

# GMRES PRECONDICIONADO CON WAVELETS. UN ALGORITMO DE SELECCIÓN DEL UMBRAL PARA LA OBTENCIÓN DEL PATRÓN DE DISPERSIÓN.

Lilian Villarín Pildaín\*, Angela León Mecías<sup>1\*\*</sup>, Marta L. Bager Díaz Romañach\*\*, Yisleidy Linares Zaila\*\*

\*Universidad de Heidelberg, Alemania

\*\*Universidad de La Habana, Facultad de Matemática y Computación, San Lázaro y L, Vedado 10400, Habana 4, Cuba,

## ABSTRACT

In order to use the GMRES method to solve dense linear equations systems in  $O(n)$  iterations, the building of a preconditioner using the no singular sparse approximation of a dense linear equations systems matrix is proposed. The sparse approximation is obtained by wavelet compression, using Haar and Daubechies wavelets basis. The novelty of this work is the proposed strategy to automatic selection of the threshold to obtain the sparse matrix from the original dense matrix, which is based on statistics concept of percentile and another contribution is the cost analysis that was done. A broad numerical experimentation allowed to evaluate the quality of the preconditioner related to some parameters like problem dimension, used wavelet base and number of non zero elements of the sparse matrix.

**KEYWORDS:** preconditioning techniques, wavelet compression, Krylov subspaces methods

**MSC:** 65F50

## RESUMEN

Se propone la obtención de un preconditionador a partir de la aproximación dispersa no singular de la matriz de un sistema de ecuaciones lineales denso de grandes dimensiones, que permite la utilización del método GMRES para la solución del sistema denso original en un número  $O(n)$  de iteraciones. La aproximación dispersa es obtenida mediante compresión wavelet, utilizando bases de Haar y de Daubechies. Los aportes novedosos del trabajo consisten en la propuesta de una estrategia de selección automática del umbral de truncamiento para la obtención de la aproximación dispersa de la matriz densa, basada en el concepto estadístico de percentil, así como en el análisis de costo computacional realizado. Una amplia experimentación numérica permitió evaluar la calidad del preconditionador sujeto a diferentes parámetros como dimensión del problema, base wavelet utilizada y cantidad de elementos no nulos de la matriz dispersa.

## 1. INTRODUCCIÓN

La solución numérica de problemas con aplicación real conduce a modelos matemáticos cada vez más complejos, que en muchos casos requieren resolver sistemas de ecuaciones lineales (SEL)

$$Ax = b, x, b \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n} \quad (1)$$

de grandes dimensiones y con matriz A densa. Para resolver SEL densos, los métodos que históricamente se han utilizado son los métodos directos como la eliminación gaussiana y las factorizaciones triangulares, que requieren  $O(n^3)$  operaciones para llegar a la solución. Estos métodos tienen como inconveniente el alto costo en tiempo de ejecución cuando se quiere resolver un sistema para n grande y en muchas aplicaciones es importante dar una respuesta en tiempo real. De ahí que usualmente se adopten técnicas iterativas para la resolución computacional eficiente y robusta de tales sistemas. Tales técnicas pueden ser por ejemplo la de los subespacios de Krylov [9]; que acompañadas de un preconditionamiento efectivo pueden alcanzar gran velocidad de convergencia, manteniendo pequeño el número de iteraciones.

---

<sup>1</sup> angela@matcom.uh.cu

El método de residuo mínimo generalizado -GMRES por sus siglas en inglés- acompañado de preconditionamiento es uno de los métodos iterativos de los subespacios de Krilov más confiable para resolver sistemas densos con matriz generalizada, como se puede ver por ejemplo en [1], [3], [6], [7], [8]. Una forma de preconditionamiento consiste en encontrar una matriz  $M \approx A$  tal que el sistema transformado  $M^{-1}Ax = M^{-1}b$  tenga características (propiedades espectrales) que aceleren la convergencia del método iterativo. Existen diversas técnicas para la construcción de preconditionadores, algunas que actúan sobre la matriz de coeficientes original para de alguna manera modificar la estructura algebraica de la matriz, ([3], [4], [5], [7], [8], [9]) y otras que se apoyan en el problema físico que originó el sistema de ecuaciones lineales. La adaptación de GMRES preconditionado para la solución de sistemas densos haciendo uso de las bases wavelets ha sido ampliamente abordada en la literatura [3], [6], [7], [8], pero para problemas bien específicos, y sin reportes detallados del costo computacional, análisis del comportamiento para diversos valores de los parámetros y en particular, del tema de selección de un umbral de truncamiento en la obtención de la matriz dispersa mediante la transformada wavelet discreta. Al respecto en [7] se presenta la técnica de los triángulos, que es muy común en compresión de imágenes y está basada en histogramas de los valores de los pixels de la imagen. En esta aplicación tiene sentido, dado que los pixels generalmente pueden tomar valores en conjuntos finitos, por ejemplo, valores entre 0 y 255. La cantidad de elementos de los histogramas es entonces finita. En el caso de matrices, los valores de cada elemento son reales, luego, obtener una histograma de valores requiere conocer el máximo y mínimo valor de la matriz, fijar a priori un número de elementos para el histograma y entonces aplicar la técnica que aparece en [7]. Fueron estas valoraciones las que nos motivaron a buscar nuevas estrategias de selección del umbral de truncamiento.

En este trabajo se propone la obtención de un preconditionador a partir de la factorización LU incompleta (ILU) de nivel 0 de la aproximación dispersa no singular  $\tilde{A}$  de la matriz  $A$  de (1); lo cual permite la utilización del método GMRES para la solución del sistema denso original, en un número  $O(n)$  de iteraciones. La aproximación dispersa  $\tilde{A}$  es obtenida mediante compresión wavelet, utilizando bases de Haar y de Daubechies, [2]. El método se denotará por  $W\_ILU\_GMRES$ .

La novedad fundamental del trabajo consiste en la propuesta de una estrategia de selección del umbral de truncamiento en la aproximación dispersa de la matriz densa, basada en el concepto estadístico de percentil, así como en el análisis de costo computacional realizado. Se muestra además cómo cuando los valores propios del sistema preconditionado forman conglomerados de radio pequeño y alrededor de 1 se acelera la convergencia de GMRES.

La estructura del artículo es como sigue; en la siguiente sección se muestra en qué consiste el algoritmo  $W\_ILU\_GMRES$ , que incluye detalles sobre la construcción del preconditionador así como de la estrategia de selección del umbral para la obtención de la matriz dispersa. En la tercera sección se presenta un análisis minucioso de costo del algoritmo  $W\_ILU\_GMRES$ , se obtiene una cota inferior para el costo computacional así como una relación entre ésta y la cantidad de elementos no nulos que debe tener la matriz dispersa con la que se construye el preconditionador. En la cuarta y última sección se presenta la experimentación numérica realizada para la obtención de los parámetros óptimos. Para finalizar se presentan conclusiones y comentarios de los resultados obtenidos.

## 2. ESTRATEGIA DEL MÉTODO $W\_ILU\_GMRES$

El método del residuo mínimo generalizado GMRES(Generalized Minimum Residual) [9], es un método iterativo de subespacios de Krilov, pensado originalmente para la resolución de sistemas de ecuaciones lineales como (1) de grandes dimensiones pero con matriz dispersa. Su aplicación a la resolución de sistemas densos con el objetivo de obtener la solución en una cantidad de iteraciones menor que  $O(n^3)$ , debe ser objeto de análisis. Es de interés que GMRES converja en un número de iteraciones muchísimo menor que  $n$  de modo que se disminuya considerablemente el número de operaciones requeridas para llegar a una aproximación de la solución. Con este objetivo se utilizan las técnicas de preconditionamiento, que aceleran la convergencia a la solución, mejorando la estructura espectral y el número de condición de la matriz del sistema a resolver.

La idea básica del preconditionamiento consiste en transformar el sistema (1) en uno equivalente,

$$M^{-1}Ax = y$$

donde  $M$  es una matriz no singular de orden  $n$  y  $y = M^{-1}b$ . Mientras más cercana sea  $M$  a  $A$ , más cerca estará  $M^{-1}A$  de la matriz identidad y por tanto más agrupados alrededor de uno estarán los valores propios de la matriz transformada. De ahí que el problema está en encontrar un preconditionador que aproxime a  $A$  lo suficiente para mejorar la convergencia de GMRES y cuya obtención sea poco costosa. En este trabajo se propone aplicar GMRES preconditionado con la factorización  $LU$  incompleta ( $ILU$ ), pero aplicando la factorización  $ILU$  a una matriz dispersa obtenida de  $A$  mediante la compresión wavelet. A continuación se expone la estrategia a seguir. Sea  $W$  una matriz ortonormal base del espacio wavelet. Se conoce que aplicar la transformada wavelet bidimensional a  $A$ , es equivalente a obtener la matriz  $\tilde{A} = WAW^T$ . Esta matriz sigue siendo densa, pero al aplicársele un umbral lo suficientemente pequeño, se puede alcanzar el patrón de dispersión buscado. Una vez aplicado el umbral se obtiene la matriz dispersa  $\tilde{A} = \text{umbral}(\tilde{A})$ . En este punto se pueden seguir 2 vías :

1. Resolver el sistema disperso mediante algún método iterativo:  
 $\tilde{A}y = Wb; \text{ con } x = W^T y;$  (2)

2. Utilizar  $\tilde{A}$  para preconditionar el sistema denso  
 $\tilde{A}y = Wb; \text{ con } x = W^T y$  (3)  
 y resolverlo mediante algún método iterativo.

El primer caso es ampliamente analizado en [7] para las bases wavelets de Haar y Daubechies. El segundo caso también se analiza en [7], pero sólo para la base wavelet de Daubechies. Nosotros haremos un análisis minucioso de la segunda vía. Para que los costos de obtención del preconditionador y su aplicación no dominen el costo de GMRES preconditionado, es necesario obtener una aproximación dispersa de  $A$ , o sea,  $\tilde{A} \approx A$ , tal que  $nnz(\tilde{A}) \ll nnz(A)$  ( $nnz$  número de entradas no nulas). Con este objetivo, se introduce la aplicación de la compresión mediante bases wavelets, particularmente de las bases wavelets de Haar y Daubechies.

## 2.1. Representación de la matriz en la base wavelet

Sea  $A$  una matriz densa de dimensión  $n = 2^L$  y sea  $r$  el nivel de la wavelet que se desea utilizar (cuando se habla del nivel, es la resolución con la que se desea ver los datos; un nivel  $r$  significa que se ha condensado la información inicial en  $2^{L-r}$  coeficientes y los  $2^r$  restantes son coeficientes de detalles obtenidos en los  $r$  pasos anteriores). Sea  $s^{(L)}$  un vector columna de  $A$  en el nivel 0 de la wavelet. Si se utiliza la transformada discreta, se va transformando la representación del vector  $s^{(L)}$ ,

$$\begin{array}{ccccccc} s^{(L)} & \rightarrow & s^{(L-1)} & \rightarrow & s^{(L-2)} & \rightarrow & \dots & \rightarrow & s^{(L-r)} \\ & & \searrow & & \searrow & & \searrow & & \searrow \\ & & d^{(L-1)} & & d^{(L-2)} & & \dots & & d^{(L-r)} \end{array}$$

donde los vectores  $s^{(k)}$  y  $d^{(k)}$  son de dimensión  $2^k$ . La suma total de los tamaños de los datos transformados  $s^{(L-r)}, d^{(L-r)}, \dots, d^{(L-2)}, d^{(L-1)}$  es  $2^L$  : La transformación del vector columna  $s^{(L)}$  se puede expresar matricialmente como un nuevo vector  $w = Ws^{(L)}$ . De forma similar, aplicar la transformada wavelet a un vector fila  $s_{(L)}$  de  $A$ , sería :

$$w = Ws_{(L)}^T, \rightarrow w = s_{(L)}W^T. \quad (4)$$

Luego, obtener la matriz que contiene los coeficientes de la transformada bidimensional para  $A$  es equivalente a obtener

$$\tilde{A} = WAW^T = P_{L-r+1}W_{L-r+1} \cdots P_LW_LAW_L^T P_L^T \cdots W_{L-r+1}^T P_{L-r+1}^T \quad (5)$$

Para bases ortogonales, la matriz  $W$  obtenida es ortogonal, característica ésta que permite utilizar compresión wavelet para resolver sistemas de ecuaciones lineales. La obtención de la matriz  $WAW^T$  se realiza implícitamente mediante la transformada wavelet rápida, a un costo del orden de las  $n^2$  operaciones. Luego, un nivel completo de la transformada discreta de la matriz  $A$  sería  $\tilde{A} = WAW^T$ . La matriz  $\tilde{A}$  es todavía densa pero puede ser transformada en dispersa seleccionando un umbral a partir del cual desechar los coeficientes menores y manteniendo

solamente los coeficientes mayores, que son los que contienen mayor información de la matriz original. La matriz obtenida luego de aplicado el umbral se llamará  $\tilde{A}$ .

## 2.2. Selección del umbral de truncamiento

La selección del umbral de truncamiento para obtener una buena aproximación dispersa  $\tilde{A}$  de la matriz  $A$  es un tema muy delicado, pues de esto depende el nivel de compresión y calidad de la aproximación obtenida. Se aborda en relativamente poca literatura, cuando se trata de aplicaciones de compresión en sistemas de ecuaciones lineales. En este trabajo se propone una estrategia para la selección del umbral de truncamiento, que al aplicarla a la matriz  $\tilde{A}$ , garantiza que se obtenga una aproximación dispersa  $\tilde{A}$  con el número de elementos distintos de cero deseado. La propuesta está inspirada en el concepto de percentiles en estadística y la forma de calcularlos es como sigue.

Dada una muestra  $V \in R^{n^2}$  y un valor  $\alpha \in [0, 100]$ , se define el percentil  $\alpha$  de la muestra, al valor  $p_\alpha$  que divide la muestra en dos clases, de tal forma que el  $\alpha\%$  del total de observaciones sea menor que  $p_\alpha$  y consecuentemente el  $100 - \alpha\%$  sea mayor o igual que él.

El cálculo de percentiles puede ser realizado simplemente ordenando la muestra, y tomando el valor que está en la posición  $\frac{\alpha}{100} * n^2 + 1$ . Existen métodos que permiten obtenerlo en orden lineal, basados en tablas de frecuencias e interpolación. Estas pueden ser explotadas en el futuro, con el objetivo de bajar el costo computacional. Aplicado a la compresión de datos, si se desea obtener una aproximación de  $V$  con el  $\beta\%$  de elementos de mayor valor no nulos, bastaría con quitar de  $V$  todos los elementos menores que  $p_{100 - \beta}$ . En nuestra aplicación, el objetivo es encontrar el valor umbral  $u_{nz}$  de la muestra tal que existan aproximadamente  $nz$  elementos mayores o iguales que dicho umbral. Haciendo  $\beta = \frac{nz}{n^2} * 100$ , el porcentaje de elementos que representa  $nz$  respecto a la cantidad total de elementos de la muestra, se tiene que  $u_{nz}$  es el percentil  $p_{100 - \beta}$ .

Entonces encontrar  $u_{nz}$  en un vector  $V \in R^{n^2}$  de  $n^2$  datos, es equivalente a obtener una ordenación descendente  $R$  de dicho vector, y tomar el valor  $u_{nz} = R(nz)$  de  $V$ , tal que aproximadamente  $nz$  elementos tienen valor absoluto mayor o igual que  $u_{nz}$ . La Figura 1 muestra este procedimiento.

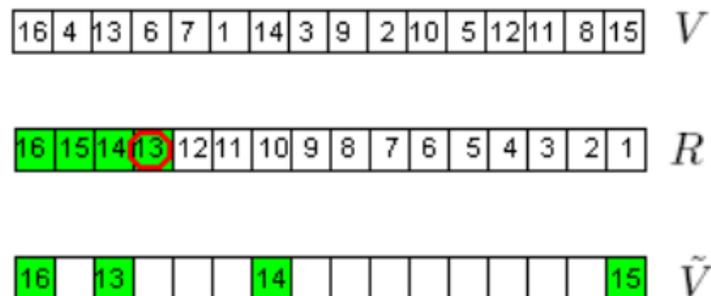


Figura 1: Algoritmo de selección del umbral

Aplicar la estrategia descrita más arriba a encontrar el umbral de truncamiento en una matriz  $A$  de dimensión  $n \times n$ , requiere representar la matriz  $A$  como un vector de  $n^2$  componentes, lo cual se puede lograr mediante la transformación:

$$V(k) = \tilde{A}\left(\left\lfloor \frac{k}{n} \right\rfloor + 1, k - \left\lfloor \frac{k}{n} \right\rfloor * n\right)$$

Una vez obtenido  $V$ , basta con aplicar  $u_{nz} = Umbral(V, nz)$ , siendo  $u_{nz}$  el valor tal que aproximadamente  $nz$  elementos de la matriz  $A$  son mayores o iguales que él. La decisión de la cantidad de elementos  $nz$  no nulos en la aproximación dispersa, define la calidad del preconditionador y su costo de obtención. El hecho de poder seleccionar  $nz = n * o(n)$ , permite realizar un análisis de costo de obtención del preconditionador, no encontrado en la literatura consultada; como se verá más adelante.

### 2.3. Precondicionador W\_ILU

La propuesta que se hace en este trabajo, utiliza la factorización  $LU$  Incompleta como preconditionador (clasifica dentro de los llamados preconditionadores algebraicos, entre los que se encuentran también la factorización incompleta de Cholesky, SPAI, entre otros). Para un estudio detallado ver [9]. Existen varias formas de realizar la factorización  $LU$  Incompleta, siguiendo diferentes estrategias de creación de relleno, como son los patrones fijos, umbrales de truncamiento y niveles de relleno, que definen hasta cuántas nuevas entradas no nulas y en qué forma se crearán. La que se utiliza en el trabajo es aquella que no introduce nuevas entradas no nulas durante la factorización, con lo que

$$nz(L) + nz(U) = nz(A) + O(n) \quad (6)$$

La factorización  $LU$  Incompleta nivel 0 de relleno, está basada en eliminaciones gaussianas; su única diferencia con la factorización completa, es que sólo realiza las operaciones donde la matriz original es distinta de 0, no incluyendo nuevas entradas no nulas. La utilización de matrices dispersas, optimizadas para el indexado de sus entradas no nulas, hace que la factorización pueda realizarse en un costo de operaciones mucho menor que la factorización  $LU$  completa, dependiendo de la cantidad de elementos no nulos que la matriz tenga. A continuación se muestra el algoritmo

Algoritmo ILU(0)	
1.	Para $i = 2; \dots; n$
2.	Para $k = 1; i - 1$ tal que $(i, k) \in nz(\tilde{A})$
3.	Hacer $\tilde{A}_{i,k} = \tilde{A}_{i,k} / \tilde{A}_{k,k}$
4.	Para $j = k + 1; \dots; n$ tal que $(i, j) \in nz(\tilde{A})$
5.	Hacer $\tilde{A}_{i,j} = \tilde{A}_{i,j} - \tilde{A}_{i,k} * \tilde{A}_{k,j}$

El preconditionador W\_ILU para sistemas densos, es una modificación que se hace a la factorización  $LU$  Incompleta, de manera que el costo computacional permita aplicarlo a sistemas densos. Para matrices densas,  $nz(\tilde{A}) \approx n^2$ , con lo que solamente obtener el preconditionador, requiere  $O(n^3)$  operaciones, dejando de ser posible su utilización para mejorar el desempeño de los métodos directos. De aquí que, la obtención de un nuevo preconditionador, W\_ILU, consiste en encontrar una aproximación dispersa  $\tilde{A}$ , basada en la compresión wavelet. A la aproximación dispersa  $\tilde{A}$  se le halla la factorización  $LU$  Incompleta, y el preconditionador  $M = LU$  obtenido, se define en este trabajo como *Precondicionador W\_ILU*.

Aplicar la transformada wavelet bidimensional estándar, para una base wavelet ortogonal cuya matriz de cambio de base es  $W$ , es equivalente a obtener la matriz  $\tilde{A} = WAW^T$  y debido a la ortogonalidad de  $W$ , los sistemas  $Ax = b \equiv \tilde{A}y = Wb$ ,  $x = W^T y$  tienen la misma solución  $x$ . La matriz dispersa  $\tilde{A}$ , obtenida de eliminar de  $\tilde{A}$  las entradas menos relevantes (es importante que no se pierdan determinadas características de la matriz al trunca elementos, como es la no singularidad del patrón de dispersión  $\tilde{A}$ , lo cual invalidaría su utilización como preconditionador) es entonces utilizada para obtener el preconditionador  $M = L * U = ILU(\tilde{A})$ , de modo que el sistema

$$(LU)^{-1} \tilde{A}y = (LU)^{-1} Wb, x = W^T y$$

tenga mejor estructura espectral, haciendo que la convergencia de **GMRES** ocurra en un número pequeño de iteraciones.

### 2.4. W\_ILU\_GMRES

La propuesta **W\_ILU\_GMRES** puede ser resumida en el siguiente algoritmo

Algoritmo WP GMRES( $A, b, w, level, nz$ )
--

- |    |  |
|----|--|
| 1. | Hacer $[L, U, \tilde{A}] = W\_ILU(A, w, level, nz)$ y<br>$\tilde{b} = Transform1D(b, w, level)$                                |
| 2. | Obtener $[\tilde{x}, flag, m] = GMRES\_ILU\_Prec(\tilde{A}, \tilde{b}, L, U)$  |
| 3. | Si $(flag \Rightarrow convergencia) \Rightarrow iTransform1D(\tilde{x}, level)$ ,<br><br>Solución del sistema denso $Ax = b$ . |

### 3. ANÁLISIS DE COSTO

El algoritmo W ILU GMRES retorna la solución aproximada  $x$  del sistema  $Ax = b$ , una bandera flag indicando el status de convergencia de GMRES y el número de iteraciones requeridas para alcanzar la precisión deseada. El costo total del algoritmo está determinado por la suma de los costos de  $W\_ILU$ ,  $GMRES\_ILU\_Prec$  y de  $2 * costo(Transform1D)$  y asciende a,

$$costo(W\_ILU\_GMRES) \approx 2n^2 \log_2 n + n * nnz(\tilde{A}) + 2m * nnz(\tilde{A}) + 2n^2 m + 4nm^2 \quad (7)$$

**Teorema:** El costo de la propuesta **W\_ILU\_GMRES** está acotado inferiormente por  $n^2 \log(n)$ , o sea

$$costo(W\_ILU(A, w, nz)) = \Omega(n^2 \log(n))$$

**Demostración:** La demostración de este teorema viene dada analizando la expresión (7), donde uno de los términos de la sumatoria es justamente  $2n^2 \log(n)$ , de donde la propuesta **W\_ILU\_GMRES** como mínimo deberá realizar  $O(n^2 \log(n))$  operaciones para encontrar el umbral. ■

El teorema establece una cota mínima para **W\_ILU\_GMRES**. Otras estrategias de selección del umbral, menos costosas pero menos potentes, pudieran bajar esta cota inferior; además esta cota está dada por la versión secuencial del algoritmo, la implementación paralela puede bajar esta cota a un  $O(n^2)$ .

**Teorema:** Si el costo de la propuesta  $W\_ILU\_GMRES$  es  $n^2 * o(n)$ , entonces  $nnz(\tilde{A}) = n * o(n)$  y  $m = o(n)$ .

**Demostración:** Si el costo total de  $W\_ILU\_GMRES$  es  $n^2 * o(n)$ , entonces cada término en (7) lo es. En particular  $n * nnz(\tilde{A}) = n^2 * o(n)$ , lo cual es posible si y sólo si  $nnz(\tilde{A}) = n * o(n)$  y  $n^2 * m = n^2 * o(n)$ , de ahí que  $m = o(n)$  y viceversa. ■

Con este teorema queda establecido que para que la propuesta sea  $n * o(n^2)$ , es necesario que  $nnz(\tilde{A}) = n * o(n)$ , lo cual da una idea de qué valor pasar como parámetro  $nz$ , si se desea que el desempeño del algoritmo sea  $n^2 * o(n)$ . Por otro lado, se puede comprobar si la propuesta es realmente  $n^2 * o(n)$ , si tomando  $nnz(\tilde{A}) = n * o(n)$ , la cantidad de iteraciones que GMRES requiere para converger es  $m = o(n)$ .

#### 3.1. Costo de GMRES

El análisis de costo de GMRES viene dado por tres partes fundamentales,

1. Obtención de la base ortogonal mediante Gram-Schmidt Arnoldi modificado

$$\sum_{j=1}^m \{ costo(Av_j) + \sum_{i=1}^j [ costo(w_j v_i) + c ] \} \quad (8)$$

cuyo costo es igual a  $mn^2 + nm^2nm$

2. Resolución del problema de mínimos cuadrados dado por  $\bar{H}_m y = \beta e_1$ , que incluye las siguientes etapas:

- Transformación de  $\bar{H}_m y = \beta e_1$ , en el sistema  $R_m y = \gamma$  mediante transformaciones equivalentes (Rotaciones de Givens) que conserven la norma del residual. Su costo total es entonces,

$$\sum_{j=1}^m 4(m-j+1) = 4 \sum_{j=1}^m (j) = 2m^2 + 2m \quad (9)$$

- Resolución del sistema  $m \times m$  procedente de tomar la submatriz triangular superior  $R_m$ , donde  $\bar{H}_m$  fue transformada en  $\begin{bmatrix} R_m \\ O_{1 \times m} \end{bmatrix}$ , con  $R_m$  matriz  $m \times m$ , triangular superior. Su costo es el costo de la sustitución retrógrada,

$$\sum_{j=1}^{m-1} (i+m) = \frac{1}{2}m^2 + \frac{1}{2}m \quad (10)$$

3. Obtención de  $x_m$ , con un costo de  $m^2$  operaciones.

Resumiendo se tiene que en total, GMRES necesita  $mn^2 + nm^2 + \frac{7}{2}m^2 + \frac{5}{2}m + nm$  operaciones. Aplicar GMRES con preconditionador, es equivalente a obtener en cada iteración, el residual para el sistema preconditionado  $rprec_k = M^{-1}r_k$  y a partir de éste realizar la iteración, con lo que al costo del GMRES se le añade  $m * costo(M^{-1}r_k)$ .

#### a. Costo de $W_{ILU}$

El costo de  $W_{ILU}$  viene dado por el costo de la transformada wavelet bidimensional, el costo de obtención de la matriz dispersa y de la factorización incompleta  $ILU$ ,

$$costo(W_{ILU}) = O(level * n^2) + 2n^2 \log_2 n + n * nnz(\tilde{A}).$$

El costo de obtener  $\tilde{A}$  tal que  $nnz(A) = nz$  es:

$$\begin{aligned} costo \left( \text{Obtener } \tilde{A}(\bar{A}, nz) \right) &= costo(u_{nz} = \text{Umbral}(\bar{A}, nz)) + \\ & \quad costo \left( \tilde{A} = \text{Drop}(\bar{A}, \text{umbral}) \right) \\ &= O(n^2 \log_2 n + n^2) \\ &= O(n^2 \log_2 n) \end{aligned}$$

El costo de obtención de la factorización incompleta (preconditionador  $ILU$ ) está acotado superiormente por

$$\frac{(n-1)n}{2} + (n-1)nnz(\tilde{A})$$

Una vez obtenido el preconditionador  $M = L * U$ , donde obtener  $rprec_k = M^{-1}r_k$  es equivalente a aplicar la sustitución hacia adelante y retrógrada, que puede hacerse eficientemente en  $nnz(L + U)$  operaciones flotantes, con lo que en total, se le añade al costo de aplicación del GMRES  $m * nnz(L + U) \approx m * nnz(\tilde{A})$ .

## 4.1 EXPERIMENTACIÓN NUMÉRICA. DISCUSIÓN

Con el objetivo de validar nuestra propuesta se seleccionaron 7 matrices de prueba de la galería de Matlab; (cauchy, circul, fiedler, lehmer, minij, parter, ris). La dimensión  $n$  se toma en el conjunto {256, 512, 1024, 2048}, por lo que se forman 4 grupos de 7 matrices cada uno. Para este grupo de matrices se realizaron tres tipos de experimentos que se describen a continuación.

### 4.1. Compresión wavelet en matrices no singulares

Para determinar si en cada grupo existe una combinación (base, nivel), tal que todas las matrices puedan ser aproximadas por un preconditionador no singular, utilizando la compresión wavelet se obtuvo el preconditionador  $W_{ILU}$  con parámetro  $nz = n * \log_2 n$ ,  $base \in \{ 'haar', 'db4' \}$  y  $nivel \in \{ 1, \dots, \log_2 n \}$ . Se analizó si éste era singular o de rango deficiente. Debido a la estrategia de selección del umbral que se propone, la cota inferior del costo computacional es  $n^2 \log_2 n$ , por lo que no tiene sentido una experimentación con órdenes menores que  $\log_2 n$ . Como resultado se obtuvo que la base wavelet de Daubechies permite obtener un mayor número de preconditionadores sin problemas de singularidad y rango deficiente.

Si bien con el aumento del nivel de compresión se obtienen mejores aproximaciones, la acumulación de errores de redondeo que dicho proceso implica provoca que a partir de determinado nivel disminuya la calidad de los preconditionadores. Una posible explicación a este comportamiento, es que los coeficientes de las transformadas wavelets de Haar y Daubechies, contienen números irracionales, con lo que un mayor número de niveles implica un mayor número de operaciones con aproximaciones racionales de estos coeficientes. Esto trae como consecuencia que los coeficientes de la aproximación dispersa estén más afectados de acumulación de errores de redondeo en cuanto mayor es el nivel de la transformada.

El nivel óptimo medio crece de acuerdo a la dimensión del problema, siendo para las matrices de prueba estudiadas de orden (256, 512, 1024, 2048) de (3, 4, 5, 1) para la base de Haar y (4, 4, 4, 9) para la base de Daubechies de orden 4. Para las matrices de dimensiones grandes con  $nz = n * \log_2 n$ , el número de matrices numéricamente no singulares se redujo, lo que indica que para grandes dimensiones, no es posible aproximar ninguna de las matrices de manera no singular con solamente  $n \log_2 n$  elementos, ver gráficos en [10]. Las matrices de Cauchy y Lehmer no reportaron preconditionadores no singulares, para ninguna de las bases y en ninguna de las dimensiones analizadas, lo cual significa que un sistema de ecuaciones lineales con estas matrices no puede ser resuelto mediante  $W_{ILU\_GMRES}$ , al menos en orden  $O(n^2 \log_2 n)$ . Tomando una cantidad de elementos no nulos mayor que  $n * \log_2 n$ , es posible que se pueda obtener la solución en orden  $o(n^3) > n^2 \log_2 n$ .

#### 4.2. Comportamiento del preconditionador $W_{ILU}$

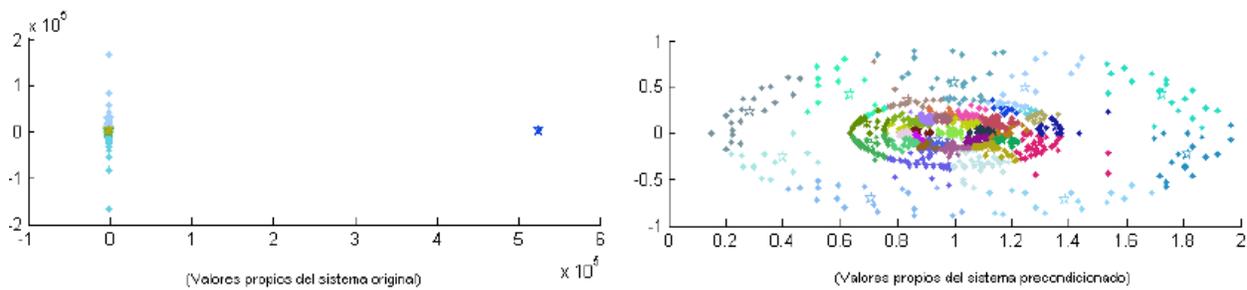
El experimento anterior mostró en varios de los casos de prueba, que  $nz = n * \log_2 n$  no permite obtener preconditionadores no singulares cuando  $n$  es grande. Un mayor número de entradas no nulas, podría ocasionar una mejor aproximación dispersa  $\tilde{A}$ , sin embargo, también haría crecer el costo de obtención del preconditionador ILU. Este experimento tiene como objetivo observar el comportamiento del preconditionador  $W_{ILU}$  cuando  $nz = n * o(n)$  crece. Para ello se tiene en cuenta: i) calidad de la aproximación dispersa no singular, que viene dado por el número de preconditionadores no singulares obtenidos, ii) análisis de costo de obtención del preconditionador  $W_{ILU}$ , donde la estrategia de selección del umbral impone una cota inferior a la propuesta  $W_{ILU\_GMRES}$ . Sin embargo, al aumentar  $nz = n * o(n)$ , siendo  $o(n) > \log_2 n$ , la obtención de la factorización LU Incompleta debe dominar el costo total, siendo este  $O(n^2 o(n))$ . Se realizaron los experimentos para las 7 matrices de prueba de dimensión  $n = 1024$ . Se utilizó la transformada wavelet de Daubechies, para los niveles  $1; \dots; \log_2 n$ . Se tomaron las funciones  $o(n) \in \{ \log_2 n, \sqrt{n}, \sqrt[3]{n^2}, \sqrt[12]{n^{11}} \}$ , y se varió el parámetro  $nz = n * o(n)$ . Se midió, para cada matriz, el número de condición y rango del preconditionador  $W_{ILU}$ , así como el tiempo de obtención del mismo, desglosado en el tiempo de la transformada wavelet, de la estrategia de selección del umbral y de obtención de la factorización LU incompleta. Esto se hizo para cada nivel de la base wavelet. Se arribó a los resultados siguientes: el aumento del número de entradas no nulas ocasiona un aumento en el número de preconditionadores no singulares, pero a su vez aumentó el costo de obtención del preconditionador. Se produce una disminución de la contribución de los tiempos de ejecución de la transformada wavelet y la estrategia de selección del umbral al tiempo total de ejecución del preconditionador, ver tablas y gráficos en el epígrafe 7.3 de [10].

#### 4.3. Comportamiento de la propuesta $W_{ILU\_GMRES}$

Este experimento tiene como objetivo analizar el comportamiento de la propuesta  $W_{ILU\_GMRES}$ , para las matrices de prueba de dimensión  $n = 1024$ , para las cuales fue posible obtener preconditionadores no singulares. Tomando  $nz = n * \sqrt[3]{n^2}$  como la cantidad de elementos no nulos en la aproximación dispersa, que representa aproximadamente el 10% de la cantidad de elementos no nulos de la matriz densa original, se obtuvieron para el nivel 5 de la wavelet de Daubechies, 6 de las 7 matrices de prueba con preconditionadores no singulares. Por esta razón, se realizan los experimentos con estos parámetros. Para que los sistemas de prueba tengan un orden de

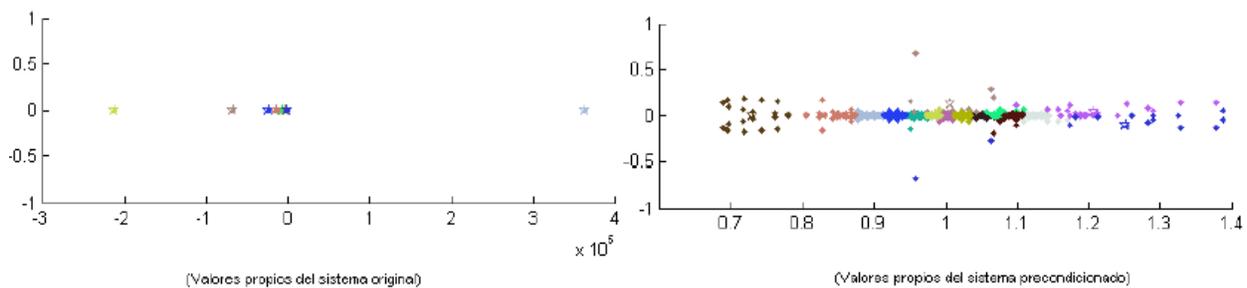
resolución  $O(n^2 * \sqrt[3]{n^2})$ , es importante que la convergencia de GMRES se alcance en un número de iteraciones menor que  $\sqrt[3]{n^2} \approx 100$ . Luego, si GMRES converge en menos de 100 iteraciones, se puede decir que se bajó el costo de resolución a  $O(n^2 * \sqrt[3]{n^2}) < O(n^3)$ .

Los resultados obtenidos para las 6 matrices de prueba fueron muy positivos; GMRES convergió en un número de iteraciones menor que  $\sqrt[3]{n^2} \approx 100$ . Se alcanzaron residuales relativos de alta precisión (menor que  $10^{-6}$ ). El costo de obtención de la factorización ILU, dominó en el algoritmo, imponiendo como costo total del mismo un  $O(n^2 * \sqrt[3]{n^2})$ . Se logró una reducción del costo de resolución mediante  $W\_ILU\_GMRES$  a menos del 32% de operaciones flotantes, respecto a la resolución mediante  $LU$ . Otra parte interesante del experimento, es el análisis de la influencia del preconditionador  $W\_ILU$  sobre la convergencia de GMRES. Para ello, se obtuvo la representación de los valores propios de las matrices del sistema original (gráfica de la izquierda en la Figura 2 y la Figura 3) y el preconditionado (gráfica de la derecha en la Figura 2 y la Figura 3). Aquí lo importante es observar que en todos los casos ocurrió un agrupamiento de los valores propios alrededor de 1 según se muestra en la Figura 2 derecha para la matriz circul y en la Figura 3 derecha para la matriz fiedler, lo que contribuyó a la enorme aceleración de GMRES, que necesitó en los 6 casos, mucho menos que 100 iteraciones para la convergencia.



**Figura 2:** Espectro de la matriz circul para  $n = 1024$

La estrategia de agrupamiento utilizada fue el k-means, pasando como cantidad de conglomerados (conocidos del inglés como clusters), la cantidad de iteraciones requeridas por GMRES para converger. Se calculó como medida de agrupamiento, el promedio de las distancias de cada valor propio, al centro de su conglomerado. Los resultados se muestran en la tabla



**Figura 3:** Espectro de la matriz fiedler para  $n = 1024$

#### 4. CONCLUSIONES

Se desarrolló una estrategia de selección automática del umbral, que permite obtener aproximaciones dispersas tan flexibles como se desee. Su único inconveniente, es que al estar basada en un algoritmo de ordenación, su costo es  $O(n^2 * \log_2 n)$ , imponiendo esto como cota inferior de la propuesta. Sin embargo, para grandes dimensiones se requieren aproximaciones dispersas con cantidad de elementos no nulos mayores que  $O(n * \log_2 n)$ ; la obtención de la factorización LU Incompleta, tiene entonces un costo mayor que  $O(n^2 * \log_2 n)$ , dominando el costo total de  $W\_ILU\_GMRES$ . Esto hace que la estrategia de selección del umbral propuesta sea de un orden aceptable.

Matrices	Iteraciones	Medida de Agrupamiento de valores propios		Residual relativo	Tiempo W ILU GMRES				W_ILU_GMRES Vs LU
		Sist. Original	Sist. Prec.		TW	Umbral	ILU	GMRES	
circul	40	9.5E+06	5.7E-04	2.2E-07	3.9E-01	2.5E-01	5.2E-01	9.2E-01	31.0%
fiedler	15	3.8E+03	4.9E-04	2.4E-09	3.9E-01	2.9E-01	4.9E-01	3.6E-01	22.2%
lehmer	11	3.2E-02	7.8E-05	1.3E-11	4.2E-01	3.4E-01	5.2E-01	3.3E-01	20.8%
minij	12	4.5E+02	3.5E-05	2.8E-09	3.9E-01	2.6E-01	5.0E-01	3.2E-01	21.2%
parter	13	1.2E-02	1.9E-05	3.0E-10	4.0E-01	2.8E-01	5.4E-01	3.5E-01	21.5%
ris	6	9.4E-07	2.5E-06	1.4E-10	3.9E-01	2.8E-01	5.3E-01	2.5E-01	19.2%

**Figura 4:** Resumen de la experimentación

La experimentación numérica realizada con el preconditionador W ILU para la selección de parámetros óptimos condujo a las siguientes aseveraciones: i) cuando crece la dimensión de los problemas, si se mantiene fijo el orden de entradas no nulas en la aproximación  $nz = n * o(n)$ , la aproximación dispersa empeora su desempeño, pudiendo volverse singular, o aumentando el número de iteraciones y residual relativo, ii) la compresión mediante la base wavelet de Daubechies, resulta más potente en niveles intermedios, permitiendo obtener preconditionadores no singulares más dispersos, con mayor probabilidad que cuando se utiliza la base wavelet de Haar, iii) cuando crece la cantidad de elementos no nulos en la aproximación dispersa, el preconditionador es de mayor calidad, sin embargo su costo de obtención crece en el mismo orden. Se formuló la propuesta de resolución de sistemas de ecuaciones lineales densos mediante  $W\_ILU\_GMRES$ , con costo total  $n^2 O(\max\{o(n), \log_2 n, it\})$ . La experimentación evidenció además que los valores propios se agruparon alrededor del 1, en clusters de radio muy pequeños, contribuyendo a la aceleración de la convergencia de GMRES que requirió un máximo de 40 iteraciones. El residual obtenido en cada caso tuvo la precisión deseada. Sin embargo el comportamiento de  $W\_ILU\_GMRES$  no fue el deseado para todas las matrices seleccionadas en la experimentación. Un trabajo futuro sería experimentar con otras clases de matrices para determinar qué características propician un buen desempeño del método; así como rediseñar de forma paralela las componentes de  $W\_ILU\_GMRES$ , con el objetivo de bajar los costos computacionales de obtención del preconditionador.

**RECEIVED, OCTOBER 2011  
REVISED, MAY, 2012**

#### REFERENCIAS

- [1] BUNSE-GERSTNER A., GUTIÉRREZ-CAÑAS I.(2006): A preconditioned GMRES for complex dense linear systems from electromagnetic wave scattering problems. **Linear Algebra and its Applications**, 416, 135-147.
- [2] DAUBECHIES I.(1992): **10 Lectures in Wavelets**, SIAM.
- [3] FORD J.M. (2001): Wavelet based preconditioning of dense linear systems. **PhD thesis**, University of Liverpool.
- [4] FORD J.M.(2003): An improved discrete wavelet transform preconditioner for dense matrix problems. **SIAM J. Matrix Anal. Appl.** 25, 642-661.
- [5] FORD J.M., TYRTYSHNIKOV E.E. (2005): Solving linear systems using wavelet compression combined with Kronecker product approximation. **Numerical Algorithms** 40, 125-135.
- [6] GONZÁLEZ P., CABALEIRO J.C., PENA T.F. (2002): Iterative solution of large linear systems with nonsmooth submatrices using partial wavelet transforms and split-matrix matrix-vector multiplication. **Technical Report**, Universidad de Santiago de Compostela, España.
- [7] GONZÁLEZ P. (2000): Técnicas de resolución de sistemas densos en multicomputadores: aplicación al crecimiento de grietas en estructuras. **PhD thesis**, Universidad de Santiago de Compostela.
- [8] LARRAZÁBAL G. (2002): Técnicas algebraicas de preconditionamiento para la resolución de sistemas lineales. **PhD thesis**, Universidad Politécnica de Cataluña.

[9] SAAD Y. (2003), **Iterative Methods for sparse linear systems**. SIAM.

[10] VILLARÍN L. (2008): W ILU GMRES: definición, análisis de costos y experimentación. **Tesis de Maestría**, Facultad Matemática y Computación, Universidad de La Habana.