DF para el esquema de resolución de la EDPFS propuesto por Li *et al.* (2010)

**2.2. Esquemas de resolución por diferencias finitas**

En este trabajo se han tenido en cuenta dos esquemas de resolución de la EDPFS mediante las Diferencias Finitas (DF). La primera de ellas, propuesta por Harbaugh (2005), considera un esquema de siete

bloques, de manera que se capturen las componentes ortogonales de la conductividad hidráulica del tensor de la ecuación (2). Este esquema de solución por DF tiene intrínseco el planteamiento del grafo que se muestra en la Figura 1 y matemáticamente se expresa como:

$$\begin{aligned} &\Delta y|_j \Delta z|_k \frac{2K_{xx}|_{i-1/2,j,k}}{\Delta x|_{i-1} + \Delta x|_i} (h_{i-1,j,k}^t - h_{i,j,k}^t) + \Delta y|_j \Delta z|_k \frac{2K_{xx}|_{i+1/2,j,k}}{\Delta x|_{i+1} + \Delta x|_i} (h_{i+1,j,k}^t - h_{i,j,k}^t) + \dots \\ &+ \Delta x|_i \Delta z|_k \frac{2K_{yy}|_{i,j-1/2,k}}{\Delta y|_{j-1} + \Delta y|_j} (h_{i,j-1,k}^t - h_{i,j,k}^t) + \Delta x|_i \Delta z|_k \frac{2K_{yy}|_{i,j+1/2,k}}{\Delta y|_{j+1} + \Delta y|_j} (h_{i,j+1,k}^t - h_{i,j,k}^t) + \dots \\ &+ \Delta x|_i \Delta y|_j \frac{2K_{zz}|_{i,j,k-1/2}}{\Delta z|_{k-1} + \Delta z|_k} (h_{i,j,k-1}^t - h_{i,j,k}^t) + \Delta x|_i \Delta y|_j \frac{2K_{zz}|_{i,j,k+1/2}}{\Delta z|_{k+1} + \Delta z|_k} (h_{i,j,k+1}^t - h_{i,j,k}^t) + \dots \\ &+ q_{i,j,k} = \Delta x|_i \Delta y|_j \Delta z|_k S_{i,j,k} \frac{h_{i,j,k-1}^t - h_{i,j,k}^{t-1}}{\Delta t} \end{aligned} \tag{3}$$

donde  $\Delta y|_j$ ,  $\Delta x|_i$  y  $\Delta z|_k$  son los tamaños del bloque localizado en las coordenadas  $(i,j,k)$  en las direcciones de los ejes cartesianos  $y$ ,  $x$  y  $z$ , respectivamente;  $K_{yy}$ ,  $K_{xx}$  y  $K_{zz}$  son las conductividades interbloque aparentes en las direcciones de los ejes cartesianos  $y$ ,  $x$  y  $z$ , respectivamente;  $h_{i,j,k}^t$  es la altura piezométrica del bloque con coordenadas  $(i,j,k)$  en el tiempo  $t$  y  $S_{i,j,k}$  es el coeficiente de almacenamiento del bloque con coordenadas  $(i,j,k)$ . El contexto espacial de estas variables se presenta en la Figura 1.

Por su parte, Li *et al.* (2010) proponen resolver la ecuación (1) por medio de un esquema de DF que considera diecinueve bloques, como se muestra en la Figura 2, de forma que se capturen las componentes no diagonales del tensor de conductividades hidráulicas. Matemáticamente, considerando un esquema implícito hacia adelante para las derivadas parciales temporales, el esquema de DF de Li *et al.* (2010) se expresa como:

$$\begin{aligned} &Ah_{i,j+1,k}^{t+1} + Bh_{i,j,k}^{t+1} + Ch_{i+1,j+1,k}^{t+1} + Dh_{i-1,j,k}^{t+1} + Eh_{i+1,j,k}^{t+1} + Fh_{i-1,j,k}^t + Gh_{i,j+1,k}^t + Hh_{i,j,k}^t + Ih_{i+1,j+1,k}^t + \dots \\ &+ Jh_{i-1,j,k}^t + Kh_{i+1,j,k}^t + Lh_{i-1,j,k}^t + Mh_{i,j+1,k}^t + Nh_{i,j,k}^t + Oh_{i+1,j+1,k}^t + Ph_{i-1,j,k}^t + Qh_{i+1,j,k}^t + Rh_{i-1,j,k}^t + \dots \\ &+ Sh_{i,j+1,k}^t + q_{i,j,k} \Delta x|_i \Delta y|_j \Delta z|_k S_{i,j,k} = S_{i,j,k} (h_{i,j,k}^t - h_{i,j,k}^{t-1}) \end{aligned} \tag{4}$$

donde los coeficientes  $(A \sim S)$   $[L^2/T]$  pueden ser consultados en el trabajo de Li *et al.* (2010) y su evaluación depende de las conductividades hidráulicas de bloque, las características geométricas de la malla usada para la discretización espacial del área de interés y del tamaño de paso de las DF

temporales. Los sub-índices  $i$ ,  $j$  y  $k$  representan la discretización espacial en las direcciones de los ejes cartesianos  $x$ ,  $y$  y  $z$ , respectivamente. En un dominio espacial discretizado por DF tridimensionales mediante  $n$  prismas rectangulares e incluyendo diferentes tipos de condiciones de contorno, se

podrá usar (3) o (4) para obtener un Sistema de Ecuaciones Lineales Disperso (SELD) en cada paso temporal, el cual se plantea de la siguiente manera:

$$\left(\mathbf{A} - \frac{\mathbf{D}}{\Delta t}\right) \mathbf{h}^t + \mathbf{q} = -\frac{\mathbf{D}}{\Delta t} \mathbf{h}^{t-1} \quad (5)$$

donde  $AER^{n \times n}$  es una matriz de conductividades equivalentes  $[L^2/T]$ ,  $h^t \in R^n$  es un vector con las alturas piezométricas al final del  $t$ -ésimo intervalo  $[L]$ ,  $h^{t-1} \in R^{n \times n}$  es un vector con las alturas piezométricas al inicio de dicho intervalo,  $D \in R^{n \times n}$ , es una matriz de almacenamientos específicos  $[L^2]$  y  $q \in R^n$  es el vector de flujos  $[L^3/T]$ . Alternativamente, la ecuación (5) puede escribirse como:

$$\mathbf{Ch}^t = \mathbf{d} \quad (6)$$

SELD que se resuelve sucesivamente durante cada intervalo de simulación para obtener el vector de alturas piezométricas  $h^t$ ;  $C$  es la matriz de coeficientes del sistema que depende de los parámetros hidráulicos del acuífero y de las condiciones de contorno, y  $d$  es un vector de términos independientes de flujos internos y externos del acuífero.

### 2.3. Resolución de los modelos de flujo

Una característica importante de la expresión (6) es que para ambos esquemas de resolución por DF considerados en este trabajo,  $C$  siempre es una matriz simétrica definida positiva y dispersa con estructura invariante. Ello permite adoptar esquemas de resolución muy eficientes basados en la proyección de (6) sobre un sub-espacio de Krylov construido como  $K_n = \langle b, Cb, C^2b, \dots, C^{n-1}b \rangle$ , particularmente el método del Gradiente Conjugado (GC, Hestenes y Stiefel, 1952).

El GC es un método iterativo para resolver SELD ampliamente utilizado en la actualidad por su simpleza y por qué, teóricamente, se alcanza la solución en, a lo sumo,  $n$  iteraciones (Trefethen y Bau III, 1997). Desde el punto de vista práctico, el uso de GC es interesante porque, primero, el número de operaciones asociadas a los productos vectoriales y matriz-vector es bajo y asegura la velocidad en la convergencia de la solución; segundo, no se modifican las estructuras de las matrices de los SELD durante las iteraciones; y tercero, las operaciones involucradas pueden implementarse paralelamente para lograr la aceleración de la ejecución del modelo. De acuerdo con Van der Vorst y Dekker (1988), la eficiencia en la ejecución del GC depende de la distribución de los autovalores de  $C$  y del número de condición espectral, es decir, la razón entre sus autovalores mínimo y máximo. Una forma de acelerar la ejecución del GC es someter a la matriz  $C$  a un preconditionamiento; técnica que mejora el número de condición  $k$  de dicha matriz. Así, sea  $M \in R^{n \times n}$  otra matriz que aproxima a  $C$ , pero es fácilmente

invertible, entonces se resuelve  $Ch^t = d$  solucionando indirectamente el sistema (Saad, 2003):

$$\mathbf{M}^{-1}\mathbf{Ch}^{t+1} = \mathbf{M}^{-1}\mathbf{d} \quad (7)$$

donde  $M$  es llamada la matriz de preconditionamiento. Si  $k(M^{-1}C) \ll k(C)$  entonces (7) tendrá una convergencia mucho más rápida en los métodos iterativos para resolver (6). Incluyendo el preconditionamiento, el algoritmo del GC Precondicionado (GCP) se plantea para un vector inicial de solución  $v_0$  y una tolerancia en el residuo  $\epsilon$ , mediante el siguiente esquema iterativo (Hill, 1990; Larabi y De Smedt, 1994; Saad, 2003; Flores, 2009):

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{b} - \mathbf{C}\mathbf{v}_0 \\ \mathbf{p}_0 &= \mathbf{z}_0 = \mathbf{M}^{-1}\mathbf{r}_0 \\ \text{Para } k &= 0, 1, 2, \dots \text{ hacer} \\ \alpha_k &= -\frac{\mathbf{r}_k \mathbf{M}^{-1} \mathbf{r}_k}{\mathbf{p}_k \mathbf{C} \mathbf{p}_k} \\ \mathbf{h}_{k+1} &= \mathbf{h}_k - \alpha_k \mathbf{p}_k \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{C} \mathbf{p}_k \\ \beta_k &= -\frac{\mathbf{r}_{k+1} \mathbf{M}^{-1} \mathbf{r}_{k+1}}{\mathbf{r}_k \mathbf{M}^{-1} \mathbf{r}_k} \\ \mathbf{p}_{k+1} &= \mathbf{M}^{-1} \mathbf{r}_k + \beta_k \mathbf{p}_k \\ \text{Hasta que } \mathbf{r}_k &< \epsilon \end{aligned} \quad (8)$$

### 2.4. Técnicas de pre-condicionamiento

En este trabajo se realiza un estudio de la eficacia computacional de diferentes esquemas de preconditionamiento del GCP para acelerar la resolución de SELD, como los descritos en la ecuación (6), por lo cual los métodos tenidos en cuenta se explican brevemente en los siguientes apartados.

#### 2.4.1 Métodos clásicos iterativos de resolución

Los métodos iterativos clásicos pueden ser vistos como preconditionamiento de los métodos iterativos. Así, sea  $Ch = d$  el problema que se quiere resolver. Para una matriz  $Q \in R^{n \times n}$  invertible, si  $h$  es un punto fijo de la aplicación  $F(h) = h + Q^{-1}(d - Ch)$ , puede plantearse la siguiente forma iterativa:

$$\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{Q}^{-1}(\mathbf{d} - \mathbf{C}\mathbf{h}_i) = \mathbf{h}_i + \quad (9)$$

Pueden plantearse diferentes preconditionamientos de acuerdo con la escogencia de la matriz de iteración  $Q$ . Para fines de notación, la matriz  $C$  se descompone como sigue (Benzi, 2002; Saad, 2003):

$$\mathbf{C} = \mathbf{D} + \mathbf{U} + \mathbf{L} \quad (10)$$

donde  $D \in R^{n \times n}$  denota la matriz diagonal cuyas entradas son las de la diagonal de  $C$ ,  $L \in R^{n \times n}$  denota la parte estrictamente inferior de  $C$  y  $U \in R^{n \times n}$  su parte estrictamente superior. En el caso del método de

Jacobi, se considera tomar Q como la matriz D. En el caso del método de Gauss-Seidel, se adopta a Q como la matriz triangular inferior de C, es decir  $Q = D + L$ . En el caso del método SOR (*Successive Overrelaxation*) Q es una variación paramétrica de Gauss-Seidel dada por  $Q = D + \omega L$ , donde  $\omega$  es el parámetro de relajación.

#### 2.4.2. Factorización LU incompleta

Meijerink y Van der Vorst (1977) fueron los primeros en reconocer el uso potencial de la factorización incompleta como preconditionamiento para resolver SELD. Una factorización incompleta es deseable porque las matrices resultantes respetan el patrón de dispersión de la matriz original, hasta cierto punto. Así, lo que se busca es controlar que la cantidad de procesamiento del preconditionamiento no se incremente excesivamente. La factorización LU Incompleta (ILU) consiste en realizar una descomposición LU, pero descartando algunos de sus elementos. Este procedimiento, en el ámbito del preconditionamiento del GCP, tiene dos importantes ventajas. Primero, la factorización ILU aplicada sobre C arroja matrices con menores radios espectrales que el asociado a la matriz C original; segundo, los requerimientos de almacenamiento son más bajos que los requeridos por la factorización LU de C. Sea P el patrón de dispersión impuesto a la factorización, así:

$$P \subset \{(i,j) : i \neq j; 1 < i, j < n\} \tag{11}$$

el cual representa un conjunto de componentes  $(i,j)$  de la matriz C que no serán descartados en el procedimiento de factorización. La primera elección de P para imponer un patrón de factorización ILU fue realizada con una matriz C bandeada, escogiendo una matriz de preconditionamiento que también era bandeada. Concretamente, si P coincide con las posiciones no nulas de C, la factorización ILU se denomina sin relleno, ILU(0). Esta aproximación es efectiva para solucionar los SELD que surgen en discretizaciones de bajo orden en EDP elípticas escalares (Benzi, 2002), como en el caso de la EDPFS resuelta por DF planteadas en la ecuación (5). Desafortunadamente, a medida que la discretización espacial de las DF usadas para resolver el flujo subterráneo se hace más fina o irregular, las aproximaciones sin relleno resultan ser muy burdas y debe adoptarse algún esquema de llenado (Hill, 1990; Naff y Banta, 2008). Gustafsson (1978) y Watts III (1981) propusieron esquemas ILU(p) de llenado basados en el cálculo de un índice por elemento a medida que avanza la eliminación gaussiana incompleta. Así, el nivel inicial de llenado de un elemento de una matriz dispersa es:

$$lev_{ij} = \begin{cases} 0 & \text{si } c_{i,j} \neq 0 \\ \infty & \text{en otro caso} \end{cases} \tag{12}$$

el cual se renueva a medida que el elemento es modificado por una eliminación gaussiana. Durante la factorización ILU(), el nivel de llenado se evalúa como en cada iteración sobre la -ésima fila; luego el patrón de dispersión para un nivel de llenado se define como el conjunto:

$$P = \{(i,j) : lev_{ij} > p\} \tag{13}$$

La motivación de este nivel de llenado es que, para matrices diagonalmente dominantes, mientras mayor sea el nivel de llenado de un elemento, menor es su valor absoluto, con lo cual la matriz de errores entre ambas matrices exhibe elementos muy cercanos a cero y la factorización es más precisa. A pesar de que en la mayoría de las ocasiones, aplicar el nivel de relleno mejora el preconditionamiento con respecto a las versiones sin llenado, la desventaja principal es que requiere almacenar muchos elementos de llenado pequeños en valor absoluto que contribuyen muy poco en la disminución del número de condición de C.

En algunos casos, un preconditionamiento más eficiente se obtiene ejecutando una factorización por descarte ILU( $\tau$ ) donde los elementos de llenado se aceptan con base en su magnitud. Así pues, se define a la tolerancia de descarte como el número entero positivo  $\tau$  que actúa como límite inferior admisible para los valores absolutos de los elementos de llenado; es decir, se permite llenar los elementos de las matrices L y U si y sólo si  $c_{ij} > \tau$ . La principal desventaja de ello es que resulta difícil escoger una tolerancia de descarte, por lo cual se hace necesario recurrir a un proceso de ensayo y error hasta encontrar un valor satisfactorio. Un problema es que no puede conocerse a priori la cantidad de almacenamiento necesaria para la factorización (Saad, 2003).

Saad (1994) propone una estrategia de preconditionamiento dual por umbral ILU( $p, \tau$ ), que opera fijando una tolerancia de descarte y un número de elementos de relleno en cada fila de las matrices factores L y U. A diferencia de las factorizaciones incompletas discutidas hasta ahora, este método dual decide el relleno de los elementos de L y U de forma dinámica aplicando una regla de eliminación sobre el elemento si su magnitud es menor que cierto umbral  $\tau$  y luego manteniendo los p elementos de relleno de mayor magnitud por fila. Por consiguiente, los parámetros  $\tau$  y p pueden modificarse convenientemente hasta alcanzar un preconditionamiento apropiado de acuerdo al problema a solucionar.

#### 2.4.3. Métodos algebraicos de múltiples mallas

Los Métodos Algebraicos de Múltiples Mallas (MAMM) se han establecido como una técnica robusta para resolver SELD. Al mismo tiempo, han sido usados como técnicas de preconditionamiento eficientes

para la ejecución del GCP (Tatebe, 1993). La idea de los MAMM es eliminar los residuales que se obtienen mediante técnicas basadas en la relajación, usando un conjunto de matrices o mallas más gruesas que se superponen sobre la matriz original, para las cuales se usa un método directo de solución de los sub-problemas resultantes. Así, un preconditionador MAMM construye una jerarquía para dichas mallas de matrices y el conjunto de operaciones de transferencia entre ellas usando únicamente la información contenida en la matriz de coeficientes. La ventaja de estas técnicas de preconditionamiento es que pueden tratar fácilmente casos en los cuales la modelación envuelve geometrías irregulares o mallas no estructuradas (Pereira *et al.*, 2006). Los pasos usados por los MAMM para el preconditionamiento dentro de un esquema iterativo de solución de SELD, definiéndose una malla como un conjunto de nodos extraídos de la matriz original, son como sigue.

Primero, se establecen los nodos de la matriz original, a partir de los cuales se extraen subconjuntos de nodos para diferentes tamaños de malla, ya sea gruesa o fina. Lo anterior equivale a extraer un conjunto de múltiples mallas sobre las cuales operará el preconditionamiento MAMM. Segundo, se establecen las operaciones de transferencia entre las múltiples mallas, de la siguiente manera:

$$\begin{array}{l} \text{Prolongación} \\ \text{Restricción} \end{array} \quad \begin{array}{l} P_k: V_{k+1} \rightarrow V_k \\ R_k = P_k^T \end{array} \quad (14)$$

donde  $V_k$  son matrices aproximación de las soluciones en los diferentes niveles de las mallas,  $R_k$  son las matrices de restricción sobre cada malla y  $P_k$  es la matriz de interpolación para cada nivel de malla. Tercero, se define la jerarquía matricial de las múltiples mallas, así:

$$A_{k+1} = R_k A_k P_k \quad (15)$$

Para terminar, se define la correspondiente técnica de suavizado, es decir, el método directo de solución de la malla gruesa que sea más conveniente a las características del problema. Más información acerca de detalles matemáticos de los MAMM puede encontrarse en Briggs *et al.* (2000).

### 3. SOFTWARE E IMPLEMENTACIÓN

La solución de grandes SELD es un problema que involucra gran cantidad de cómputos, el cual, inicialmente, fue tratado manualmente y luego programando computadoras para que realizaran dichas operaciones automáticamente. La creación de lenguajes de programación permitió a la comunidad científica unificar esfuerzos y desarrollar una colección de funciones y rutinas eficientes y portables para operaciones básicas, llamada BLAS (*Basic Linear Algebra Subprograms*, Lawson *et al.*, 1979). A partir de BLAS se crearon programas más

sofisticados para solucionar SELD por factorización directa (LU, Cholesky, QR, SVD, etc.); en particular se pueden destacar los paquetes de librerías de EISPACK (Smith *et al.*, 1976), LINPACK (Dongarra *et al.*, 1979) y LAPACK (Anderson *et al.*, 1999), anotando que este último es el más ampliamente aceptado en la comunidad científica debido a que es mucho más eficiente con el uso de la memoria que EISPACK y LINPACK. Hoy en día, con la posibilidad de trabajar paralelamente con muchos procesadores, librerías como BLAS y LAPACK, no bastan para hacer eficiente la solución de grandes SELD, y por eso se utilizan herramientas como PETSc (Balay *et al.*, 2012) o MUMPS (Amestroy *et al.*, 1998) para abordar estos problemas.

El PETSc, desarrollado por el *Agronne National Laboratory* de los Estados Unidos, es un software pensado como una herramienta para la computación científica portable y extensible, compuesta por una colección de rutinas y estructuras de datos que proveen bloques de construcción para la implementación de códigos de gran escala en computadoras en paralelo. Una de las grandes ventajas de PETSc es que puede ser usado en sistemas de procesadores que trabajen en paralelo (clusters, supercomputadoras o computadoras personales). Ello se logra a través de librerías especializadas de programación paralela como MPICH u OpenMPI, que se encargan de transmitir información entre procesadores de una forma ordenada y eficiente, a través de los lenguajes de programación tradicionales como FORTRAN, C, C++ y Python. Las librerías del PETSc permiten la creación de familias de objetos de matrices, vectores e índices de conjuntos, en las que el mismo PETSc se encarga de realizar operaciones eficientes entre ellos, incluso si se trabaja en paralelo, lo que permite crear códigos más limpios y eficientes para la solución de sistemas de ecuaciones.

Si bien el PETSc es un entorno de alto nivel para solucionar grandes SELD, también puede considerarse como un software que requiere un entendimiento avanzado de sus herramientas para lograr un código adecuado en términos de eficiencia. Debido a esto, en el marco de este documento se han desarrollado dos implementaciones del PETSc a la solución de la EDP de flujo subterráneo. La primera implementación se trata de la adecuación un código de flujo subterráneo llamado FDPACK (Álvarez-Villa, 2014) a las librerías del PETSC. FDPACK simula numéricamente el comportamiento del flujo en dos dimensiones en un medio poroso heterogéneo. La implementación del FDPACK bajo las librerías de PETSc puede ser consultada en la página web <http://soilros.github.io/FDPACK/>. Por otro lado, la implementación llamada ANISOFLOW es un programa implementado en lenguaje FORTRAN 90, que utiliza las herramientas computacionales de paralelismo proporcionadas por PETSc, enfocado a usar y evaluar los elementos teóricos mostrados a lo largo de este documento y que, además, se busca

que sirva como alternativa al MODFLOW cuando las soluciones secuenciales de éste no sean lo suficientemente eficientes para la modelación del flujo subterráneo. El código fuente del programa puede ser encontrado en el repositorio online del programa <http://github.com/SoilRos/ANISOFLOWPACK>.

#### 4. EXPERIMENTOS NUMÉRICOS

##### 4.1. Modelo simple bidimensional

Este modelo consiste en un dominio computacional bidimensional con 11 de bloques de lado y profundidad unitaria en régimen permanente de flujo con conductividad hidráulica uniforme en el espacio de 0.5 m/d, para diferentes condiciones de anisotropía. Este modelo se ha implementado para comprobar que las características de la anisotropía de la conductividad hidráulica del medio se estén representando de una manera cualitativamente adecuada en FDPACK y ANISOFLOW, tanto considerando las expresiones propuestas por Harbaugh (2005), como las recomendadas por Li *et al.*, (2010), de acuerdo con los esquemas de linealización por DF presentados en la sección 2.2. El borde más exterior del dominio tiene alturas piezométricas impuestas de 10 m, mientras que en el centro del dominio se tiene una altura piezométrica impuesta de 5 m. La Figura 3 muestra una representación de las características del acuífero considerado.

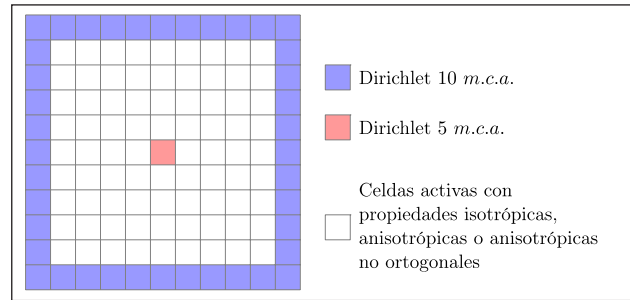


Figura 3. Dominio y condiciones de contorno, modelo bidimensional

En los experimentos computacionales se estudiaron tres medios homogéneos con conductividad hidráulica uniforme y diferentes condiciones de anisotropía: (i) medio isótropo, (ii) medio anisótropo con ejes de anisotropía paralelos a los ejes cartesianos, y (iii) medio anisótropo con ejes de anisotropía rotados un ángulo de 45° con respecto al Norte, con conductividades cruzadas de 0.4 m/d. Las alturas piezométricas simuladas se presentan en la Figura 4, donde se observa la coherencia de la variabilidad espacial de las alturas piezométricas simuladas con respecto a los planteamientos de anisotropía en los modelos. Cabe notar que los ejes de la elipse de descensos se orientan de acuerdo a la dirección de anisotropía impuesta a los medios de prueba.

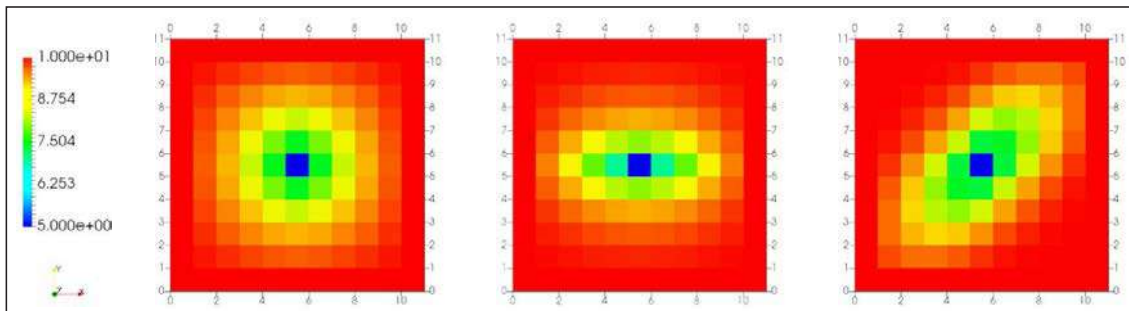
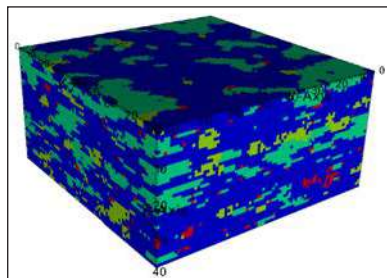


Figura 4. Solución del modelo de prueba (alturas piezométricas) para medio homogéneo isótropo (izquierda), anisótropo con ejes de anisotropía paralelos a los ejes cartesianos (centro) y anisótropo con ejes de anisotropía rotados 45°

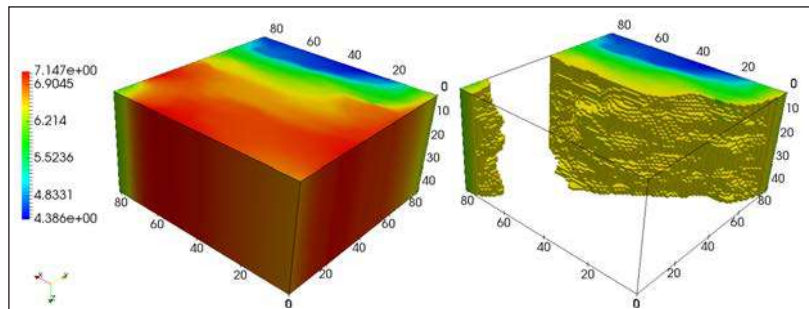
##### 4.2. Medio heterogéneo por facies geológicas de roca fracturada

La información considerada en el presente trabajo para medio fracturado corresponde a la utilizada por Blessent *et al.* (2011), la cual se ha levantado en la isla de Olkiluoto, Finlandia, sitio potencial para disponer residuos nucleares de alta actividad. Los datos de campo hacen alusión a roca fracturada, donde las propiedades físicas del material parental, la geometría, la extensión y la conectividad de las fracturas fueron caracterizadas a partir de núcleos de perforación, pruebas hidráulicas, estudios geofísicos y datos de pozos. Las principales zonas de fractura identificadas tienen una transmisividad mayor de  $1 \times 10^{-8}$  m<sup>2</sup>/s y una extensión de varios cientos de

metros. Con base en esta información se plantearon modelos igualmente probables de facies para un Medio Poroso Equivalente (MPE); planteamiento que ha requerido la definición de categorías de rocas, las cuales se obtuvieron analizando las conductividades hidráulicas obtenidas en los ensayos de interferencia y registros de densidad de fracturamiento. De acuerdo a ello, los MPE generados por simulación secuencial basada en la probabilidad de transición (Carle y Fogg, 1996; Carle, 1999) se componen de las siguientes categorías: (i) roca madre poco fracturada (SFB), (ii) lecho de roca escasamente a moderadamente fracturado (SMFB), (iii) lecho de roca de moderado a altamente fracturado (MHFB) y (iv) lecho de roca altamente fracturado (HFB).



**Figura 5.** Esquema de la zonificación de propiedades hidráulicas del medio (Azul: SFB, Verde: SMFB, Amarillo: MHFB y Rojo: HFB)



**Figura 6.** Alturas piezométricas simuladas por ANISOFLOW para el modelo tridimensional de facies planteado por Blessent *et al.* (2011) y contornos de dicha solución

El resultado del análisis geoestadístico de facies es un planteamiento de cuatro zonas de conductividades hidráulicas sobre un dominio espacial que tiene una longitud de 400 m en dirección Norte y Este, y 200 m en dirección vertical, con discretización regular de 5 m en cada dirección (ver la Figura 5), mientras que las características hidráulicas de cada una de las cuatro zonas se muestran en la Tabla 1. Las condiciones de contorno consideradas consisten en niveles piezométricos impuestos en los bordes laterales de la primera capa.

Tabla 1. Conductividades hidráulicas (m/s) según las facies del dominio espacial

Facies	Conductividad hidráulica según las facies		
	$K_x$	$K_y$	$K_z$
Facies 1 – SFB	$6.34 \times 10^{-8}$	$3.17 \times 10^{-9}$	$3.17 \times 10^{-9}$
Facies 2 – SMFB	$5.77 \times 10^{-7}$	$5.77 \times 10^{-7}$	$5.77 \times 10^{-7}$
Facies 3 – MHFB	$1.37 \times 10^{-6}$	$1.37 \times 10^{-6}$	$1.37 \times 10^{-6}$
Facies 4 – HFB	$8.00 \times 10^{-6}$	$8.00 \times 10^{-6}$	$8.00 \times 10^{-6}$

La EDPFS se solucionó en régimen permanente mediante el código ANISOFLOW para el medio heterogéneo por facies de roca fracturada acá considerado mediante el GC con preconditionamiento ILU y esquema dual por descarte ( $=0.0005$ ) y nivel de llenado ( $=2$ ); solución que se comparó con la obtenida aplicando el código MODFLOW (Harbaugh, 2005). Los resultados obtenidos se presentan la Figura 6, donde se muestran los niveles piezométricos y los contornos de estos. Un análisis cuantitativo

mostró que existe únicamente un error relativo máximo del 0.105 % entre ambas soluciones, el cual corresponde a una diferencia absoluta cercana a 0.016 m en los bloques donde se tienen las mayores alturas piezométricas. Este resultado permite validar indirectamente la aplicación del código ANISOFLOW en medios heterogéneos y anisótropos.

### 4.3. Medio altamente heterogéneo y anisótropo

Este modelo considera un acuífero cúbico de 400 m de lado, formado por un medio heterogéneo con zonas isotropas y anisotropas no paralelas a los ejes cartesianos, el cual fue generado pseudo-aleatoriamente con una discretización regular, cuyo tamaño de bloque depende del caso de análisis. Con el fin de realizar los experimentos de escalabilidad computacional se generaron discretizaciones de 25, 50, 100 y 200 bloques de lado que, respectivamente, son equivalentes a sistemas matriciales compuestos por 15625 (16K), 125000 (125K), 1000000 (1M) y 8000000 (8M) ecuaciones. Las respectivas configuraciones espaciales de cada modelo se muestran en la Figura 7. En dicha figura se observan tres zonas con diferentes características de anisotropía. Los bloques de color rojo (zona 1) corresponden a material isotropo y uniforme, los bloques de color verde (zona 2) corresponden a material isotropo con diferentes conductividades en las diferentes direcciones de los ejes coordenados; finalmente, los bloques de color azul corresponden a material totalmente anisotropo, es decir, se tienen componentes no nulos en todas las entradas del tensor de conductividades hidráulicas. Dichas conductividades hidráulicas, para cada una de las tres zonas, se plantearon como sigue:

$$K_1 = \begin{bmatrix} 5a & 0 & 0 \\ 0 & 5a & 0 \\ 0 & 0 & 5a \end{bmatrix}; K_2 = \begin{bmatrix} 5a & 0 & 0 \\ 0 & 5b & 0 \\ 0 & 0 & 5c \end{bmatrix}; K_3 = \begin{bmatrix} 5a & 2d & 0.1e \\ 2d & 5b & 0.1f \\ 0.1e & 0.1f & 5c \end{bmatrix} \quad (16)$$

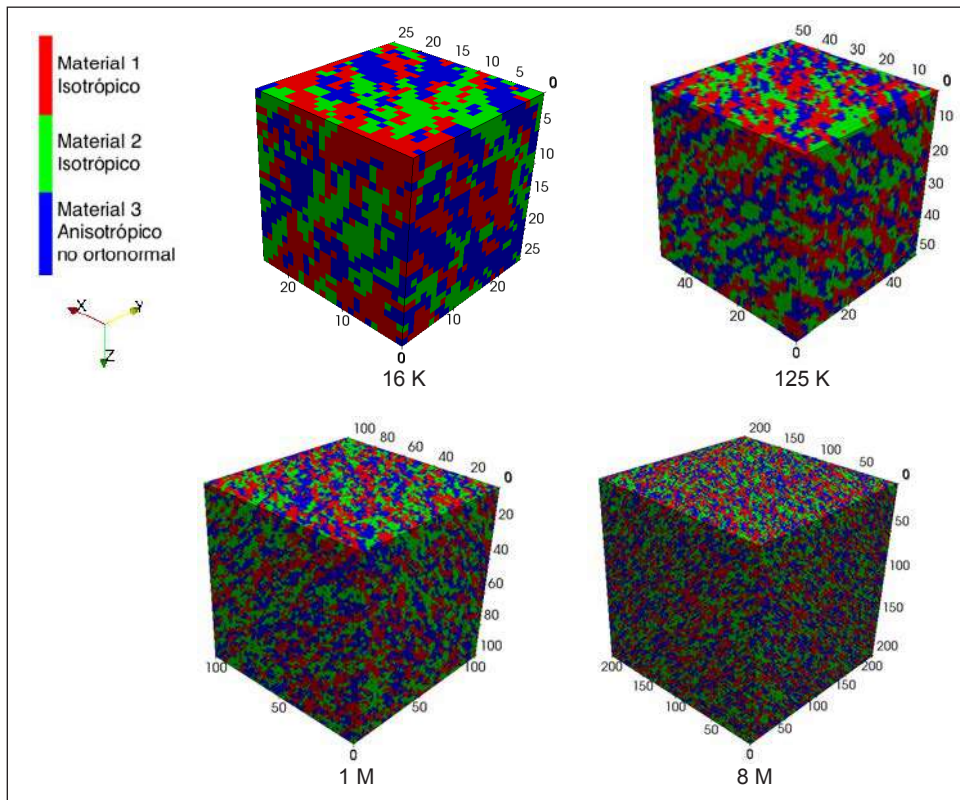
donde las variables  $a \sim f$  son numeros aleatorios uniformes entre cero y uno. Para seleccionar el tipo de zona que le corresponde a cada bloque se eligió copiar pseudo-aleatoriamente entre el índice

de zona de un bloque adyacente o un número pseudo-aleatorio uniforme, para así provocar un agrupamiento de zonas. Si el vecino no tiene asignado valor, éste no se toma en cuenta. Las



condiciones de contorno para resolver los modelos en régimen permanente consisten en una función sinusoidal que varía entre 5 y 10 m entre cada par de paredes opuestas del dominio sobre la primera capa de bloques de DF. Estas soluciones se utilizan como condiciones iniciales de los modelos resueltos en régimen transitorio y se presentan en la Figura 8. Para las simulaciones en régimen transitorio, a lo largo del horizonte temporal también se modifican las condiciones de contorno en los bordes del dominio espacial de manera sinusoidal, modificándose entre los niveles impuestos para las condiciones iniciales y 0 m cada 365 días; alteraciones que representan una variabilidad hidrológica controlada por un ciclo anual. Además, en el centro del dominio, es decir, en el bloque con coordenadas (200 m; 200 m) en

direcciones Este y Norte, y a profundidades de 0, 200 y 400 m, se imponen niveles piezométricos nulos durante un período de seis meses, seguido por un intervalo de la misma longitud en los que no se imponen niveles en dichos bloques. Este mecanismo representa indirectamente una extracción impuesta de agua mediante, por ejemplo, bombeo. Pasado un año de simulación, en los mismos lugares se repite la secuencia de descensos y recuperación de niveles, pero imponiendo 20 m en las celdas centrales. Es decir, un ciclo completo de descenso y recuperación del acuífero dura dos años. Las simulaciones realizadas con este modelo sintético tienen una duración total de cuatro años, con resolución temporal de un día.



**Figura 7.** Configuración de los medios heterogéneos usados para los experimentos de escalabilidad computacional de la solución por DF planteada para ANISOFLOW. Debajo de cada gráfico se muestra el número de ecuaciones asociado a la discretización

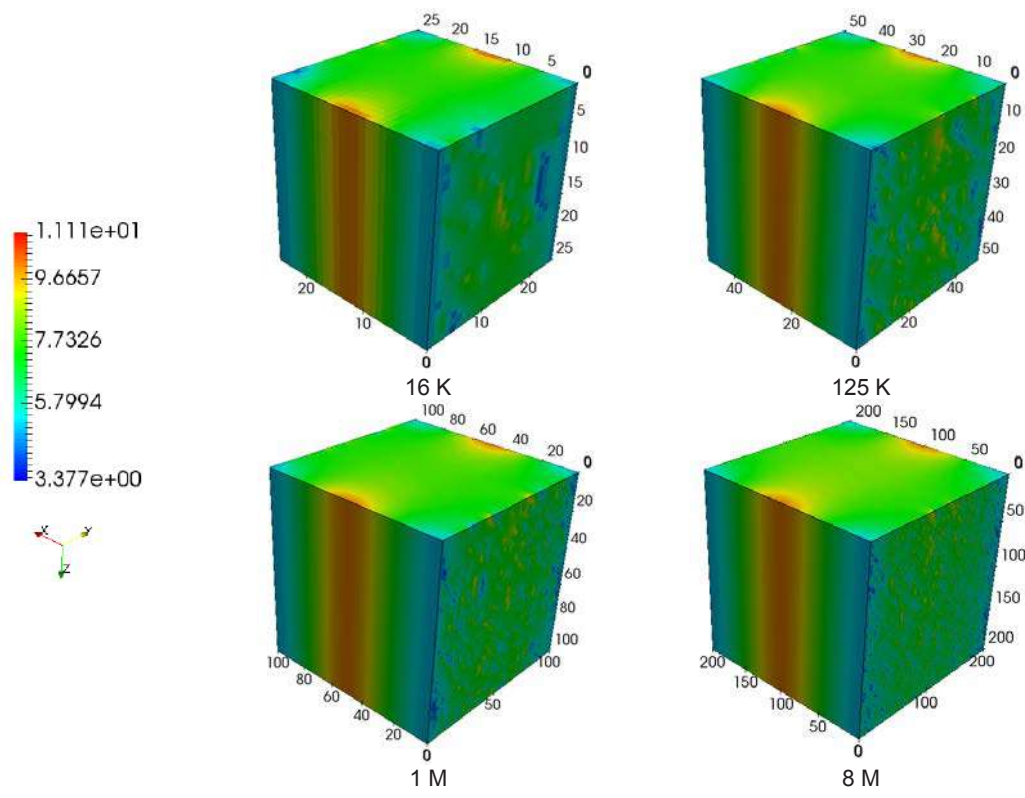
Como se ha dicho antes, se desea evaluar la influencia de los esquemas de preconditionamiento en el aumento o disminución de la eficiencia de resolución por el GCP de los SELD planteados por las diferentes soluciones de la EDPFS mediante las DF (sección 2.2) implementados en ANISOFLOW usando las herramientas de paralelización proporcionadas por PETSc. Dado lo anterior, los modelos de flujo subterráneo construidos para las diferentes discretizaciones espaciales (16K, 125K, 1M y 8M, ver la Figura 7) del acuífero sintético planteado, se han resuelto utilizando los siguientes preconditionadores en paralelo: Jacobi, Jacobi

Bloque y Algebraico de Múltiple malla (Multi-malla). También se considera la resolución mediante el GC sin preconditionamiento.

Debido a que los preconditionadores del tipo ILU no son eficientes cuando se paralelizan, los desarrolladores de PETSc implementaron otras formas para que sus beneficios computacionales se puedan usar en códigos en paralelo. Esto consiste en resolver la factorización ILU empleando las matrices que se encuentran guardadas localmente en cada procesador y usar el conjunto de las sub-matrices resultantes como preconditionador del sistema a resolver. Con ello se minimizan los mensajes enviados

y recibidos entre procesadores, y por consiguiente se puede aumentar considerablemente la eficiencia de la ejecución de los modelos de flujo subterráneo. Aunque es usual llamar a este preconditionador simplemente como Jacobi Bloque, en realidad se trata de Jacobi Bloque que usa factorización ILU en los hilos de resolución enviados a cada procesador. Por otro lado, si bien las técnicas de Múltiple Malla

son preconditionadores normalmente usados para resolver EDP mucho más complejas que la tratada aquí sobre dominios irregulares, se incluyó dentro de las pruebas para resaltar la flexibilidad que presenta haber escrito el código en PETSc y mostrar que una mala elección del preconditionador no solo puede no mejorar el tiempo de ejecución, sino que puede llegar a empeorarlo.



**Figura 8.** Alturas piezométricas iniciales obtenidas de resolver los diferentes modelos en régimen permanente con condiciones de contorno sinusoidales. Debajo de cada gráfico se muestra el número de ecuaciones asociado a la discretización

Los experimentos computacionales fueron realizados en un sistema con las siguientes características: Intel® Xeon® CPU E5-2698 v3 @ 2.30GHz de 32 procesadores. Es importante mencionar que este ordenador no está diseñado para llevar a cabo matemática computacional en paralelo, pues sus conexiones entre procesadores no son óptimas y se limita la velocidad de transferencia de información de procesador a procesador. Los modelos descritos a lo largo de este apartado se resolvieron usando diferentes cantidades de procesadores y diferentes tipos de preconditionamientos. A cada uno de los experimentos se le midió su tiempo de ejecución

y su aceleración al considerar un aumento de la cantidad de procesadores usados para la ejecución. Los resultados (ver la Figura 9, columnas izquierda y derecha para tiempo de ejecución y aceleraciones, respectivamente) se graficaron relacionando la cantidad de procesadores con el tiempo de ejecución requerido y la aceleración alcanzada con respecto al uso de un sólo procesador, de la siguiente manera: (i) sin preconditionamiento, líneas azules, (ii) Jacobi clásico, línea verde, (iii) Jacobi Bloque y factorización ILU, línea roja, y (iv) Múltiple malla, línea naranja. Los gráficos obtenidos permiten plantear las siguientes observaciones.

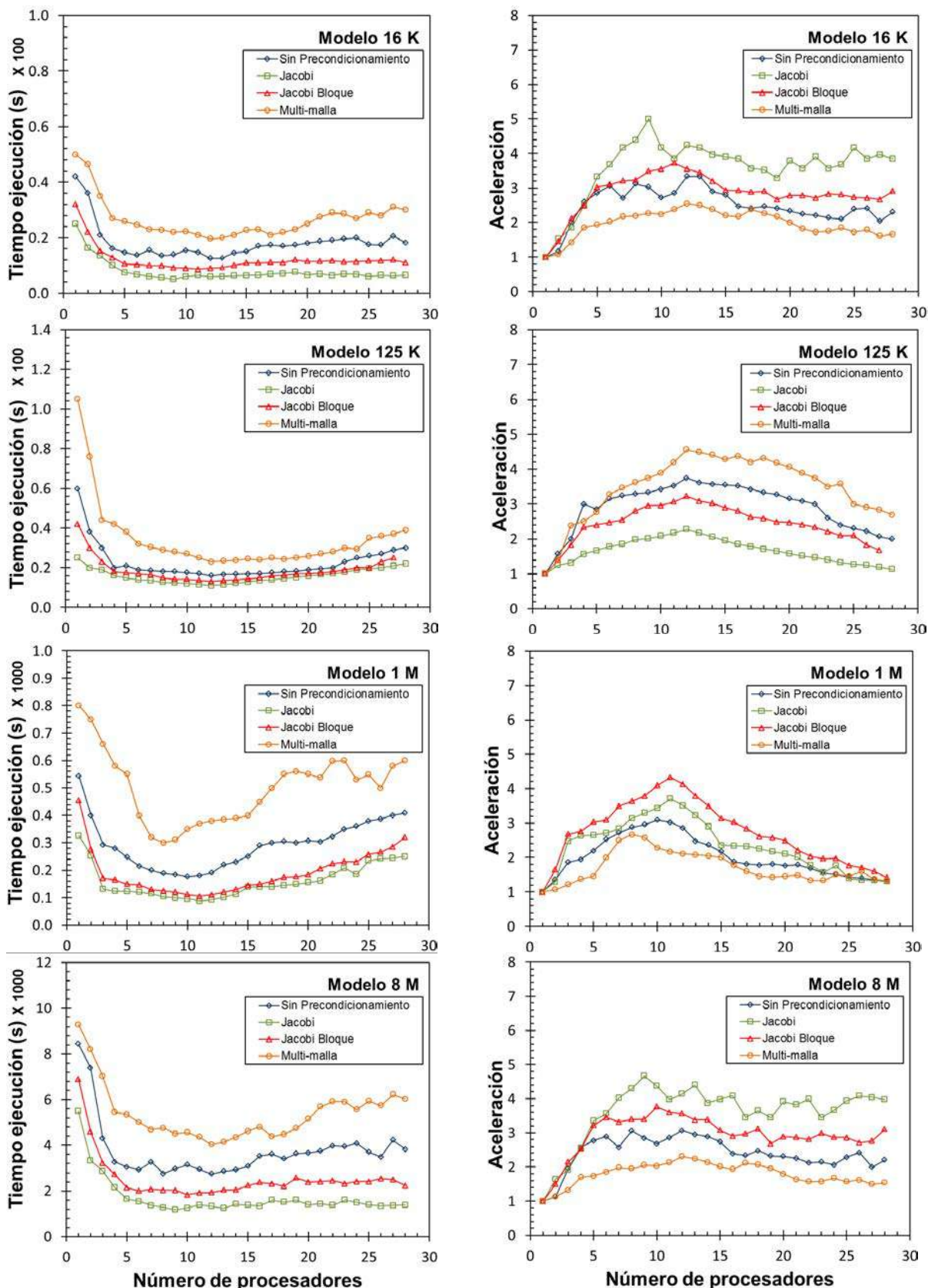


Figura 9. Tiempos de solución (izquierda) y aceleración (derecha) por paralelismos de los diferentes modelos con diferentes preconditionamientos por el método de GCP

Primero, en general, el preconditionamiento que disminuye en menor medida el tiempo de ejecución de los modelos de flujo subterráneo es Jacobi clásico, seguido de Jacobi Bloque con factorización

ILU. Las mayores aceleraciones de los modelos acá planteados se lograron usando entre dos y doce procesadores; límite que indica el progreso insignificante o disminución de dicha aceleración.

Aún así, se observa que, en casi todos los casos, el tiempo de ejecución necesario para resolver los modelos usando diez procesadores se disminuye hasta cerca de una quinta parte del requerido al emplear únicamente un solo procesador. Esta observación aplica para todas las discretizaciones espaciales consideradas y es la primera muestra de la escalabilidad de ejecución computacional que proporciona la incorporación de las herramientas de alto rendimiento ofrecidas por PETSc en el programa ANISOFLOW.

Segundo, los gráficos de velocidad y aceleración muestran que el recurso computacional disponible influye en la optimización de las simulaciones puesto que, a pesar de que en cada modelo hay diferente cantidad de ecuaciones asociadas, el número de procesadores para solucionar los modelos de flujo con rendimiento computacional óptimo aparece invariante alrededor de 12. Usar más de dicha cantidad de recurso computacional aumenta ligeramente la duración de las ejecuciones de los modelos, al mismo tiempo que reduce su aceleración. Este problema de desempeño se atribuye a las características del hardware usado durante los experimentos computacionales; máquina que, como se dijo antes, no optimiza la comunicación entre procesadores durante las ejecuciones en paralelo, lo cual interfiere en el óptimo desarrollo de las simulaciones ejecutadas mediante el ANISOFLOW.

Tercero, los resultados muestran que los preconditionadores más complejos como Jacobi Bloque con factorización ILU o Múltiple Malla toman más tiempo en construir las matrices asociadas a su operación computacional, que el tiempo que podría requerirse para ejecutar el número de iteraciones que se ahorran. Esto sucede porque el radio espectral de las matrices que forman los SELD asociados a la solución por DF de los modelos de flujo planteados en este trabajo no se mejora considerablemente como efecto de ejecutar los preconditionamientos sobre el GC. Es más, para la geometría del acuífero analizado, las condiciones de contorno impuestas sobre éste y el esquema de DF empleado para la linealización de la EDPFS, el preconditionador más simple (Jacobi) de todos los acá utilizados es el más efectivo y es suficiente. Lo anterior se traduce en que no es necesario el uso de preconditionamientos más elaborados que tendrían, teóricamente, mayor impacto sobre el radio espectral de las matrices asociadas a los modelos de flujo, dado que dichas matrices no son tan mal condicionadas como en otros casos, por ejemplo, cuando se consideraran geometrías irregulares del acuífero. Los resultados muestran que, más que el preconditionamiento en sí, el verdadero impacto sobre la rapidez de solución de la EDPFS radica en el paralelismo de la implementación. Esta conclusión no es generalizable y aplica únicamente para la configuración geométrica acá considerada.

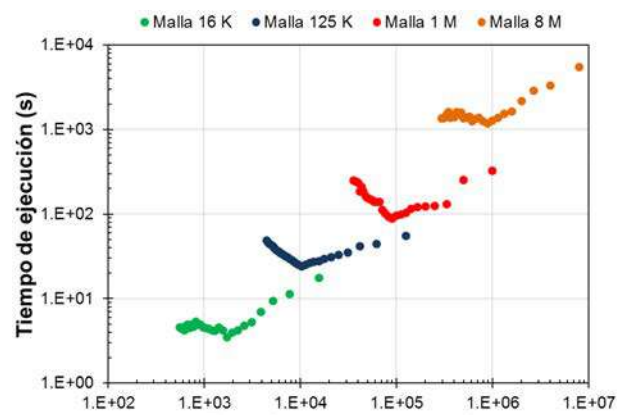


Figura 10. Escalabilidad en la ejecución de los diferentes modelos con preconditionamiento por Jacobi clásico

Por último, en la Figura 10 se muestran los resultados del tiempo de ejecución en función de la cantidad de ecuaciones almacenadas por cada procesador utilizado durante la ejecución, considerando únicamente el preconditionamiento por el método de Jacobi. Cabe notar que el tiempo de ejecución disminuye potencialmente con la cantidad de transacciones por procesador a medida que se aumentan la cantidad de procesadores disponibles de uno a doce; en otras palabras, en este intervalo de disponibilidad de recurso computacional el escalamiento de la solución por efecto de la paralelización es adecuado. Al usar más de doce procesadores, el desempeño disminuye notablemente. De la anterior observación se concluye que cuando el ANISOFLOW no está sujeto a restricciones de hardware, por ejemplo la poca optimización en los envíos de mensajes entre procesadores, como sucede en la máquina usada durante el desarrollo de este trabajo, el escalamiento de la ejecución debería obtenerse de forma adecuada y clara. Estas limitaciones operativas se resolverían fácilmente ejecutando los modelos de flujo subterráneo en máquinas orientadas al procesamiento numérico en paralelo.

## 5. CONCLUSIONES

Se ha implementado ANISOFLOW, un código computacional en paralelo para resolver la EDPFS en tres dimensiones y transitoria mediante DF, considerando medios homogéneos y anisótropos. Este desarrollo computacional se ejecutó empleando convenientemente las herramientas de altas prestaciones contenidas en la interfaz del software PETSc. La ventaja principal de adoptar dichas librerías es la posibilidad de lograr el escalamiento en la ejecución de los modelos de flujo subterráneo, especialmente cuando éstos han sido finamente discretizados en el tiempo y el espacio y pretenden representar con cierto detalle la complejidad

intrínseca a los medios geológicos. Propiedad de escalamiento que puede ser aprovechada para embeber los modelos en esquemas de optimización donde se requiera su ejecución repetida, como puede ser la optimización de sistemas de uso conjunto o la resolución del problema inverso.

Ahora bien, aunque en este trabajo no se han presentado geometrías complejas, estas se han considerado en el desarrollo del software por medio de etiquetas de celdas activas e inactivas. Evidentemente, las pruebas aquí realizadas no permiten evaluar la incidencia de la complejidad de la geometría en la escalabilidad del software, sin embargo, se espera que aunque los tiempos de solución sean mayores la escalabilidad del software permanezca.

Así pues, ANISOFLOW es un software flexible porque permite considerar explícitamente las diferentes características hidráulicas del medio hidrogeológico, la heterogeneidad y la anisotropía de la conductividad hidráulica y la heterogeneidad del coeficiente de almacenamiento. De la misma forma, permite representar la variabilidad espacio-temporal de las condiciones de contorno del acuífero y las acciones exteriores actuando sobre éste; al tiempo que abre la posibilidad de seleccionar diferentes esquemas de resolución de los modelos, de acuerdo con la configuración que se haga sobre las librerías del PETSc. Por ejemplo, seleccionar cierto tipo de preconditionamiento, usar un grafo de 7 o 19 bloques y elegir el solucionador de GCP; configuraciones que pueden modificarse directamente al momento de la ejecución.

De modo que, con la serie de experimentos computacionales presentados en este trabajo ha quedado demostrada la eficacia y eficiencia computacional del software ANISOFLOW, resaltándose la importancia de usarlo convenientemente en sistemas de computación que posean hardware orientado a optimizar las comunicaciones entre procesadores para acelerar las ejecuciones en paralelo. Como trabajo futuro se espera depurar el código, tal que pueda ser usando directamente por usuarios no especializados en computación en paralelo, y desarrollar una interfaz gráfica de usuario que permita su libre distribución.

## 6. REFERENCIAS BIBLIOGRÁFICAS

Aggelopoulos C.A., D.C. Tsakiroglou (2012). Effects of micro-heterogeneity and hydrodynamic dispersion on the dissolution rate of carbon dioxide in water-saturated porous media. *International Journal of Greenhouse Gas Control*, 10, 341–350.

Álvarez-Villa, O. D. (2014). *Simulación Eficiente de las Relaciones Río-Acuífero en Sistemas de Utilización Conjunta Mediante Técnicas de Reducción de Modelos Lineales Invariantes en el Tiempo*. Tesis Doctoral. Universitat Politècnica de València.

Amestoy P.R., I.S. Duff, J.Y. L'Excellent (1998) *MUMPS Multifrontal Massively Parallel Solver*. Version 2.0.

Anderman, E.R., K.L. Kipp, M.C Hill, J. Valstar, R.M. Neupauer (2002) *Documentation of the Model-Layer Variable-Direction Horizontal Anisotropy (LVDA) Capability of the Hydrogeologic-Unit Flow (HUF) Package. MODFLOW-2000, the U.S. Geological Survey Modular Groundwater Model*. U.S. Geological Survey Open-File Report 02-409. Denver, Colorado.

Anderson E., Z. Bai, C. Bischof, L.S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen (1999) *LAPACK Users' Guide, Third Edition*. SIAM, Pennsylvania.

Anderson M.P., W.W. Woessner (1992). *Applied Groundwater Modeling: Simulation of Flow and Advective Transport*. Second Edition. Elsevier Science.

Balay S., S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, L. Curfman McInnes, K. Rupp, B. Smith, S. Zampini, H. Zhang, and H. Zhang Mathematics and Computer Science Division, Argonne National Laboratory (2012) *PETSc User's Manual*. Revision 3.3. Argonne, Illinois.

Bear J., A.H.D. Cheng (2010). *Modeling Groundwater Flow and Contaminant Transport*. Volume 23. Springer Science and Business Media.

Benzi M. (2002) Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2), 418–477.

Blessent D., R. Therrien, C.W. Gable (2011). Large-scale numerical simulation of groundwater flow and solute transport in discretely-fractured crystalline bedrock. *Advances in Water Resources*, 34(12), 1539–1552.

Briggs W.L., V.E. Henson, S.F. McCormick (2000) *A Multigrid Tutorial*. Second Edition, SIAM, California.

Carle S.F. (1999) *T-PROGS: Transition Probability Geostatistical Software*. University of California, Davis, CA.

Carle S.F., G.E. Fogg (1996) Transition probability-based indicator geostatistics. *Mathematical geology*, 28(4), 453–476.

Dongarra J.J., C.B. Moler, J.R. Bunch, G.W. Stewart (1979) *LINPACK Users' Guide*. SIAM. Philadelphia, Pennsylvania.

Gustafsson I. (1978). A class of first order factorization methods. *BIT Numerical Mathematics*, 18(2), 142–156.

Flores O. (2008) *Estudio de Librerías Paralelas de Libre Distribución y Algoritmos Paralelos Iterativos Multipaso para la Resolución de Sistemas de*

- Ecuaciones Lineales Dispersos*. Aplicación a la Ecuación de Difusividad Neutrónica. Tesis Doctoral. Universitat Politècnica de València.
- Harbaugh A.W. (2005) *MODFLOW-2005, the US Geological Survey Modular Ground-water Model: The Ground-Water Flow Process*. U.S. Department of the Interior, U.S. Geological Survey Techniques and Methods 6–A16. Reston, Virginia.
- Hestenes M.R., E. Stiefel (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 409–436.
- Hill M.C. (1990) *Preconditioned Conjugate-Gradient 2 (PCG2), A Computer Program for Solving Ground-water Flow Equations*. U.S. Department of the Interior, U.S. Geological Survey Water-Resources Investigations Report 90-4048. Denver, Colorado.
- Larabi A., F. De Smedt (1994). Solving three-dimensional hexahedral finite element groundwater models by preconditioned conjugate gradient methods. *Water Resources Research*, 30(2), 509–521.
- Lawson C.L., R.J. Hanson, D.R. Kincaid, F.T. Krogh (1979) Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software (TOMS)*, 5(3), 308–323.
- Li L., H. Zhou, J.J. Gómez-Hernández (2010) Steady-state saturated groundwater flow modeling with full tensor conductivities using finite differences. *Computers and Geosciences*, 36(10), 1211–1223.
- Meijerink J.A., H.A. van der Vorst (1977) An iterative solution method for linear systems of which the coefficient matrix is a symmetric  $M$ -matrix. *Mathematics of Computation*, 31(137), 148–162.
- Naff R.L., E.R. Banta (2008) *The US Geological Survey Modular Ground-water Model - PCGN: a Preconditioned Conjugate Gradient Solver with Improved Nonlinear Control*. U.S. Geological Survey Open-File Report 2008-1331.
- Pereira F.H., S.L. Lopes, S. Ikuyo (2006) A fast algebraic multigrid preconditioned conjugate gradient solver. *Applied Mathematics and Computation* 179, 344–351.
- Saad Y. (1994a) ILUT: A dual threshold incomplete LU factorization. *Numerical Linear Algebra with Applications*, 1(4), 387–402.
- Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Segunda Edición. SIAM. Philadelphia, Pennsylvania.
- Salamon P. (2006). *On Modeling Contaminant Transport in Complex Porous Media Using Random Walk Particle Tracking*. Tesis Doctoral. Universitat Politècnica de València.
- Sahuquillo A., E. Custodio, M.R. Llamas (2009) La gestión de las aguas subterráneas. Panel Científico Técnico de Seguimiento de las Políticas de Aguas, Fundación Nueva Cultura del Agua, Sevilla, 24 de Enero de 2008. Publicado en *Tecnología del Agua*, 305, 60-67.
- Sánchez L. (2009). *Modelación de Efectos Anómalos de Transporte de Masa en Tanque de Laboratorio: Análisis Compartivo de la Aplicación de Distintas Metodologías*. Trabajo Final de Máster. Universitat Politècnica de València
- Smith B.T., J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema, C.B. Moler (1976) *Matrix Eigensystem Routines – EISPACK - Guide*. Springer Lecture Books in Computer Science Volume 6.
- Tatebe O. (1993) The Multigrid Preconditioned Conjugate Gradient Method, vol. 3224, *NASA Conference Publication*, pp. 621–634.
- Trefethen L.N., D. Bau III (1997). *Numerical Linear Algebra*. Volume 50. SIAM. Philadelphia, Pennsylvania.
- Van der Vorst, H.A., K. Dekker (1988) Conjugate gradient type methods and preconditioning. *Journal of Computational and Applied Mathematics*, 24(1–2), 73–87.
- Watts III J.W. (1981). A conjugate gradient-truncated direct method for the iterative solution of the reservoir simulation pressure equation. *Society of Petroleum Engineers Journal*, 21(3), 345–353.