UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE MATEMÁTICA

# ESTUDIO NUMÉRICO DE LA APLICACIÓN DE TÉCNICAS *WAVELETS* EN LOS SISTEMAS DE ECUACIONES LINEALES PROVENIENTES DE LA SOLUCIÓN NUMÉRICA DE ECUACIONES DIFERENCIALES MEDIANTE EL USO DE MÉTODOS IMPLÍCITOS DE DIFERENCIAS FINITAS DE ORDEN SUPERIOR

**Trabajo presentado ante el Ilustre Consejo de la Facultad de Ingeniería de la Universidad de Carabobo como credencial de mérito para el ascenso dentro del escalafón del Personal Docente y de Investigación a la categoría de Profesor Titular**

**Autor:** Ing° Enrique V. Flores C., MSc.

Bárbula, 13 de Enero del 2014

# ÍNDICE GENERAL

# LISTA DE FIGURAS

# INTRODUCCIÓN

La solución numérica de problemas con aplicación real, los cuáles en diversos casos son modelados a través de ecuaciones diferenciales, conduce a modelos matemáticos cada vez más complejos. En el caso de la resolución de problemas de valores en la frontera, existen diversos métodos para determinar la solución numérica de la ecuación diferencial (o el sistema de ecuaciones diferenciales) que modela el problema, donde se genera un sistema de ecuaciones lineales cuya resolución emplea gran cantidad de tiempo del proceso de aproximación.

El presente trabajo de investigación muestra un estudio del uso de técnicas *wavelets* en la resolución de sistemas de ecuaciones lineales provenientes de la solución numérica de ecuaciones diferenciales mediante el uso de métodos implícitos de diferencias finitas de orden superior. En virtud que la discretización generada a través de métodos implícitos de diferencias finitas de orden superior conduce a un sistema de ecuaciones lineales, y que en la resolución de dicho sistema se invierte una apreciable cantidad de tiempo, se propone establecer entre un conjunto de familia *wavelets* y un conjunto de métodos iterativos, cual combinación entre ellos logra una reducción del tiempo empleado en dicha resolución, en relación a paramétros tales como: número de iteraciones, error en la aproximación y tiempo de procesamiento.

Los resultados del presente trabajo representarían un aporte en relación a la selección de métodos iterativos para la resolución de sistemas de ecuaciones lineales; abriría un campo de futuros trabajos de investigación, en relación a la resolución de problemas de aplicación que involucren la resolución de sistemas de ecuaciones lineales; y a su vez representaría un aporte a la búsqueda de la soberanía tecnológica, ya que podría facilitar el desarrollo de proyectos que son modelados a través de ecuaciones diferenciales en diversos ámbitos cómo la industría petrolera, entre otros.

# CAPÍTULO I

# EL PROBLEMA DE INVESTIGACIÓN

## 1. Planteamiento del problema

La solución numérica de problemas con aplicación real, los cuáles en diversos casos son modelados a través de ecuaciones diferenciales, conduce a modelos matemáticos cada vez más complejos. En el caso de la resolución de problemas de valores en la frontera, el uso de esquemas implícitos en los métodos de diferencias finitas de orden superior lleva a la resolución de sistemas de ecuaciones lineales de la forma:

$$Ax = b \ , \ A \in R^{n \times n} \ , \ x \in R^{n \times 1} \ , \ b \in R^{n \times 1} \tag{1}$$

sistemas que en forma general son de grandes dimensiones y con posibilidad de matriz $A$ densa. Villarín et al. (2012) indica que es usual adoptar técnicas iterativas para la resolución computacional eficiente y robusta de dichos sistemas. Además, Villarín también señala que tales técnicas pueden ser, por ejemplo, el uso de subespacios de Krylov acompañadas de un precondicionamiento efectivo; donde se puede alcanzar gran velocidad de convergencia, manteniendo pequeño el número de iteraciones. Villarín propone una técnica de precondicionamiento que aproxime a la matriz $A$ de coeficientes del sistema, la cual es densa, a otra matriz $\hat{A}$ dispersa lo suficiente para mejorar la convergencia del método iterativo GMRES. Dicho precondicionamiento se hace mediante compresión de la matriz $A$ mediante el uso de *wavelets*. Se emplearon para las pruebas numéricas matrices de prueba del software comercial MATLAB®, dejando como trabajo futuro la prueba con otras clases de matrices para determinar las características que garantizan un buen desempeño del método propuesto. Es de destacar que en este trabajo se emplea un solo método iterativo.

Acevedo (2009) , plantea que un sistema tal como el descrito por la ec. (1) puede transformarse a un sistema de la forma:

$$\hat{A}\hat{x} = \hat{b} \tag{2}$$

donde:

$$\hat{A} = WAW^T \tag{3}$$

$$\hat{x} = Wx \tag{4}$$

$$\hat{b} = Wb \tag{5}$$

Obtenida la solución del sistema descrito por la ec. (2) la solución $\hat{x}$ del sistema descrito por la ec. (1) se obtiene mediante:

$$x = W^T \hat{x} \tag{6}$$

El procedimiento descrito se puede emplear solo para sistemas cuyo tamaño sea $n = q \cdot 2^k$ , para algún entero $k > 0$. El trabajo descrito propone como trabajo futuro el uso de la técnica descrita en la solución de problemas en otras áreas tales como modelos en electromagentismo mediante el método de los momentos, cuya formulación conduce a sistemas de ecuaciones lineales del tipo descrito por la ec. (1).

Flores et al. (2012) , indica que las tecnicas *wavelets* propuestas por Acevedo (citado anteriormente) tienen aplicabilidad en la resolución de sistemas de ecuaciones lineales que surgen de la solución numérica de problemas de valores en la frontera mediante el uso de métodos miméticos, métodos que guardan cierta similitud en sus esquemas numéricos con los métodos en diferencias finitas tradicionales. Los resultados del trabajo de Flores arrojan una reducción en el número de iteraciones necesarias para la obtención de soluciones aproximadas en dichos sistemas, a expensas de un incremento del error relativo en la aproximación de dicha solución. Se emplea como método iterativo el método BICGSTAB sin precondicionamiento y con precondicionamiento, y se concluye que el uso de precondicionadores junto a las técnicas *wavelets* reducen apreciablemente la cantidad de iteraciones. Es de destacar que en este trabajo se emplea un solo tipo de método iterativo.

Guo y Wang (2005) , proponen el uso de métodos explícitos en diferencias finitas de orden superior en la aproximación numérica de la solución numérica de la ecuación de convección-difusión, en el caso particular de un modelo de simulación en la industria petrolera. Las pruebas numéricas realizadas indican estabilidad numérica. En éste trabajo, queda abierto realizar la aproximación numérica usando métodos implícitos de diferencias finitas de orden superior.

En los trabajos descritos se observa el uso de un solo método iterativo en aquellos casos donde se requiere la resolución de sistema de ecuaciones lineales. Adicionalmente, el uso de métodos explícitos de diferencias finitas de orden superior no involucra la resolución de sistemas de ecuaciones lineales; resolución que se hace necesaria en el caso del uso de métodos implícitos.

Por todo lo antes expuesto, se propone realizar un estudio numérico del uso de las técnicas *wavelets* propuestas por Villarín et al. (citados anteriormente) y Acevedo (citado anteriormente) en la resolución de los sistemas de ecuaciones lineales resultantes del uso de métodos implícitos de diferencias finitas de orden superior propuestos por Guo y Wang (citados anteriormente) mediante una serie de métodos iterativos que incluyen a los usados en los trabajos citados y métodos alternativos a éstos; con el fin de establecer el desempeño de uso de las mencionadas técnicas combinadas con una serie de métodos iterativos para la resolución de sistemas de ecuaciones lineales, y determinar cual de las combinaciones tiene el mejor desempeño.

# 2. Objetivos de la investigación

## 2.1 Objetivo general

Realizar el estudio numérico de la aplicación de técnicas *wavelets* en los sistemas de ecuaciones lineales provenientes de la solución numérica de ecuaciones diferenciales mediante el uso de métodos implícitos de diferencias finitas de orden superior.

## 2.2 Objetivos específicos

1. Determinar los esquemas numéricos que surgen del uso de métodos implícitos de diferencias finitas de orden superior en la solución numérica de ecuaciones diferenciales.
2. Seleccionar los métodos iterativos a emplear en los esquemas numéricos que surgen del uso de métodos implícitos de diferencias finitas de orden superior.
3. Combinar las técnicas *wavelets* con los métodos iterativos seleccionados.
4. Evaluar el desempeño de las combinación de técnicas *wavelets* con los métodos iterativos seleccionados.

# 3. Justificación de la investigación

## 3.1 Conveniencia

Los resultados de esta investigación podrían ser empleado en investigaciones que se centren en el desarrollo de aplicaciones científicas en las cuales se encuentren presentes la resolución de sistemas de ecuaciones lineales provenientes del uso de esquemas implícitos en diferencias finitas de orden superior.

## 3.2 Implicaciones prácticas

La investigación mostraría resultados que permitirían a los investigadores que desean simular procesos de ingeniería contar con un método que arroja resultados precisos y eficientes.

## 3.3 Utilidad metodológica

Los resultados de ésta investigación podrían representar un aporte para establecer lineamientos y rutinas en el desarrollo de algoritmos optimizados para el mejoramiento de ejecución de los programas.

## 3.4 Relevancia social

Los resultados de ésta investigación servirían de recurso de utilidad tanto para los estudiantes que podrían simular aplicaciones científicas como para los investigadores ya que podrían mejorarse los tiempos de ejecución en la simulación de aplicaciones en la ingeniería y en la ciencia. Scheid (1968), citado por Hernández (2005), asegura que: "muchos de los problemas de la vida real pueden ser modelados mediante sistemas de ecuaciones en derivadas parciales, para los cuales usualmente es complicado obtener la solución analítica y por ello necesitan ser resueltos en forma aproximada mediante métodos numéricos".

# CAPÍTULO II

# MARCO TEÓRICO REFERENCIAL

Realizada la descripción del problema de investigación, y expuesta la justificación de la misma, a continuación se muestran los aspectos teóricos que le darán sustento. La intención del marco teórico referencial que se desarrolla para éste trabajo es situar el problema de investigación de estudio dentro de ciertos parámetros que servirían de soporte y orientación para el mismo. En el desarrollo de éste marco referencial, se hace una breve referencia a trabajos realizados de donde se extraen ideas que sirven de punto de partida para éste trabajo. Posteriormente, se hace una breve referencia a los aspectos téoricos que dan sustento al presente trabajo, los cuales están relacionados con la solución numérica de ecuaciones diferenciales mediante métodos implícitos de diferencias finitas; con lo relativo a los métodos iterativos para la resolución de sistemas de ecuaciones lineales y técnicas de precondicionamiento; y con lo relativo a la transformada *wavelet* y su uso en la resolución de sistemas de ecuaciones lineales.

## 1. Antecedentes

Guo y Wang (2005) , en su trabajo titulado: **"Testing of New High-Order Finite Difference Methods for Solving the Convection-Diffusion Equation"**, determinan las formulaciones explícitas de diferencias finitas con precisión de segundo orden y con precisión de tercer orden para resolver la ecuación de convección-difusión en el caso particular de un modelo de simulación en la industria petrolera. Los resultados indicaron que el uso de las formulaciones de diferencias finitas precisión de segundo y tercer orden pueden acelerar las simulaciones numéricas. Dejan como trabajo en desarrollo las formulaciones implícitas en diferencias finitas para la ecuación de convección-difusión, las cuáles conducen a un sistema de ecuaciones lineales de la forma $Ax = b$. Éste trabajo servirá de punto de partida para dichas formulaciones implícitas, y de éste trabajo se tomarán las condiciones iniciales, las condiciones en la frontera y los valores de los parámetros de algunas de las pruebas numéricas hechas para la realización de los experimentos numéricos de éste trabajo.

Acevedo (2009) , en su trabajo titulado: **"Computación Paralela de la Transformada Wavelet; Aplicaciones de la Transformada Wavelet al álgebra Lineal Numérica"**, muestran el estudio de la computación paralela de la Transformada *Wavelet* Discreta (DWT), y el estudio de las aplicaciones de la DWT en el campo del álgebra lineal numérica. Se estudiaron distintas variantes de construcción

de precondicionadores basados en la DWT para acelerar la convergencia de métodos iterativos en la resolución de sistemas de ecuaciones lineales. Deja como trabajo futuro el uso de la técnica descrita en la solución de problemas en otras áreas tales como modelos en electromagnetismo mediante el método de los momentos. De éste trabajo se tomará la metodología empleada para la transformación de un sistema de ecuaciones lineales $Ax = b$ a la base *wavelet*.

Villarín et al. (2012) , en su trabajo titulado: **"GMRES precondicionado con wavelets. Un algoritmo de selección del umbral para la obtención del patrón de dispersión"** proponen una técnica de precondicionamiento que a partir del sistema de ecuaciones lineales del sistema $Ax = b$, aproxime a la matriz de coeficientes $A$, la cual es densa, a otra matriz $\hat{A}$ dispersa, que aproxima a $A$ lo suficiente para mejorar la convergencia del método iterativo GMRES. Dicho precondicionamiento se hace mediante compresión de la matriz $A$ mediante el uso de *wavelets*. Como precondicionador se empleó la factorización LU incompleta. Se emplearon para las pruebas numéricas matrices de prueba del software comercial MATLAB®, dejando como trabajo futuro la prueba con otras clases de matrices para determinar las características que garantizan un buen desempeño del método propuesto. De éste trabajo, se tomará la idea de usar la factorización LU incompleta como precondicionador de un sistema de ecuaciones lineales en la base *wavelet*.

Hernández (2005) , en su trabajo titulado: **"Estudio de los Sistemas Lineales Dispersos provenientes de Discretizaciones Miméticas. Caso: Operador de Convección-Difusión"**, muestra un estudio numérico de distintos esquemas de discretización en diferencias finitas correspondientes al operador escalar de convección difusión en una y dos dimensiones para la aproximación de sus derivadas, esquemas que son generados a partir del método de operadores de soporte, a partir de las técnicas de Castillo-Grone-Yasuda, y del método tradicional en diferencias finitas, entre otros. En éste último método, se recurre al uso de nodos fantasmas en las fronteras del mallado con el propósito de aproximar a las derivadas con fórmulas centradas. De éste trabajo, se tomará la idea del uso de nodos fantasmas en las fronteras del mallado.

# 2. Bases teóricas

## 2.1 Ecuación diferencial parcial y la ecuación de convección-difusión

Una ecuación diferencial parcial (e.d.p) se define como toda ecuación diferencial donde aparecen dos o más variables independientes. Asi mismo, se define una e.d.p. lineal, o cuasi-lineal, a una ecuación de

la forma:

$$A\frac{\partial^2\Phi}{\partial x^2}+B\frac{\partial^2\Phi}{\partial y\partial x}+C\frac{\partial^2\Phi}{\partial y^2}=f\left(x,y,\Phi,\frac{\partial\Phi}{\partial x},\frac{\partial\Phi}{\partial y}\right) \tag{7}$$

donde $A$, $B$, y $C$ son constantes. Las e.d.p. cuasi-lineales se clasifican en: elípticas si $B^2-4AC<0$; parabólicas si $B^2-4AC=0$; e hiperbólicas si $B^2-4AC>0$. Problemas en ciencias e ingeniería tales como el régimen permanente de distribución de temperaturas en una placa plana rectangular, el flujo de calor unidimensional en un alambre de longitud finita aislado en los extremos, y la vibración de una cuerda unidimensional son modelados a través de e.d.p. (Mathews y Fink, 2000) .

Un caso particular de e.d.p. es la ecuación de convección-difusión unidimensional, la cual es una ecuación de la forma:

$$\phi\frac{\partial C(x,t)}{\partial t}=D\frac{\partial^2 C(x,t)}{\partial x^2}-u\frac{\partial C(x,t)}{\partial x} \tag{8}$$

donde $\phi$ es la porosidad, $C(x,t)$ es la concentración normalizada, $D$ es el coeficiente difusivo, $u$ es la velocidad de flujo (coeficiente convectivo), $x$ es la posición, $t$ es el tiempo (Peaceman, 1977) . La relación entre los coeficientes difusivo y convectivo le otorga al problema un carácter desde el punto de vista físico (Hernández, 2005) ; para $D$ pequeño el problema se considera fundamentalmente convectivo, mientras que para $D$ grande el problema se considera fundamentalmente difusivo. Es de hacer notar que la ecuación de convección-difusión se clasifica como una e.d.p. de tipo parabólico.

La ecuación de convección-difusión exhibe muchas de las características de ecuaciones de simulación de yacimientos de petróleo: $C(x,t)$ puede ser la función de presión y la función de saturación, y la constante $D=\frac{k}{\mu}$ puede representar la transmisibilidad en el reservorio, donde $k$ es la permeabilidad y $\mu$ es la viscosidad del fluido (Guo y Wang, 2005) . La ecuación de convección-difusión será la e.d.p. a tomar para el desarrollo del presente trabajo.

## 2.2 Condiciones de frontera

Las condiciones de frontera estan asociadas a variables que representan alguna dimensión espacial y que se hallan restringidas a una cierta región $\Omega$. Si la ecuación en derivadas parciales es de segundo orden, va a ser necesario conocer el valor de la solución, de su derivada o de una combinación lineal de ellas, en la frontera $\partial\Omega$ de la región conocida. Así pues se considerarán tres tipos de condiciones de frontera (Zill, 2000) :

1. **Condición de Dirichlet:** Es aquella donde se determina el valor de la solución en la frontera y suele representarse como:
$$C(x,t)|_{\partial\Omega} = g(t)$$
2. **Condición de Neumann:** es aquella donde se determina el valor de la frontera según la dirección normal de la solución en la frontera, es decir:
$$\frac{\partial C}{\partial \eta}(x,t)|_{\partial\Omega} = g(t)$$
3. **Condiciones de Robin:** esta condición también es conocida como condición mixta, ya que la misma es una combinación de las dos anteriores:
$$C(x,t)|_{\partial\Omega} + k\frac{\partial C}{\partial \eta}(x,t)|_{\partial\Omega} = g(t), \quad (k \in I\!R)$$

$x$ representa el conjunto de las variables que definen la región $\Omega$ del espacio.

Las condiciones en la frontera a emplear en el presente trabajo son las establecidas por Guo y Wang (2005) , las cuáles corresponden a condiciones de frontera de tipo Dirichlet.

## 2.3 Solución numérica de una ecuación diferencial parcial mediante diferencias finitas. Métodos implíctos aplicados a la ecuación de convección-difusión

A menudo, al intentar resolver una ecuación diferencial, se da el caso que no hay una solución analítica conocida, por lo que se requiere realizar una aproximación numérica de la solución (Mathews y Fink, 2000) . Para determinar ésta aproximación numérica, se cuenta con diversos métodos, entre los cuáles se pueden indicar el método de diferencias finitas, el método de elementos finitos, el método de volúmenes finitos, los métodos miméticos, entre otros.

El método de diferencias finitas consiste en convertir una función de valores continuos en una función de valores discretos, mediante el reemplazo del dominio continuo donde se desea determinar la solución de la ecuación diferencial por un conjunto de puntos discretos (mallado), y las derivadas en la ecuación diferencial son reemplazadas por operadores discretos que surgen del truncamiento de la serie de Taylor en el punto donde se desea hacer la aproximación (Causon y Mingham, 2010) . El método de diferencias finitas tiene como ventajas, entre otras, que es muy simple de utilizar, no requiere de integración numérica y es usado extensivamente en mecánica de fluidos y problemas de turbulencia (Cerrolaza, 2006) .

El método de diferencias finitas puede resumirse en la realización de los siguientes pasos (Kincaid y Cheney, 1994) :

i. Se realiza una discretización del dominio continuo, reemplazando dicho dominio por un conjunto discreto de puntos (nodos). El conjunto resultante de ésta discretización recibe el nombre de mallado. El objetivo de éste método consiste en determinar los valores aproximados, en los puntos del mallado, de la función solución de la ecuación diferencial a resolver.

ii. Se determinan las expresiones en diferencias que surgen de aproximar dichas derivadas de las ecuación diferencial a través del truncamiento de la serie de Taylor. Ésto también se hace en las condiciones iniciales y de frontera (de ser necesario).

iii. Se sustituyen tanto en la ecuación diferencial como en las condiciones iniciales y de frontera las versiones discretizadas de las derivadas obtenidas en el paso (ii). Ésto se hace para cada punto del mallado, lo cual genera un sistema de ecuaciones, el cual pudiese ser lineal o no lineal. En el caso de las e.d.p. cuasi-lineales, el sistema de ecuaciones es lineal. El sistema resultante recibe el nombre de esquema en diferencias finitas de la ecuación diferencial dada inicialmente.

iv. Se resuelve el sistema de ecuaciones resultantes. La solución de dicho sistema representa la aproximación numérica buscada.

En el caso de la ecuación de convección-difusión unidimensional objeto de éste trabajo:

$$\phi\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - u\frac{\partial C(x,t)}{\partial x} \tag{9}$$

si la discretización se hace de la siguiente forma:

$$\phi\frac{C_i^{n+1} - C_i^n}{\triangle t} = D\frac{C_{i-1}^n - 2C_i^n + C_{i+1}^n}{\triangle x^2} - u\frac{C_i^n - C_{i-1}^n}{\triangle x} \tag{10}$$

donde:

$C_i^n = C(x,t)$

$C_i^{n+1} = C(x,t+\triangle t)$

$C_{i-1}^n = C(x-\triangle x,t)$

$C_{i+1}^n = C(x+\triangle x,t)$

se afirma que el la formulación en diferencias finitas es explícita; mientras que si la discretización se hace de la siguiente forma:

$$\phi\frac{C_i^{n+1}-C_i^n}{\triangle t}=D\frac{C_{i-1}^{n+1}-2C_i^{n+1}+C_{i+1}^{n+1}}{\triangle x^2}-u\frac{C_i^{n+1}-C_{i-1}^{n+1}}{\triangle x} \tag{11}$$

donde:

$$C_{i-1}^{n+1}=C(x-\triangle x,t+\triangle t)$$

$$C_{i+1}^{n+1}=C(x+\triangle x,t+\triangle t)$$

se afirma que el la formulación en diferencias finitas es implícita. En forma general, los métodos explícitos tienen una cota superior en el valor de $\triangle t$ para estabilidad numérica, y requieren disminuir $\triangle t$ a medida que $\triangle x$ disminuye; mientras que los métodos implícitos son siempre estables, independiente del valor de $\triangle t$ (Nakamura, 1992) ; razón por la cual se emplearán éstos últimos en el desarrollo del presente trabajo. Los métodos implícitos conducen a un sistema de ecuaciones de la forma:

$$Ax=b\ ,\ A\in R^{n\times n}\ ,\ x\in R^{n\times 1}\ ,\ b\in R^{n\times 1} \tag{12}$$

## 2.4 La transformada *wavelet* y su aplicación en la solución de sistemas de ecuaciones lineales

En términos generales, la transformada *wavelet* es una herramienta que divide datos en diferentes componentes de frecuencia y luego estudia cada componente en resoluciones que se ajustan a su escala. La aplicación de la transformada *wavelet* a una serie de datos numéricos se puede hacer a través de un algoritmo basado en un banco de filtros que permite obtener la transformada *wavelet*, a partir de los datos de interés; dicho algoritmo se conoce como la transformada *wavelet* discreta (DWT por sus siglas en inglés) (Acevedo, 2009) .

Determinar la transformada discreta *wavelet* (DWT) de un vector *v* de *n* elementos consiste en obtener un vector transformado que tiene en una mitad, conocida como parte alta, la misma información de alta frecuencia que el vector original y en otra mitad, conocida como parte baja, la información de baja frecuencia. La información de baja frecuencia se obtiene aplicando un filtro *G*, denominado pasa-bajo, determinado por los coeficientes $\{g_i\}_{i=1}^{L}$, convolucionándolo con los datos; similarmente para obtener la

información de alta frecuencia se aplica un filtro de alta frecuencia $H$, denominado pasa-alto, $\{h_i\}_{i=1}^{L}$ . La representación matricial de los filtros $H$ y $G$ se muestra a continuación:

$$H = \begin{pmatrix} h_1 & h_2 & \cdots & h_L & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & h_1 & h_2 & \cdots & h_L & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_3 & h_4 & \cdots & h_L & 0 & \cdots & \cdots & 0 & h_1 & h_2 \end{pmatrix}_{\frac{n}{2}\times n} \tag{13}$$

$$G = \begin{pmatrix} g_1 & g_2 & \cdots & g_L & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & g_1 & g_2 & \cdots & g_L & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_3 & g_4 & \cdots & g_L & 0 & \cdots & \cdots & 0 & g_1 & g_2 \end{pmatrix}_{\frac{n}{2}\times n} \tag{14}$$

La matriz de transformación wavelet se define como:

$$W = \begin{pmatrix} H \\ G \end{pmatrix}_{n\times n} \tag{15}$$

Y la DWT aplicada al vector $v$ de $n$ elementos se determina a través de:

$$Wv = \begin{pmatrix} Hv \\ Gv \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_{n/2} \\ s_1 \\ \vdots \\ s_{n/2} \end{pmatrix} \tag{16}$$

Es de resaltar que la matriz de transformación *wavelet* $W$ es una matriz ortogonal, lo que garantiza que $W^{-1} = W^T$ .

Mediante el uso de la matriz de transformación *wavelet*, un sistema de ecuaciones lineales de la forma:

$$Ax = b \tag{17}$$

puede transformarse a un sistema de la forma:

$$\hat{A}\hat{x} = \hat{b} \tag{18}$$

donde:

$$\hat{A} = WAW^T \tag{19}$$

$$\hat{x} = Wx \tag{20}$$

$$\hat{b} = Wb \tag{21}$$

Obtenida la solución el sistema descrito por la (21) la solución $\hat{x}$ del sistema descrito por la (17) se obtiene mediante:

$$x = W^T\hat{x} \tag{22}$$

El procedimiento descrito se puede emplear solo para sistemas cuyo tamaño sea $n = q \cdot 2^k$ , para algún entero $k > 0$.

Es de resaltar que existen diferentes familias de *wavelets* con características que la definen y diferencian entre si, las cuales permiten adaptarlas a determinadas aplicaciones, dentro de las cuales se tienen: Daubechies, Symlets, Coiflets, Meyer, entre otras (Jiménez y Jiménez, 2010) .

## 2.5 Métodos iterativos para la resolución de sistemas de ecuaciones lineales. Métodos basados en subespacios de Krylov

En el cálculo científico la mayor parte de tiempo de cómputo se consume en la resolución de sistemas de ecuaciones lineales. Estos sistemas pueden ser bastante grandes; es el caso de los problemas de dinámica de fluidos computacional, donde cada ecuación se describe en términos del valor de un parámetro desconocido local (por ejemplo la velocidad local del flujo) dependiente de los valores

(desconocidos) en las cercanías (Van Der Vorst, 2003) . En general, las técnicas de discretización de ecuaciones en derivadas parciales suelen dar lugar a matrices grandes y dispersas. Una matriz dispersa se define, en forma general, como una matriz que tiene muy pocos elementos distintos de cero (Saad, 2003).

En general, para resolver un sistema de ecuaciones lineales $Ax = b$, siendo $A$ una matriz de tamaño $n \times n$ no singular, se tiene posibilidad de elegir entre los métodos directos y los métodos iterativos (Van Der Vorst, 2003) . Si bien una característica fundamental de los métodos directos es llegar a la solución exacta en un número finito de pasos, para el caso de matrices dispersas, los métodos directos presentan serias limitaciones, en relación al uso de espacio en memoria y en la exactitud numérica (Larrazabal, 2003) ; por otra parte, los métodos iterativos tiene como característica general que no alteran la estructura de la matriz de coeficientes del sistema; no requieren de tanta memoria como los métodos directos (Eliminación gaussiana, factorización LU,...) y si bien la convergencia de los métodos iterativos es, en general, lenta, siempre existe la posibilidad de ajustarse a exactitud deseada. La aceleración en la convergencia de los métodos iterativos puede lograrse mediante el empleo de precondicionadores (Larrazabal, 2003) . Los métodos iterativos pueden ser atractivos incluso cuando la matriz $A$ es densa (Van Der Vorst, 2003) .

Los métodos iterativos pueden dividirse en dos tipos: los métodos clásicos y los métodos de proyección (Larrazabal, 2003) . Dentro de los métodos clásicos pueden nombrarse la iteración de Jacobi, la iteración de Gauss-Seidel, y el método SOR. Éstos métodos son de concepción relativamente simple; sin embargo, son útiles en una clase restringida de problemas matriciales (Chen, 2005) . Los métodos de proyección consisten en extraer las soluciones aproximadas a partir de un subespacio previamente establecido. Dentro de los métodos de proyección, se encuentran los basados en subespacios de Krylov, que se consideran como una de las técnicas iterativas más importantes para la resolución de grandes sistemas lineales (Saad, 2003) . En el presente trabajo, se emplearán los métodos de proyección basados en subespacios de Krylov, por lo que a continuación se hace una breve descripción de los mismos.

Dado un vector $v$ de un espacio vectorial de dimensión $n$, y una matriz $A$ cuadrada de tamaño $n$, un subespacio de Krylov de orden $m$ se denota $K_m(A,v)$ y se define como (Van Der Vorst, 2003) :

$$K_m(A,v) = \left\{v, Av, A^2v, \cdots, A^{m-1}v\right\} \tag{23}$$

Un método iterativo basado en un subespacio de Krylov consiste en encontrar la solución del sistema:

$$Ax = b \tag{24}$$

en el subespacio: $K_k(A,r_0) = \left\{r_0, Ar_0, A^2r_0 \cdots, A^{k-1}r_0\right\} = \{r_0, r_1, r_2, \cdots, r_{k-1}\}$ ; con $r_0 = b - Ax_0$,

para cierto criterio de parada. Entre los métodos iterativos basados en subespacios de Krylov se tienen: Gradiente Conjugado (CG, *Conjugated Gradient*), Generalizado de Residuo Minimo (GMRES, *Generalized Minimum RESidual*), Gradiente BiConjugado (Bi-CG), Gradiente Conjugado al Cuadrado (CGS), Gradiente BiConjugado Estabilizado (BiCGStab), QMR, entre otros (Van Der Vorst, 2003) .

## 2.6 Precondicionamiento

La convergencia de los métodos iterativos es, en general, lenta. Sin embargo, siempre existe la posibilidad de ajustarse a exactitud deseada. La aceleración en la convergencia puede lograrse mediante el empleo de precondicionadores (Larrazabal, 2003) .

La idea básica de un precondicionador es buscar una matriz $M$ que sea una buena aproximación de $A$, y que sea fácil de invertir. Luego, se emplea la matriz $M^{-1}$ para corregir las sucesivas aproximaciones de un método numérico. Para precondicionar el sistema (24) , la matriz $M^{-1}$ debe ser tal que el número de condición de la matriz precondicionada debe ser mejor que el número de condición de la matriz $A$ (Larrazabal, 2003) . Existen fundamentalmente tres formas de precondicionar un sistema: por la izquierda, por la derecha y por ambos lados:

$$M^{-1}Ax = M^{-1}b \tag{25}$$

$$AM^{-1}Mx = b \tag{26}$$

$$M^{-1/2}AM^{-1/2}M^{1/2}x = M^{-1/2}b \tag{27}$$

El precondicionador ideal es la matriz $A^{-1}$, el cual es muy difícil de conseguir sin un esfuerzo computacional considerable. Por esto, la idea es encontrar una matriz $M^{-1}$que se aproxime lo suficiente a la matriz $A^{-1}$ a un costo computacional lo menor posible. Las técnicas para hallar $M^{-1}$ se conocen cómo técnicas de construcción de precondicionadores.

Las técnicas de construcción de precondicionadores clásicas son (Larrazabal, 2003) :

i. Precondicionadores polinomiales: se basan en hallar $M^{-1}$mediante un polinomio de grado $m$ en la matriz $A$:

$$M^{-1} = P_m(A) = q_0 + q_1 A + \cdots + q_m A^m \tag{28}$$

La elección de los coeficientes del polinomio se hace mediante la minimización de la función:

$$f(\lambda) = \|1 - P_m(\lambda)\| \tag{29}$$

en el intervalo $E$ que incluye al espectro de $A$. Ésta estrategia requiere de buenas cotas para el intervalo $E$, lo que en cierto modo limita su aplicabilidad.

ii. Precondicionadores Tipo SPAI (*SParse Approximate Inverse*, aproximación dispersa de la inversa): consiste en hallar una matriz $M$ tal que el producto $AM$ sea lo mas próximo posible a la matriz identidad. De gran aplicación en conjunto con métodos iterativos.

iii. Factorizaciones incompletas: consiste en factorizar la matriz $A$ sin introducir todo el llenado que produce el proceso de factorización. Representa una de las formas más sencillas de definir un precondicionador . Dentro de éste tipo de factorización, se considera la factorización incompleta ILU(0), que consiste en una descomposición de la forma $A = LU - R$ donde $L$ y $U$ tienen la misma estructura de elementos distintos de cero tanto en la parte triangular inferior y en la parte triangular superior de $A$, respectivamente, y $R$ es el residuo o error de la factorización. Éste tipo de factorización incompleta es bastante fácil de calcular y de bajo costo computacional (Saad, 2003), por tal razón, será empleada como técnica de precondicionamiento en el presente trabajo.

# CAPÍTULO III

# MARCO METODOLÓGICO

Una vez formulado el problema de investigación, definidos sus objetivos y establecidas las bases teóricas que orientarán el sentido de la misma, en el presente capítulo se describen los distintos métodos y procedimientos para llevar a cabo la investigación. A lo largo del presente capítulo se definen el tipo de investigación, el diseño de investigación, las técnicas e instrumentos de recolección de datos, y los procedimientos previstos en el desarrollo de la investigación.

## 1. Tipo y modalidad de investigación

El presente trabajo se considera como una investigación del tipo pura o básica, ya que se apoya dentro de un contexto teórico y se caracteriza por recabar datos para ampliar y evaluar la teoría de resolución de sistemas de ecuaciones lineales mediante el uso de métodos iterativos. (Rodríguez et al, 2008) .

Asi mismo, la modalidad de ésta investigación es de tipo descriptiva, en virtud que busca precisar el desempeño de los métodos iterativos en la resolución de sistemas de ecuaciones lineales que surgen del uso de métodos implícitos de diferencias finitas de orden superior cuando se emplean técnicas *wavelets* en dicho proceso de resolución (Rodríguez et al, 2008) .

## 2. Diseño de investigación

El diseño de ésta investigación se enmarca dentro del tipo experimental, debido a que se realizarán modificaciones en un grupo de variables del problema de investigación previamente definidas (variables independientes), para describir el efecto que tienen dichas modificaciones en otro grupo de variables dentro del mismo problema de investigación (variables dependientes) (Rodríguez et al, 2008) .

A su vez, el diseño de investigación para ésta propuesta está fundamentado en métodos cuantitativos, en virtud que se realizarán mediciones de desempeño, necesarias para garantizar la validez y confiabilidad de la investigación.

Las variables independientes en éste trabajo son:

- Condiciones físicas del problema de convección-difusión descrito por la ecuación diferencial dada: se tomarán diversos casos del trabajo realizado por Guo y Wang (2005).
- Tamaño del sistema de ecuaciones lineales: se variará el tamaño del sistema de ecuaciones a obtener producto de la discretización en diferencias finitas.
- Familia *wavelet* a emplear y número de coeficientes: se emplearán distintas familias *wavelet*, variando a su vez la cantidad de coeficientes en cada familia.
- Método iterativo a emplear en la resolución del sistema de ecuaciones lineales: se emplearán distintos métodos iterativos basados en subespacios de Krylov, en función de los sistemas de ecuaciones lineales obtenidos.
- Precondicionamiento: se resolverán los sistemas con precondicionamiento y sin precondicionamiento.

Las variables dependientes en éste trabajo son:

- Número de elementos no nulos en el sistema de ecuaciones lineales original y en el sistema de ecuaciones en la base *wavelet* .
- Número de iteraciones para lograr la convergencia en la resolución del sistema de ecuaciones lineales original y en el sistema en base *wavelet*.
- Tiempo de procesamiento en la resolución del sistema de ecuaciones lineales original y en el sistema en la base *wavelet* .
- Error relativo en la aproximación de la solución en la resolución del sistema de ecuaciones lineales original y en la resolución del sistema en la base *wavelet* .

## 3. Técnicas de recolección y de análisis de los datos

En cuanto a la técnica de recolección se utilizará la observación directa, debido a que el responsable de la investigación recoge los datos obtenidos mediante la medición de las variables dependientes e independientes a medida que se van realizando los experimentos numéricos (Rodríguez et al, 2008) . El

análisis de los datos recogidos se basará en técnicas de estadística descriptiva; mediante la representación gráfica de dichos datos, para la posterior descripción de los comportamientos observados en cada experimento realizado.

## 4. Procedimientos previstos

1. Se determinarán los esquemas numéricos de los sistemas de ecuaciones lineales provenientes del uso de métodos implícitos de diferencias finitas de orden superior.
2. Se seleccionarán un conjunto de métodos iterativos basados en subespacios de Krylov para la resolución de los sistemas de ecuaciones lineales obtenidos.
3. Se seleccionarán un conjunto de familias *wavelet* y la cantidad de coeficientes en cada familia para la transformación de los sistemas de ecuaciones lineales obtenidos.
4. Se determinará el número de elementos no nulos, el número de iteraciones, el tiempo de procesamiento y el error en la aproximación en la resolución los sistemas de ecuaciones lineales originales y en la base *wavelet*, con distintos métodos iterativos; sin precondicionamiento y con precondicionamiento en la base *wavelet*.
5. Se realizará la representación gráfica de los datos recogidos para realizar el análisis y medición del desempeño del uso de las técnicas *wavelets* en conjunto con los métodos iterativos seleccionados.

# CAPÍTULO IV

# RESULTADOS OBTENIDOS

En el presente capítulo se presentan los resultados obtenidos en la investigación, en términos de los objetivos planteados y los procedimientos previstos.

## 1. Determinación de los esquemas numéricos de los sistemas de ecuaciones lineales provenientes del uso de métodos implícitos de diferencias finitas de orden superior

Se determinó el esquema numérico proveniente del uso de métodos implícitos de diferencias finitas de orden 2, partiendo de las formulación propuesta por Guo y Wang (2005):

$$\phi\frac{\partial C}{\partial t} = \alpha_1\frac{\partial C}{\partial x} + \alpha_2\frac{\partial^2 C}{\partial x^2} + \alpha_3\frac{\partial^3 C}{\partial x^3} + \alpha_4\frac{\partial^4 C}{\partial x^4} \tag{30}$$

donde:

$\alpha_1 = -u$

$\alpha_2 = D - \frac{\phi u \triangle x - u^2 \triangle t}{2\phi}$

$\alpha_3 = -\frac{uD\triangle t}{\phi}$

$\alpha_4 = \frac{D^2\triangle t}{2\phi}$

La formulación implícita en diferencias finitas de la ec. (30) es:

$$C_i^{n+1} - C_i^n = \frac{\alpha_1\triangle t}{\phi\triangle x}C_1^* + \frac{\alpha_2\triangle t}{\phi\triangle x^2}C_2^* + \frac{\alpha_3\triangle t}{\phi\triangle x^3}C_3^* + \frac{\alpha_4\triangle t}{\phi\triangle x^4}C_4^* \tag{31}$$

donde:

$C_1^* = C_i^{n+1} - C_{i-1}^{n+1}$

$$C_2^* = C_{i+1}^{n+1} - 2C_i^{n+1} + C_{i-1}^{n+1}$$

$$C_3^* = C_{i+1}^{n+1} - 3C_i^{n+1} + 3C_{i-1}^{n+1} - C_{i-2}^{n+1}$$

$$C_4^* = C_{i+2}^{n+1} - 4C_{i+1}^{n+1} + 6C_i^{n+1} - 4C_{i-1}^{n+1} + C_{i-2}^{n+1}$$

la cual puede reescribirse como:

$$k_1 C_{i-2}^{n+1} + k_2 C_{i-1}^{n+1} + k_3 C_i^{n+1} + k_4 C_{i+1}^{n+1} + k_5 C_{i+2}^{n+1} = k_6 C_i^n \tag{32}$$

donde:

$$k_1 = \frac{\alpha_4 \triangle t}{\phi \triangle x^4} - \frac{\alpha_3 \triangle t}{\phi \triangle x^3}$$

$$k_2 = -4\frac{\alpha_4 \triangle t}{\phi \triangle x^4} + 3\frac{\alpha_3 \triangle t}{\phi \triangle x^3} + \frac{\alpha_2 \triangle t}{\phi \triangle x^2} - \frac{\alpha_1 \triangle t}{\phi \triangle x}$$

$$k_3 = 6\frac{\alpha_4 \triangle t}{\phi \triangle x^4} - 3\frac{\alpha_3 \triangle t}{\phi \triangle x^3} - 2\frac{\alpha_2 \triangle t}{\phi \triangle x^2} + \frac{\alpha_1 \triangle t}{\phi \triangle x} - 1$$

$$k_4 = -4\frac{\alpha_4 \triangle t}{\phi \triangle x^4} + \frac{\alpha_3 \triangle t}{\phi \triangle x^3} + \frac{\alpha_2 \triangle t}{\phi \triangle x^2}$$

$$k_5 = \frac{\alpha_4 \triangle t}{\phi \triangle x^4}$$

$$k_6 = -1$$

La ecuación (32) define el esquema numérico en diferencias finitas de orden 2 obtenido a partir de métodos implícitos, para $3 \leq i \leq N+2$ , $2 \leq n \leq N+1$ ; siendo $N$ el tamaño del sistema de ecuaciones a resolver. Las condiciones iniciales y en la frontera son de tipo Dirichlet, establecidas por Peaceman (1977) , las cuáles se indican a continuación:

$$C(x,0) = 0;\ x > 0. \tag{33}$$

$$C(0,t) = 1;\ C(\infty,t) = 0;\ t > 0. \tag{34}$$

En los nodos ubicados en la frontera ($i = 3$ , $i = N+2$), se agregarán nodos fantasmas siguiendo lo propuesto por Causon y Mingham (2010) , que consiste en asignar a los nodos fantasmas inmediatos los valores en la frontera.

La representación de los elementos no nulos del esquema numérico en diferencias finitas de orden 2 obtenido a partir de métodos implícitos para el caso $N = 64$, se muestra en la figura (1):

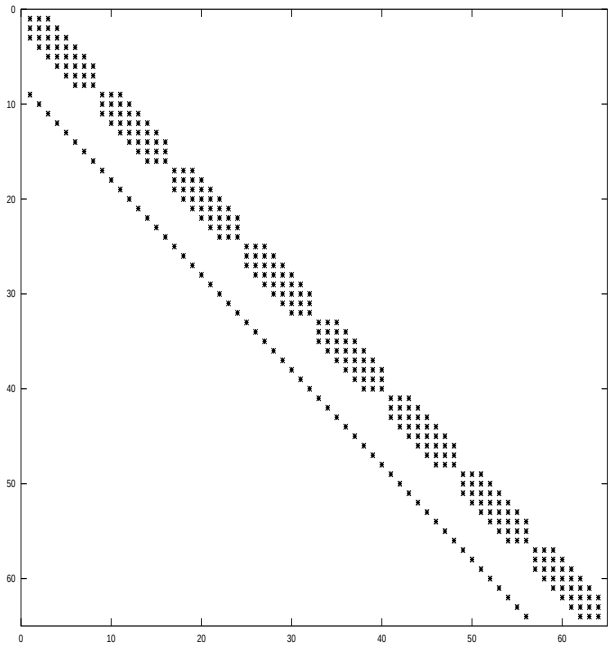


**Figura 1:** Elementos no nulos en el esquema numérico en diferencias finitas de orden 2 obtenido a partir de métodos implícitos

## 2. Selección de métodos iterativos basados en subespacios de Krylov para la resolución de los sistemas de ecuaciones lineales obtenidos

Como puede observarse en las figura (1), el esquema numérico obtenido corresponde a una matriz no simétrica; por ésta causa, deben emplearse métodos iterativos formulados para matrices generales. Saad (2003) propone cómo métodos para matrices generales no simétricas los indicados a continuación: Generalized Minimal Residual (GMRES), BiConjugate Gradient (BiCG), Quasi Minimal Residual (QMR), Conjugate Gradient Squared (CGS), y BiConjugate Gradient Stabilized (BiCGStab). El método GMRES ya ha sido estudiado en conjunto con técnicas *wavelet* (ver Villarín et al., 2012 ), por lo que en ésta investigación se trabajará con el resto de los métodos (BiCG, BiCGStab, CGS y QMR).

## 3. Selección de familias *wavelet* con su respectiva cantidad de coeficientes

Jiménez y Jiménez (2010) , señalan que existen diferentes familias de *wavelets* con características que la definen y diferencian entre si, las cuales permiten adaptarlas a determinadas aplicaciones. Trabajos

como los de Villarín et al. (2012) y Acevedo (2009) , desarrollan sus trabajos basados en la famila Daubechies con distinta cantidad de coeficientes. En ésta investigación, se trabajará con las familias Daubechies, Symlets, Biorthogonal y Reverse Biorthogonal, en los casos de 4, 6, 8 y 12 coeficientes.

## 4. Experimentos numéricos realizados y discusión de los resultados obtenidos

Se realizó la resolución de los esquemas numéricos de orden 2, para dos grupos de valores de los parámetros en dichos esquemas, tomados del trabajo de Guo y Wang (2005) :

1. Caso 1: $L = 1000; \phi = 0.30; u = 0.05; D = 0.1; \triangle x = 10; \triangle t = 42$ .
2. Caso 2: $L = 1000; \phi = 0.30; u = 1; D = 0.1; \triangle x = 10; \triangle t = 1.5$ .

En cada caso se hicieron las siguientes pruebas:

1. variando el tamaño del sistema: $N = 16$ , 64 , 256 .
2. variando la familia *wavelet* a emplear: Daubechies, Symlets, Biorthogonal y Reverse Biorthogonal.
3. variando la cantidad de coeficientes en cada familia *wavelet* empleada: 4 , 6 , 8 y 12 coeficientes.
4. variando el tipo de precondicionamiento: sin precondicionamiento, con el precondicionador $ILU(0)$ en la base *wavelet*.

En cada caso se determina la solución exacta del sistema original, la solución del sistema original con los métodos iterativos indicados y la solución del del sistema transformado a la base *wavelet* con los métodos iterativos indicados, realizando la medición de las siguientes variables:

- Número de elementos no nulos en la matriz de coeficientes del sistema.
- Número de iteraciones para lograr la convergencia.
- Tiempo de procesamiento.
- Error relativo en la aproximación de la solución.

El precondicionador empleado es: $M = LU$, donde $L$ y $U$ se obtienen de aplicar la factorización $ILU(0)$ a la matriz de coeficientes del sistema $A$.

Transformado el sistema $Ax = b$ al sistema $\hat{A}\hat{x} = \hat{b}$ ( sistema en la base *wavelet* con: $\hat{A} = WAW^T$ , $\hat{x} = Wx$ , $\hat{b} = Wb$ ; y siendo $W$ : matriz de transformación *wavelet* definida por la ec. (15) , ver capítulo 2) , la solución aproximada del sistema original partiendo de su cálculo en la base *wavelet* se obtiene mediante:

$$x = W^T \hat{x}$$

La generación de los esquemas numéricos con sus respectivas condiciones de borde, la generación de la matriz de transformación *wavelet*, y la transformación del esquema original a la base *wavelet* fue implementada computacionalmente en el lenguaje de programación Octave, el cual es software libre, y cuya sintáxis es equivalente a la empleada por el software comercial MATLAB®. La solución exacta se obtiene mediante la instrucción de Octave:

$$x = A \backslash b$$

y las implementaciones computacionales de los métodos iterativos fueron tomadas directamente del enlace: http://www.netlib.org/templates/matlab// . Los coeficientes *wavelet* de las familias empleadas fueron tomados directamente del enlace: http://wavelets.pybytes.com .

Se utilizó como criterios de parada para los métodos iterativos indicados: número máximo de iteraciones: 10000 ; tolerancia para la convergencia: $1 \times 10^{-6}$.

Los resultados de las pruebas realizadas son los siguientes:

**Caso 1, sin precondicionamiento**

Familia Daubechies

Los resultados obtenidos se observan en las figuras (2) a (17), que se muestran a continuación:

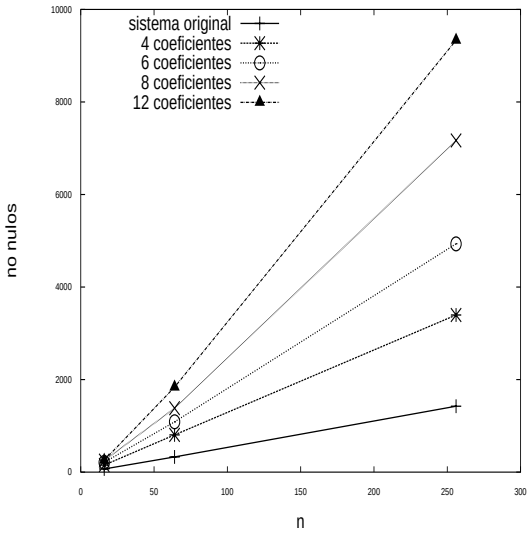


**Figura 2:** Caso 1 orden 2 Elementos no nulos Daubechies-BiCG sin precondicionamiento

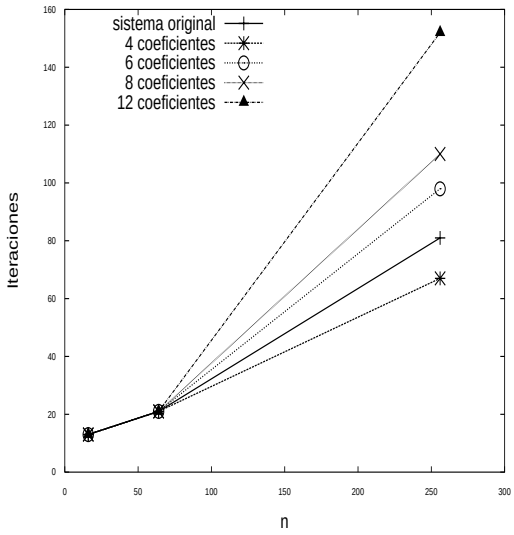


**Figura 3:** Caso 1 orden 2 Iteraciones Daubechies-BiCG sin precondicionamiento

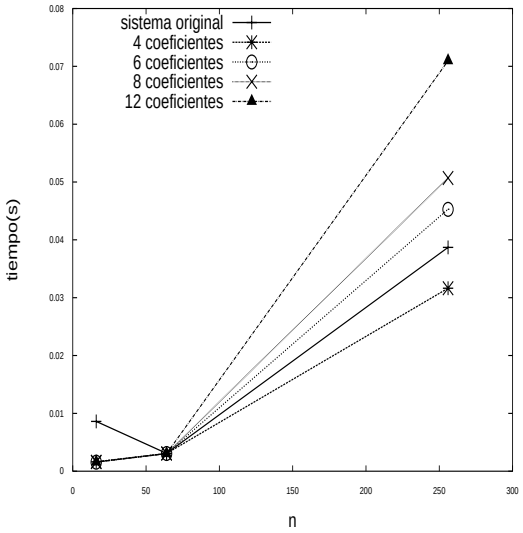


**Figura 4:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-BiCG sin precondicionamiento

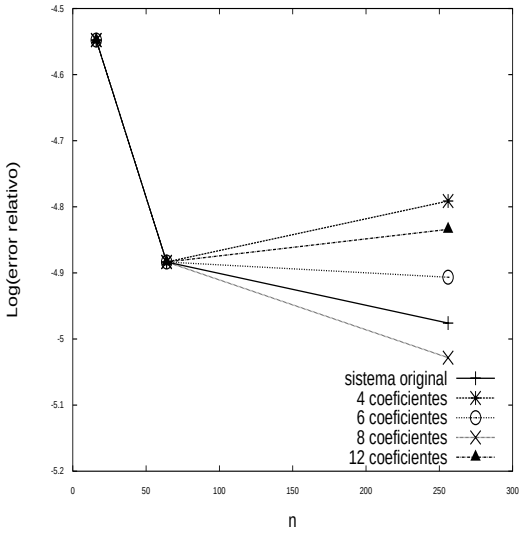


**Figura 5:** Caso 1 orden 2 Error relativo Daubechies-BiCG sin precondicionamiento

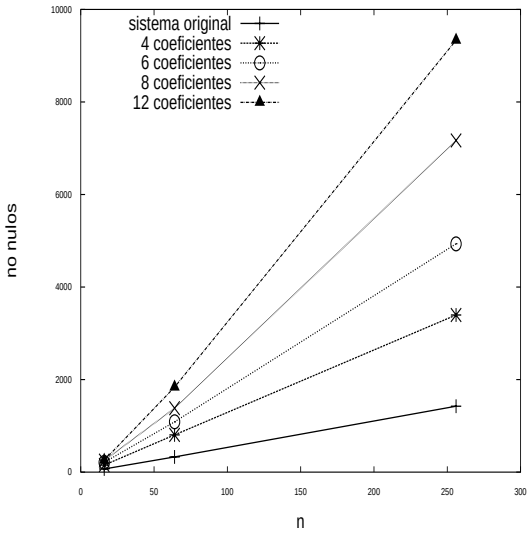


**Figura 6:** Caso 1 orden 2 Elementos no nulos Daubechies-BiCGStab sin precondicionamiento

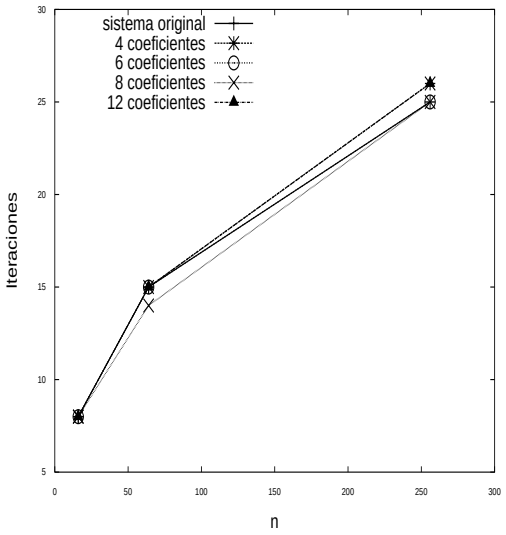


**Figura 7:** Caso 1 orden 2 Iteraciones Daubechies-BiCGStab sin precondicionamiento

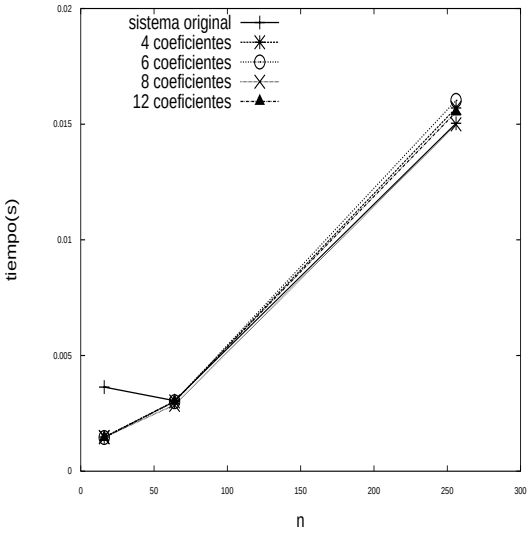


**Figura 8:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-BiCGStab sin precondicionamiento

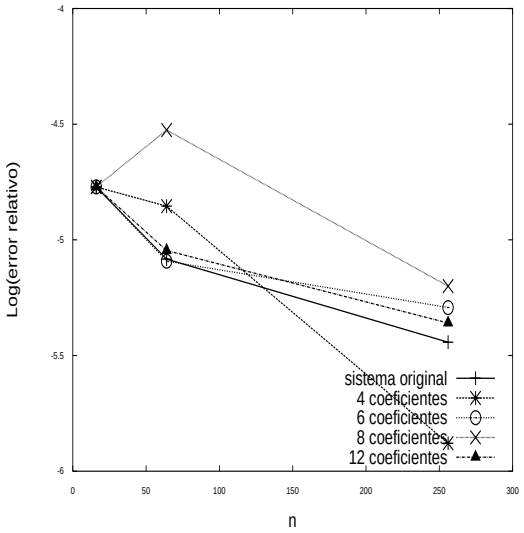


**Figura 9:** Caso 1 orden 2 Error relativo Daubechies-BiCGStab sin precondicionamiento

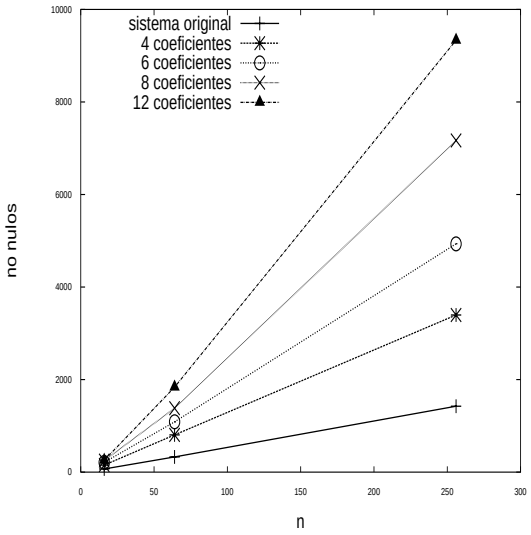


**Figura 10:** Caso 1 orden 2 Elementos no nulos Daubechies-CGS sin precondicionamiento

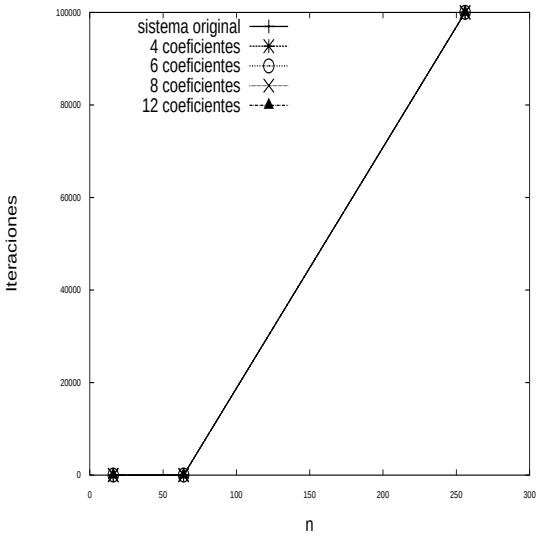


**Figura 11:** Caso 1 orden 2 Iteraciones Daubechies-CGS sin precondicionamiento

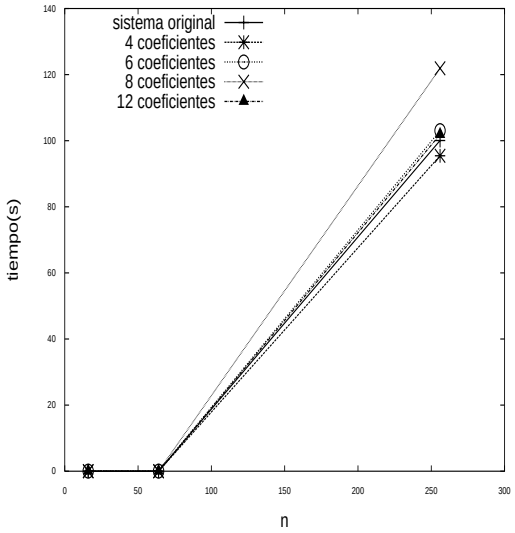


**Figura 12:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-CGS sin precondicionamiento

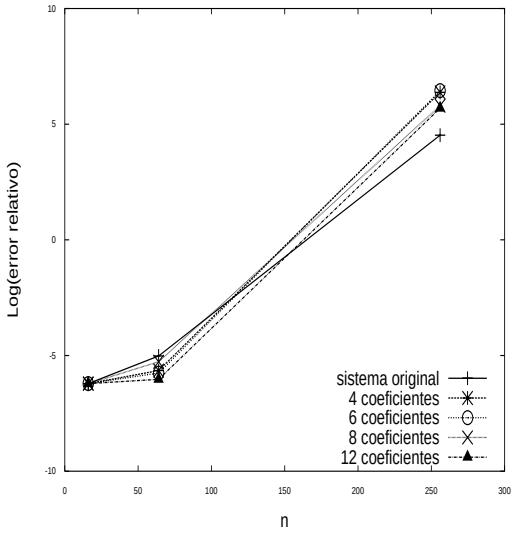


**Figura 13:** Caso 1 orden 2 Error relativo Daubechies-CGS sin precondicionamiento

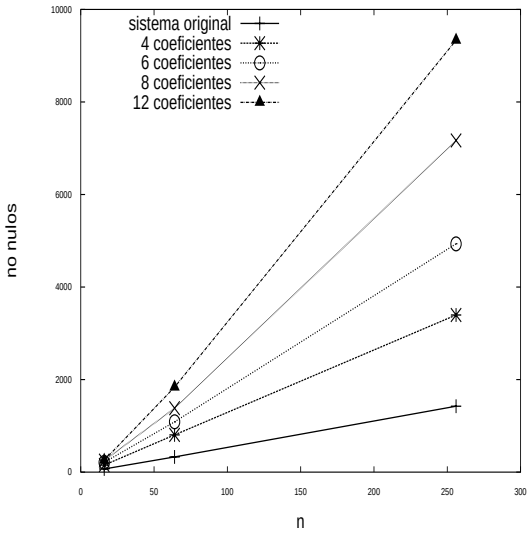


**Figura 14:** Caso 1 orden 2 Elementos no nulos Daubechies-QMR sin precondicionamiento

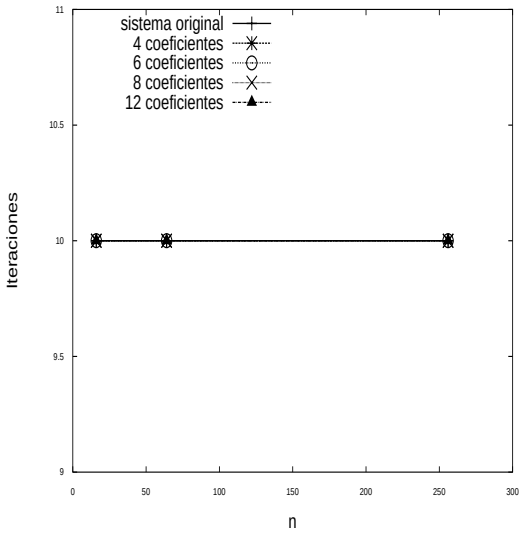


**Figura 15:** Caso 1 orden 2 Iteraciones Daubechies-QMR sin precondicionamiento

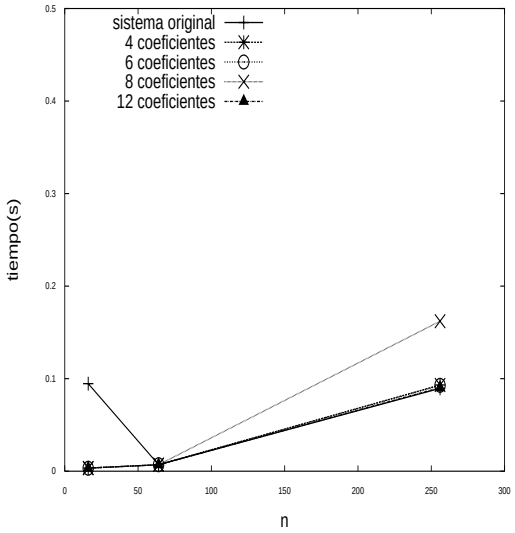


**Figura 16:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-QMR sin precondicionamiento

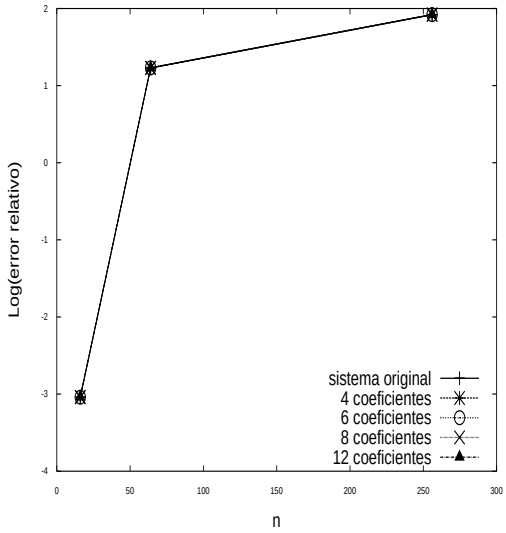


**Figura 17:** Caso 1 orden 2 Error relativo Daubechies-QMR sin precondicionamiento

En las figuras (2) a (17) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*. En el método BiCG la menor cantidad de iteraciones en el caso de 4 coeficientes; en el método BiCGStab, la menor cantidad de iteraciones se tiene en el caso de 8 coeficientes. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en los métodos CGS y QMR.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor tiempo se logra con 4 coeficientes; para el método BiCGStab, el menor tiempo se logra con 4 coeficientes; para el método CGS, el menor tiempo se logra con 4 coeficientes; para el método QMR, el menor tiempo se logra con 12 coeficientes.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor error se logra con 8 coeficientes; para el método BiCGStab, el menor error se logra con 4 coeficientes; para el método CGS, el menor error se logra en el sistema original; para el método QMR, los tiempos son fundamentalmente iguales en todos los casos.

En las figuras (2) a (17) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método BiCGStab con 4 coeficientes, en virtud que presenta la menor cantidad de tiempo de procesamiento y un error relativo que tiende a disminuir con el aumento del tamaño de la matriz de coeficientes del sistema (ver figuras 8 y 9).

Familia Symlets

Los resultados obtenidos se observan en las figuras (18) a (33), que se muestran a continuación:

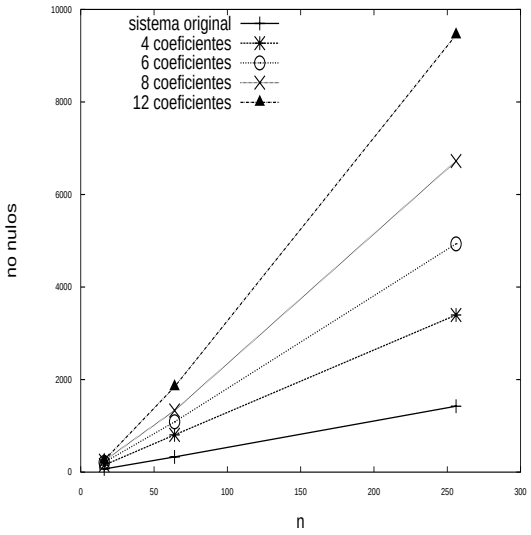


**Figura 18:** Caso 1 orden 2 Elementos no nulos Symlets-BiCG sin precondicionamiento

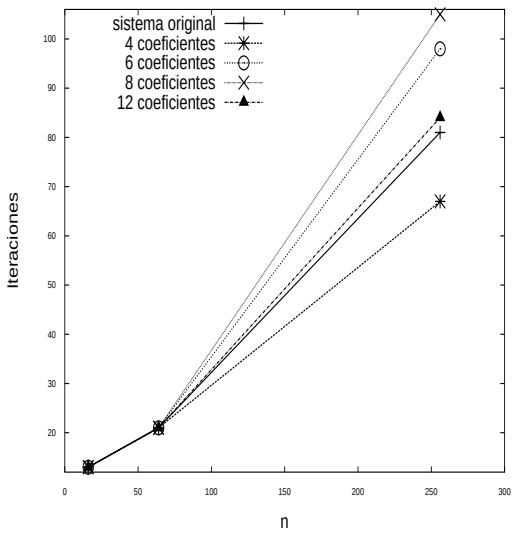


**Figura 19:** Caso 1 orden 2 Iteraciones Symlets-BiCG sin precondicionamiento

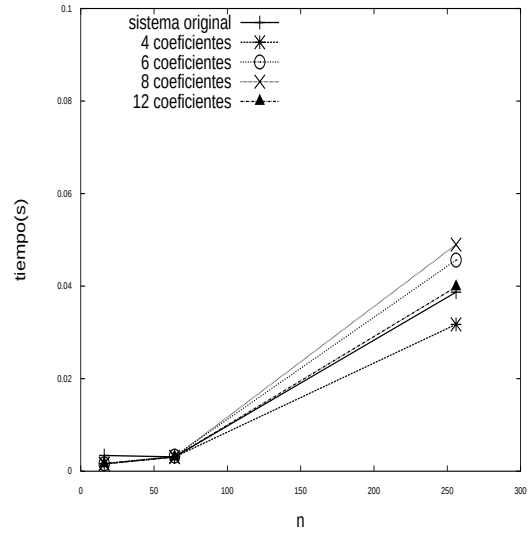


**Figura 20:** Caso 1 orden 2 Tiempo de procesamiento Symlets-BiCG sin precondicionamiento

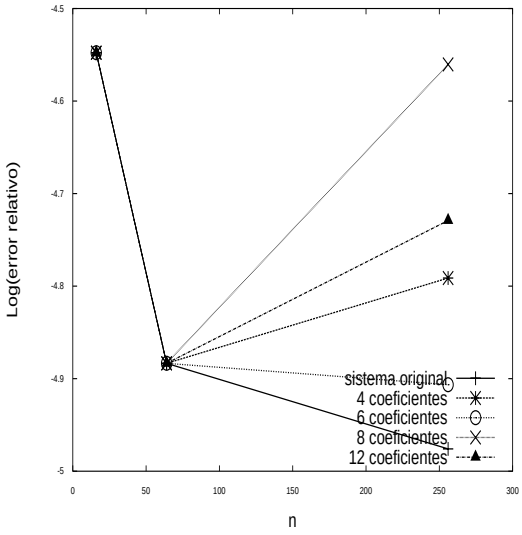


**Figura 21:** Caso 1 orden 2 Error relativo Symlets-BiCG sin precondicionamiento

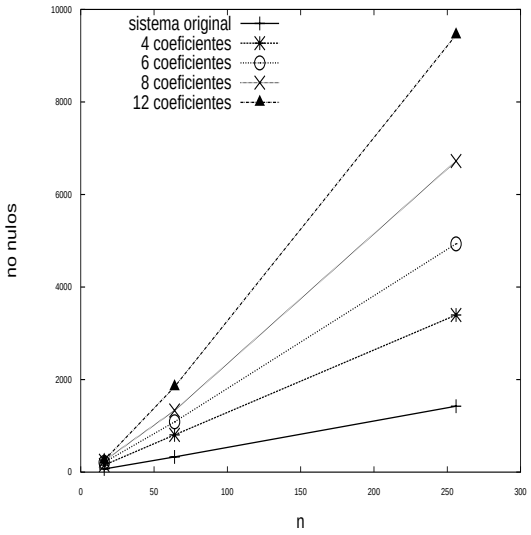


**Figura 22:** Caso 1 orden 2 Elementos no nulos Symlets-BiCGStab sin precondicionamiento

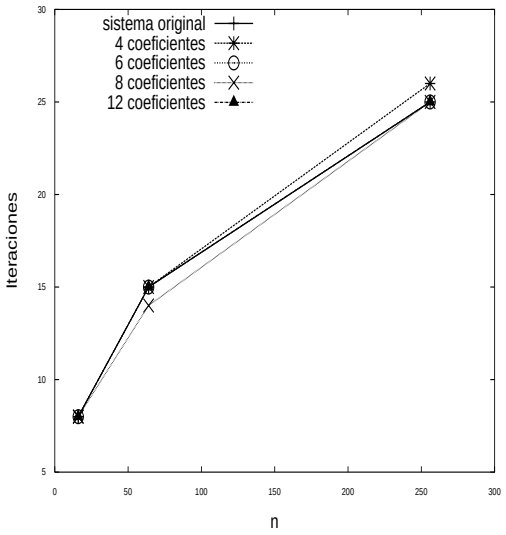


**Figura 23:** Caso 1 orden 2 Iteraciones Symlets-BiCGStab sin precondicionamiento

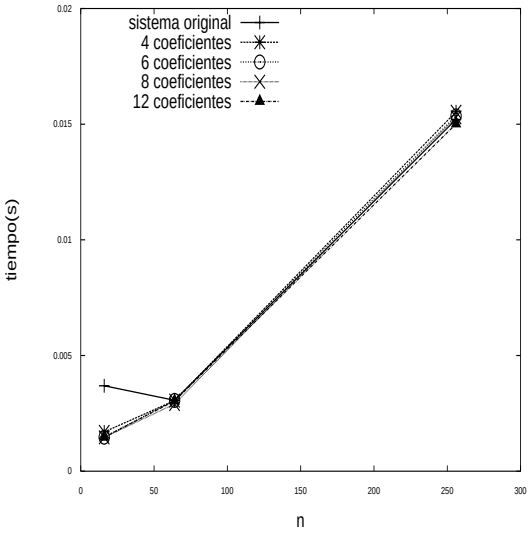


**Figura 24:** Caso 1 orden 2 Tiempo de procesamiento Symlets-BiCGStab sin precondicionamiento

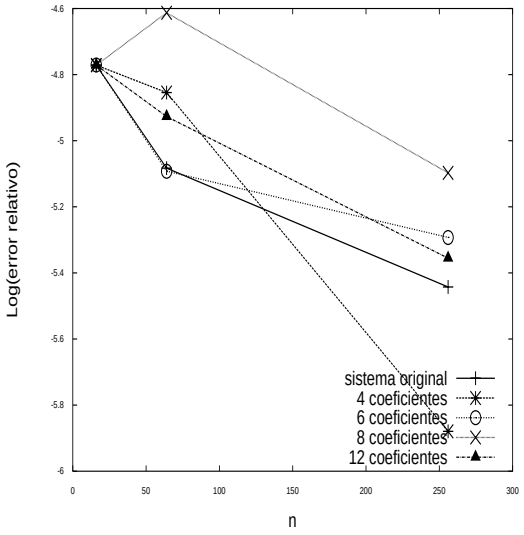


**Figura 25:** Caso 1 orden 2 Error relativo Symlets-BiCGStab sin precondicionamiento

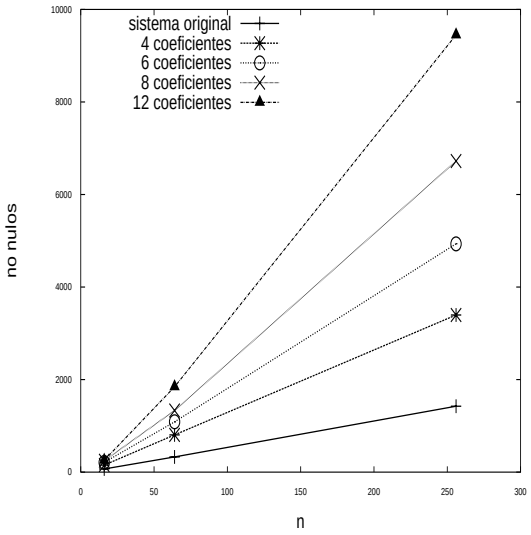


**Figura 26:** Caso 1 orden 2 Elementos no nulos Symlets-CGS sin precondicionamiento

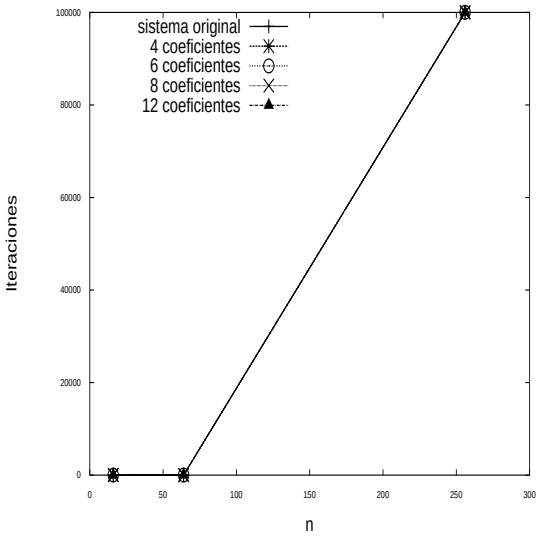


**Figura 27:** Caso 1 orden 2 Iteraciones Symlets-CGS sin precondicionamiento

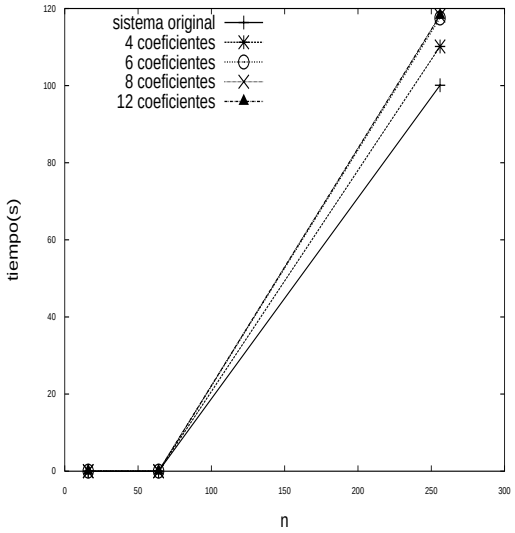


**Figura 28:** Caso 1 orden 2 Tiempo de procesamiento Symlets-CGS sin precondicionamiento

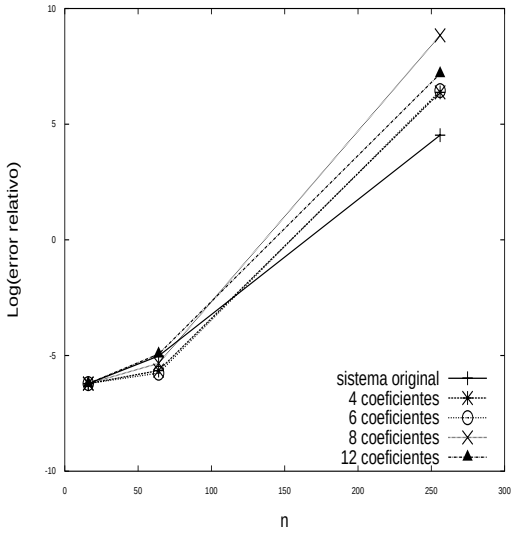


**Figura 29:** Caso 1 orden 2 Error relativo Symlets-CGS sin precondicionamiento

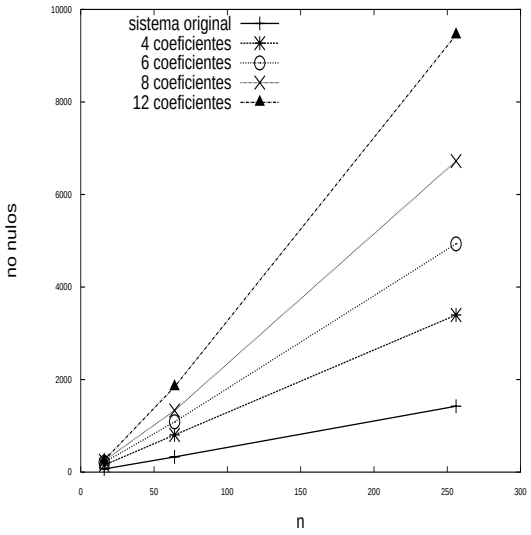


**Figura 30:** Caso 1 orden 2 Elementos no nulos Symlets-QMR sin precondicionamiento

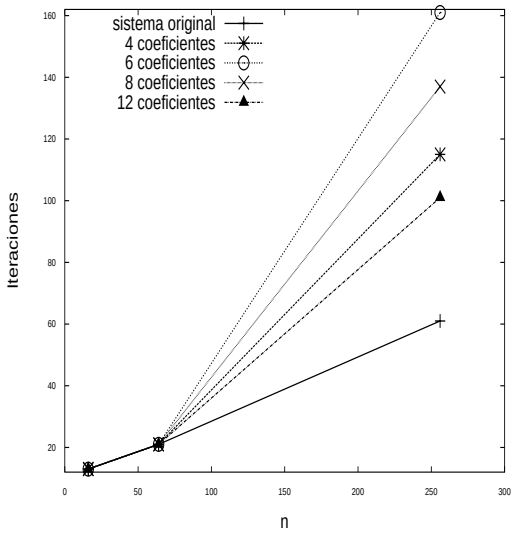


**Figura 31:** Caso 1 orden 2 Iteraciones Symlets-QMR sin precondicionamiento

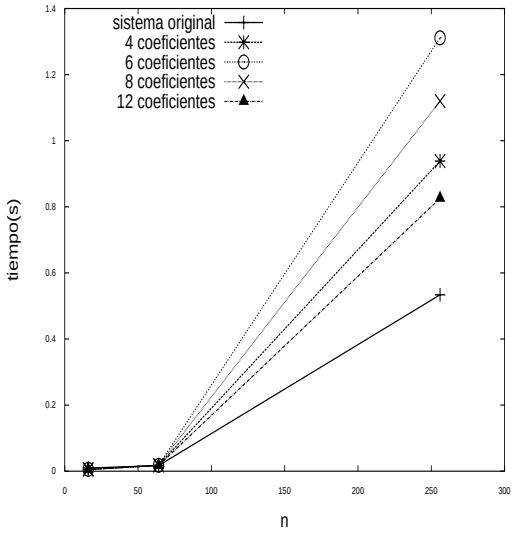


**Figura 32:** Caso 1 orden 2 Tiempo de procesamiento Symlets-QMR sin precondicionamiento

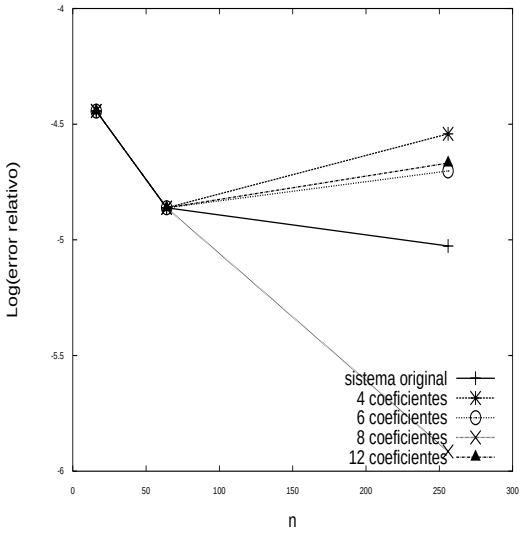


**Figura 33:** Caso 1 orden 2 Error relativo Symlets-QMR sin precondicionamiento

En las figuras (18) a (33) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*. En el método BiCG la menor cantidad de iteraciones se tiene en el caso de 4 coeficientes; en el método BiCGStab, la menor cantidad de iteraciones se tiene en el caso de 8 coeficientes; en el método CGS la cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en el método QMR, se tiene la menor cantidad de iteraciones en el sistema original.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor tiempo se logra con 4 coeficientes; para el método BiCGStab, los tiempos son muy similares para las distintas cantidades de coeficientes; para el método CGS, el menor tiempo se logra en el sistema original; para el método QMR, el menor tiempo se logra en el sistema original.

- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor error se logra en el sistema original; para el método BiCGStab, el menor error se logra con 4 coeficientes; para el método CGS, el menor error se logra en el sistema original; para el método QMR, el menor error se logra con 8 coeficientes.

En las figuras (18) a (33) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método BiCGStab con 4 coeficientes, en virtud que presenta la menor cantidad de tiempo de procesamiento y un error relativo que tiende a disminuir con el aumento del tamaño de la matriz de coeficientes del sistema (ver figuras 24 y 25).

Familia Biorthogonal

Los resultados obtenidos se observan en las figuras (34) a (49), que se muestran a continuación:

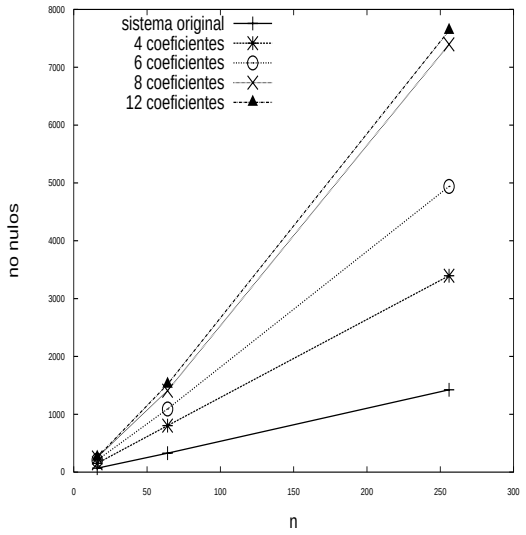


**Figura 34:** Caso 1 orden 2 Elementos no nulos Biorthogonal-BiCG sin precondicionamiento

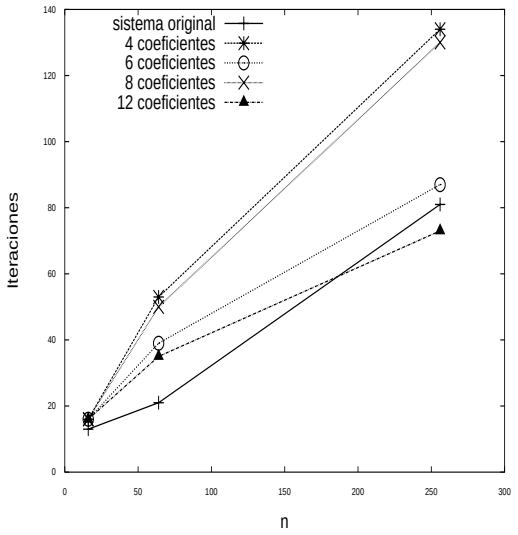


**Figura 35:** Caso 1 orden 2 Iteraciones Biorthogonal-BiCG sin precondicionamiento

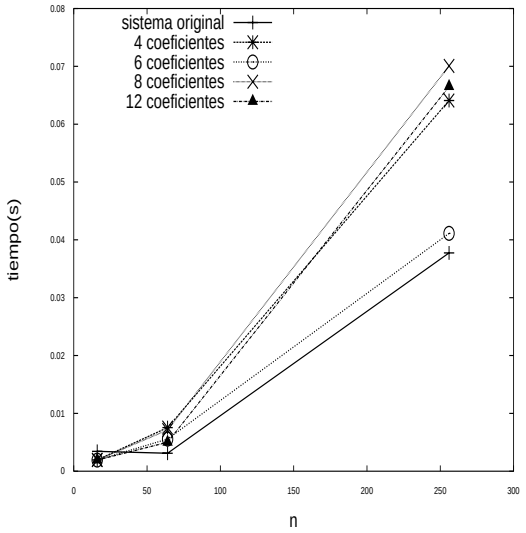


**Figura 36:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-BiCG sin precondicionamiento

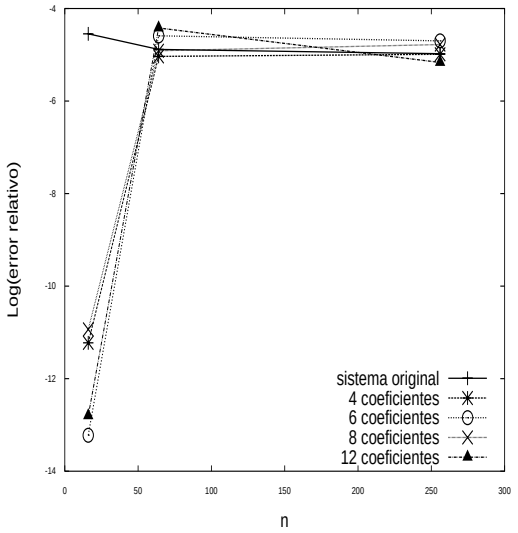


**Figura 37:** Caso 1 orden 2 Error relativo Biorthogonal-BiCG sin precondicionamiento

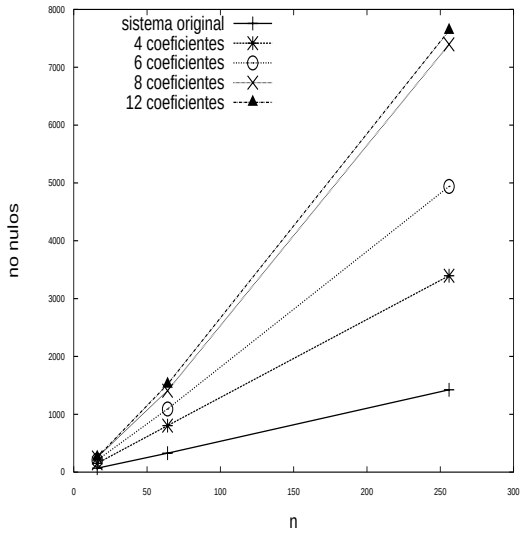


**Figura 38:** Caso 1 orden 2 Elementos no nulos Biorthogonal-BiCGStab sin precondicionamiento

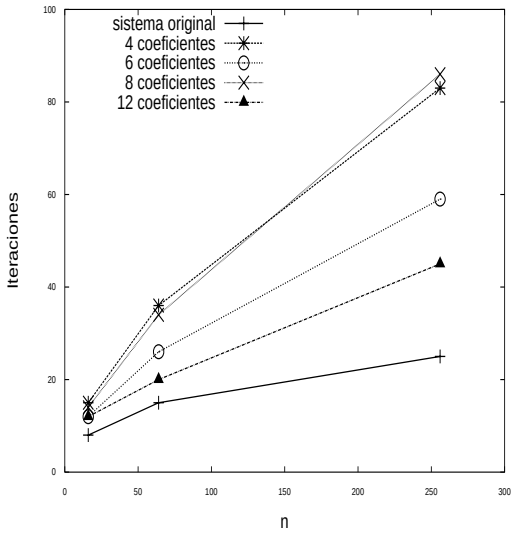


**Figura 39:** Caso 1 orden 2 Iteraciones Biorthogonal-BiCGStab sin precondicionamiento

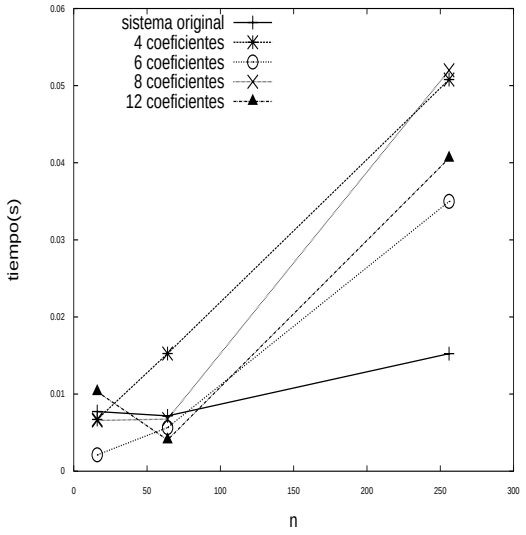


**Figura 40:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-BiCGStab sin precondicionamiento

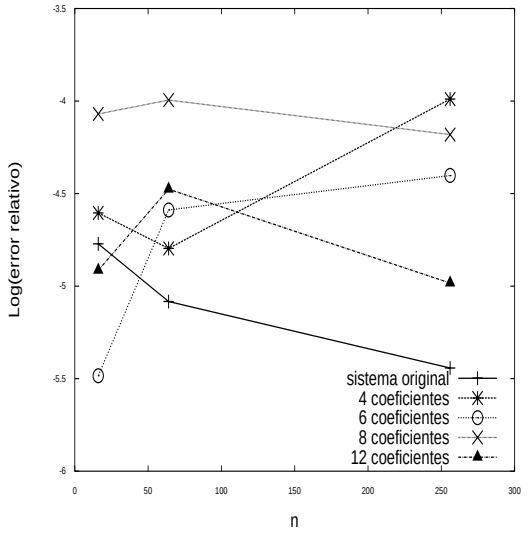


**Figura 41:** Caso 1 orden 2 Error relativo Biorthogonal-BiCGStab sin precondicionamiento

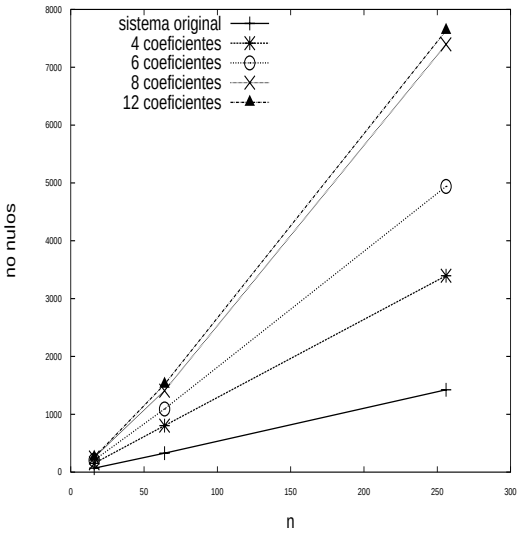


**Figura 42:** Caso 1 orden 2 Elementos no nulos Biorthogonal-CGS sin precondicionamiento

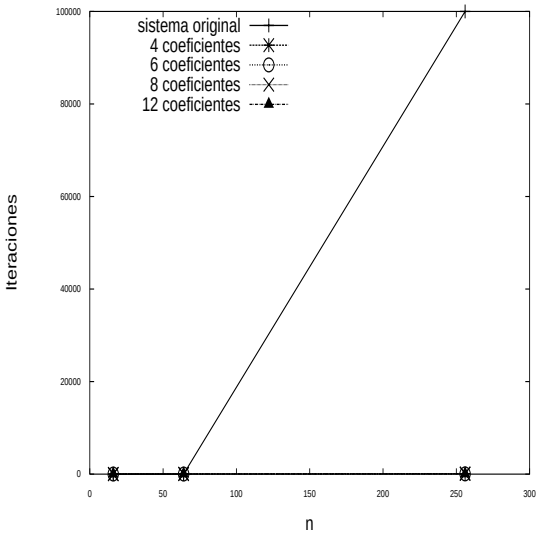


**Figura 43:** Caso 1 orden 2 Iteraciones Biorthogonal-CGS sin precondicionamiento

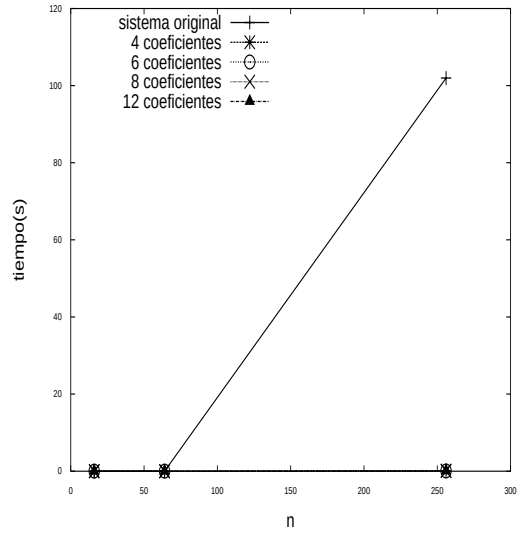


**Figura 44:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-CGS sin precondicionamiento

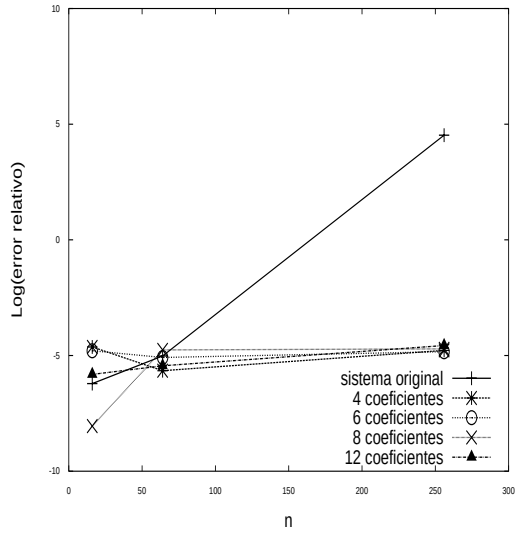


**Figura 45:** Caso 1 orden 2 Error relativo Biorthogonal-CGS sin precondicionamiento

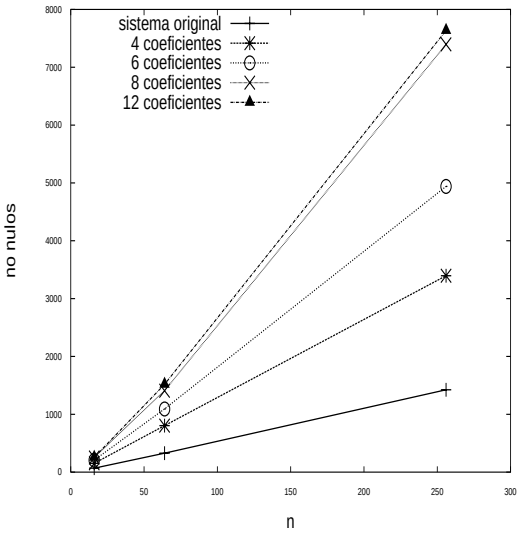


**Figura 46:** Caso 1 orden 2 Elementos no nulos Biorthogonal-QMR sin precondicionamiento

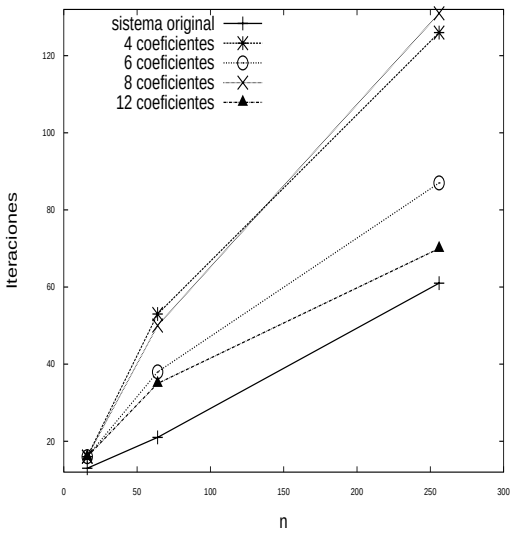


**Figura 47:** Caso 1 orden 2 Iteraciones Biorthogonal-QMR sin precondicionamiento

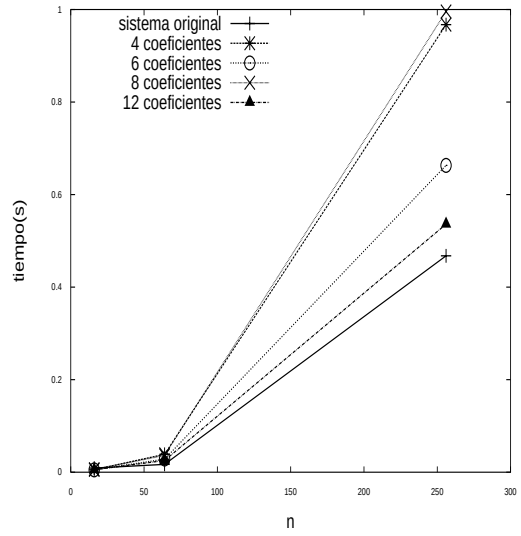


**Figura 48:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-QMR sin precondicionamiento

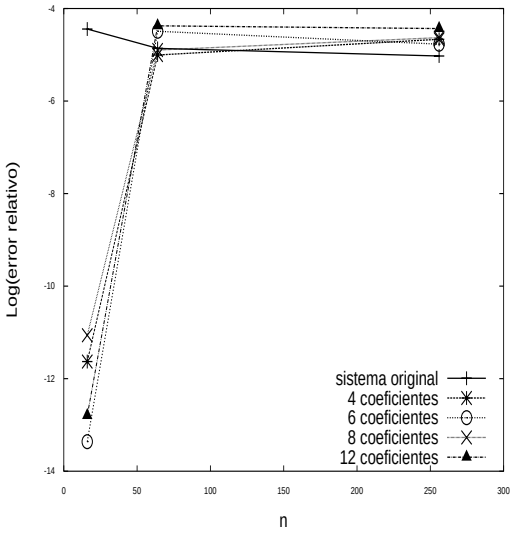


**Figura 49:** Caso 1 orden 2 Error relativo Biorthogonal-QMR sin precondicionamiento

En las figuras (34) a (49) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en los métodos BiCG, BiCGStab y QMR; en el método CGS no se presenta en general tal variación. La menor cantidad de iteraciones en todos los casos de los métodos BiCG, BiCGStab y QMR se presenta en el sistema original.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG, BiCGStab y QMR, el menor tiempo se logra en el sistema original; para el método CGS, prácticamente no hay diferencia en los tiempos de procesamiento en los distintos casos.
- Los errores relativos son en la mayoría de los casos del mismo orden, a excepción del método BiCGStab, donde el menor error se alcanza en el sistema original.

En las figuras (34) a (49) también puede observarse que en los experimentos numéricos realizados, ninguna combinación de métodos iterativos con distinta cantidad de coeficientes para la familia *wavelet* ofrece un mejor desempeño que los métodos iterativos en el sistema original.

Familia Reverse Biorthogonal

Los resultados obtenidos se observan en las figuras (50) a (65), que se muestran a continuación:

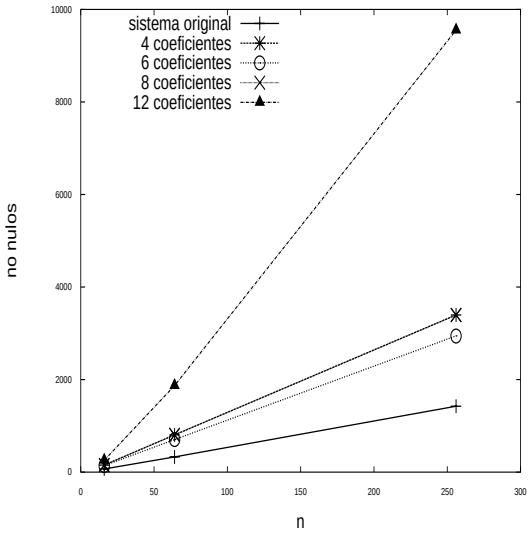


**Figura 50:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-BiCG sin precondicionamiento

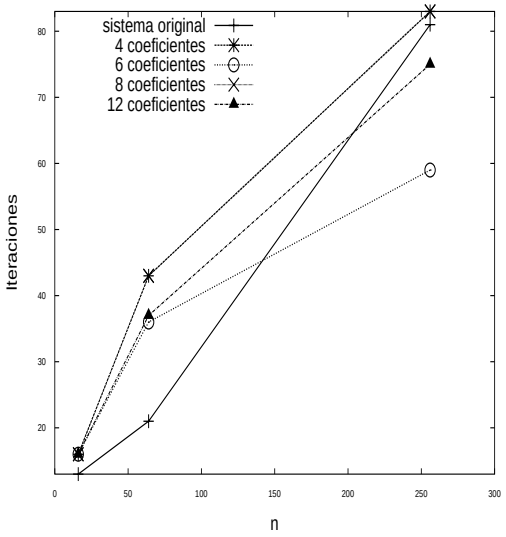


**Figura 51:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-BiCG sin precondicionamiento

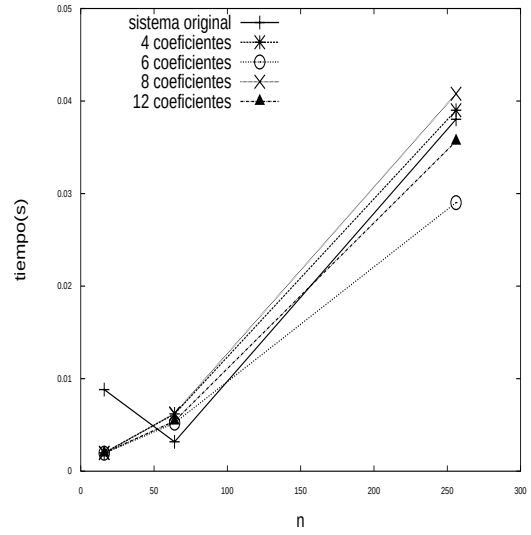


**Figura 52:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCG sin precondicionamiento

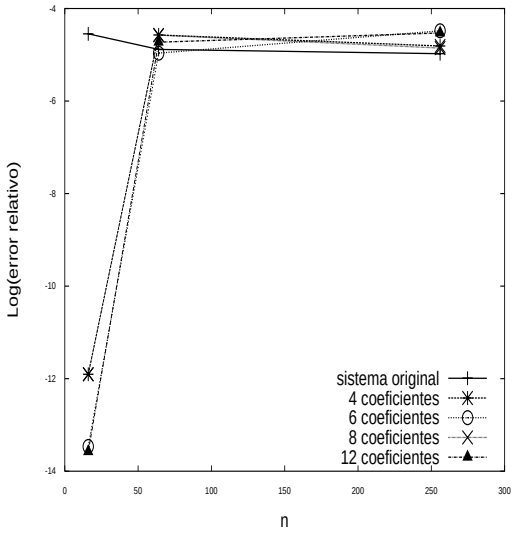


**Figura 53:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-BiCG sin precondicionamiento

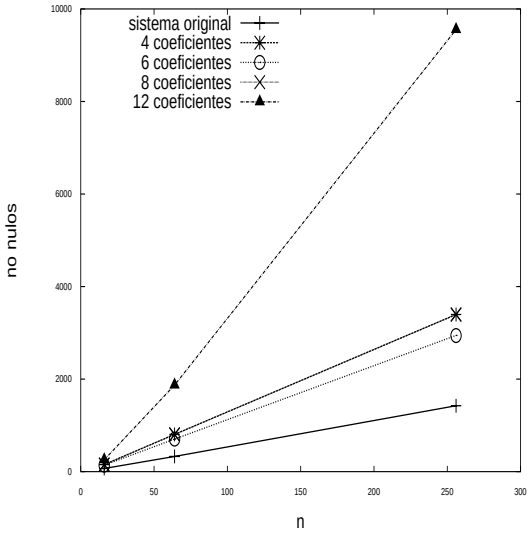


**Figura 54:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-BiCGStab sin precondicionamiento

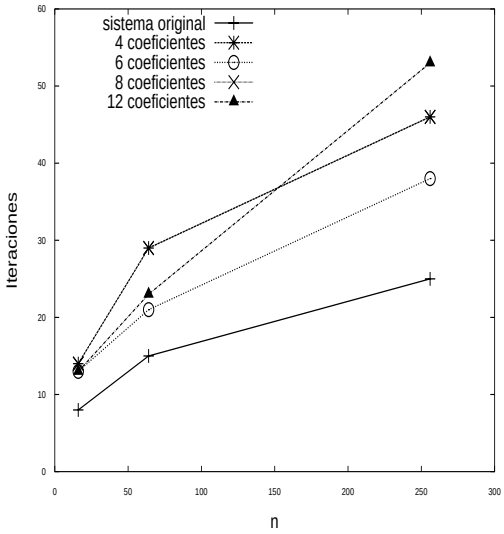


**Figura 55:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-BiCGStab sin precondicionamiento

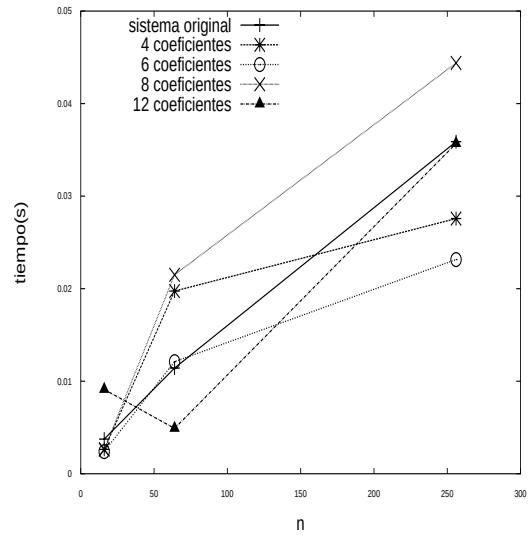


**Figura 56:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCGStab sin precondicionamiento

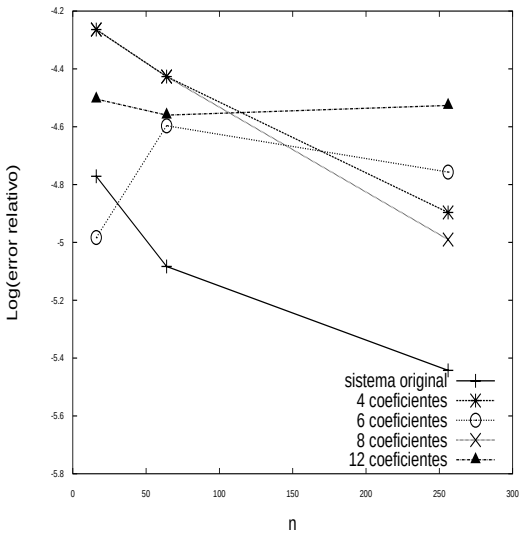


**Figura 57:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-BiCGStab sin precondicionamiento

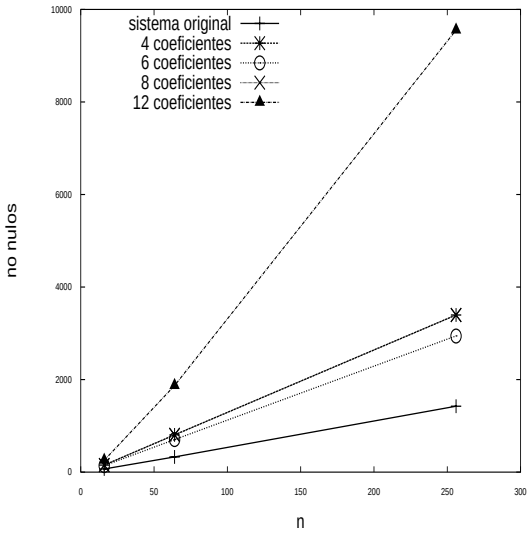


**Figura 58:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-CGS sin precondicionamiento

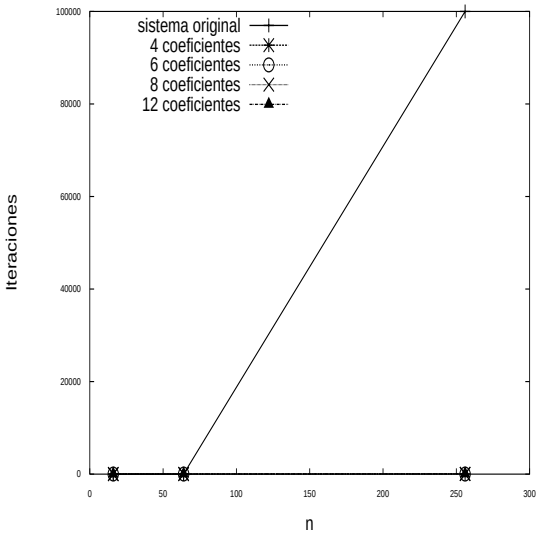


**Figura 59:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-CGS sin precondicionamiento

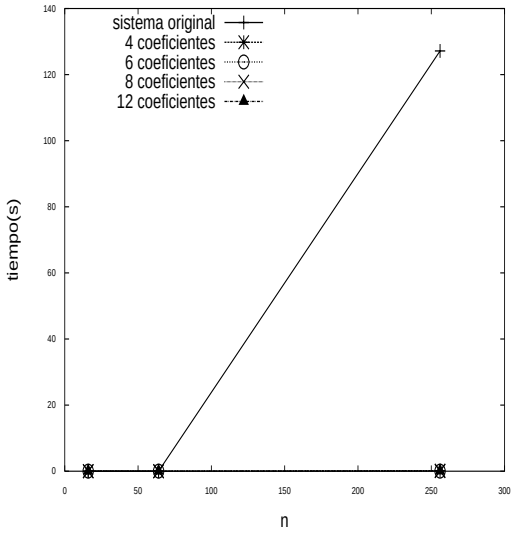


**Figura 60:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-CGS sin precondicionamiento

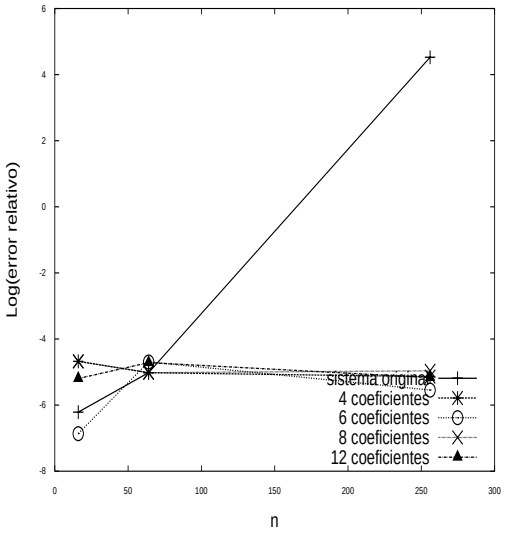


**Figura 61:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-CGS sin precondicionamiento

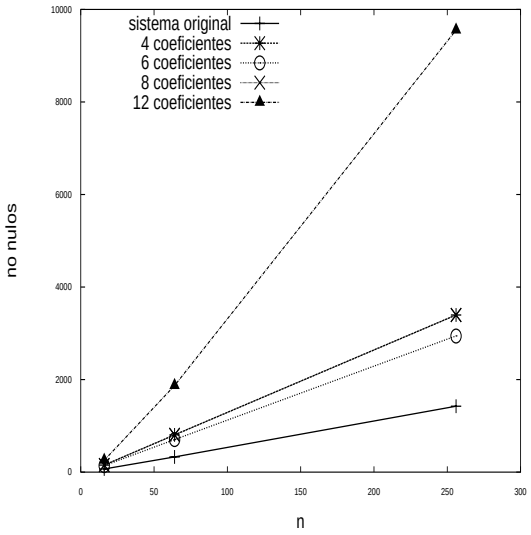


**Figura 62:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-QMR sin precondicionamiento

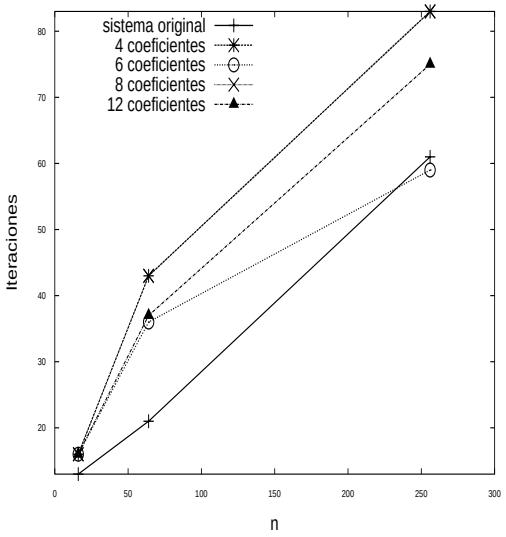


**Figura 63:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-QMR sin precondicionamiento

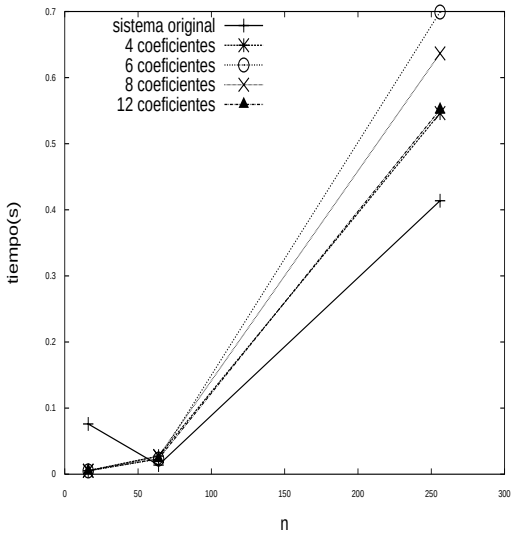


**Figura 64:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-QMR sin precondicionamiento

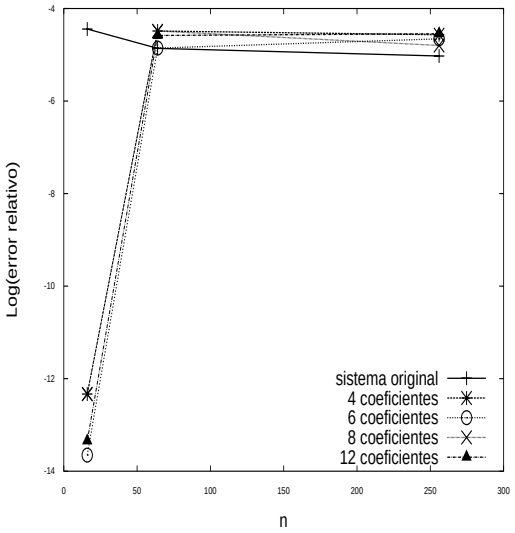


**Figura 65:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-QMR sin precondicionamiento

En las figuras (50) a (65) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en los métodos BiCG, BiCGStab y QMR; en el método CGS no se presenta en general tal variación. La menor cantidad de iteraciones de la mayoría de los casos en los métodos BiCG, BiCGStab y QMR se presenta en el sistema original.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra con 6 coeficientes; para el método QMR, el menor tiempo se logra en el sistema original; para el método CGS, prácticamente no hay diferencia en los tiempos de procesamiento en los distintos casos.
- Los errores relativos son en la mayoría de los casos del mismo orden, a excepción del método BiCGStab, donde el menor error se logra en el sistema original.

En las figuras (50) a (65) también puede observarse que en los experimentos numéricos realizados, ninguna combinación de métodos iterativos con distinta cantidad de coeficientes para la familia *wavelet* ofrece un mejor desempeño que los métodos iterativos en el sistema original.

**Caso 1, con precondicionamiento**

Familia Daubechies

Los resultados obtenidos se observan en las figuras (66) a (81), que se muestran a continuación:

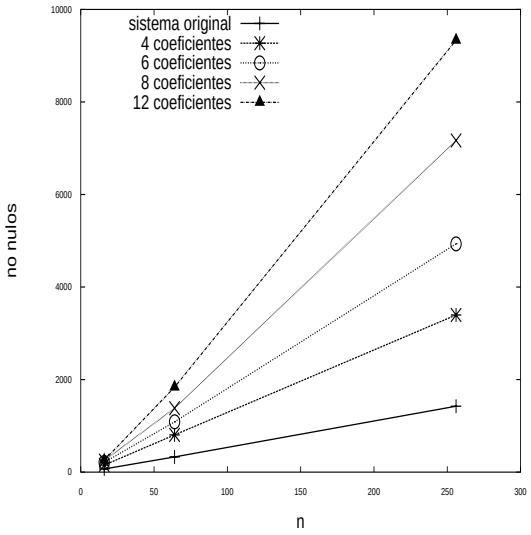


**Figura 66:** Caso 1 orden 2 Elementos no nulos Daubechies-BiCG con precondicionamiento

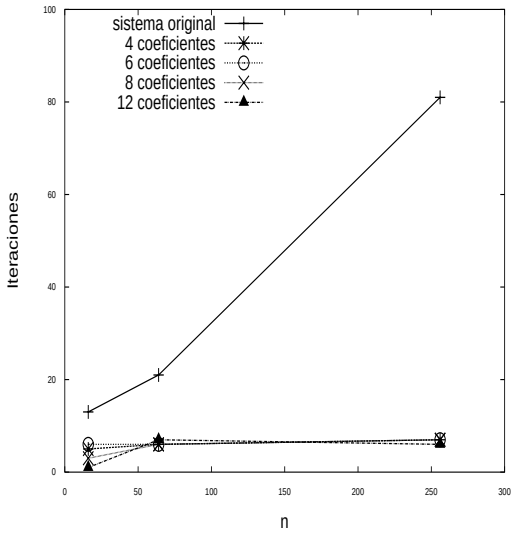


**Figura 67:** Caso 1 orden 2 Iteraciones Daubechies-BiCG con precondicionamiento

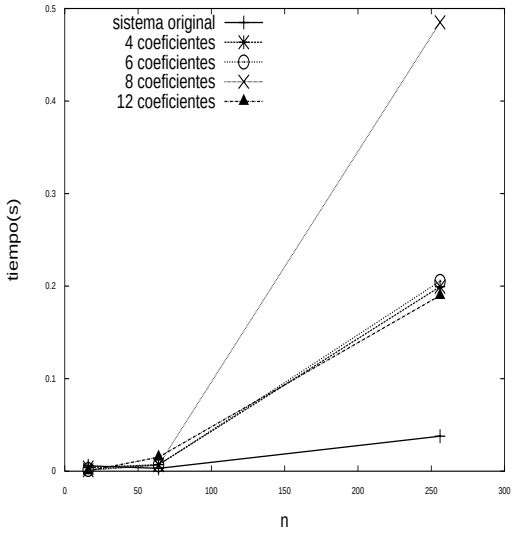


**Figura 68:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-BiCG con precondicionamiento

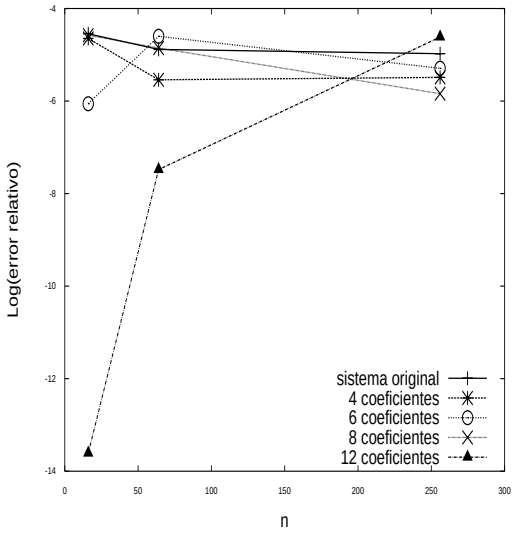


**Figura 69:** Caso 1 orden 2 Error relativo Daubechies-BiCG con precondicionamiento

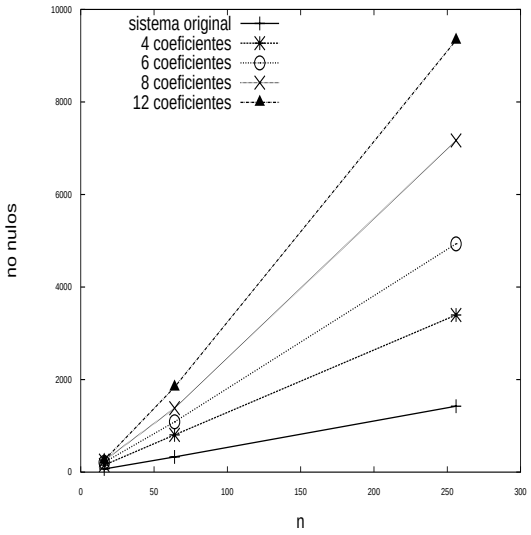


**Figura 70:** Caso 1 orden 2 Elementos no nulos Daubechies-BiCGStab con precondicionamiento

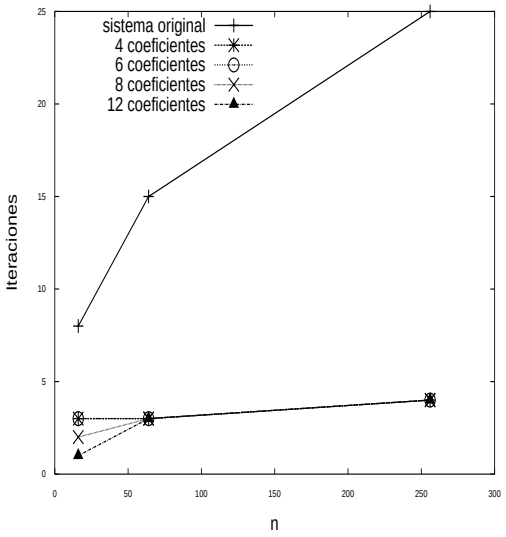


**Figura 71:** Caso 1 orden 2 Iteraciones Daubechies-BiCGStab con precondicionamiento

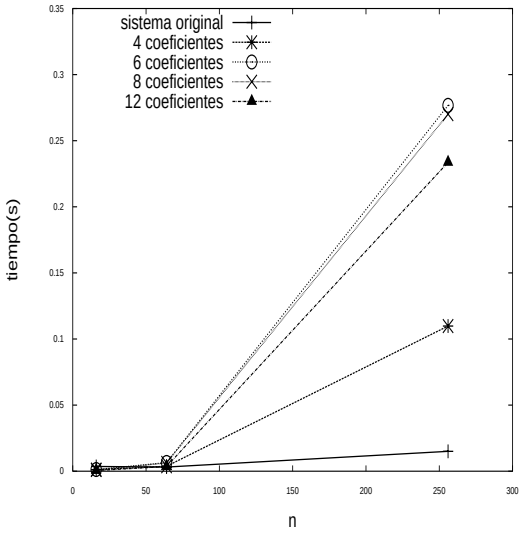


**Figura 72:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-BiCGStab con precondicionamiento

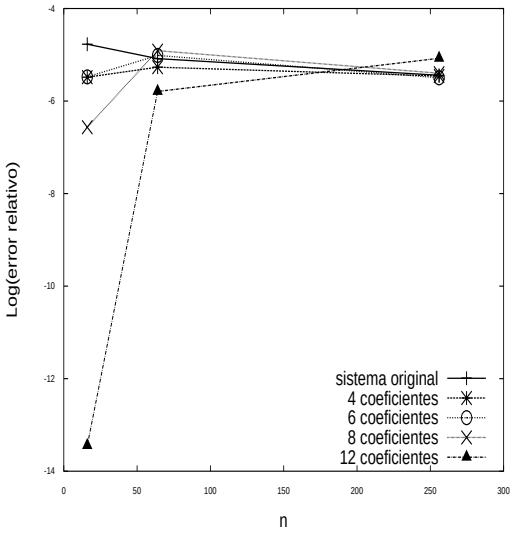


**Figura 73:** Caso 1 orden 2 Error relativo Daubechies-BiCGStab con precondicionamiento

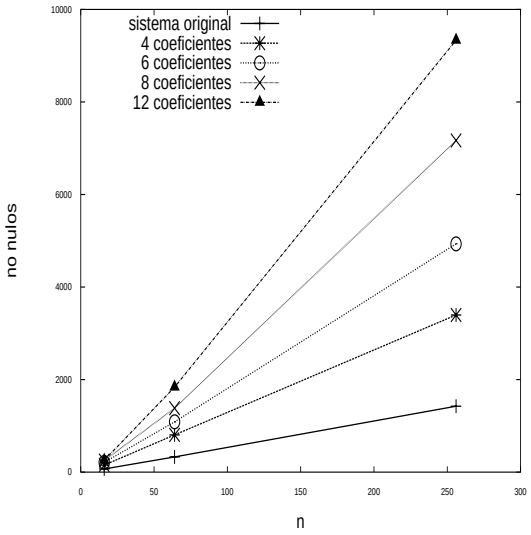


**Figura 74:** Caso 1 orden 2 Elementos no nulos Daubechies-CGS con precondicionamiento

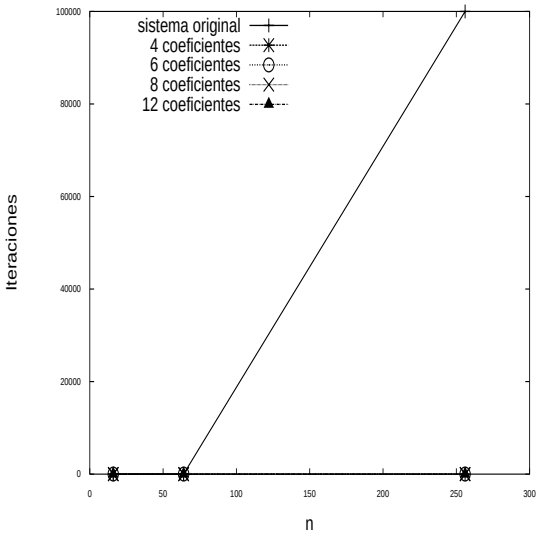


**Figura 75:** Caso 1 orden 2 Iteraciones Daubechies-CGS con precondicionamiento

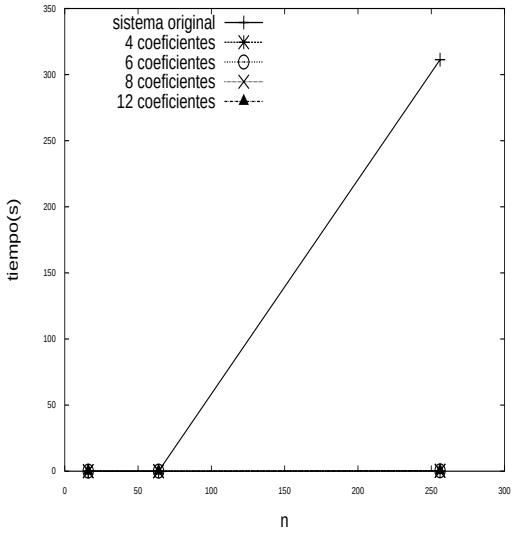


**Figura 76:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-CGS con precondicionamiento

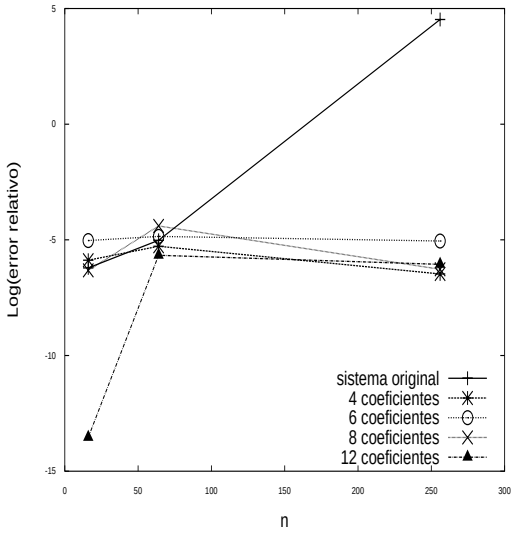


**Figura 77:** Caso 1 orden 2 Error relativo Daubechies-CGS con precondicionamiento

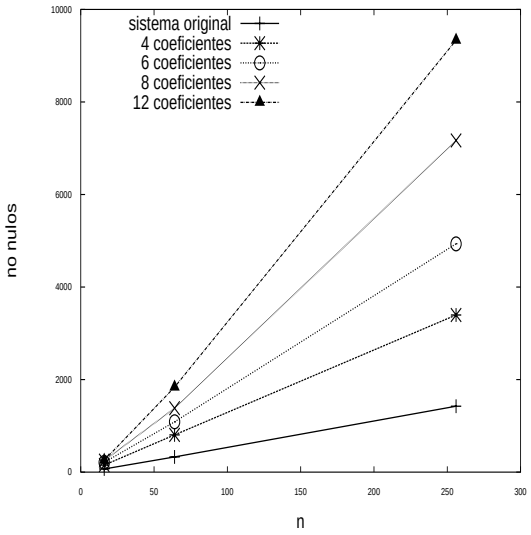


**Figura 78:** Caso 1 orden 2 Elementos no nulos Daubechies-QMR con precondicionamiento

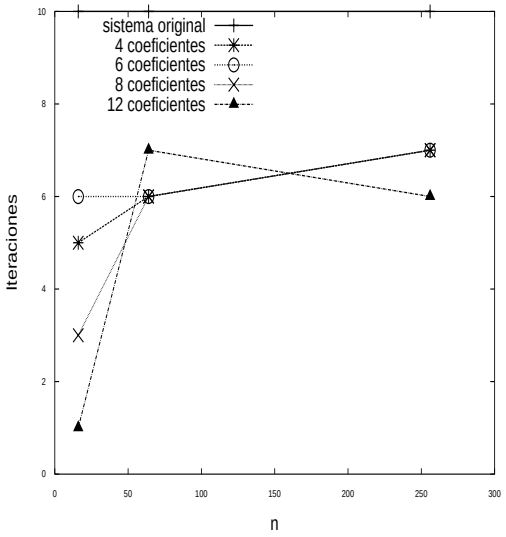


**Figura 79:** Caso 1 orden 2 Iteraciones Daubechies-QMR con precondicionamiento

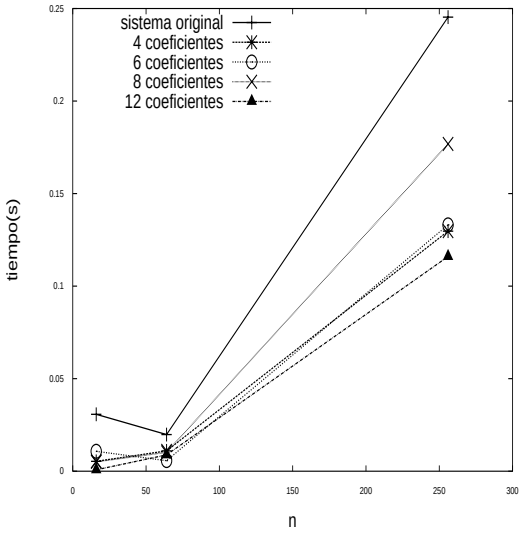


**Figura 80:** Caso 1 orden 2 Tiempo de procesamiento Daubechies-QMR con precondicionamiento

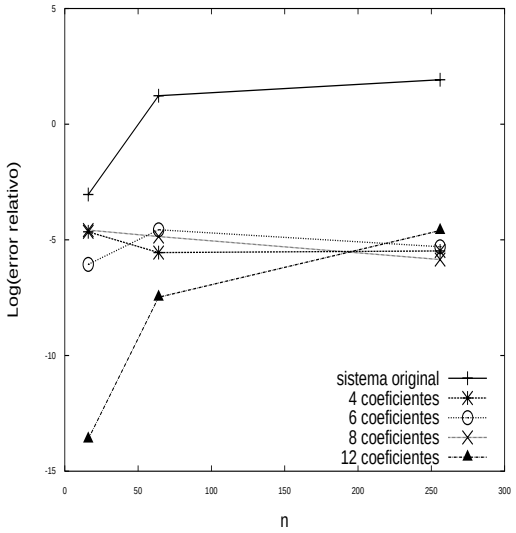


**Figura 81:** Caso 1 orden 2 Error relativo Daubechies-QMR con precondicionamiento

En las figuras (66) a (81) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta en los métodos BiCG y BiCGStab; en el método QMR la menor cantidad de iteraciones se logra con 12 coeficientes. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 12 coeficientes.

- Los menores errores relativos se presentan para los casos de 12 coeficientes en los distintos métodos.

En las figuras (66) a (81) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en todos los casos (ver figuras 80 y 81).

Familia Symlets

Los resultados obtenidos se observan en las figuras (82) a (97), que se muestran a continuación:

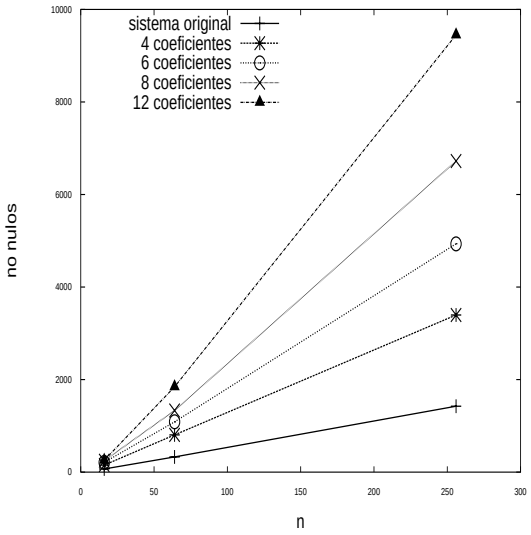


**Figura 82:** Caso 1 orden 2 Elementos no nulos Symlets-BiCG con precondicionamiento

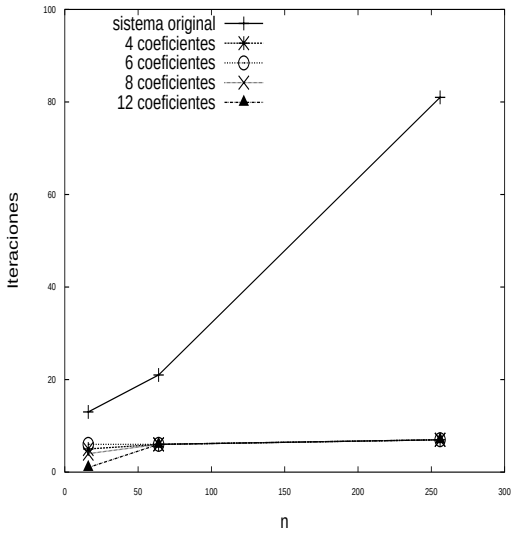


**Figura 83:** Caso 1 orden 2 Iteraciones Symlets-BiCG con precondicionamiento

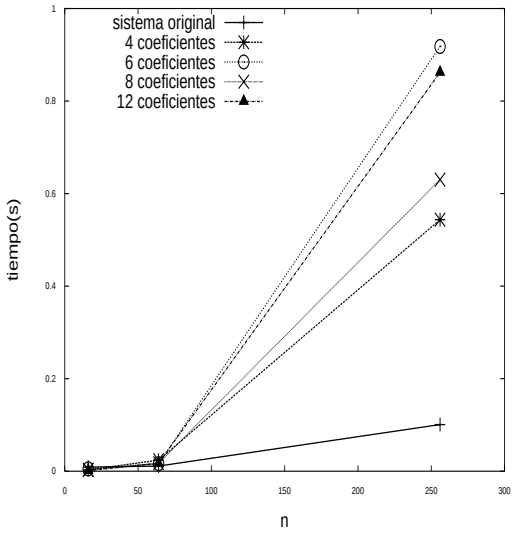


**Figura 84:** Caso 1 orden 2 Tiempo de procesamiento Symlets-BiCG con precondicionamiento

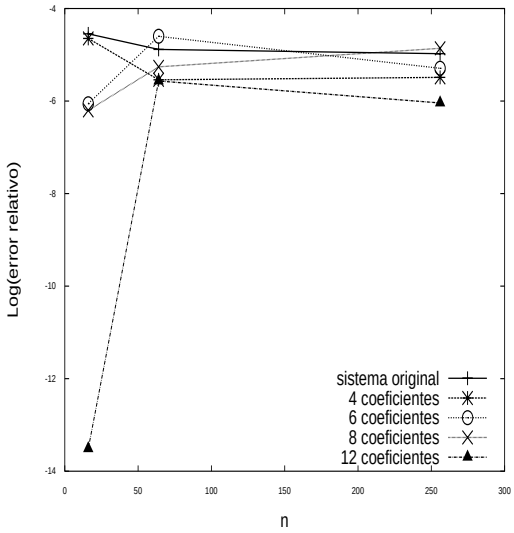


**Figura 85:** Caso 1 orden 2 Error relativo Symlets-BiCG con precondicionamiento

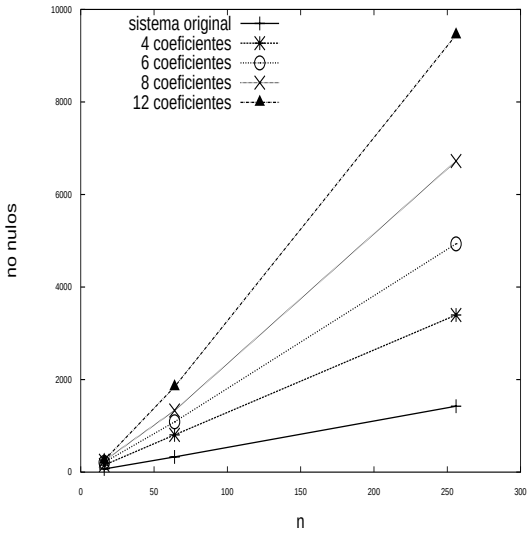


**Figura 86:** Caso 1 orden 2 Elementos no nulos Symlets-BiCGStab con precondicionamiento

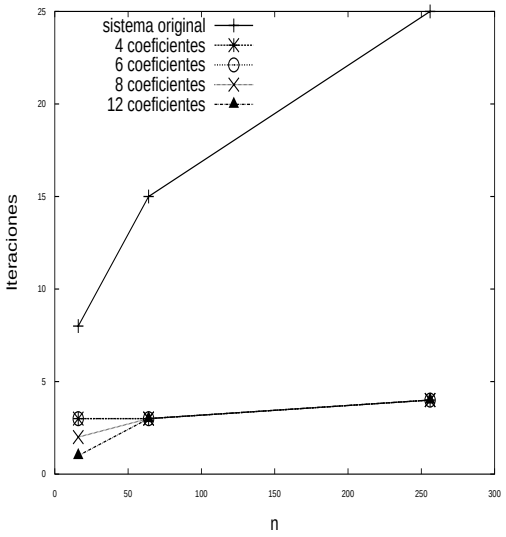


**Figura 87:** Caso 1 orden 2 Iteraciones Symlets-BiCGStab con precondicionamiento

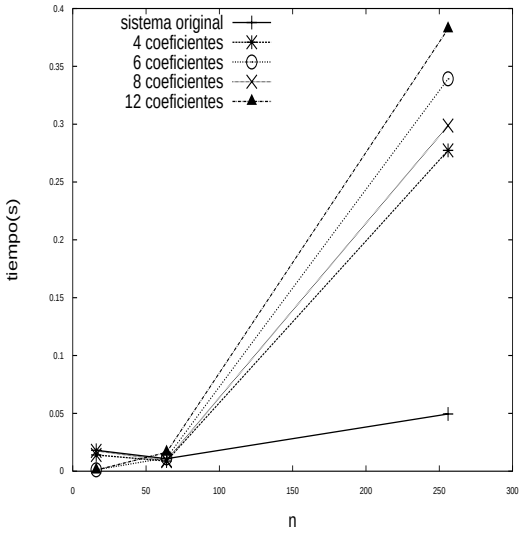


**Figura 88:** Caso 1 orden 2 Tiempo de procesamiento Symlets-BiCGStab con precondicionamiento

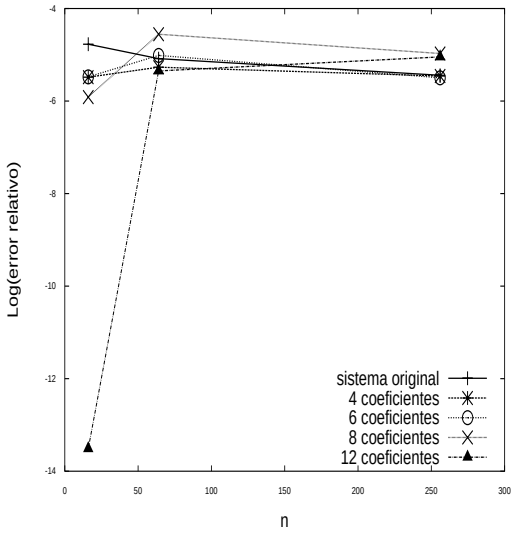


**Figura 89:** Caso 1 orden 2 Error relativo Symlets-BiCGStab con precondicionamiento

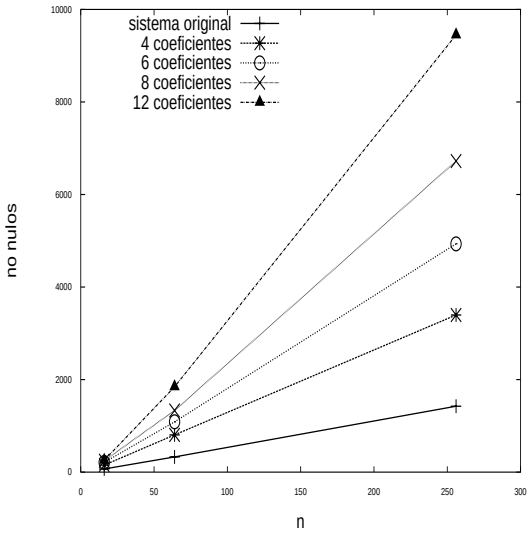


**Figura 90:** Caso 1 orden 2 Elementos no nulos Symlets-CGS con precondicionamiento

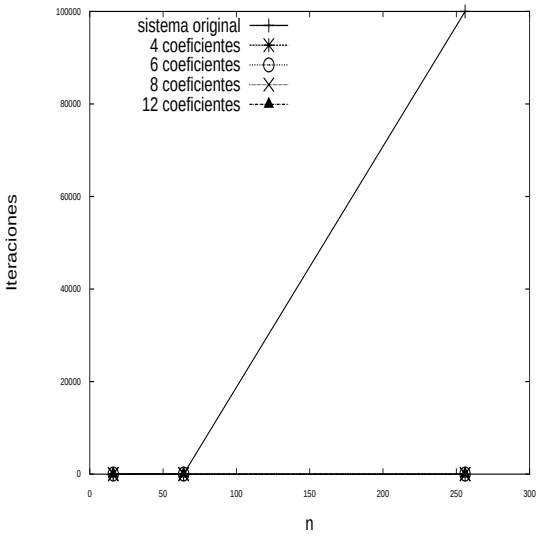


**Figura 91:** Caso 1 orden 2 Iteraciones Symlets-CGS con precondicionamiento

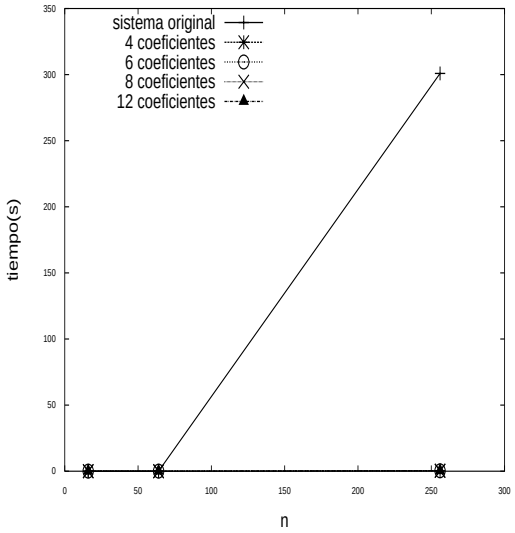


**Figura 92:** Caso 1 orden 2 Tiempo de procesamiento Symlets-CGS con precondicionamiento

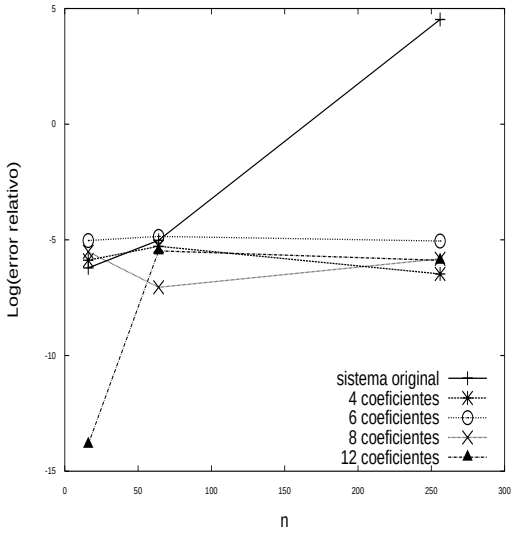


**Figura 93:** Caso 1 orden 2 Error relativo Symlets-CGS con precondicionamiento

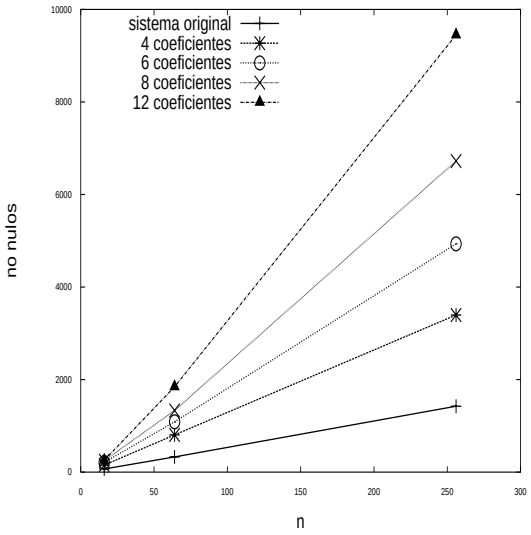


**Figura 94:** Caso 1 orden 2 Elementos no nulos Symlets-QMR con precondicionamiento

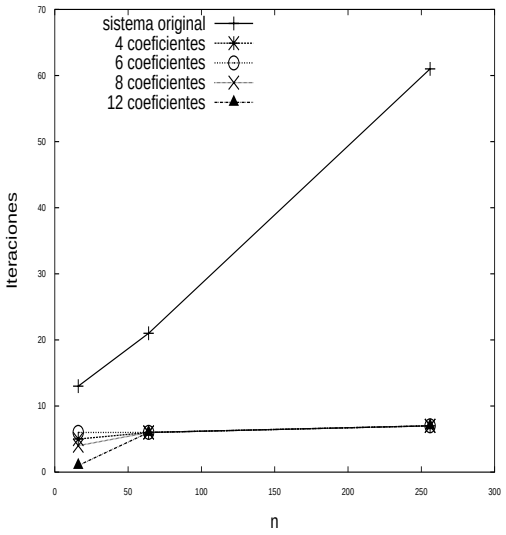


**Figura 95:** Caso 1 orden 2 Iteraciones Symlets-QMR con precondicionamiento

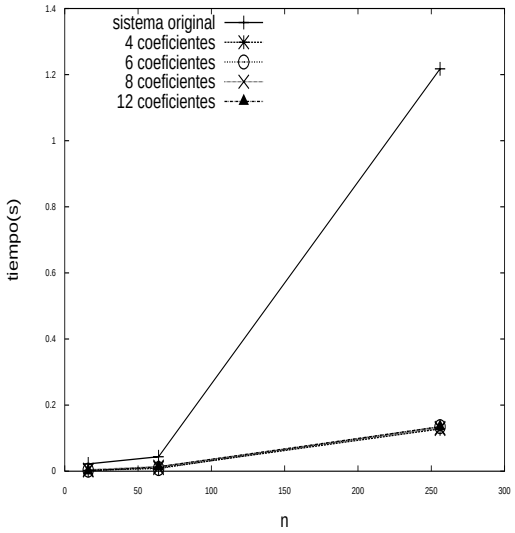


**Figura 96:** Caso 1 orden 2 Tiempo de procesamiento Symlets-QMR con precondicionamiento

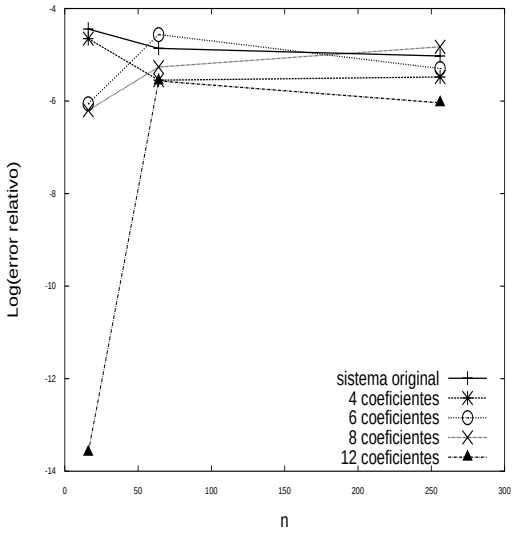


**Figura 97:** Caso 1 orden 2 Error relativo Symlets-QMR con precondicionamiento

En las figuras (82) a (97) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 12 coeficientes.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, para 12 coeficientes se presenta en la mayoría de los casos los menores errores relativos.

En las figuras (82) a (97) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en todos los casos (ver figuras 96 y 97).

Familia Biorthogonal

Los resultados obtenidos se observan en las figuras (98) a (113), que se muestran a continuación:

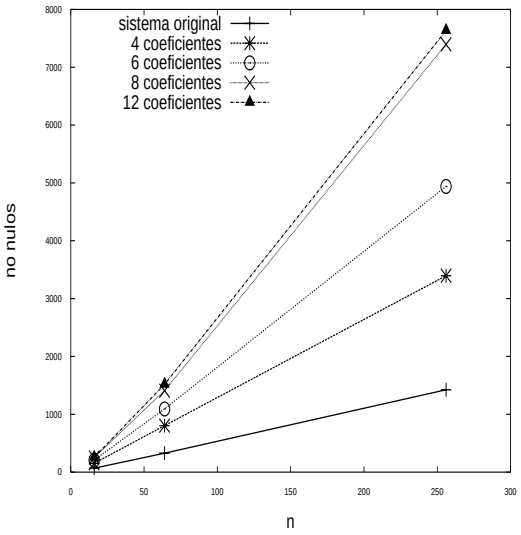


**Figura 98:** Caso 1 orden 2 Elementos no nulos Biorthogonal-BiCG con precondicionamiento

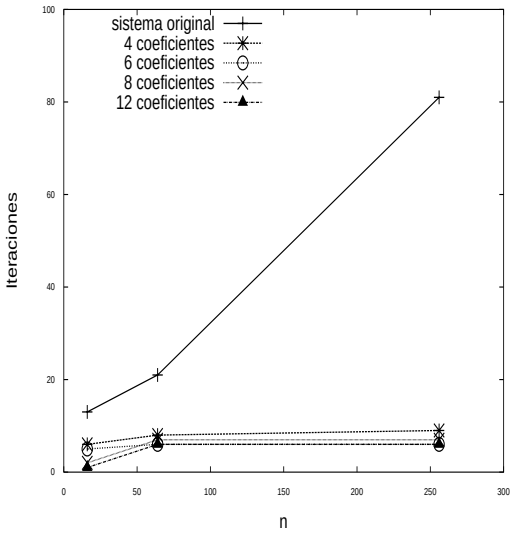


**Figura 99:** Caso 1 orden 2 Iteraciones Biorthogonal-BiCG con precondicionamiento

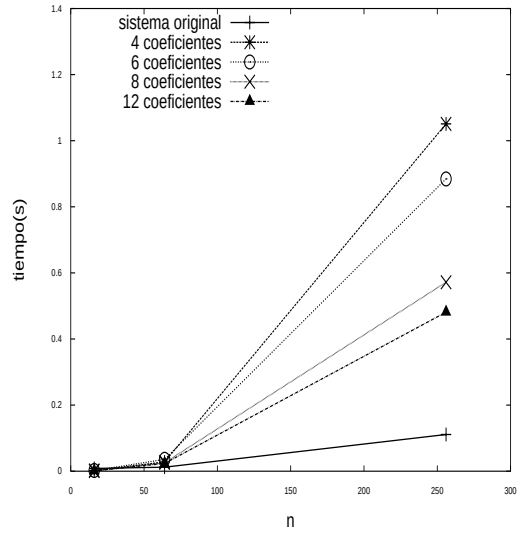


**Figura 100:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-BiCG con precondicionamiento

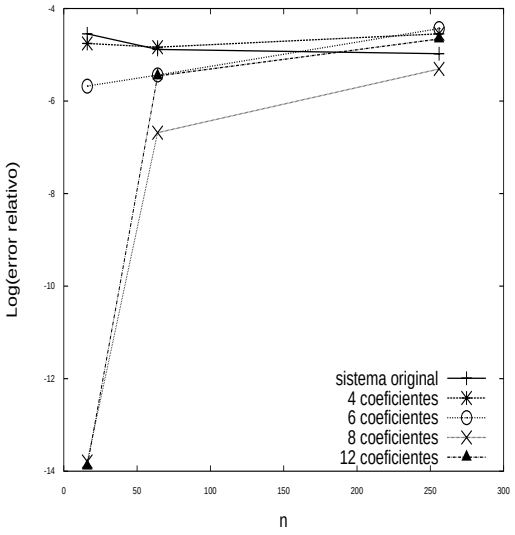


**Figura 101:** Caso 1 orden 2 Error relativo Biorthogonal-BiCG con precondicionamiento

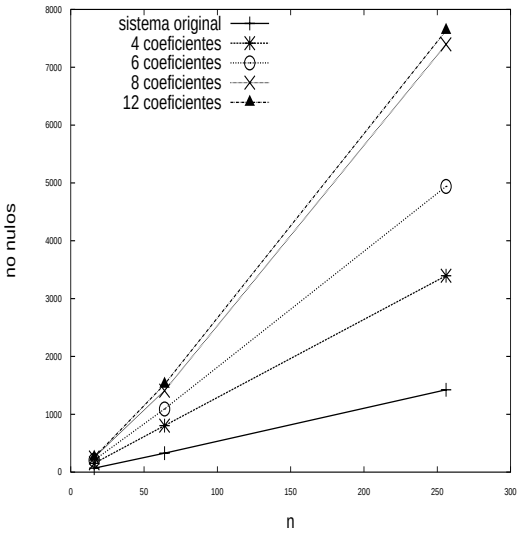


**Figura 102:** Caso 1 orden 2 Elementos no nulos Biorthogonal-BiCGStab con precondicionamiento

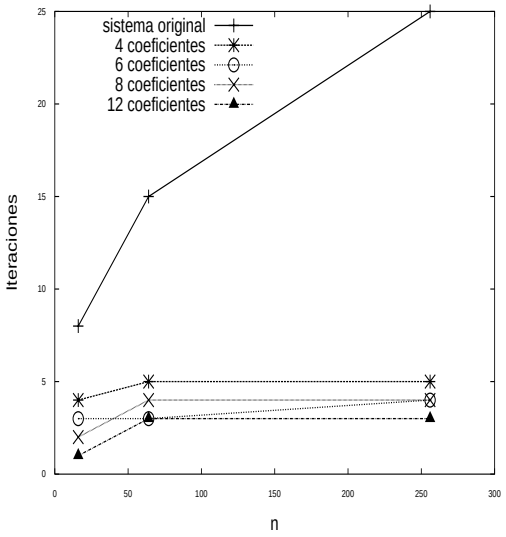


**Figura 103:** Caso 1 orden 2 Iteraciones Biorthogonal-BiCGStab con precondicionamiento

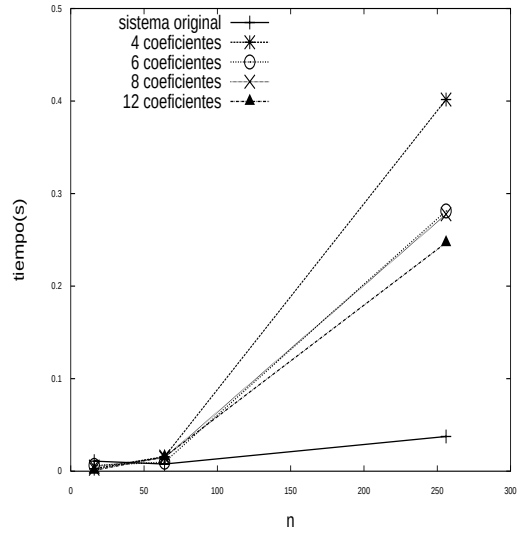


**Figura 104:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-BiCGStab con precondicionamiento

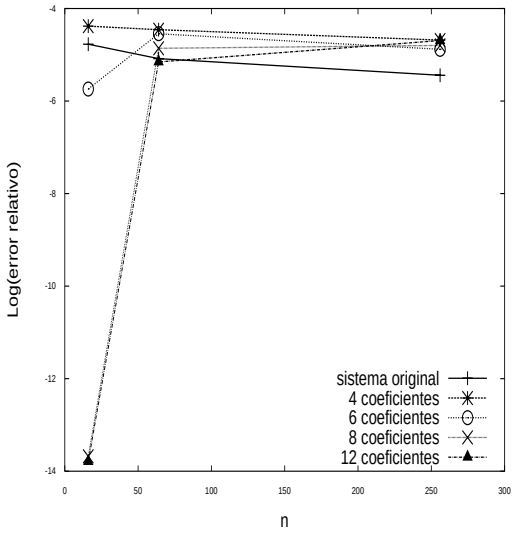


**Figura 105:** Caso 1 orden 2 Error relativo Biorthogonal-BiCGStab con precondicionamiento

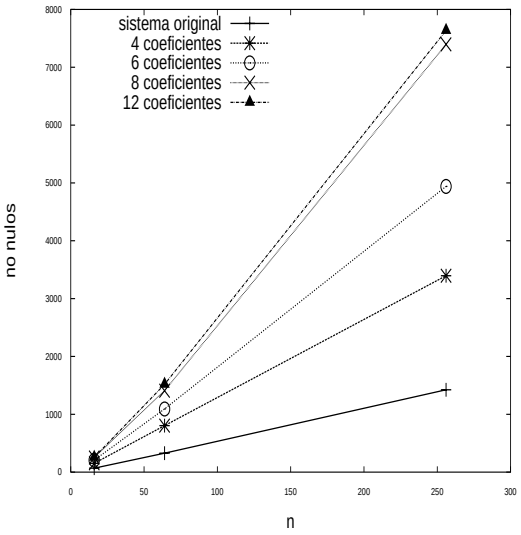


**Figura 106:** Caso 1 orden 2 Elementos no nulos Biorthogonal-CGS con precondicionamiento

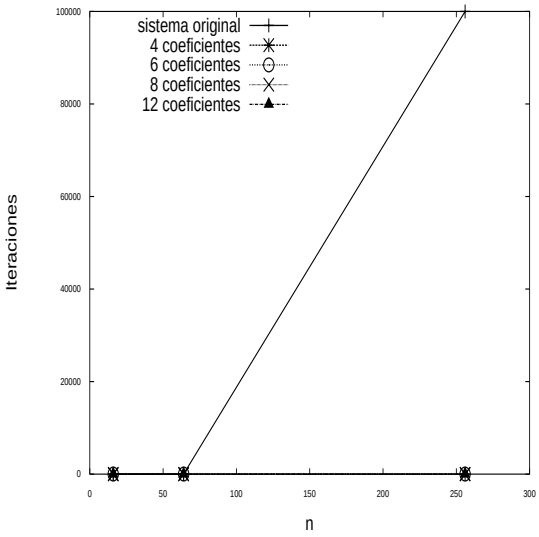


**Figura 107:** Caso 1 orden 2 Iteraciones Biorthogonal-CGS con precondicionamiento

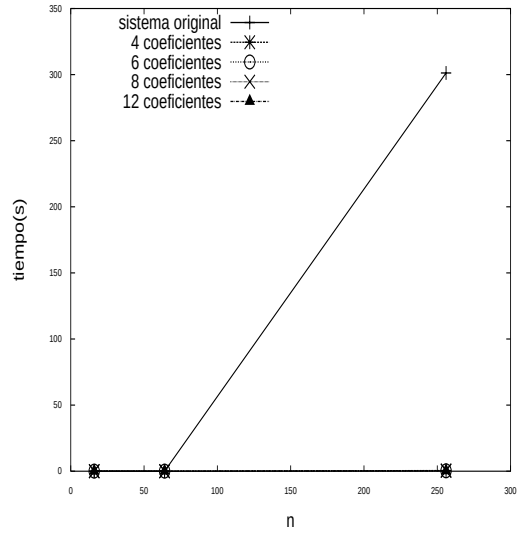


**Figura 108:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-CGS con precondicionamiento

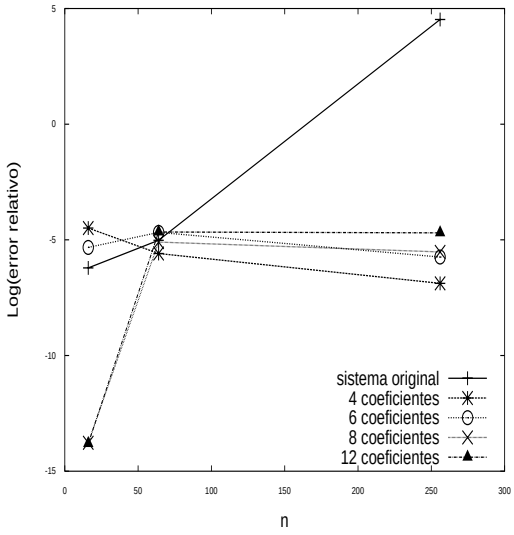


**Figura 109:** Caso 1 orden 2 Error relativo Biorthogonal-CGS con precondicionamiento

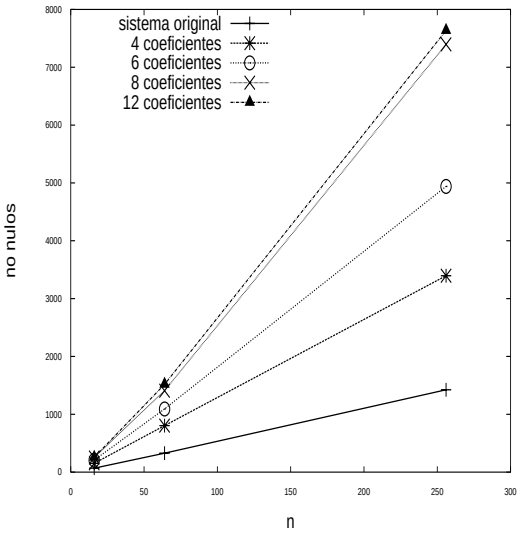


**Figura 110:** Caso 1 orden 2 Elementos no nulos Biorthogonal-QMR con precondicionamiento

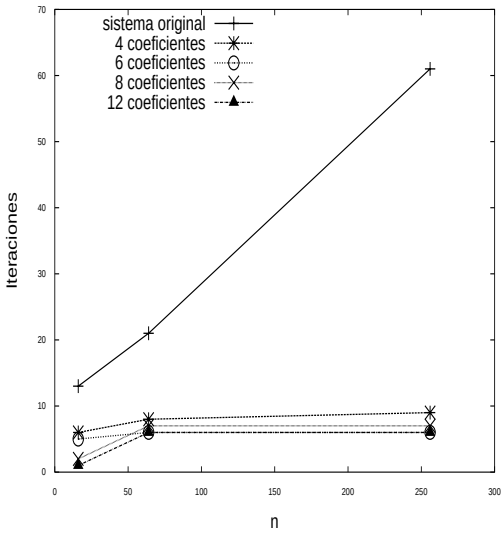


**Figura 111:** Caso 1 orden 2 Iteraciones Biorthogonal-QMR con precondicionamiento

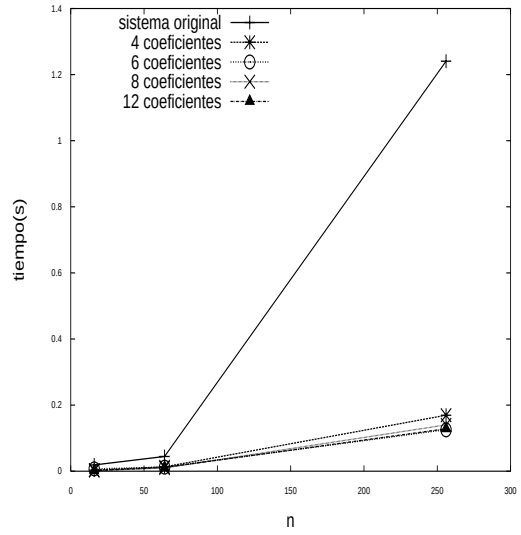


**Figura 112:** Caso 1 orden 2 Tiempo de procesamiento Biorthogonal-QMR con precondicionamiento

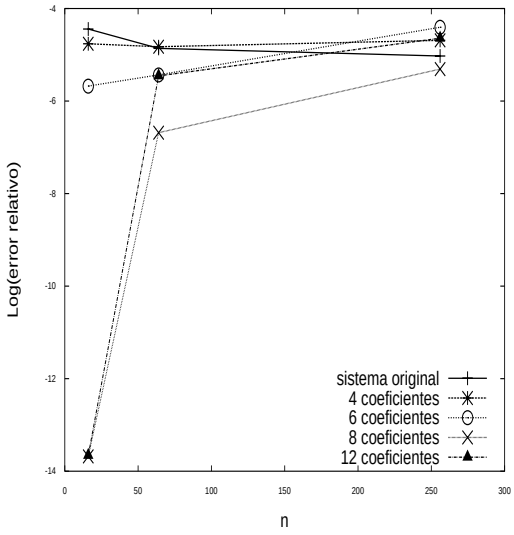


**Figura 113:** Caso 1 orden 2 Error relativo Biorthogonal-QMR con precondicionamiento

En las figuras (98) a (113) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 6 coeficientes, teniéndose tiempos bastantes cercanos al menor con 12 y 8 coeficientes.

- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, en los métodos BiCG y QMR con 8 coeficientes se presentan los menores errores relativos.

En las figuras (98) a (113) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 8 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en todos los casos (ver figuras 112 y 113).

Familia Reverse Biorthogonal

Los resultados obtenidos se observan en las figuras (114) a (129), que se muestran a continuación:

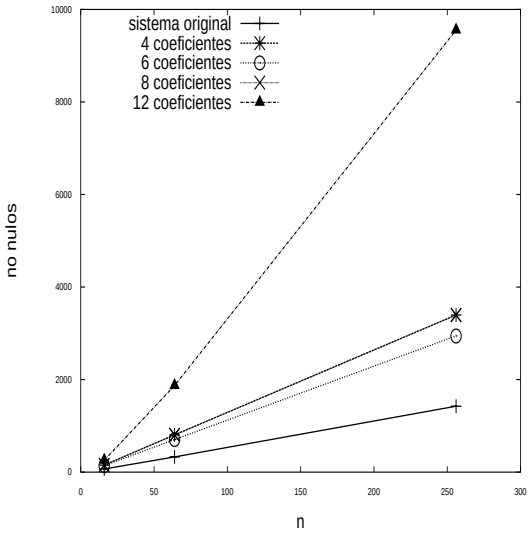


**Figura 114:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-BiCG con precondicionamiento

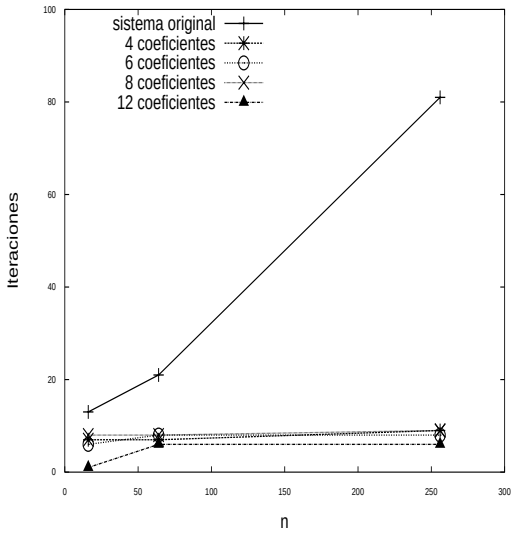


**Figura 115:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-BiCG con precondicionamiento

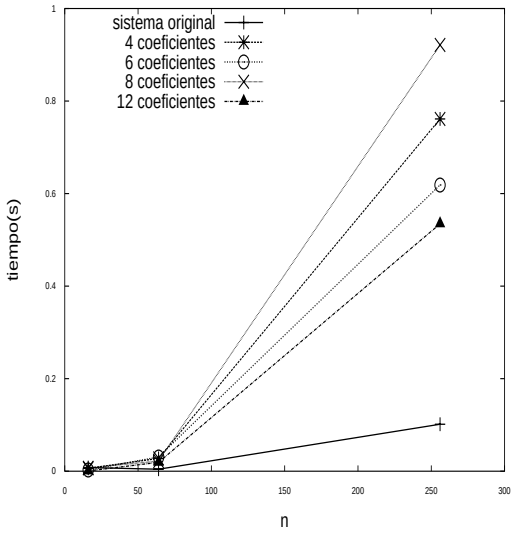


**Figura 116:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCG con precondicionamiento

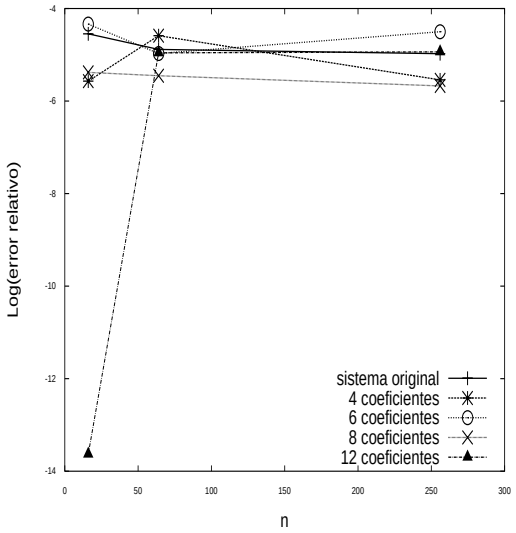


**Figura 117:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-BiCG con precondicionamiento

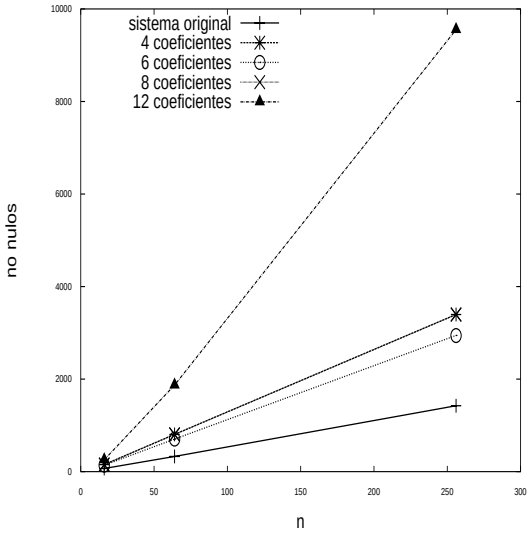


**Figura 118:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-BiCGStab con precondicionamiento

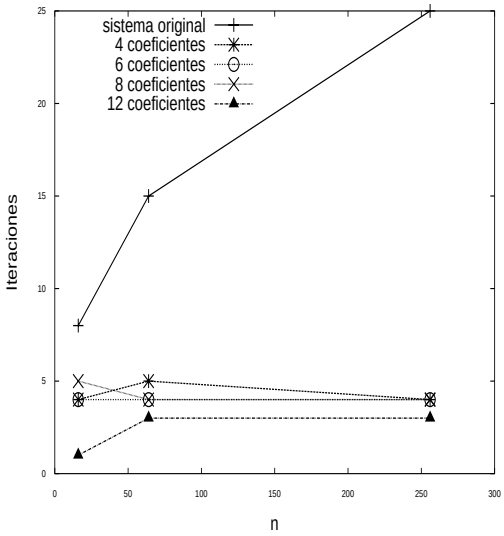


**Figura 119:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-BiCGStab con precondicionamiento

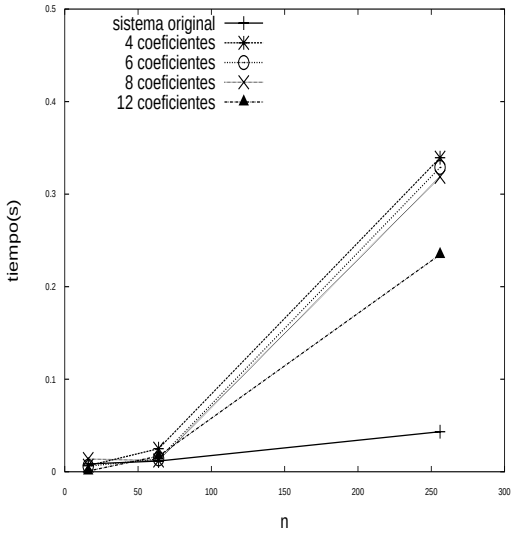


**Figura 120:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCGStab con precondicionamiento

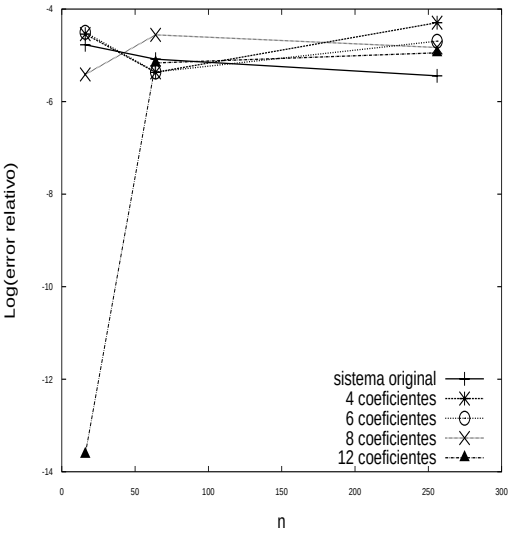


**Figura 121:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-BiCGStab con precondicionamiento

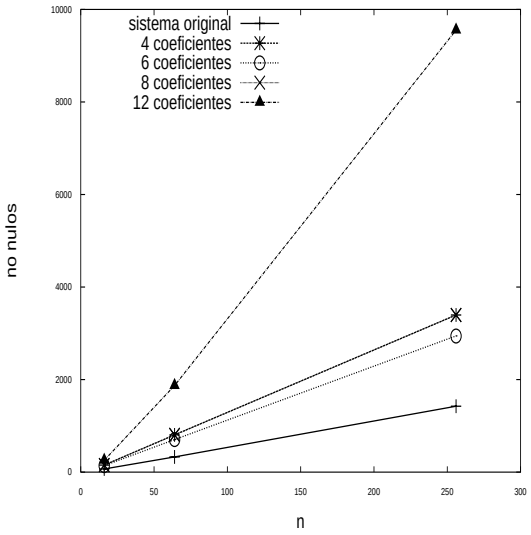


**Figura 122:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-CGS con precondicionamiento

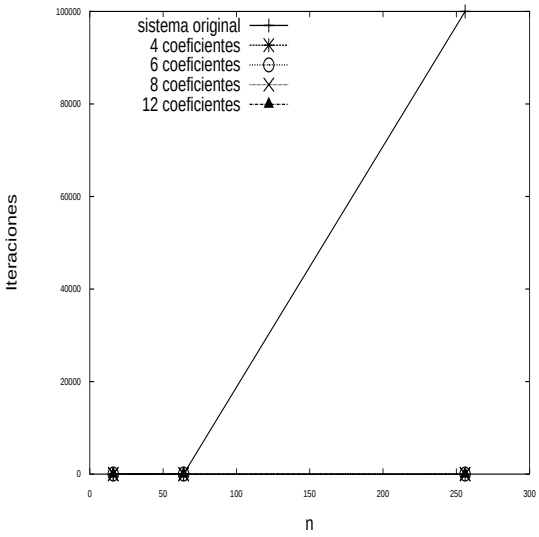


**Figura 123:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-CGS con precondicionamiento

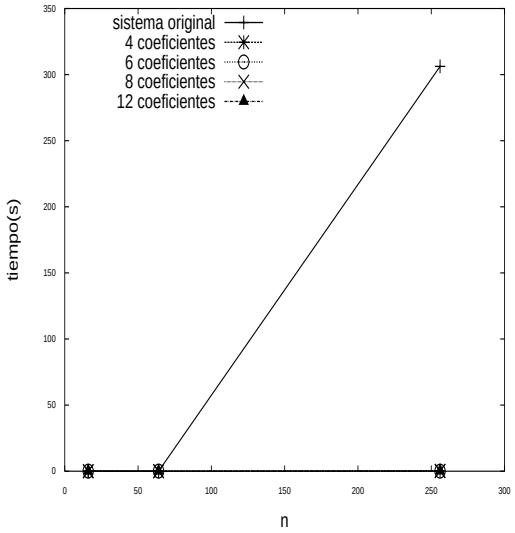


**Figura 124:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-CGS con precondicionamiento

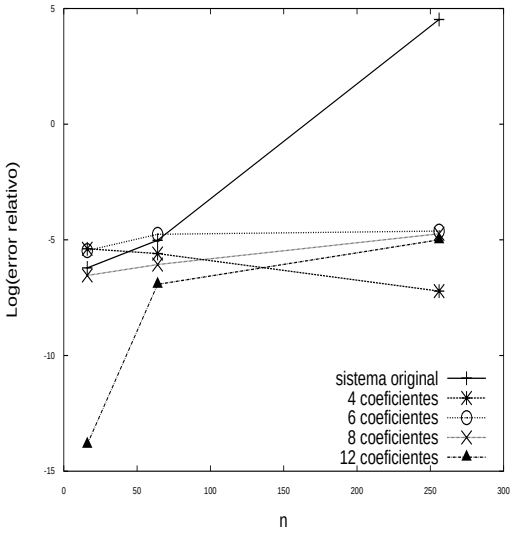


**Figura 125:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-CGS con precondicionamiento

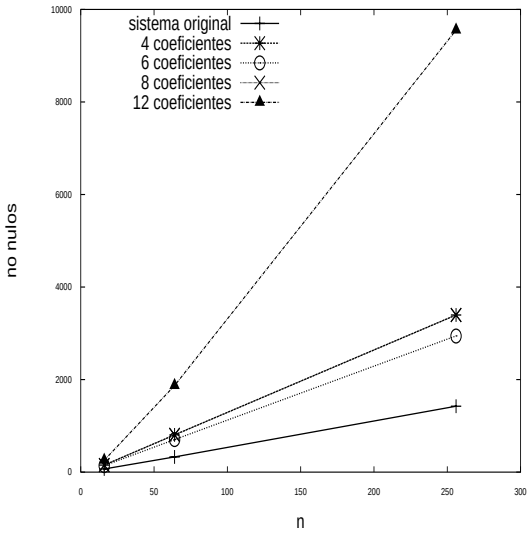


**Figura 126:** Caso 1 orden 2 Elementos no nulos Reverse Biorthogonal-QMR con precondicionamiento

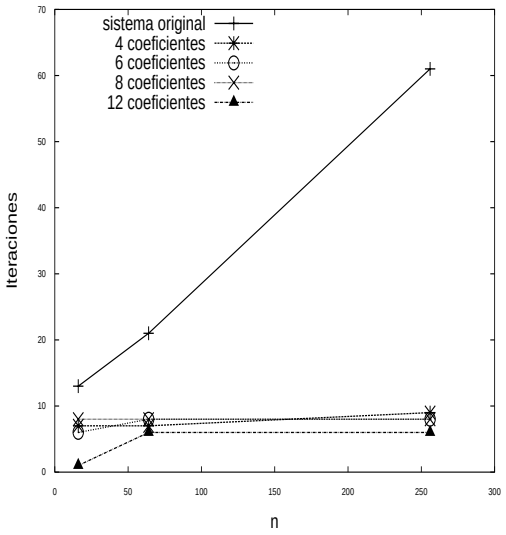


**Figura 127:** Caso 1 orden 2 Iteraciones Reverse Biorthogonal-QMR con precondicionamiento

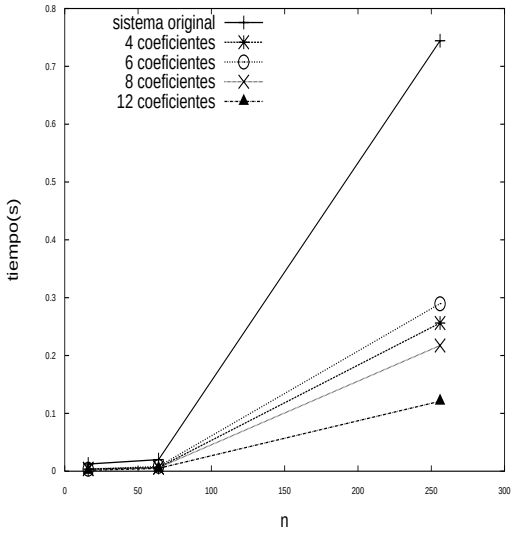


**Figura 128:** Caso 1 orden 2 Tiempo de procesamiento Reverse Biorthogonal-QMR con precondicionamiento

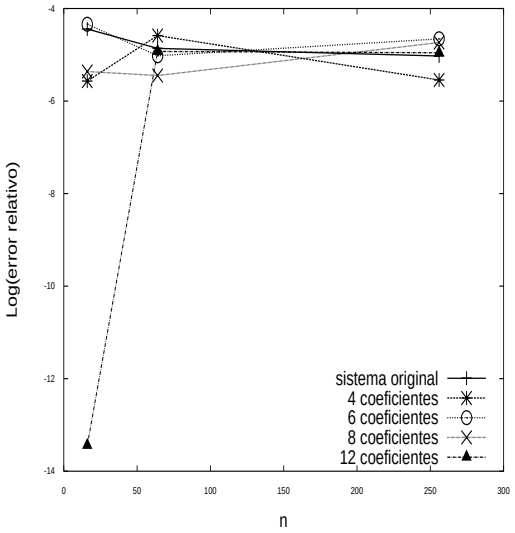


**Figura 129:** Caso 1 orden 2 Error relativo Reverse Biorthogonal-QMR con precondicionamiento

En las figuras (114) a (129) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose la menor cantidad de iteraciones en el caso de 12 coeficientes. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 12 coeficientes.

- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, en el método QMR con 12 coeficientes se presenta en la mayoría de los casos los menores errores relativos.

En las figuras (114) a (129) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en la mayoría de los casos ( ver figuras 128 y 129).

**Caso 2, sin precondicionamiento**

Familia Daubechies

Los resultados obtenidos se observan en las figuras (130) a (145), que se muestran a continuación:

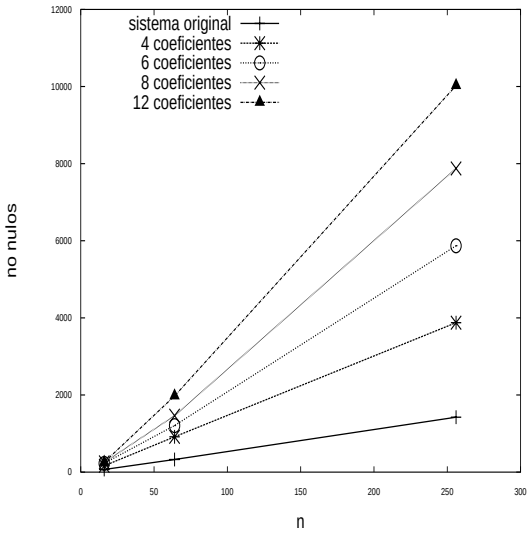


**Figura 130:** Caso 2 orden 2 Elementos no nulos Daubechies-BiCG sin precondicionamiento

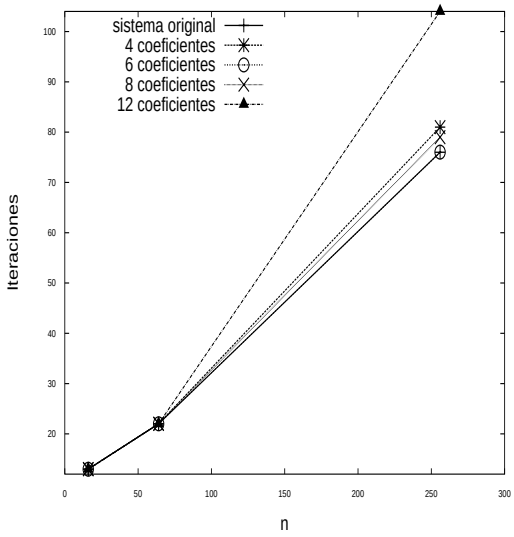


**Figura 131:** Caso 2 orden 2 Iteraciones Daubechies-BiCG sin precondicionamiento

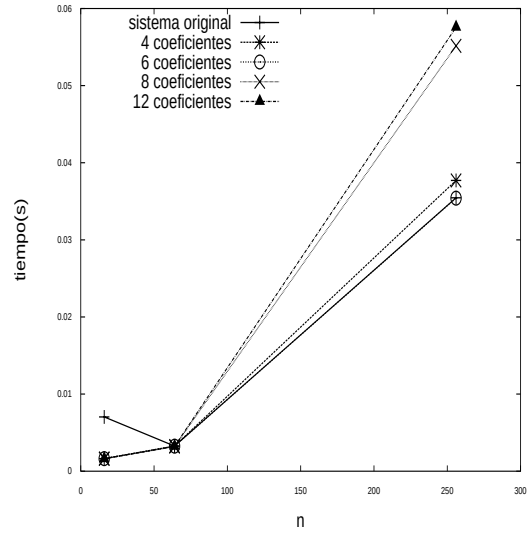


**Figura 132:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-BiCG sin precondicionamiento

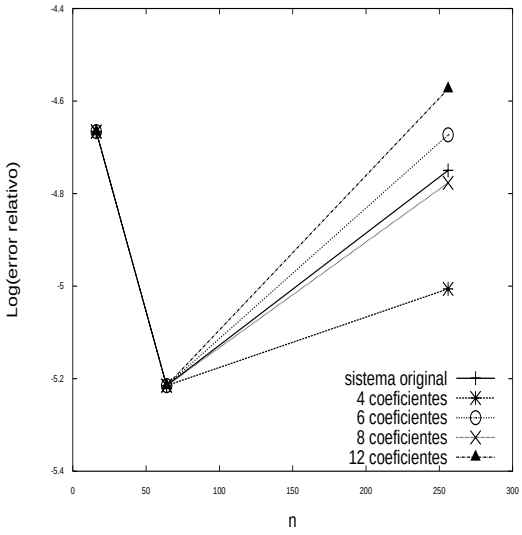


**Figura 133:** Caso 2 orden 2 Error relativo Daubechies-BiCG sin precondicionamiento

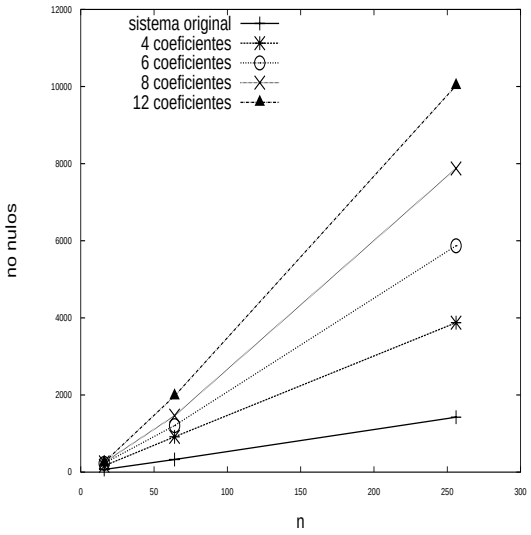


**Figura 134:** Caso 2 orden 2 Elementos no nulos Daubechies-BiCGStab sin precondicionamiento

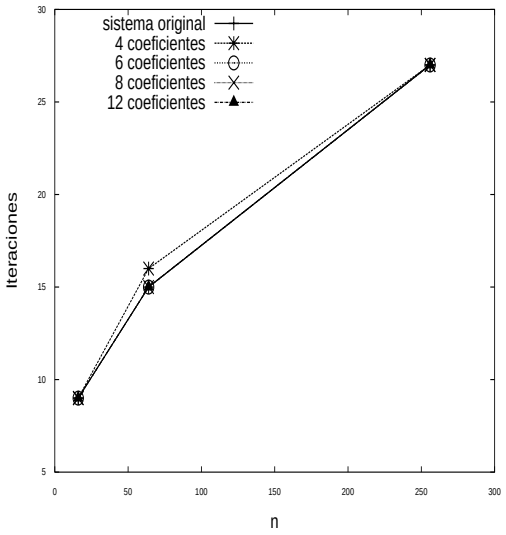


**Figura 135:** Caso 2 orden 2 Iteraciones Daubechies-BiCGStab sin precondicionamiento

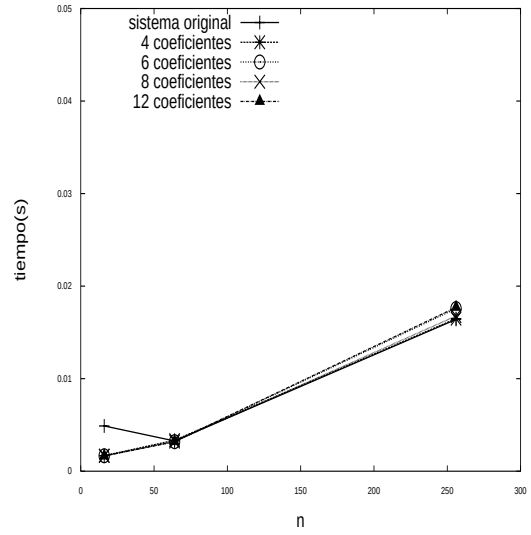


**Figura 136:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-BiCGStab sin precondicionamiento

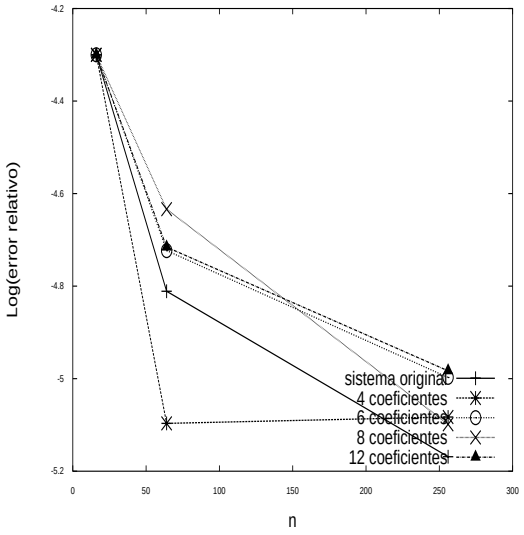


**Figura 137:** Caso 2 orden 2 Error relativo Daubechies-BiCGStab sin precondicionamiento

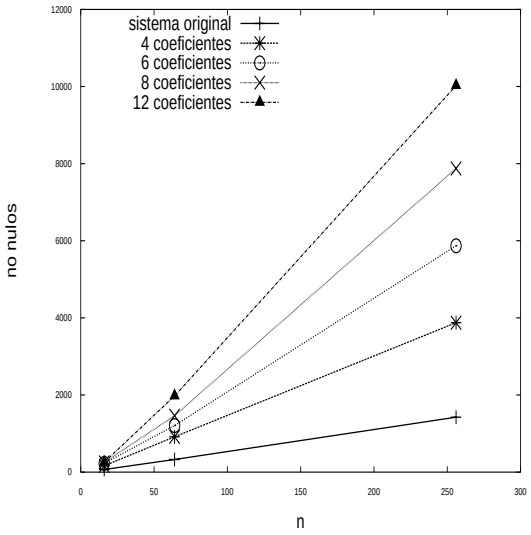


**Figura 138:** Caso 2 orden 2 Elementos no nulos Daubechies-CGS sin precondicionamiento

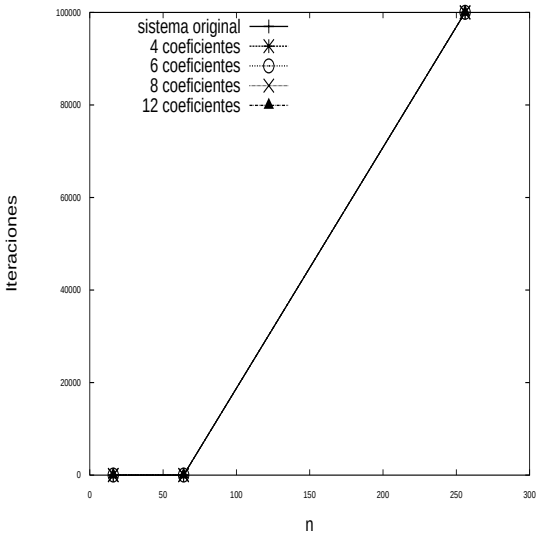


**Figura 139:** Caso 2 orden 2 Iteraciones Daubechies-CGS sin precondicionamiento

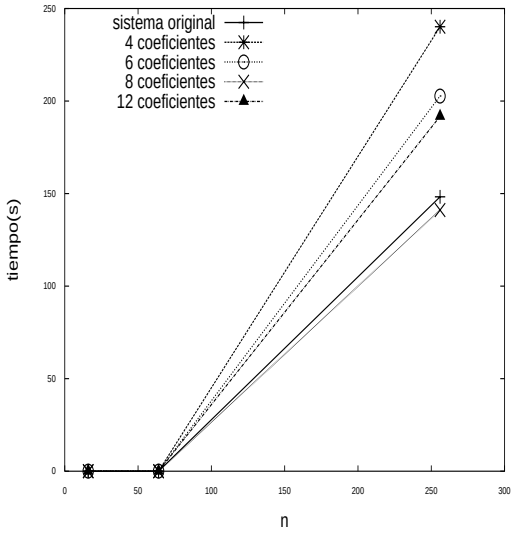


**Figura 140:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-CGS sin precondicionamiento

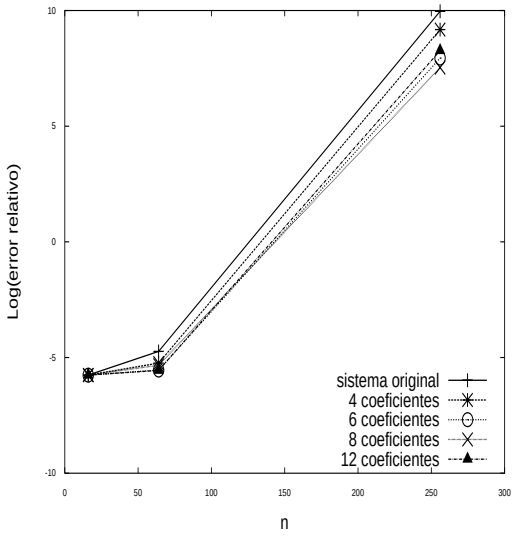


**Figura 141:** Caso 2 orden 2 Error relativo Daubechies-CGS sin precondicionamiento

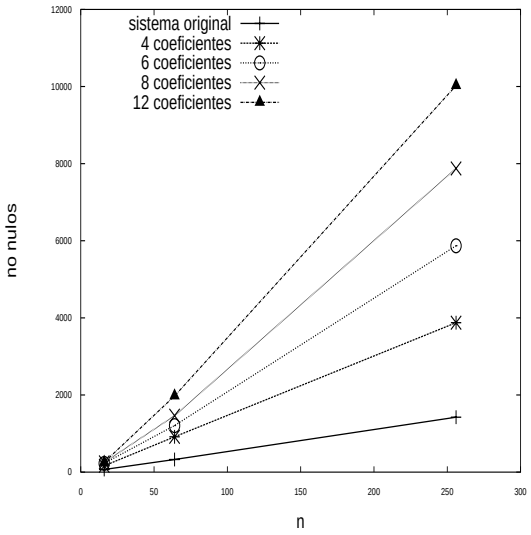


**Figura 142:** Caso 2 orden 2 Elementos no nulos Daubechies-QMR sin precondicionamiento

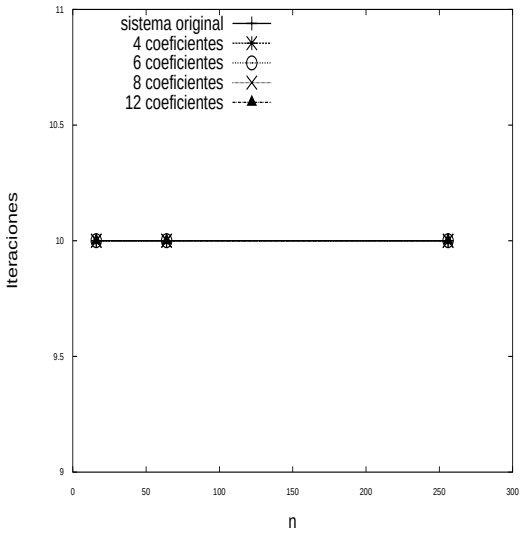


**Figura 143:** Caso 2 orden 2 Iteraciones Daubechies-QMR sin precondicionamiento

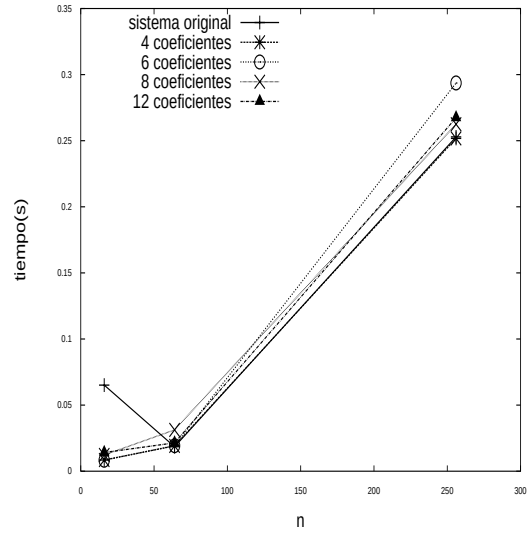


**Figura 144:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-QMR sin precondicionamiento

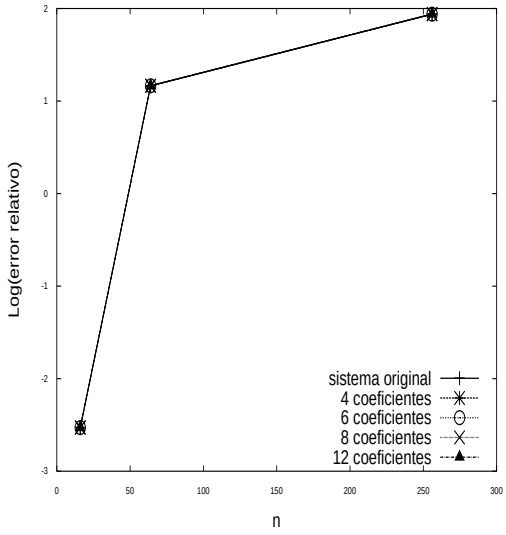


**Figura 145:** Caso 2 orden 2 Error relativo Daubechies-QMR sin precondicionamiento

En las figuras (130) a (145) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en los métodos BiCG y BiCGStab; no se presenta tal variación en los métodos CGS y QMR. En los métodos BiCG y BiCGStab la menor cantidad de iteraciones se obtiene en el caso de 6 coeficientes.
- Los tiempos de procesamiento varían dependiendo de las distintas combinaciones entre los métodos iterativos y la cantidad de coeficientes de la familia *wavelet*. En el método BiCG, el menor tiempo se obtiene con 6 coeficientes; en el método BiCGStab el menor tiempo se obtiene con 4 coeficientes; en el método CGS el menor tiempo se obtiene con 8 coeficientes; en el método QMR el menor tiempo se obtiene en el sistema original.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor error se logra con 4 coeficientes; para el método BiCGStab, el menor error se logra con 4 coeficientes; para el método CGS el menor error se logra con 8 coeficientes; para el método QMR, los errores son fundamentalmente iguales en todos los casos.

En las figuras (130) a (145) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método BiCGStab con 4 coeficientes, en virtud que presenta la menor cantidad de tiempo de procesamiento y un error relativo que tiende a disminuir con el aumento del tamaño de la matriz de coeficientes del sistema (ver figuras 136 y 137).

Familia Symlets

Los resultados obtenidos se observan en las figuras (146) a (161), que se muestran a continuación:

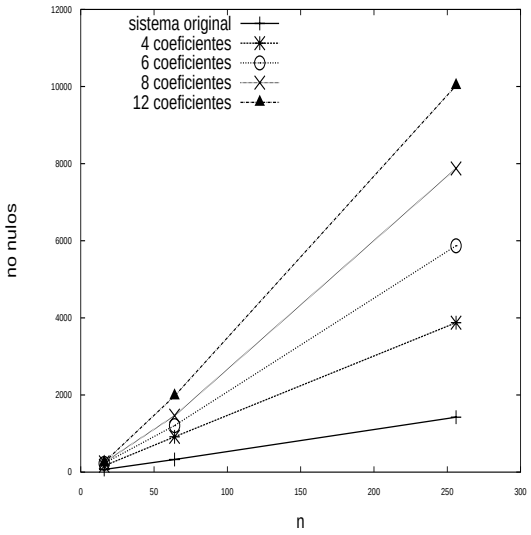


**Figura 146:** Caso 2 orden 2 Elementos no nulos Symlets-BiCG sin precondicionamiento

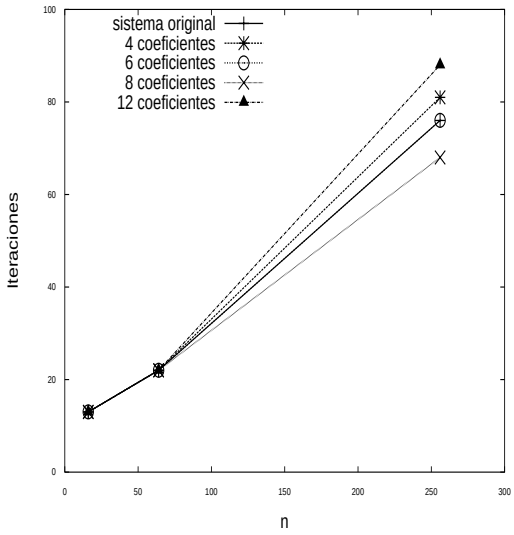


**Figura 147:** Caso 2 orden 2 Iteraciones Symlets-BiCG sin precondicionamiento

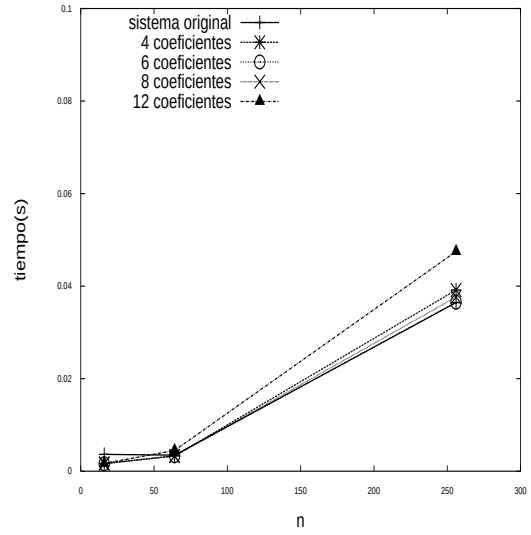


**Figura 148:** Caso 2 orden 2 Tiempo de procesamiento Symlets-BiCG sin precondicionamiento

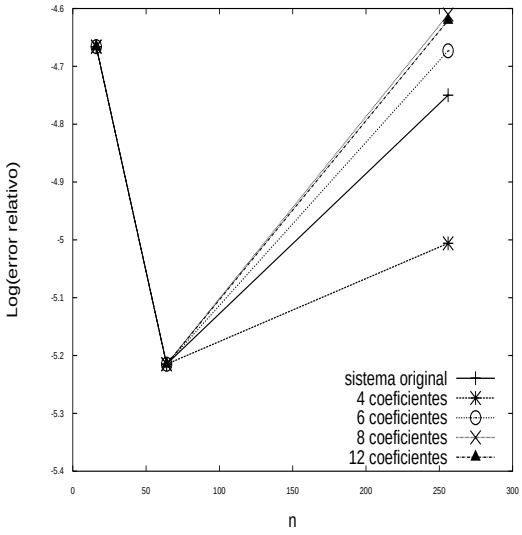


**Figura 149:** Caso 2 orden 2 Error relativo Symlets-BiCG sin precondicionamiento

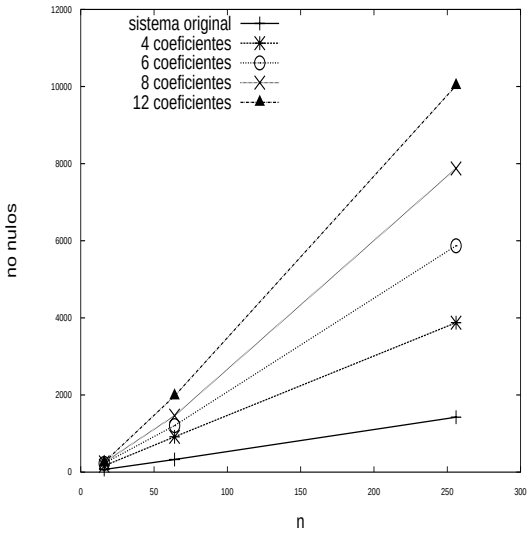


**Figura 150:** Caso 2 orden 2 Elementos no nulos Symlets-BiCGStab sin precondicionamiento

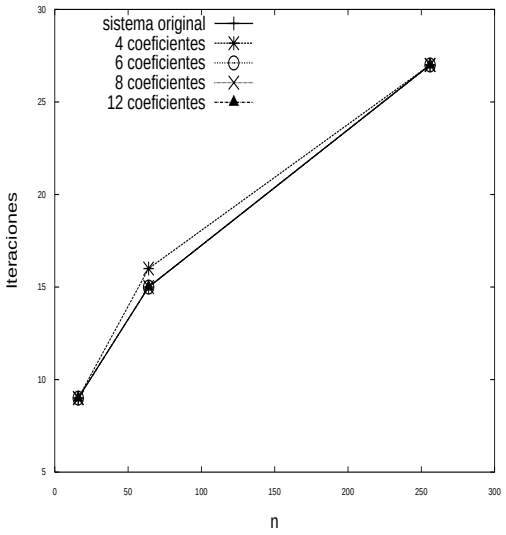


**Figura 151:** Caso 2 orden 2 Iteraciones Symlets-BiCGStab sin precondicionamiento

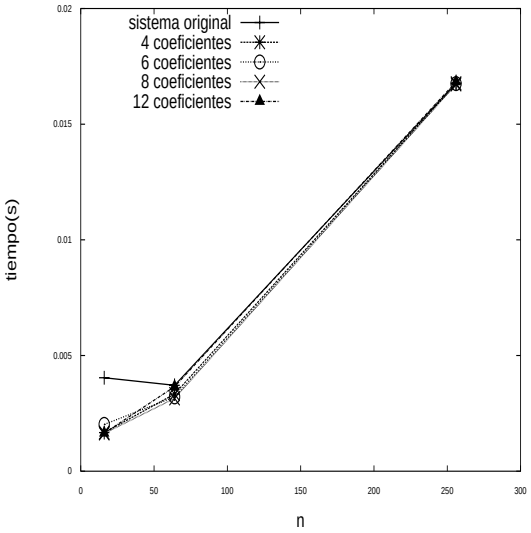


**Figura 152:** Caso 2 orden 2 Tiempo de procesamiento Symlets-BiCGStab sin precondicionamiento

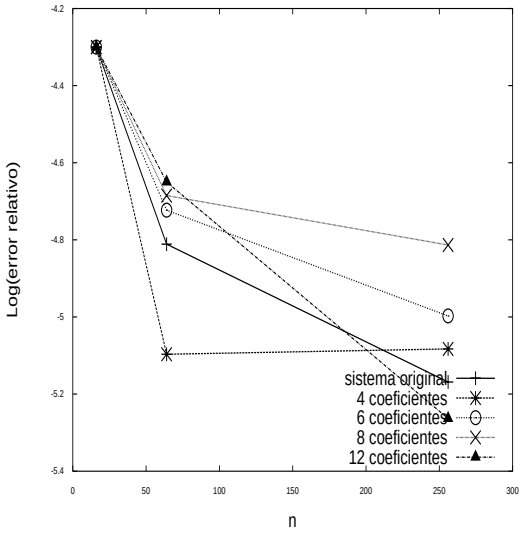


**Figura 153:** Caso 2 orden 2 Error relativo Symlets-BiCGStab sin precondicionamiento

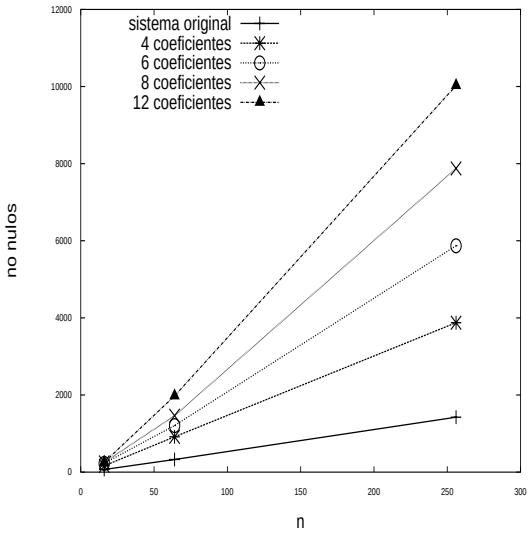


**Figura 154:** Caso 2 orden 2 Elementos no nulos Symlets-CGS sin precondicionamiento

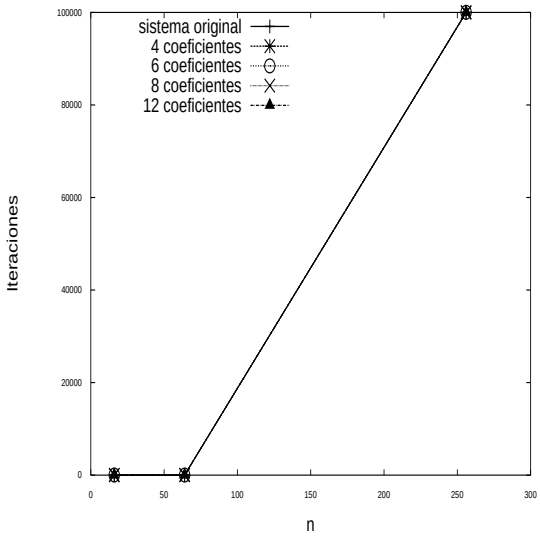


**Figura 155:** Caso 2 orden 2 Iteraciones Symlets-CGS sin precondicionamiento

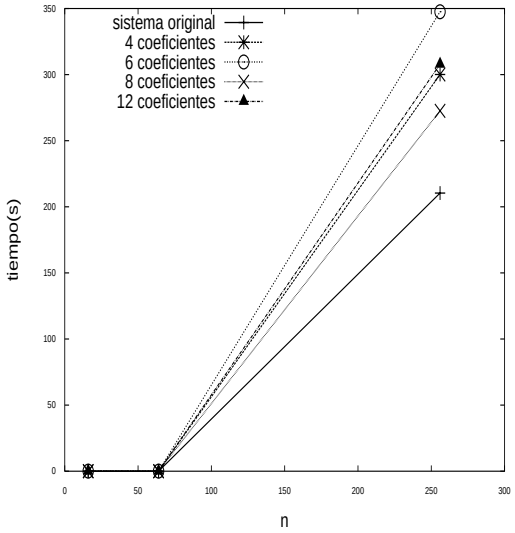


**Figura 156:** Caso 2 orden 2 Tiempo de procesamiento Symlets-CGS sin precondicionamiento

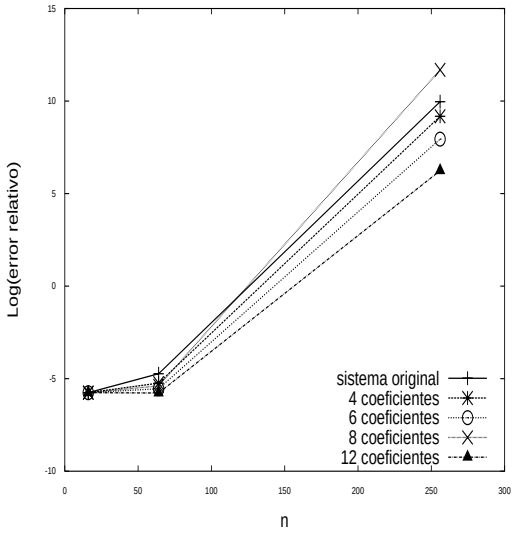


**Figura 157:** Caso 2 orden 2 Error relativo Symlets-CGS sin precondicionamiento

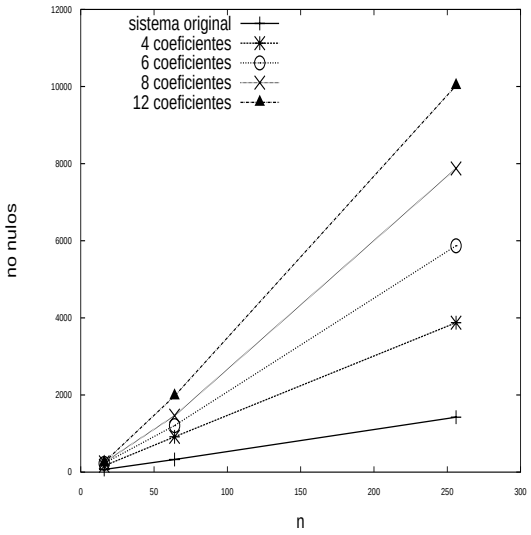


**Figura 158:** Caso 2 orden 2 Elementos no nulos Symlets-QMR sin precondicionamiento

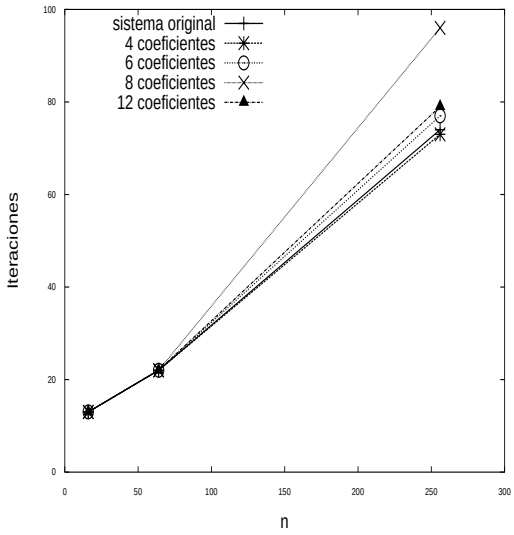


**Figura 159:** Caso 2 orden 2 Iteraciones Symlets-QMR sin precondicionamiento

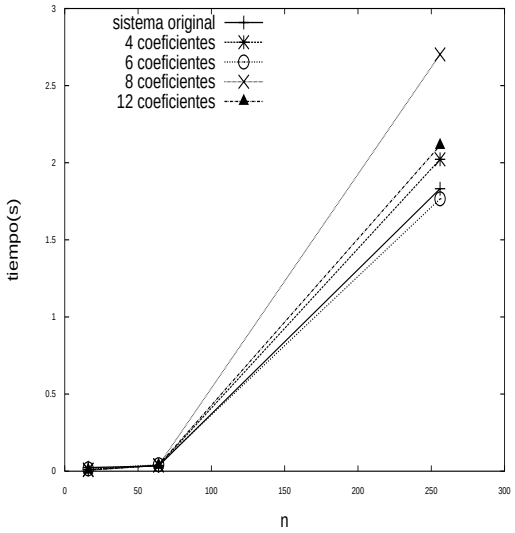


**Figura 160:** Caso 2 orden 2 Tiempo de procesamiento Symlets-QMR sin precondicionamiento

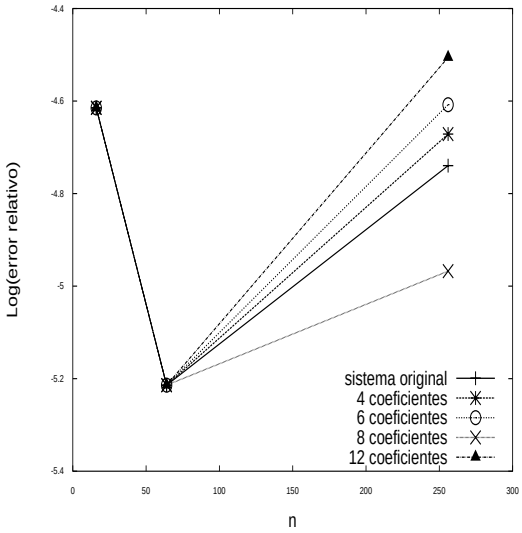


**Figura 161:** Caso 2 orden 2 Error relativo Symlets-QMR sin precondicionamiento

En las figuras (146) a (161) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en el método BiCG, se tiene la menor cantidad de iteraciones en el caso de 8 coeficientes; en el método BiCGStab, la cantidad de iteraciones es muy similar en todos los casos; en el método CGS la cantidad de iteraciones es prácticamente la misma en todos los casos; en el método QMR, se tiene la menor cantidad de iteraciones en el caso de 4 coeficientes.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor tiempo se logra con 6 coeficientes; para el método BiCGStab, los tiempos en todos los casos son muy similares; para el método CGS, el menor tiempo se logra en el sistema original; para el método QMR, el menor tiempo se logra con 6 coeficientes.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para el método BiCG, el menor error se logra con 4 coeficientes; para el método BiCGStab, el mejor comportamiento del error se logra con 4 coeficientes; para el método CGS, el el mejor comportamiento del error se logra con 12 coeficientes; para el método QMR, el menor error se logra con 8 coeficientes.

En las figuras (146) a (161) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método BiCGStab con 4 coeficientes, en virtud que presenta la menor cantidad de tiempo de procesamiento y un error relativo que tiende a disminuir con el aumento del tamaño de la matriz de coeficientes del sistema (ver figuras 152 y 153).

Familia Biorthogonal

Los resultados obtenidos se observan en las figuras (162) a (177), que se muestran a continuación:

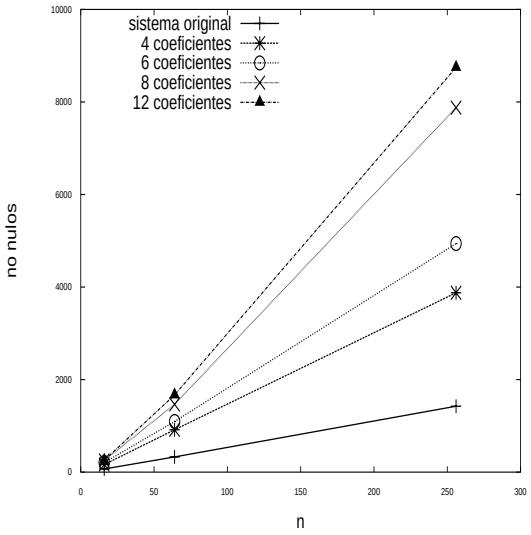


**Figura 162:** Caso 2 orden 2 Elementos no nulos Biorthogonal-BiCG sin precondicionamiento

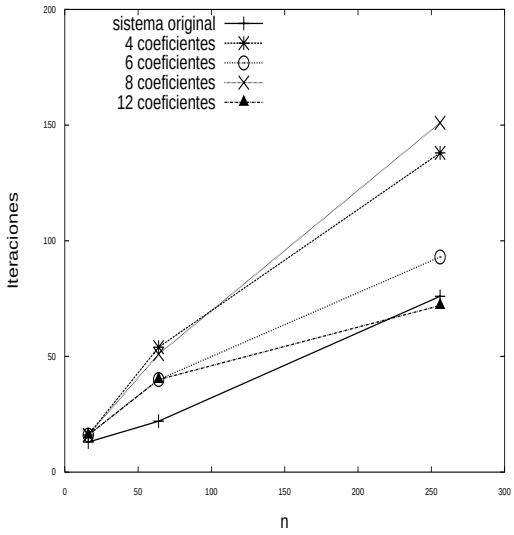


**Figura 163:** Caso 2 orden 2 Iteraciones Biorthogonal-BiCG sin precondicionamiento

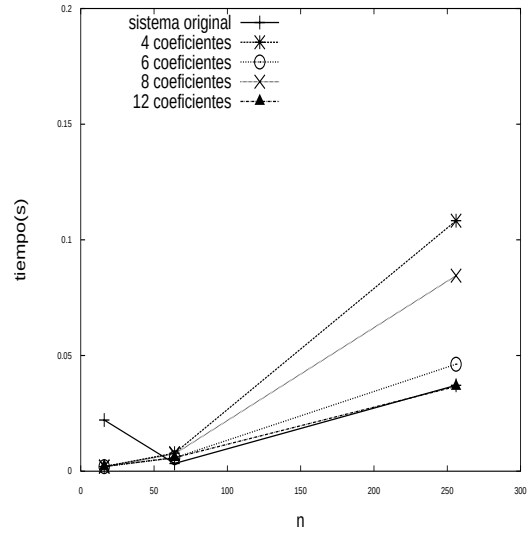


**Figura 164:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-BiCG sin precondicionamiento

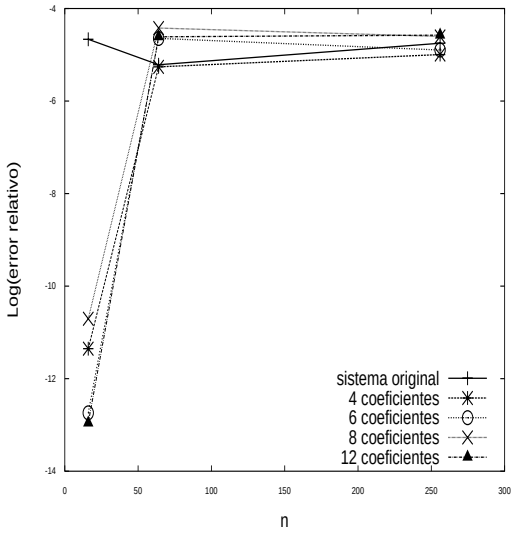


**Figura 165:** Caso 2 orden 2 Error relativo Biorthogonal-BiCG sin precondicionamiento

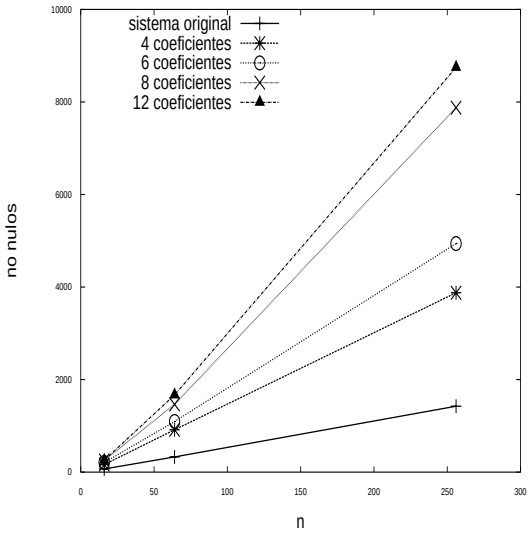


**Figura 166:** Caso 2 orden 2 Elementos no nulos Biorthogonal-BiCGStab sin precondicionamiento

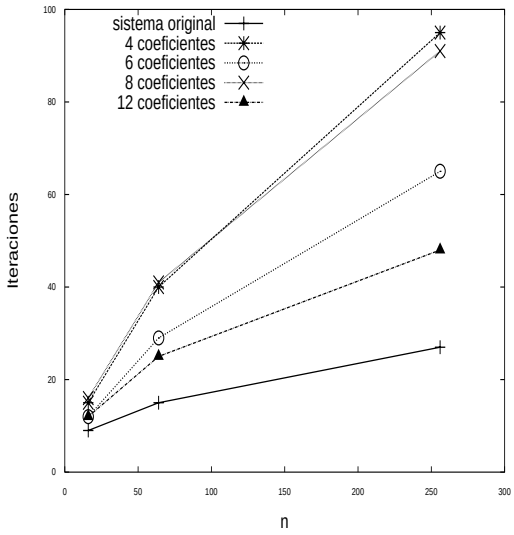


**Figura 167:** Caso 2 orden 2 Iteraciones Biorthogonal-BiCGStab sin precondicionamiento

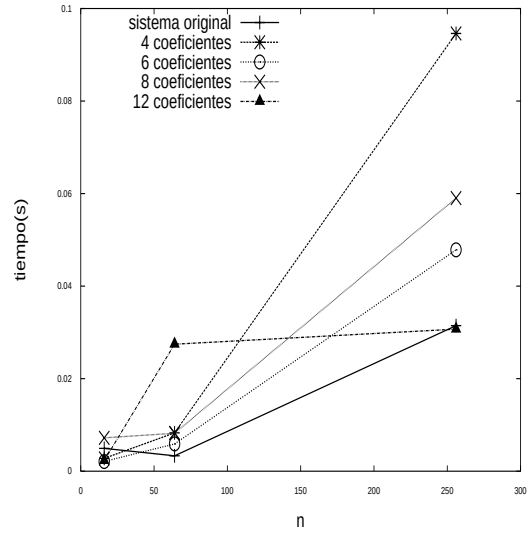


**Figura 168:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-BiCGStab sin precondicionamiento

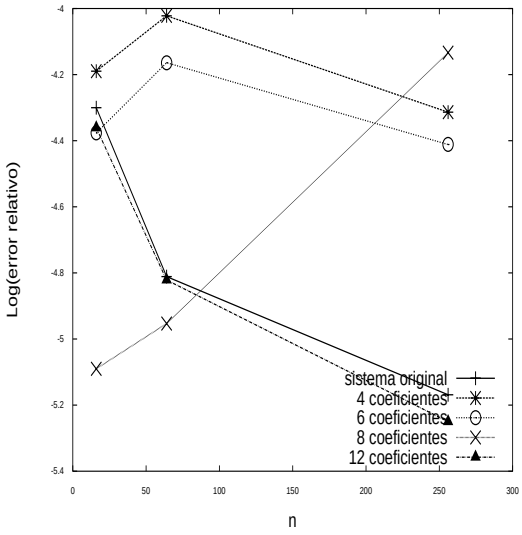


**Figura 169:** Caso 2 orden 2 Error relativo Biorthogonal-BiCGStab sin precondicionamiento

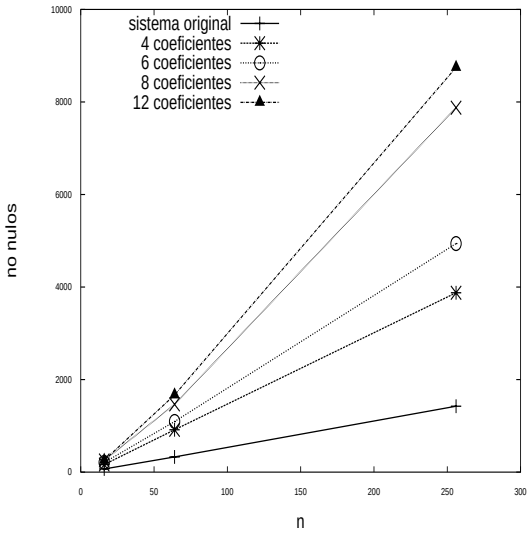


**Figura 170:** Caso 2 orden 2 Elementos no nulos Biorthogonal-CGS sin precondicionamiento

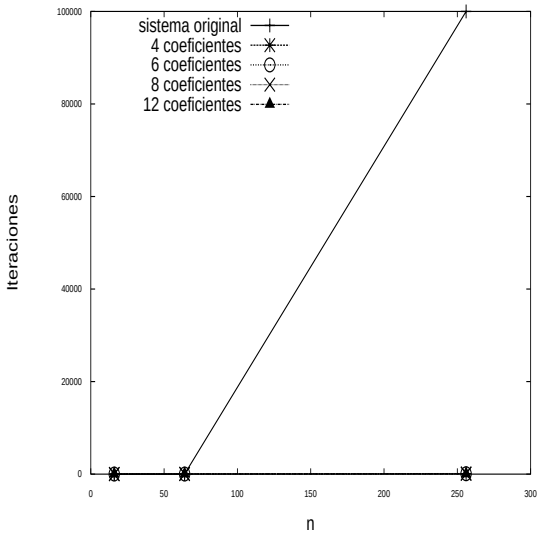


**Figura 171:** Caso 2 orden 2 Iteraciones Biorthogonal-CGS sin precondicionamiento

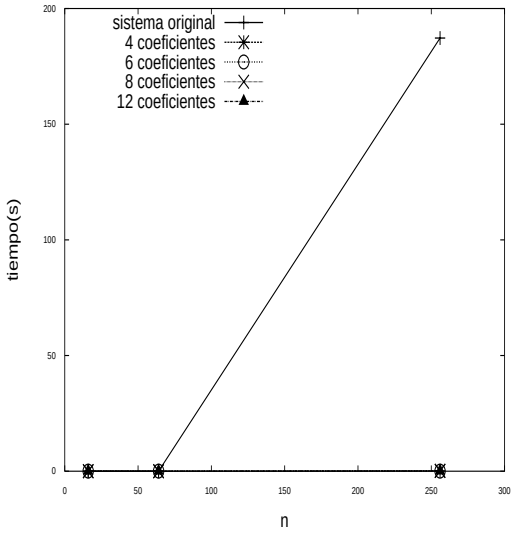


**Figura 172:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-CGS sin precondicionamiento

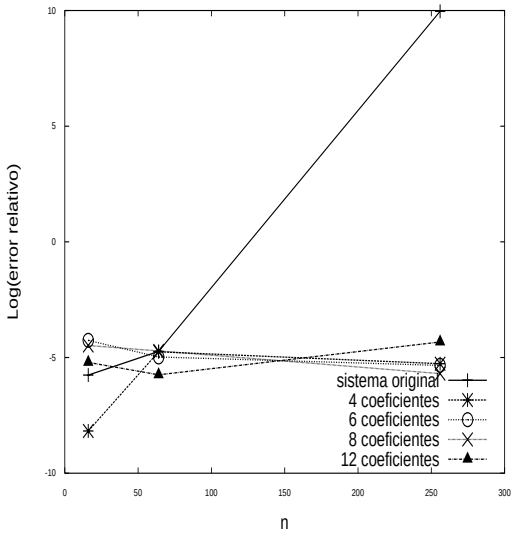


**Figura 173:** Caso 2 orden 2 Error relativo Biorthogonal-CGS sin precondicionamiento

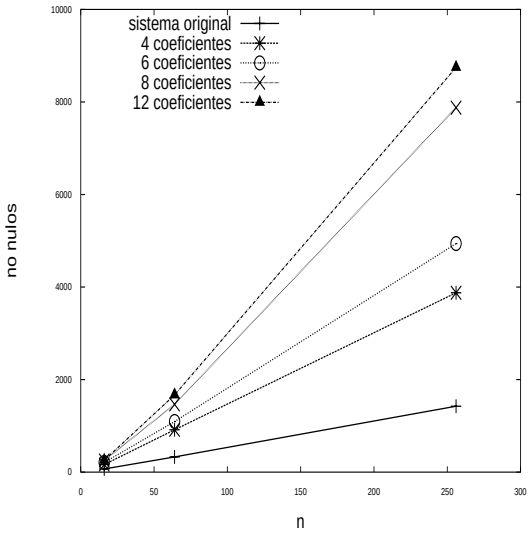


**Figura 174:** Caso 2 orden 2 Elementos no nulos Biorthogonal-QMR sin precondicionamiento

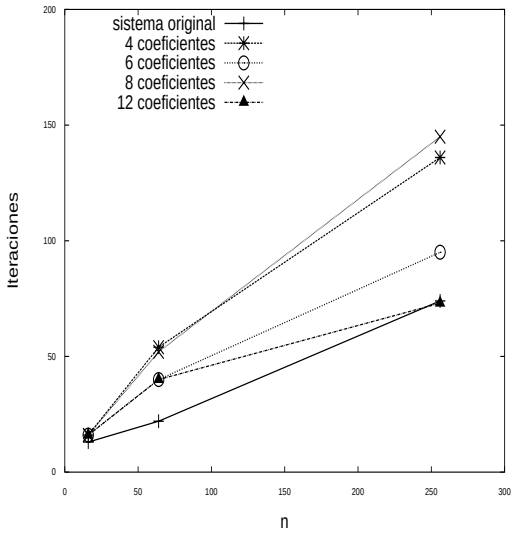


**Figura 175:** Caso 2 orden 2 Iteraciones Biorthogonal-QMR sin precondicionamiento

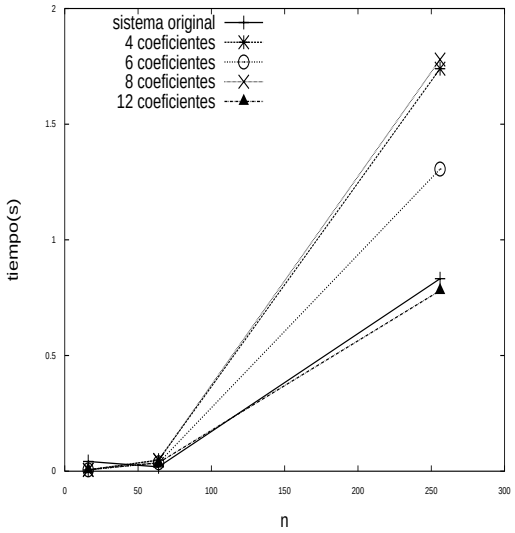


**Figura 176:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-QMR sin precondicionamiento

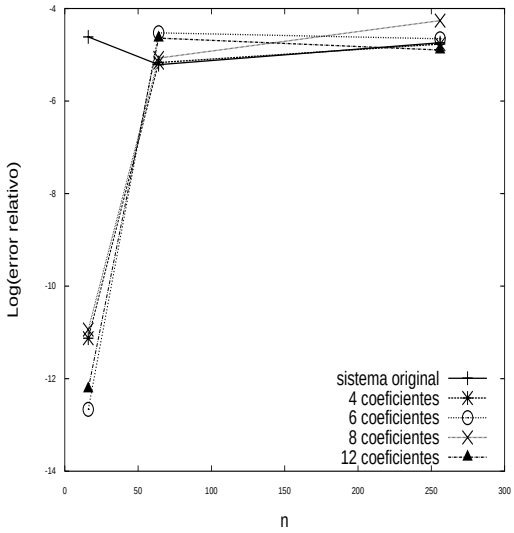


**Figura 177:** Caso 2 orden 2 Error relativo Biorthogonal-QMR sin precondicionamiento

En las figuras (162) a (177) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*. En los métodos BiCG, BiCGStab y QMR la menor cantidad de iteraciones se logra en el sistema original; en el método CGS no se presenta en general tal variación.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, prácticamente no hay diferencia en los tiempos de procesamiento en los distintos casos; para el método QMR, el menor tiempo se logra con 12 coeficientes.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG, CGS y QMR los errores son aproximadamente del mismo orden; para el método BiCGStab, los menores errores se logran con 12 coeficientes.

En las figuras (162) a (177) también puede observarse que en los experimentos numéricos realizados, ninguna combinación de métodos iterativos con distinta cantidad de coeficientes para la familia *wavelet* ofrece un mejor desempeño que los métodos iterativos en el sistema original.

Familia Reverse Biorthogonal

Los resultados obtenidos se observan en las figuras (178) a (193), que se muestran a continuación:

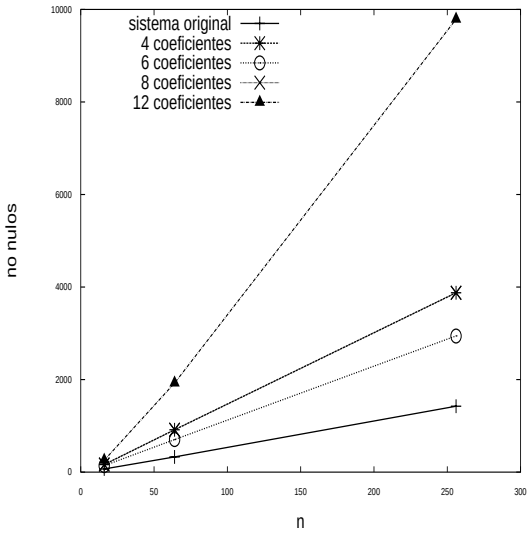


**Figura 178:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-BiCG sin precondicionamiento

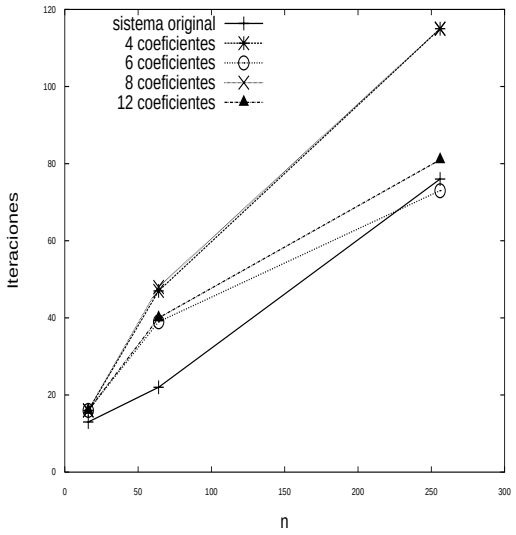


**Figura 179:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-BiCG sin precondicionamiento

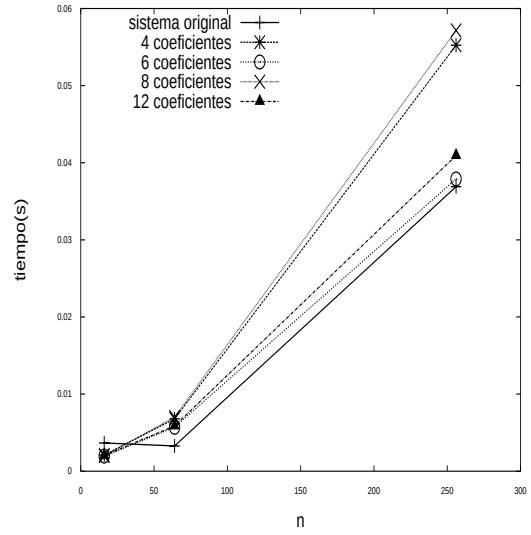


**Figura 180:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCG sin precondicionamiento

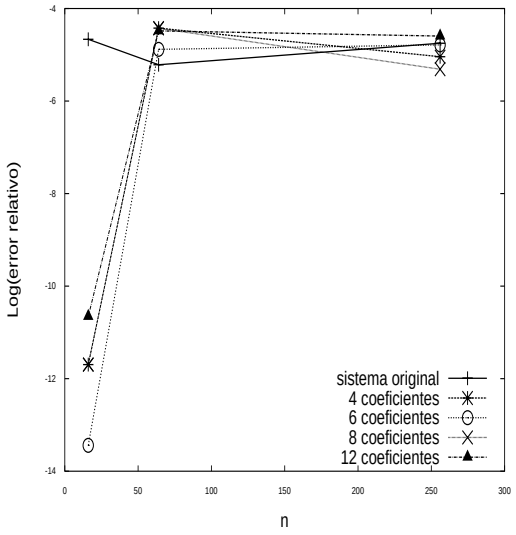


**Figura 181:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-BiCG sin precondicionamiento

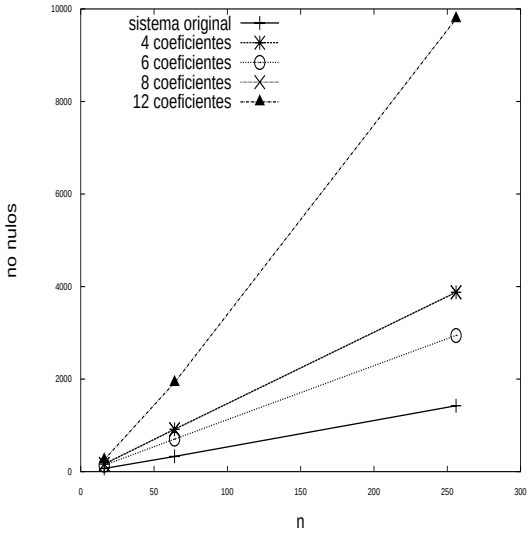


**Figura 182:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-BiCGStab sin precondicionamiento

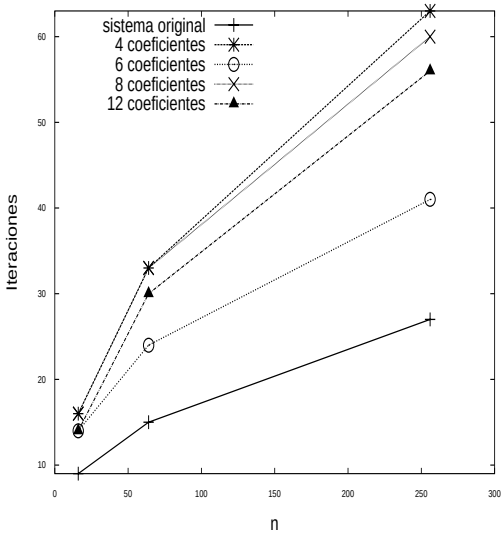


**Figura 183:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-BiCGStab sin precondicionamiento

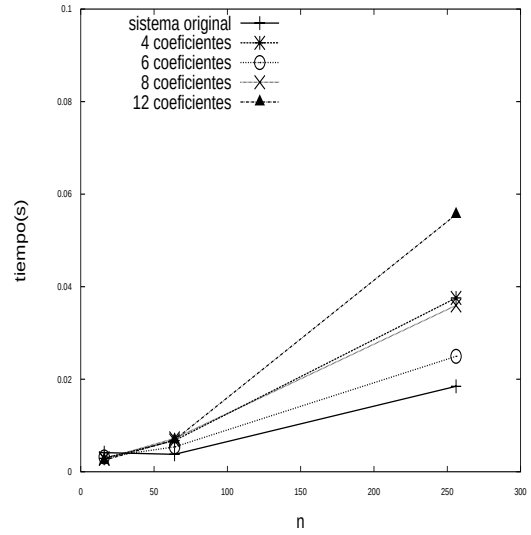


**Figura 184:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCGStab sin precondicionamiento

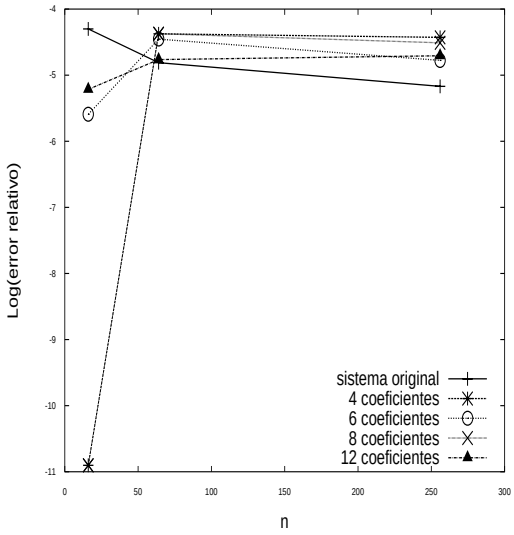


**Figura 185:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-BiCGStab sin precondicionamiento

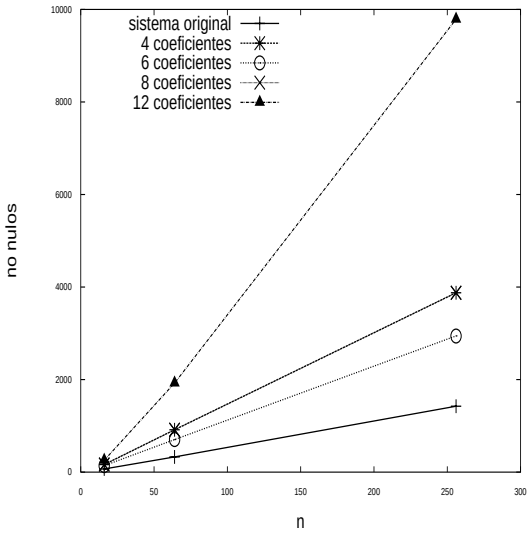


**Figura 186:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-CGS sin precondicionamiento

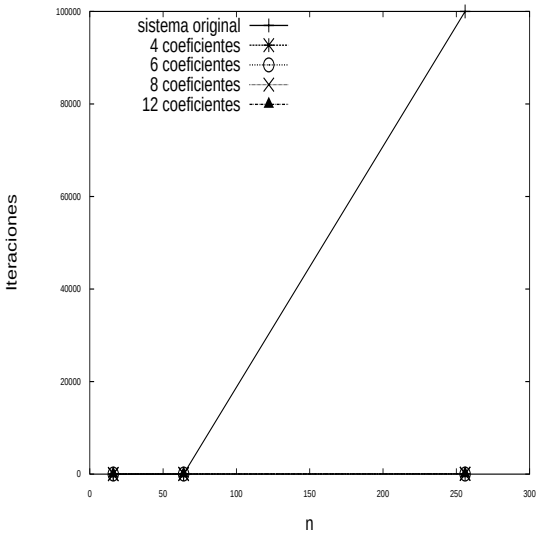


**Figura 187:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-CGS sin precondicionamiento

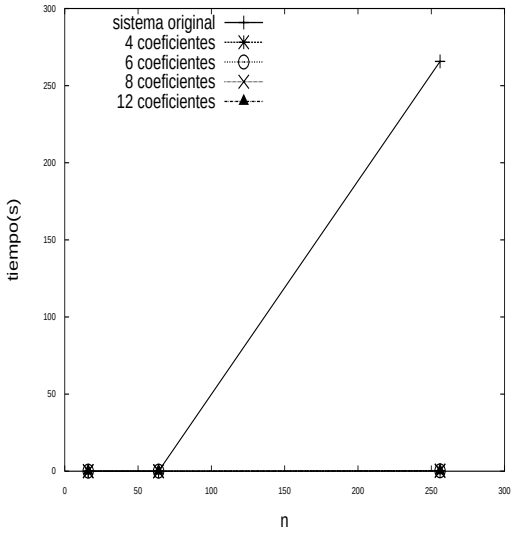


**Figura 188:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-CGS sin precondicionamiento

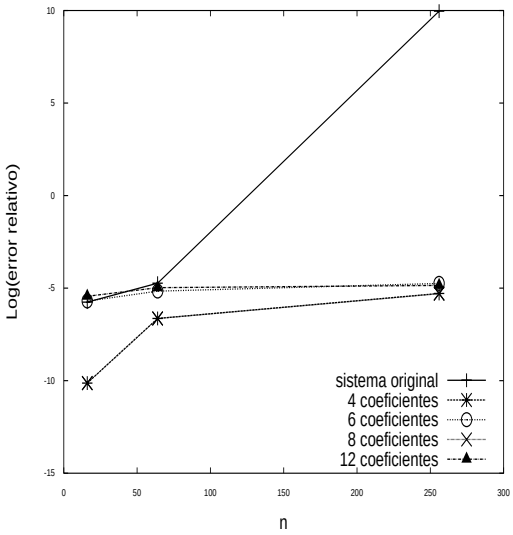


**Figura 189:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-CGS sin precondicionamiento

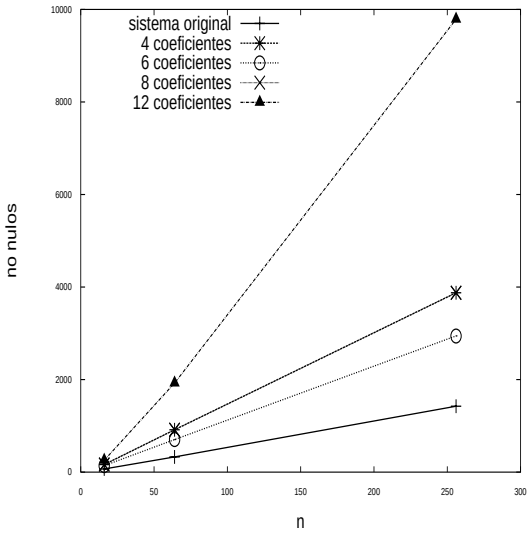


**Figura 190:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-QMR sin precondicionamiento

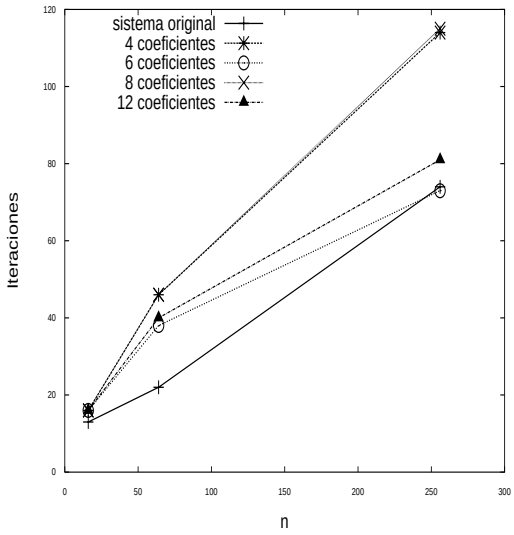


**Figura 191:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-QMR sin precondicionamiento

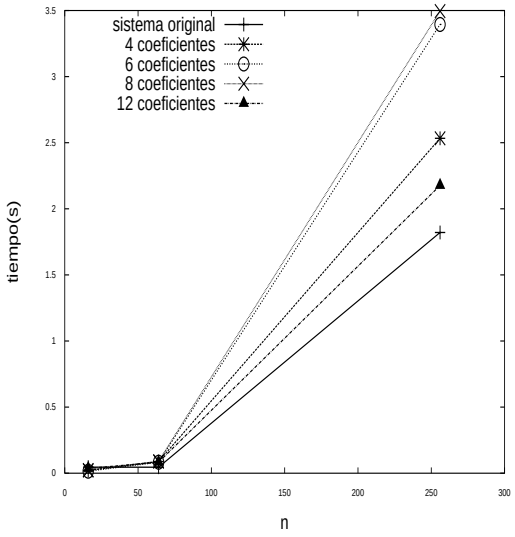


**Figura 192:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-QMR sin precondicionamiento

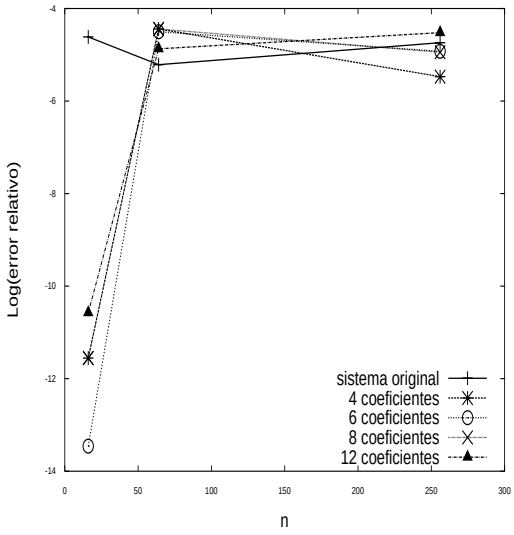


**Figura 193:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-QMR sin precondicionamiento

En las figuras (178) a (193) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, la menor cantidad de iteraciones se logra en el sistema original; en el método CGS la cantidad de iteraciones prácticamente no tiene variaciones.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG, BiCGStab y QMR, los menores tiempos se logran en el sistema original; para el método CGS, prácticamente no hay diferencia en los tiempos de procesamiento en los distintos casos.
- Los errores relativos son en la mayoría de los casos del mismo orden en todos los métodos.

En las figuras (178) a (193) también puede observarse que en los experimentos numéricos realizados, ninguna combinación de métodos iterativos con distinta cantidad de coeficientes para la familia *wavelet* ofrece un mejor desempeño que los métodos iterativos en el sistema original.

**Caso 2, con precondicionamiento**

Familia Daubechies

Los resultados obtenidos se observan en las figuras (194) a (209), que se muestran a continuación:

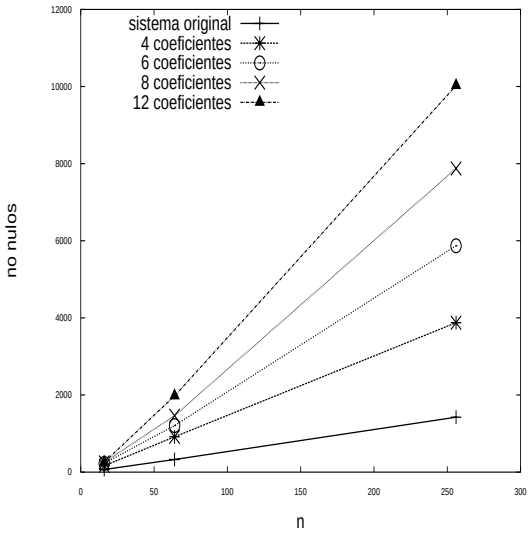


**Figura 194:** Caso 2 orden 2 Elementos no nulos Daubechies-BiCG con precondicionamiento

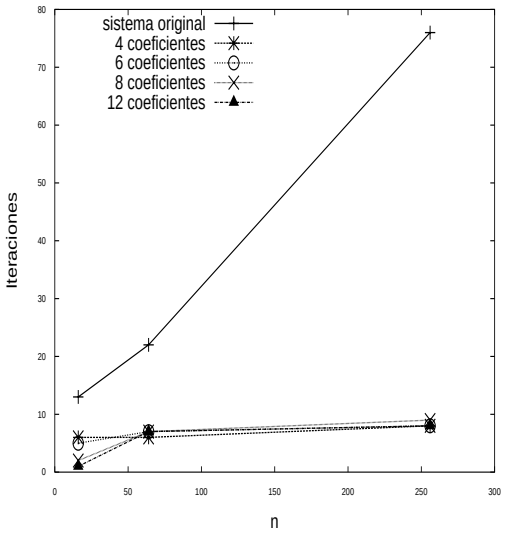


**Figura 195:** Caso 2 orden 2 Iteraciones Daubechies-BiCG con precondicionamiento

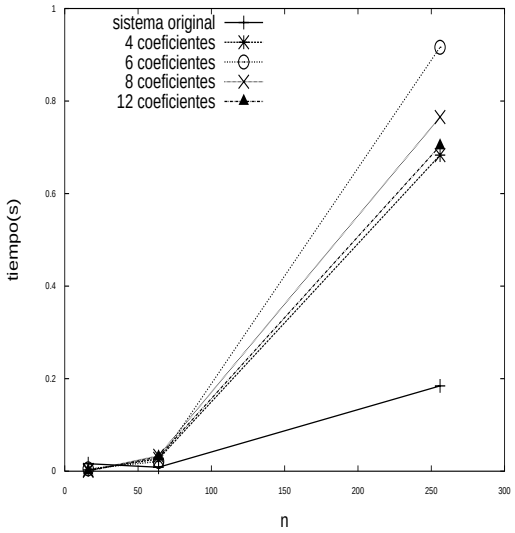


**Figura 196:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-BiCG con precondicionamiento

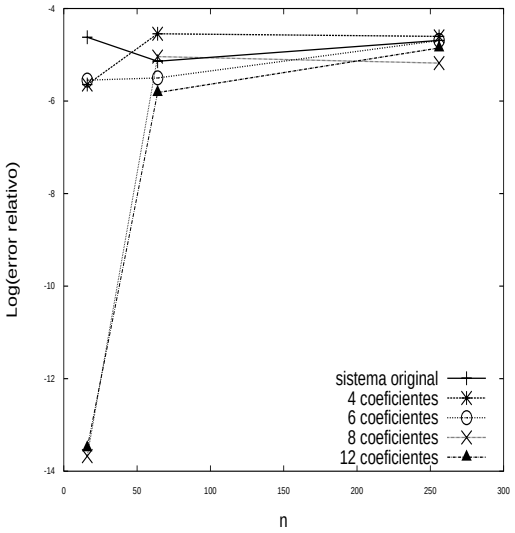


**Figura 197:** Caso 2 orden 2 Error relativo Daubechies-BiCG con precondicionamiento

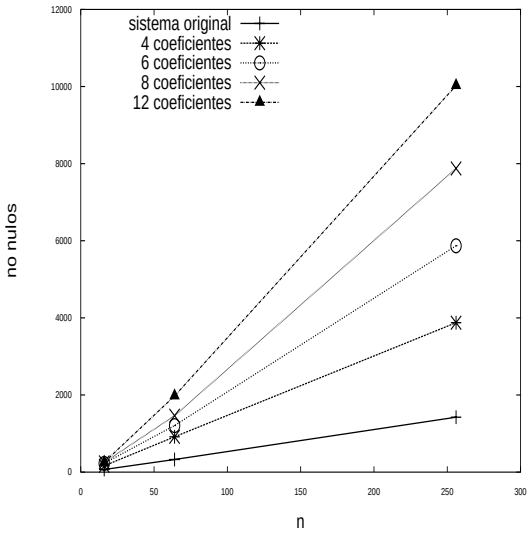


**Figura 198:** Caso 2 orden 2 Elementos no nulos Daubechies-BiCGStab con precondicionamiento

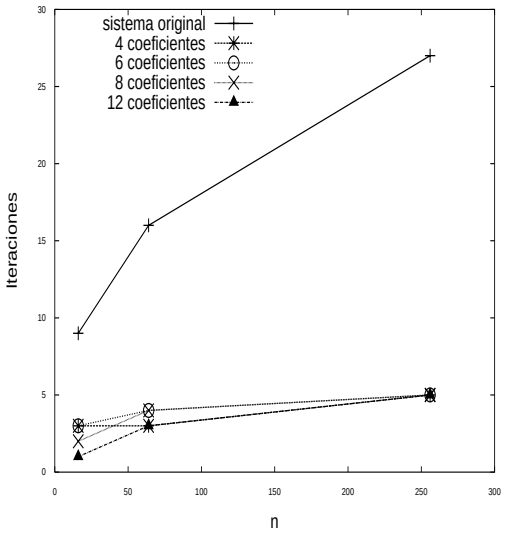


**Figura 199:** Caso 2 orden 2 Iteraciones Daubechies-BiCGStab con precondicionamiento

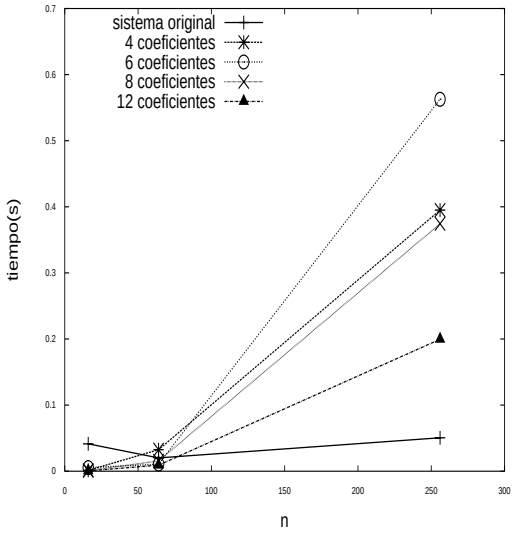


**Figura 200:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-BiCGStab con precondicionamiento

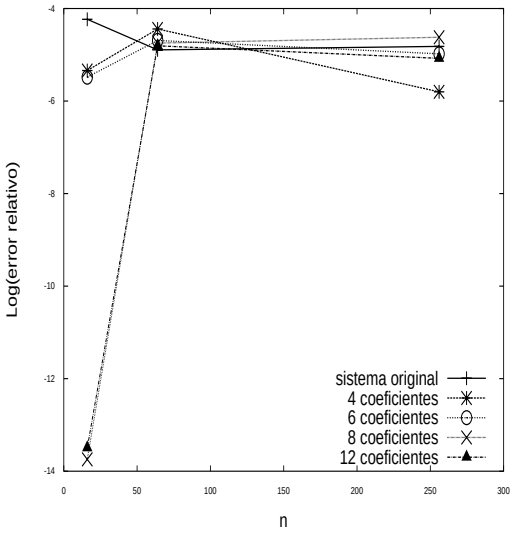


**Figura 201:** Caso 2 orden 2 Error relativo Daubechies-BiCGStab con precondicionamiento

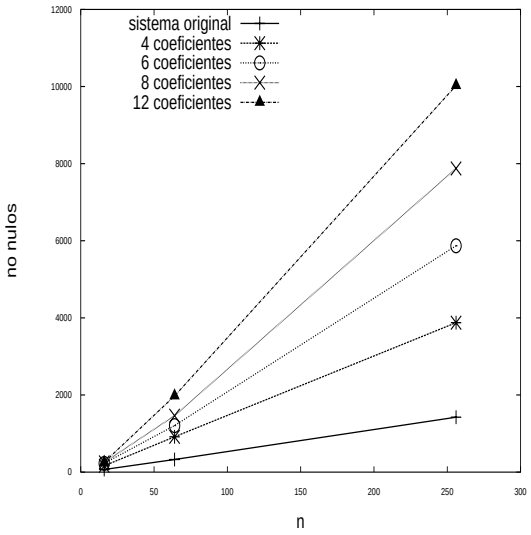


**Figura 202:** Caso 2 orden 2 Elementos no nulos Daubechies-CGS con precondicionamiento

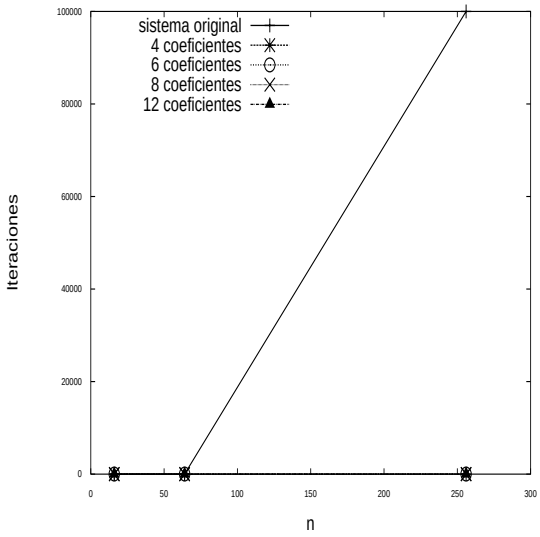


**Figura 203:** Caso 2 orden 2 Iteraciones Daubechies-CGS con precondicionamiento

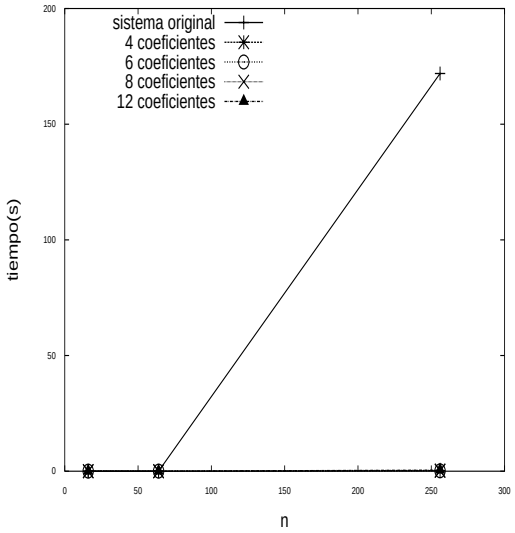


**Figura 204:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-CGS con precondicionamiento

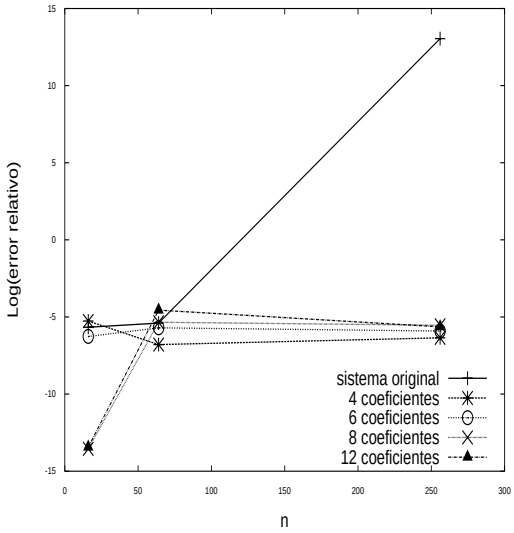


**Figura 205:** Caso 2 orden 2 Error relativo Daubechies-CGS con precondicionamiento

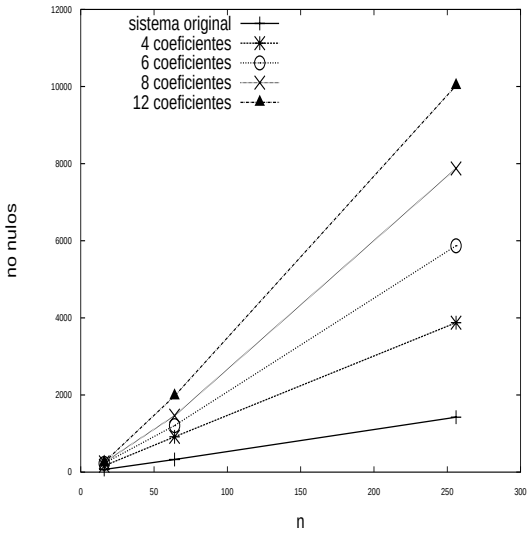


**Figura 206:** Caso 2 orden 2 Elementos no nulos Daubechies-QMR con precondicionamiento

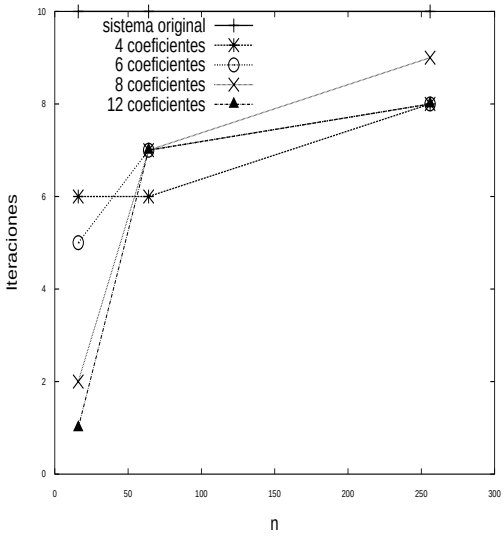


**Figura 207:** Caso 2 orden 2 Iteraciones Daubechies-QMR con precondicionamiento

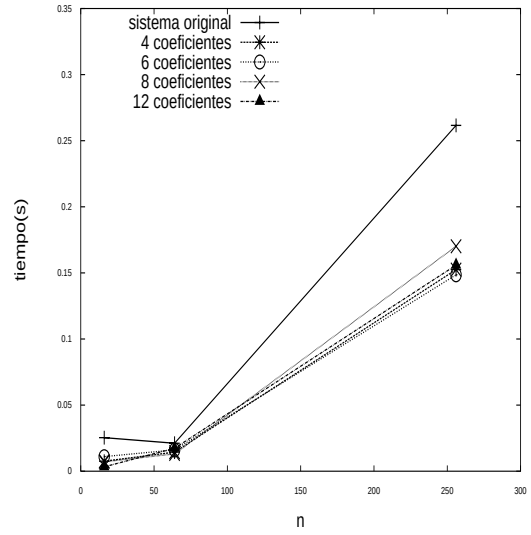


**Figura 208:** Caso 2 orden 2 Tiempo de procesamiento Daubechies-QMR con precondicionamiento

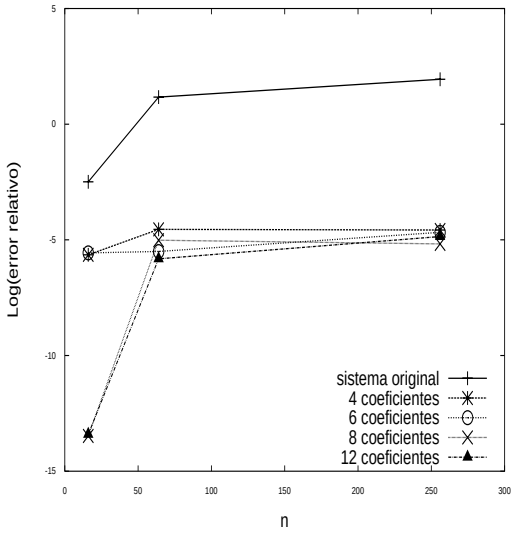


**Figura 209:** Caso 2 orden 2 Error relativo Daubechies-QMR con precondicionamiento

En las figuras (194) a (209) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 12 coeficientes.
- Los errores relativos son aproximadamente del mismo orden para las distintas cantidades de coeficientes *wavelet* empleados, y en algunos casos son menores a los errores alcanzados en el sistema original; presentándose el mejor comportamiento en el método QMR con 12 coeficientes.

En las figuras (194) a (209) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en todos los casos (ver figuras 208 y 209).

Familia Symlets

Los resultados obtenidos se observan en las figuras (210) a (225), que se muestran a continuación:

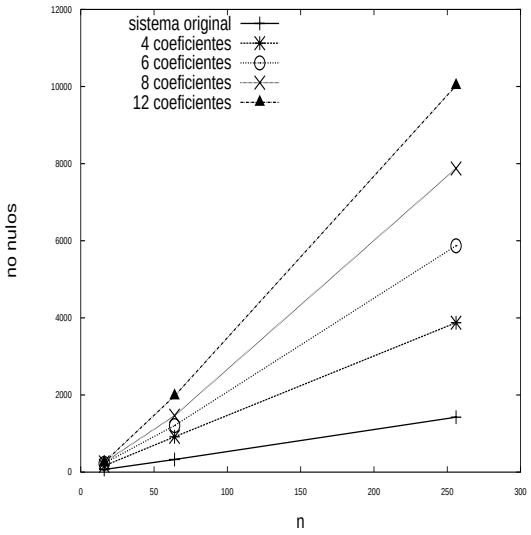


**Figura 210:** Caso 2 orden 2 Elementos no nulos Symlets-BiCG con precondicionamiento

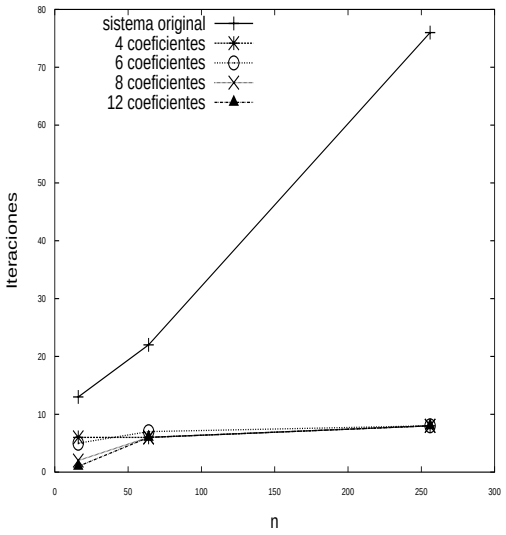


**Figura 211:** Caso 2 orden 2 Iteraciones Symlets-BiCG con precondicionamiento

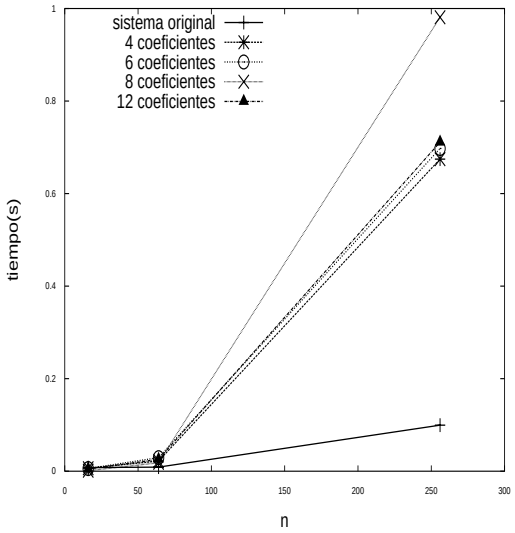


**Figura 212:** Caso 2 orden 2 Tiempo de procesamiento Symlets-BiCG con precondicionamiento

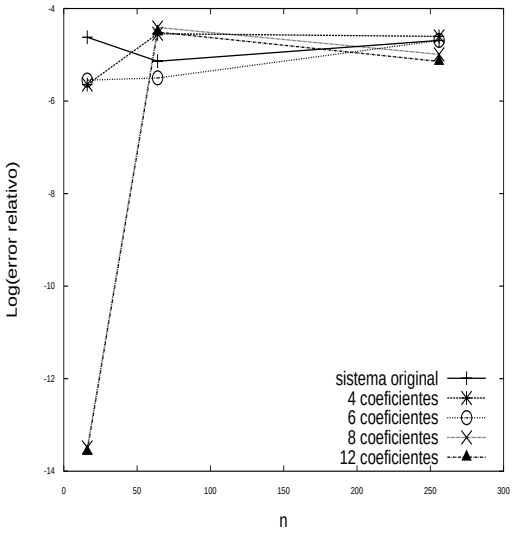


**Figura 213:** Caso 2 orden 2 Error relativo Symlets-BiCG con precondicionamiento

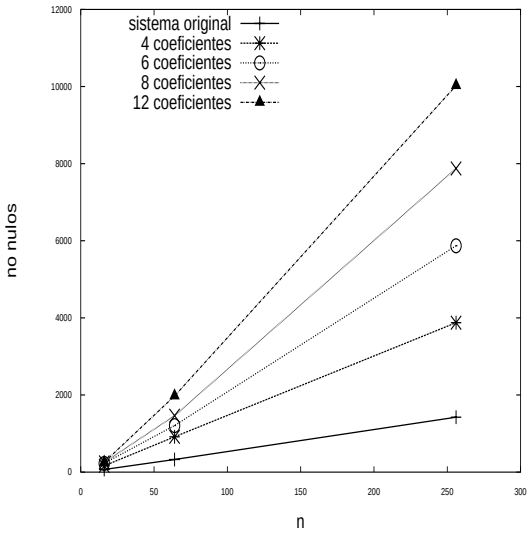


**Figura 214:** Caso 2 orden 2 Elementos no nulos Symlets-BiCGStab con precondicionamiento

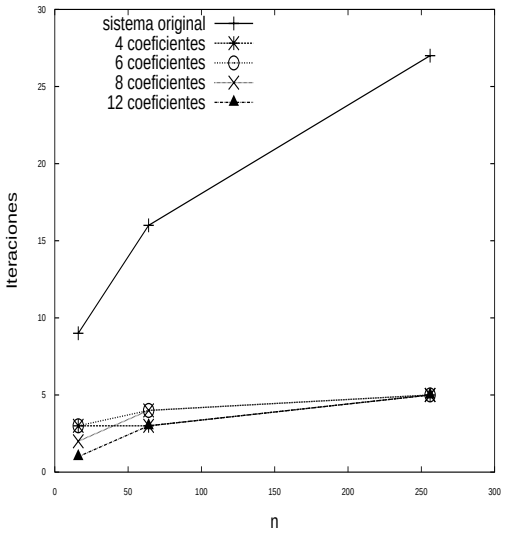


**Figura 215:** Caso 2 orden 2 Iteraciones Symlets-BiCGStab con precondicionamiento

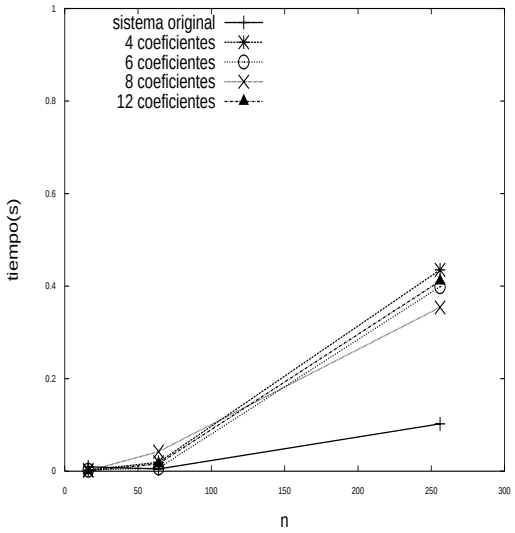


**Figura 216:** Caso 2 orden 2 Tiempo de procesamiento Symlets-BiCGStab con precondicionamiento

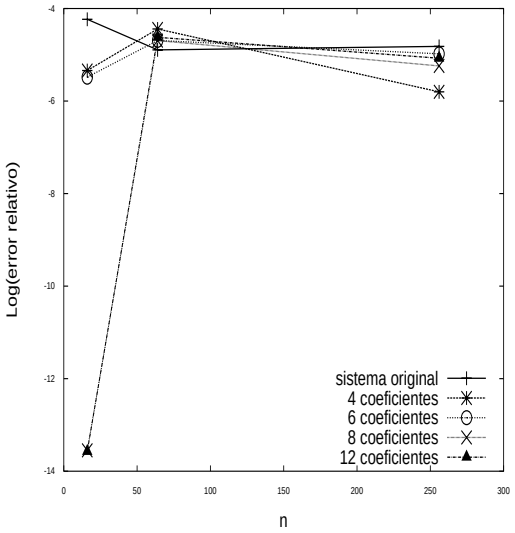


**Figura 217:** Caso 2 orden 2 Error relativo Symlets-BiCGStab con precondicionamiento

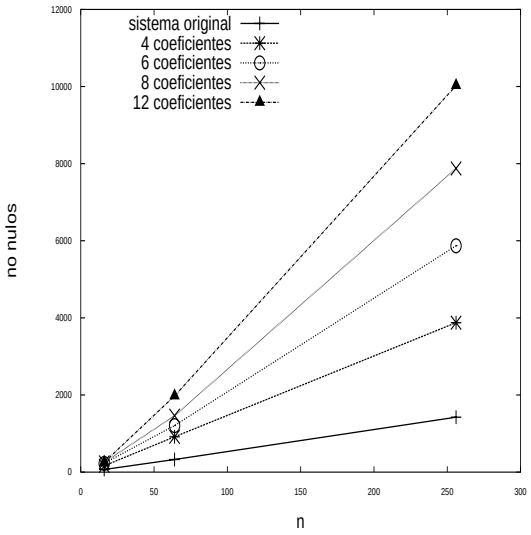


**Figura 218:** Caso 2 orden 2 Elementos no nulos Symlets-CGS con precondicionamiento

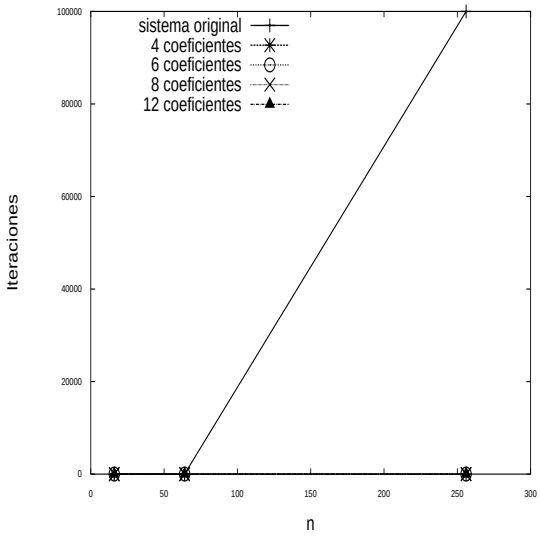


**Figura 219:** Caso 2 orden 2 Iteraciones Symlets-CGS con precondicionamiento

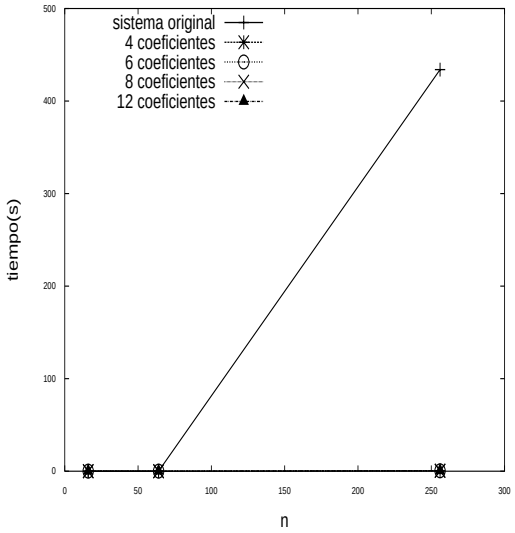


**Figura 220:** Caso 2 orden 2 Tiempo de procesamiento Symlets-CGS con precondicionamiento

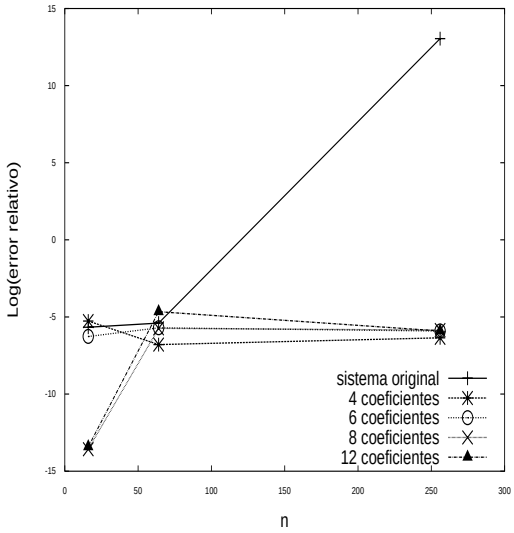


**Figura 221:** Caso 2 orden 2 Error relativo Symlets-CGS con precondicionamiento

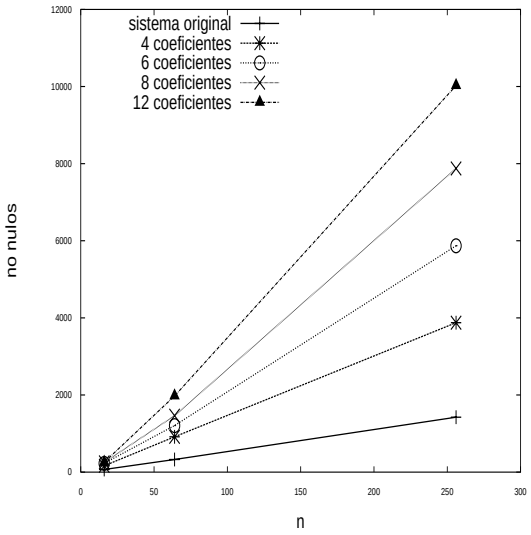


**Figura 222:** Caso 2 orden 2 Elementos no nulos Symlets-QMR con precondicionamiento

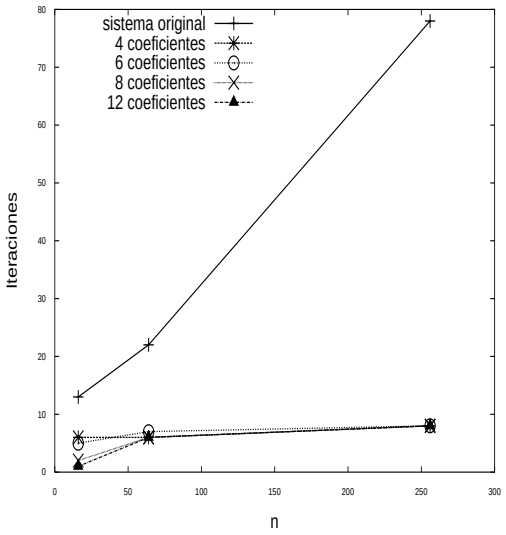


**Figura 223:** Caso 2 orden 2 Iteraciones Symlets-QMR con precondicionamiento

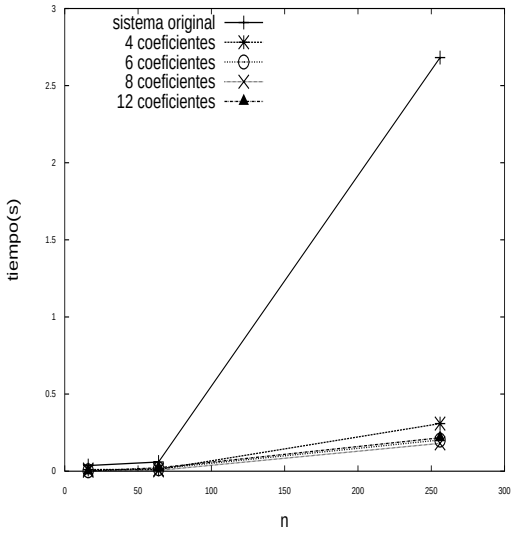


**Figura 224:** Caso 2 orden 2 Tiempo de procesamiento Symlets-QMR con precondicionamiento

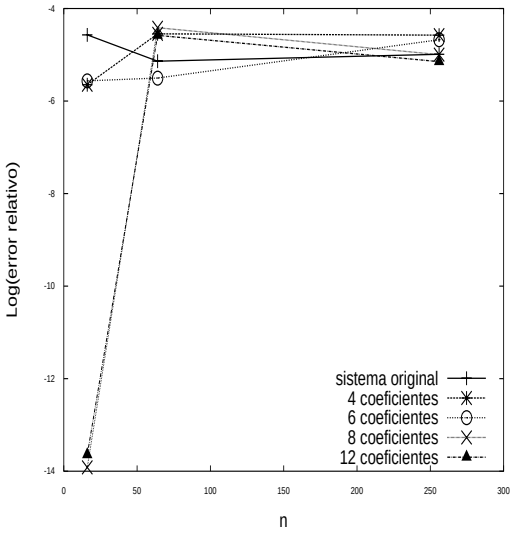


**Figura 225:** Caso 2 orden 2 Error relativo Symlets-QMR con precondicionamiento

En las figuras (210) a (225) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 8 coeficientes, teniéndose tiempos bastantes cercanos al menor con 6 y 12 coeficientes.

- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, para 12 coeficientes se presenta el mejor comportamiento del error en la mayoría de los casos.

En las figuras (210) a (225) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los mejores comportamientos de errores relativos. (ver figuras 224 y 225).

Familia Biorthogonal

Los resultados obtenidos se observan en las figuras (226) a (241), que se muestran a continuación:

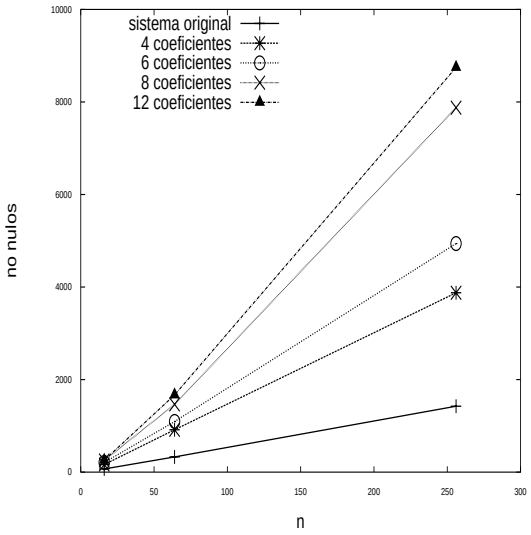


**Figura 226:** Caso 2 orden 2 Elementos no nulos Biorthogonal-BiCG con precondicionamiento

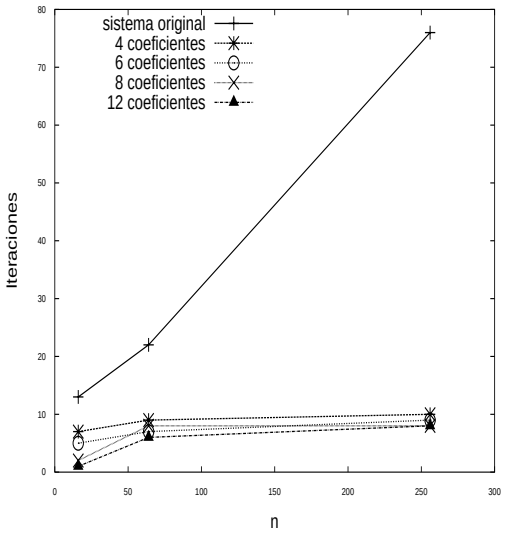


**Figura 227:** Caso 2 orden 2 Iteraciones Biorthogonal-BiCG con precondicionamiento

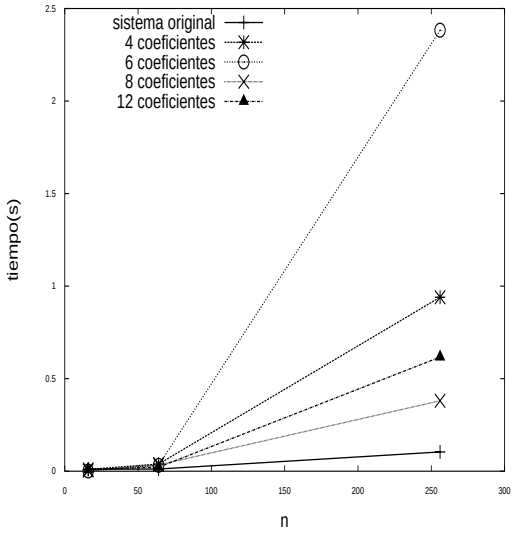


**Figura 228:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-BiCG con precondicionamiento

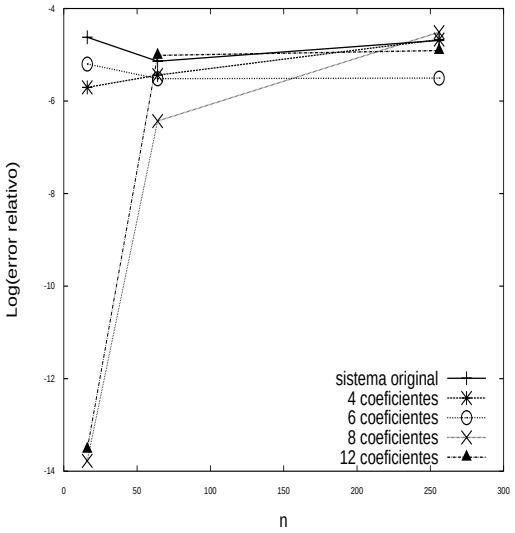


**Figura 229:** Caso 2 orden 2 Error relativo Biorthogonal-BiCG con precondicionamiento

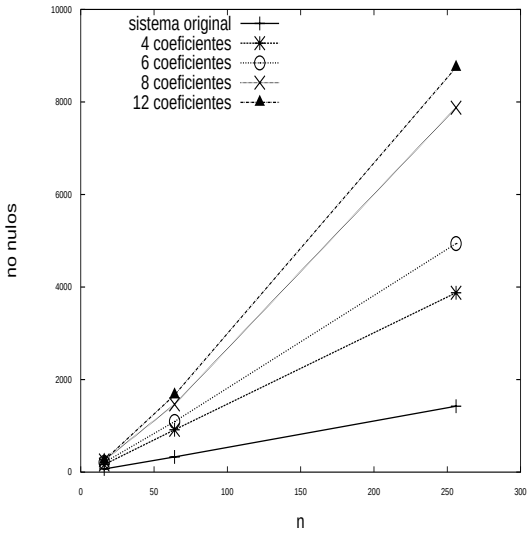


**Figura 230:** Caso 2 orden 2 Elementos no nulos Biorthogonal-BiCGStab con precondicionamiento

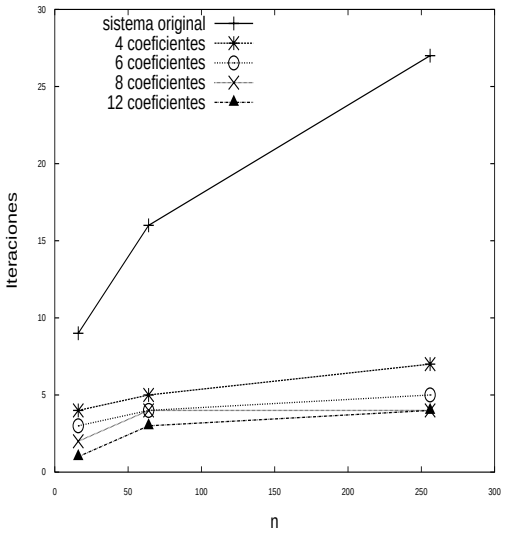


**Figura 231:** Caso 2 orden 2 Iteraciones Biorthogonal-BiCGStab con precondicionamiento

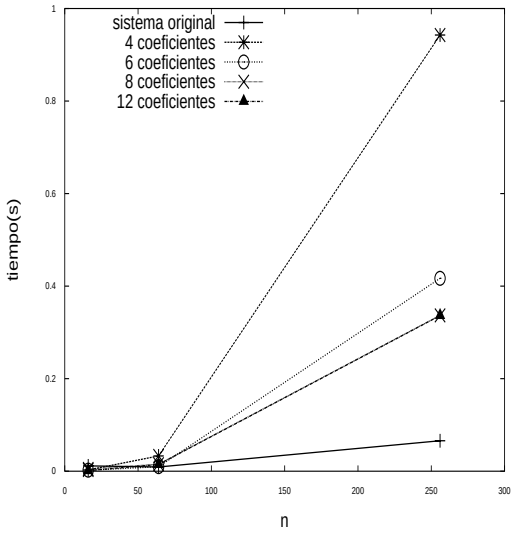


**Figura 232:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-BiCGStab con precondicionamiento

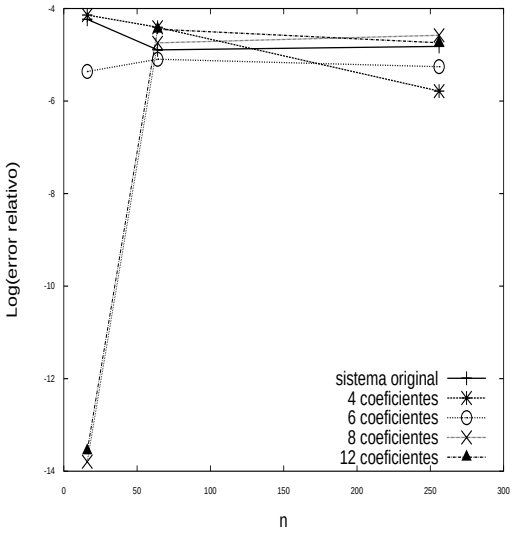


**Figura 233:** Caso 2 orden 2 Error relativo Biorthogonal-BiCGStab con precondicionamiento

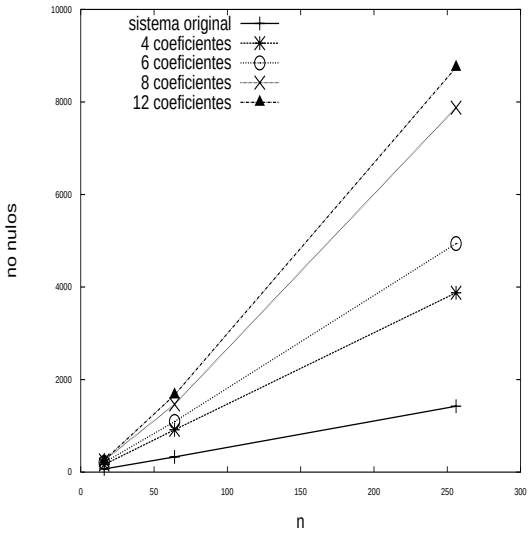


**Figura 234:** Caso 2 orden 2 Elementos no nulos Biorthogonal-CGS con precondicionamiento

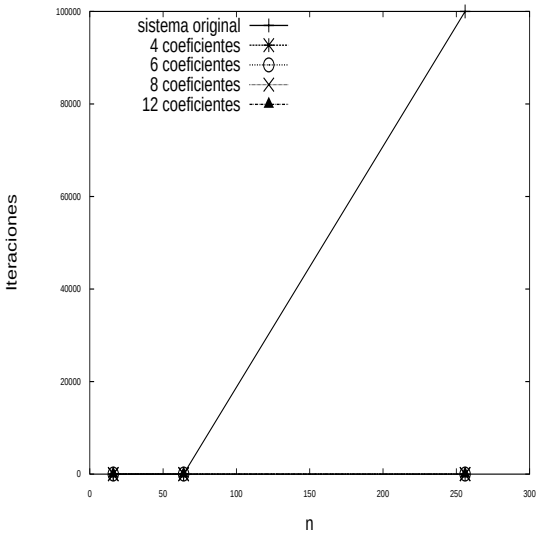


**Figura 235:** Caso 2 orden 2 Iteraciones Biorthogonal-CGS con precondicionamiento

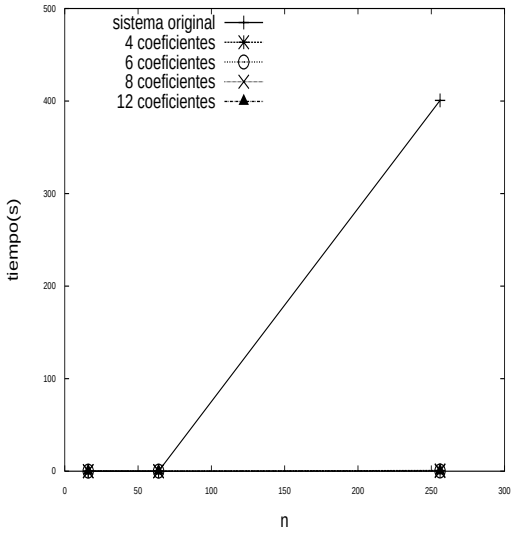


**Figura 236:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-CGS con precondicionamiento

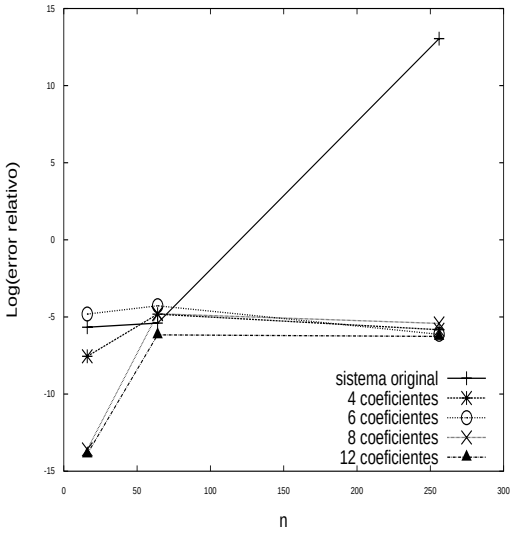


**Figura 237:** Caso 2 orden 2 Error relativo Biorthogonal-CGS con precondicionamiento

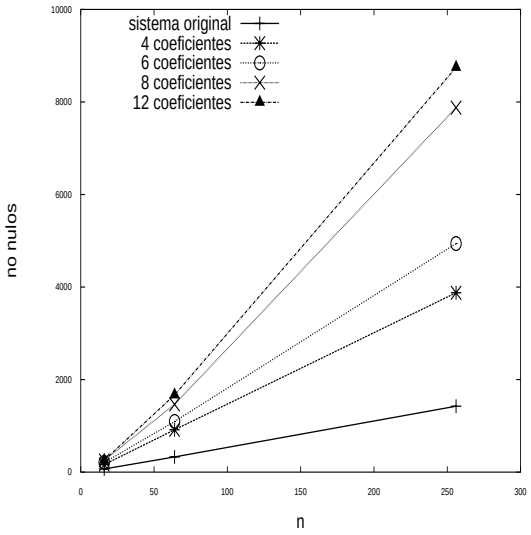


**Figura 238:** Caso 2 orden 2 Elementos no nulos Biorthogonal-QMR con precondicionamiento

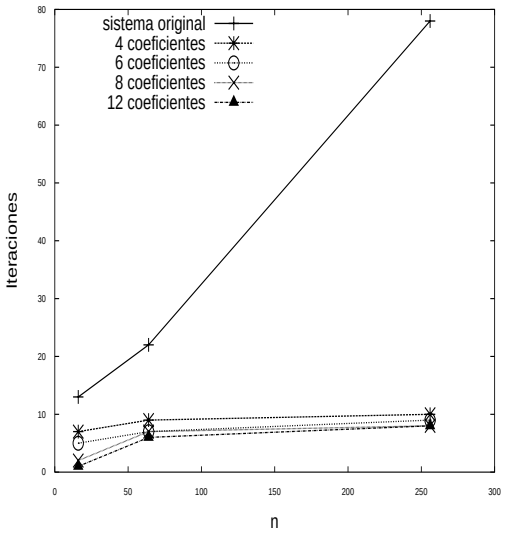


**Figura 239:** Caso 2 orden 2 Iteraciones Biorthogonal-QMR con precondicionamiento

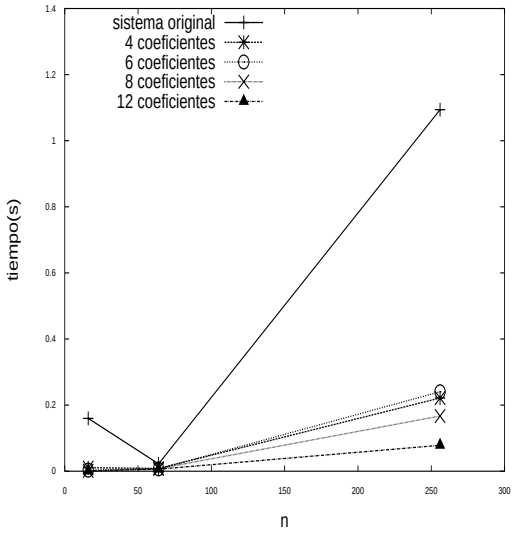


**Figura 240:** Caso 2 orden 2 Tiempo de procesamiento Biorthogonal-QMR con precondicionamiento

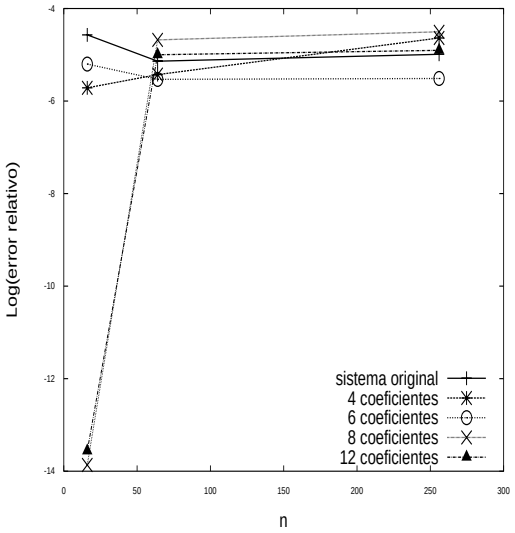


**Figura 241:** Caso 2 orden 2 Error relativo Biorthogonal-QMR con precondicionamiento

En las figuras (226) a (241) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.
- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose aproximadamente la misma cantidad de iteraciones independiente de la cantidad de coeficientes *wavelet* empleados a medida que el tamaño de la matriz aumenta. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.
- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 12 coeficientes.
- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, en el método QMR con 12 coeficientes presenta el mejor comportamiento en los errores relativos en la mayoría de los casos.

En las figuras (226) a (241) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta el mejor comportamiento en los errores relativos en la mayoría de los casos (ver figuras 240 y 241).

Familia Reverse Biorthogonal

Los resultados obtenidos se observan en las figuras (242) a (257), que se muestran a continuación:

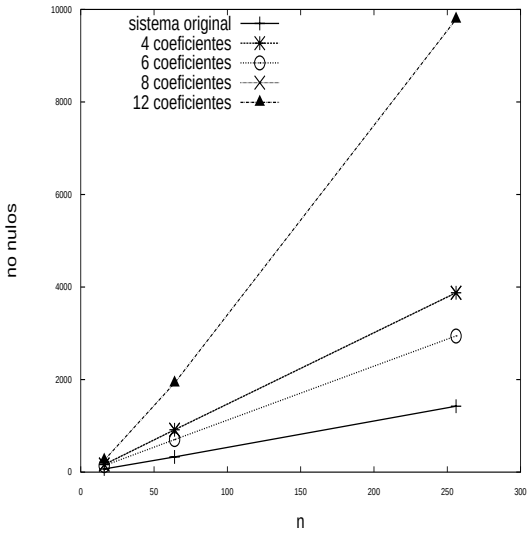


**Figura 242:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-BiCG con precondicionamiento

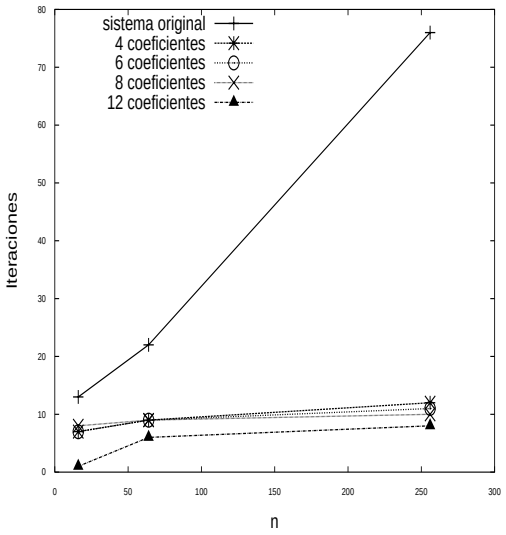


**Figura 243:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-BiCG con precondicionamiento

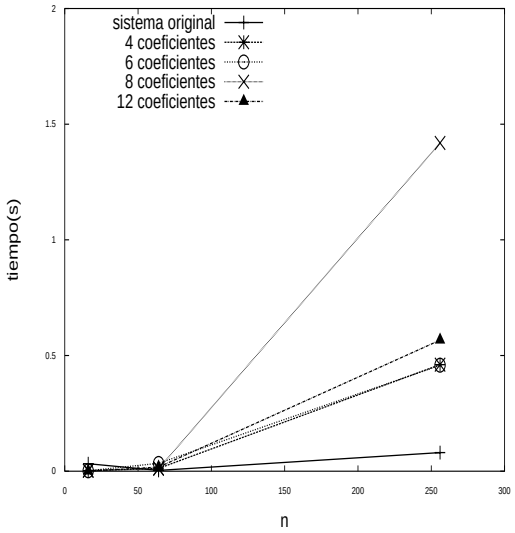


**Figura 244:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCG con precondicionamiento

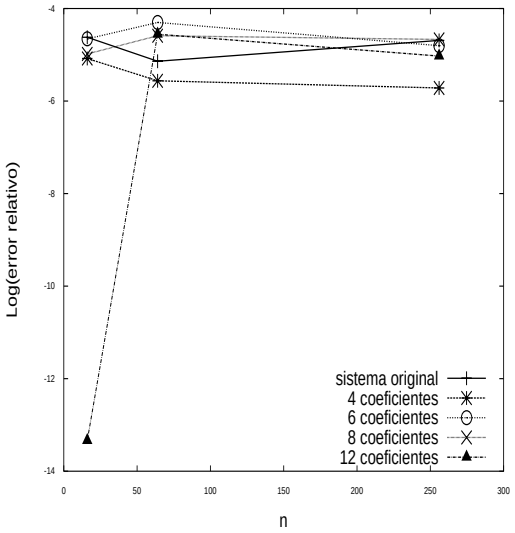


**Figura 245:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-BiCG con precondicionamiento

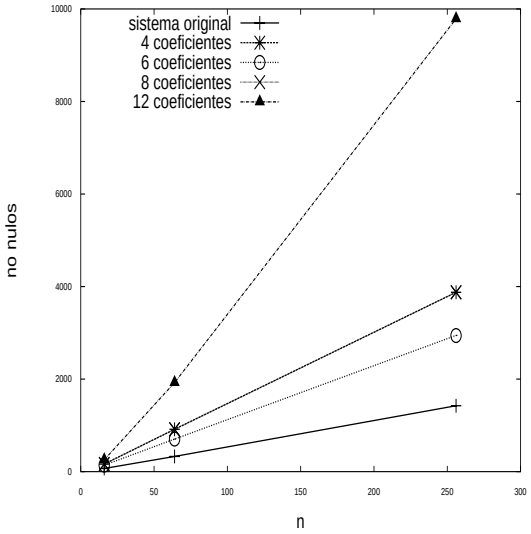


**Figura 246:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-BiCGStab con precondicionamiento

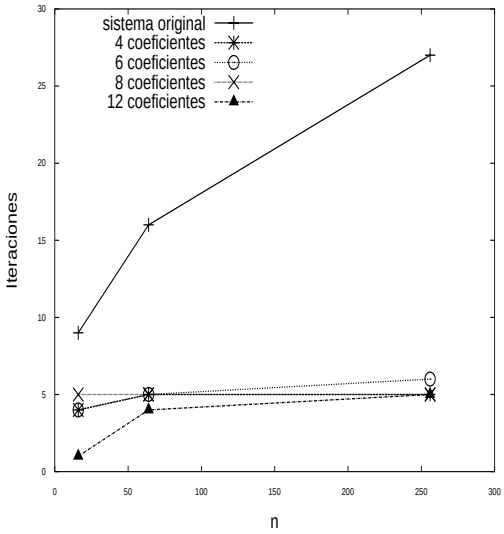


**Figura 247:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-BiCGStab con precondicionamiento

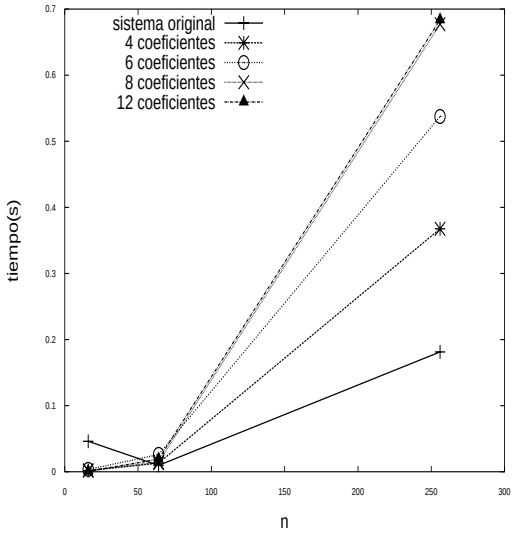


**Figura 248:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-BiCGStab con precondicionamiento

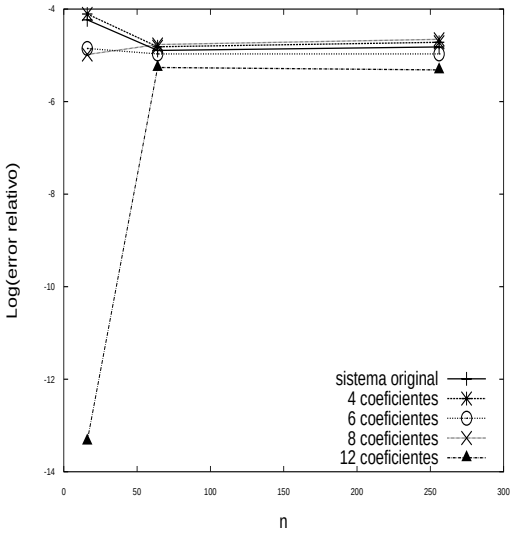


**Figura 249:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-BiCGStab con precondicionamiento

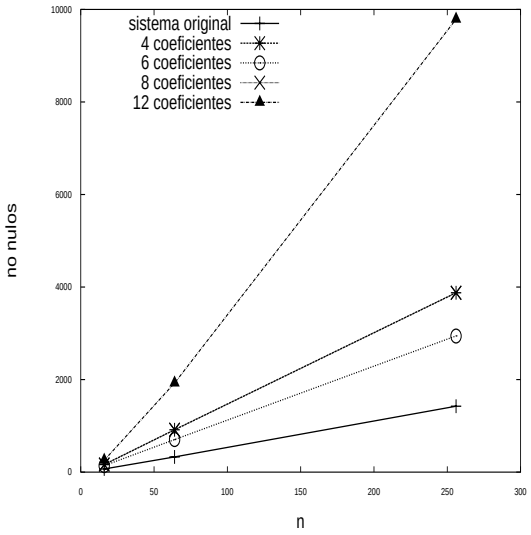


**Figura 250:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-CGS con precondicionamiento

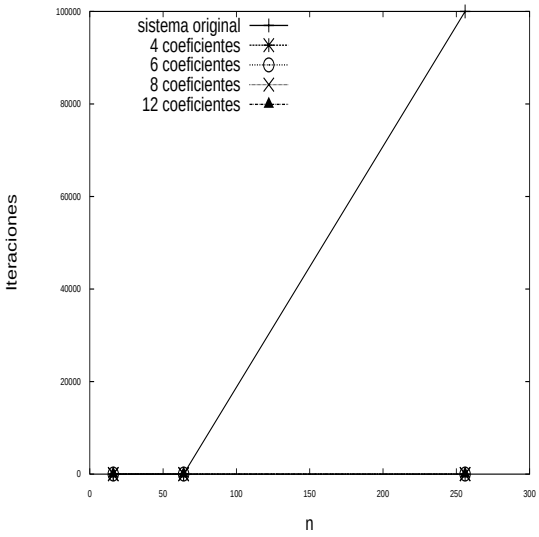


**Figura 251:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-CGS con precondicionamiento

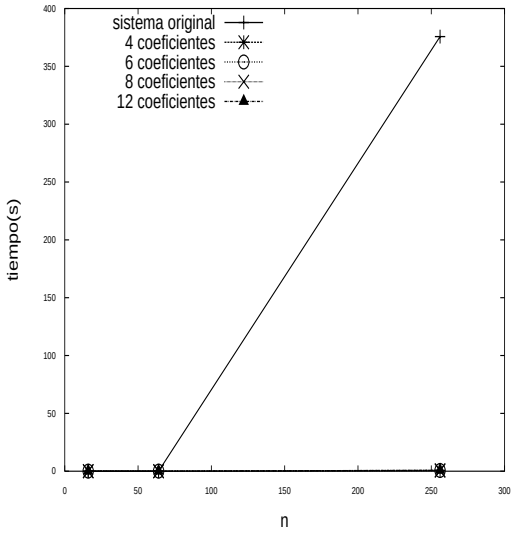


**Figura 252:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-CGS con precondicionamiento

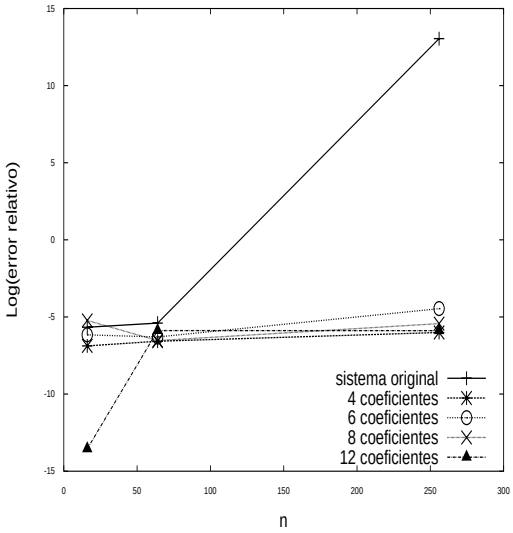


**Figura 253:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-CGS con precondicionamiento

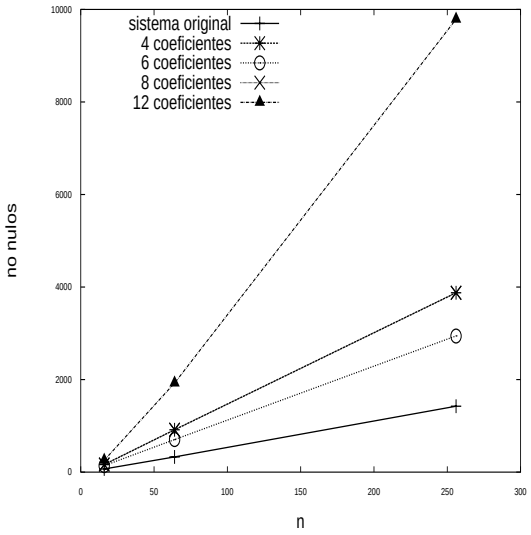


**Figura 254:** Caso 2 orden 2 Elementos no nulos Reverse Biorthogonal-QMR con precondicionamiento

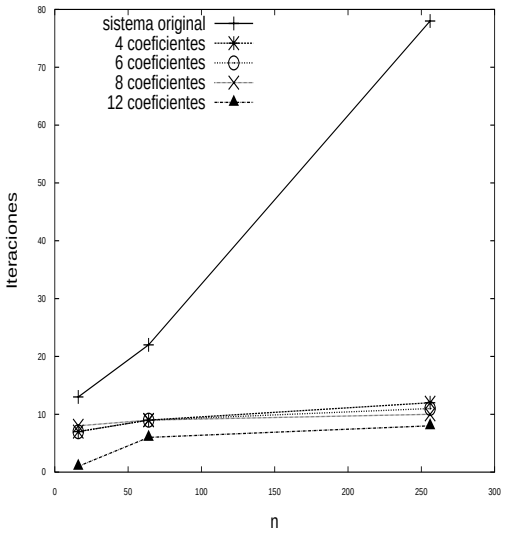


**Figura 255:** Caso 2 orden 2 Iteraciones Reverse Biorthogonal-QMR con precondicionamiento

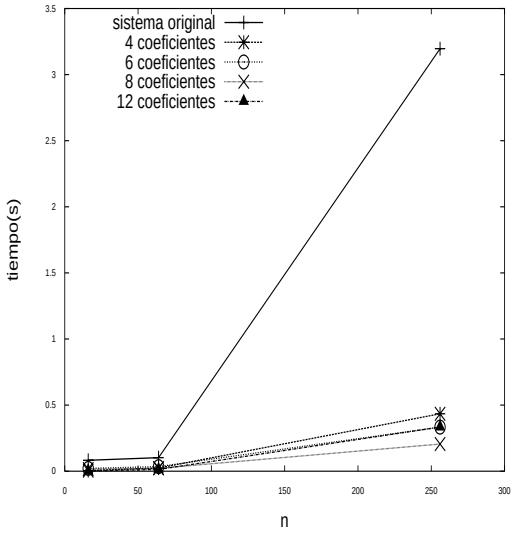


**Figura 256:** Caso 2 orden 2 Tiempo de procesamiento Reverse Biorthogonal-QMR con precondicionamiento

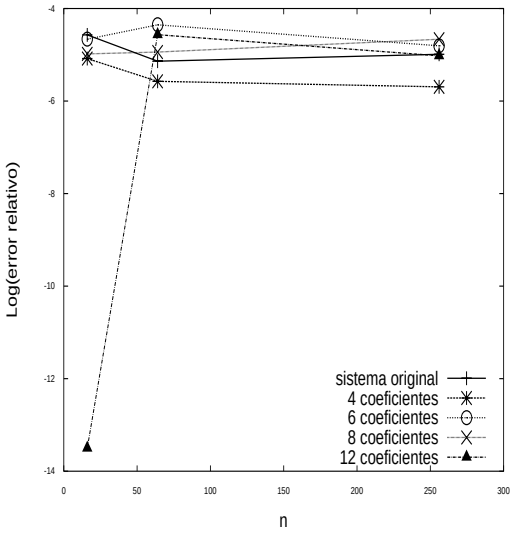


**Figura 257:** Caso 2 orden 2 Error relativo Reverse Biorthogonal-QMR con precondicionamiento

En las figuras (242) a (257) puede observarse que en los experimentos numéricos realizados:

- La transformación del esquema original al esquema en la base *wavelet* produce un aumento de elementos no nulos en la matriz de coeficientes del sistema en todos los casos; a mayor cantidad de coeficientes en la familia *wavelet*, mayor es la cantidad de elementos no nulos en la matriz de coeficientes del sistema en la base *wavelet*.

- La cantidad de iteraciones para la convergencia varía dependiendo de la cantidad de coeficientes de la famila *wavelet*; en los métodos BiCG, BiCGStab y QMR, se tiene una reducción significativa con la aplicación del precondicionador en la base *wavelet* en relación al sistema original, teniéndose la menor cantidad de iteraciones en el caso de 12 coeficientes. La cantidad de iteraciones para la convergencia no varía dependiendo de la cantidad de coeficientes de la famila *wavelet* en el métodos CGS.

- Los tiempos de procesamiento varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*. Para los métodos BiCG y BiCGStab, el menor tiempo se logra en el sistema original; para el método CGS, los tiempos no varían significativamente en la mayoría de los casos ; para el método QMR, el menor tiempo se logra con 8 coeficientes, teniéndose tiempos bastantes cercanos al menor con 6 y 12 coeficientes.

- Los errores relativos varían dependiendo de la combinación entre el método iterativo y el número de coeficientes de la familia *wavelet*; sin embargo, en el método QMR con 12 coeficientes se presenta en la mayoría de los casos los menores errores relativos.

En las figuras (242) a (257) también puede observarse que en los experimentos numéricos realizados, el método que presenta mejor relación entre el tiempo de procesamiento y el error relativo es el método QMR con 12 coeficientes, en virtud que presenta entre los menores tiempos de procesamiento y presenta los menores errores relativos en la mayoría de los casos ( ver figuras 256 y 257).

# CONCLUSIONES Y TRABAJOS FUTUROS

Realizada la presentación de los resultados obtenidos, a partir del análisis de dichos resultados se generan las siguientes conclusiones:

1. De la construcción de los esquemas numéricos provenientes de diferencias finitas de orden superior, se concluye que en forma general la cantidad de elementos no nulos en las matrices resultantes es pequeña en relación a la cantidad total de elementos de la matriz, lo que la categoriza dentro de las matrices de tipo dispersa. Así mismo, dichas matrices son no simétricas.

2. En todos los casos, la transformación los esquemas numéricos provenientes de diferencias finitas de orden superior a la base *wavelet*, genera un aumento en la cantidad de elementos no nulos en relación al esquema original.

3. En los casos estudiados sin precondicionamiento, tanto en el primer caso, en el cual predomina el término difusivo; como en el segundo caso, en el cual predomina el término convectivo; las familias Daubechies y Symlets ofrecen una buena relación entre tiempo de procesamiento y error relativo en combinación con el método BiCGStab.

4. En los casos estudiados con precondicionamiento, la cantidad de iteraciones se ve sustancialmente reducida, en todos los casos. El método QMR presenta mejor relación entre el tiempo de procesamiento y el error relativo en combinación con todas las familias *wavelet* empleadas en las pruebas correspondientes; en la mayoría de las pruebas realizadas el mejor comportamiento se observa con 12 coeficientes, presentando menores tiempos de procesamiento a medida que aumenta el tamaño de la matriz de coeficientes del sistema en comparación con los casos donde se emplea el método QMR sin precondicionamiento. Eśtos resultados se presentan tanto en el caso donde predomina el término difusivo como en el caso donde predomina el término convectivo.

Como trabajos futuros, se proponen los siguientes:

i. Realizar los experimientos numéricos planteados en éste trabajo empleando otro tipo de precondicionadores, tales como el precondicionador tipo SPAI.

ii. Realizar el estudio de la aplicacíon de técnicas de umbralización en los sistemas generados en el presente trabajo una vez transformados a la base *wavelet*.

iii. Realizar la deducción de los esquemas numéricos provenientes de métodos implícitos de orden 3, y realizar los experimientos numéricos planteados en éste trabajo con dichos esquemas de orden 3, para establecer el comportamiento correspondiente.

# REFERENCIAS BIBLIOGRÁFICAS


Acevedo, L. (2009). Computación paralela de la transformada wavelet; aplicaciones de la transformada wavelet al álgebra lineal numérica. Tesis Doctoral, Universidad Politécnica de Valencia, España.

Causon D. y Mingham, C. (2010). *Introductory Finite Difference Methods for PDEs*. Ventus Publishing ApS.

Cerrolaza, M. (2007). *El método de los elementos finitos para ingeniería y ciencias aplicadas: teoría y programas*. Universidad Central de Venezuela, Consejo de Desarrollo Científico Humanístico.

Chen, K. (2005). *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, Cambridge.

Flores, E., Vargas, E. y Rey, D. (2012). Aplicación de técnicas wavelets en los sistemas de ecuaciones lineales resultantes de la aproximación numérica de solución de ecuaciones diferenciales usando discretizaciones miméticas. *Avances en simulación numérica y modelado computacional, SVMNI*.

Guo, B. y Wang, X. (2005). Testing of new high-order finite difference methods for solving the convection-diffusion equation. *E-Journal of Reservoir Engineering*.

Hernández, F. (2005). Estudio de los sistemas lineales dispersos provenientes de discretizaciones miméticas, caso: Operador de convección-difusión. Trabajo de Grado de Maestría no publicado.

Jiménez C. y Jiménez, J. (2010). Clasificación de señales de voz utilizando transformada de wavelet y máquinas de vectores de soporte. Trabajo de ascenso no publicado, Universidad de Carabobo.

Kincaid, D. y Cheney, W. (1994). *Análisis Numérico, las matemáticas del cálculo científico*. Addison-Wesley Iberoamericana.

Larrazábal, G. (2002). Técnicas algebraicas de precondicionamiento para la resolución de sistemas lineales. Tesis Doctoral , Universidad Politecnica de Cataluña, Barcelona, España.

Mathews, J. y Fink, K. (2000). *Métodos Numéricos con Matlab*. Prentice Hall.

Nakamura, S. (1992). *Métodos Numéricos Aplicados con Software*. Pearson Educación.

Peaceman, D. (1977). *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Publishing Company.

Rodriguez, Y., Ochoa M. y Pineda, M. (2008). *La Experiencia De Investigar. Recomendaciones Precisas Para Realizar Una Investigación Y No Morir En El Intento*. Dirección de Medios y Publicaciones de la Universidad de Carabobo.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM.

Van Der Vorst, H. (2003). *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge.

Villarín, L., León, A., Baguer, M., y Linares, Y. (2012). GMRES precondicionado con wavelets, un algoritmo de selección del umbral para la obtención del patrón de dispersión. *Revista Investigación Operacional*, 33(3):222–232.

Zill, D. (2000). *Ecuaciones diferenciales con aplicaciones de modelado*. International Thomson Editores.

# APÉNDICE

## Código en Octave para la generación del esquema numérico a partir de métodos implícitos de diferencias finitas de orden 2 en el problema de convección-difusión

```
function [A,b] = genera_sistema_orden_2(N,L,phi,u,D,deltax,deltat,C)
% Genera la matriz de coeficientes del sistema (A) y el vector columna
% de terminos independientes (b)
% provenientes del uso de metodos implicitos de diferencias finitas de orden 2
% N es la cantidad de filas y columnas en el mallado, N*N el numero de
% ecuaciones del sistema
% L, phi, u, D, deltax, delta t son parametros que representan las condiciones
% fisicas del problema modelado por la ecuacion de conveccion-difusion
% (ver Guo y Wang, 2005).
% C es la matriz que contiene el mallado inicial, contemplando condiciones
% iniciales y de frontera
% Elaborado por: Enrique Flores, 2014.

alpha1= -u;
alpha2= D-(phi*u*deltax-u*u*deltat)/(2*phi);
alpha3= -u*D*deltat/phi;
alpha4= D*D*deltat/(2*phi);

k1=(alpha4/(deltax*deltax*deltax*deltax)
    -alpha3/(deltax*deltax*deltax))*(deltat/phi);
k2=(-4*alpha4/(deltax*deltax*deltax*deltax)
    +3*alpha3/(deltax*deltax*deltax)
    +alpha2/(deltax*deltax)-alpha1/(deltax))*(deltat/phi);
k3=(6*alpha4/(deltax*deltax*deltax*deltax)
     -3*alpha3/(deltax*deltax*deltax)-2*alpha2/(deltax*deltax)
     +alpha1/(deltax)-phi/(deltat))*(deltat/phi);
k4=(-4*alpha4/(deltax*deltax*deltax*deltax)
      +alpha3/(deltax*deltax*deltax)+alpha2/(deltax*deltax))*(deltat/phi);
```

```
k5=(alpha4/(deltax*deltax*deltax*deltax))*(deltat/phi);
k6=(-phi/(deltat))*(deltat/phi);

A=zeros(N*N,N*N);
b=zeros(N*N,1);

C=cond_inic_bord_orden_2(N);

i=1;
j=1;


while i <= N
if i==1
A(i,j)=k3;
A(i,j+1)=k4;
A(i,j+2)=k5;
b(i)=k6*C(i+2,1)-k1*C(i,2)-k2*C(i+1,2);

end
        if i==2
A(i,j)=k2;
A(i,j+1)=k3;
A(i,j+2)=k4;
A(i,j+3)=k5;
        b(i)=k6*C(i+2,1)-k1*C(i,2);
end
if(i>=3 & i<=N-2)
A(i,j)=k1;
A(i,j+1)=k2;
A(i,j+2)=k3;
A(i,j+3)=k4;
A(i,j+4)=k5;
        b(i)=k6*C(i+2,1);
        j=j+1;
```

```
end
        if i==N-1
A(i,j)=k1;
A(i,j+1)=k2;
A(i,j+2)=k3;
A(i,j+3)=k4;
b(i)=k6*C(i+2,1)-k5*C(i+4,2);
        j=j+1;

end
        if i==N
A(i,j)=k1;
A(i,j+1)=k2;
A(i,j+2)=k3;
        b(i)=k6*C(i+2,1)-k4*C(i+3,2)-k5*C(i+4,2);
end
        i=i+1;
end



j=1;

for (lambda=1:N)
for (i=N+1:N*N)
    if i==lambda*N+1
A(i,j)=-k6;
A(i,j+N)=k3;
A(i,j+N+1)=k4;
A(i,j+N+2)=k5;
b(i)=-k1*C(1,lambda+2)-k2*C(2,lambda+2);

j=j+1;



end
```

```
        if i==lambda*N+2
A(i,j)=-k6;
A(i,j+N-1)=k2;
A(i,j+N)=k3;
A(i,j+N+1)=k4;
A(i,j+N+2)=k5;
b(i)=-k1*C(2,lambda+2);

j=j+1;



end
if(i>=lambda*N+3) & (i<=lambda*N+N-2)
A(i,j)=-k6;
A(i,j+N-2)=k1;
A(i,j+N-1)=k2;
A(i,j+N)=k3;
A(i,j+N+1)=k4;
        A(i,j+N+2)=k5;
        b(i)=0;

j=j+1;



end
        if i==lambda*N+N-1
A(i,j)=-k6;
A(i,j+N-2)=k1;
A(i,j+N-1)=k2;
A(i,j+N)=k3;
        A(i,j+N+1)=k4;
b(i)=-k5*C(N+3,lambda+2);
    j=j+1;



end
```

```
        if i==lambda*N+N
A(i,j)=-k6;
A(i,j+N-2)=k1;
A(i,j+N-1)=k2;
A(i,j+N)=k3;
        b(i)=-k4*C(N+3,lambda+2)-k5*C(N+4,lambda+2);
j=j+1;


end


end
end

if(det(A)==0)
exit;
end
% FIN genera_sistema_orden_2.m
```

## Código en Octave para la generación de las condiciones iniciales y de frontera en el problema de convección-difusión

```
function [C] = cond_inic_bord_orden_2(N)
% Genera la matriz C que contiene el mallado inicial,
% contemplando condiciones iniciales y de frontera
% provenientes del uso de metodos implicitos de diferencias finitas de orden 2
% N es la cantidad de filas y columnas en el mallado,
% N*N el numero de ecuaciones del sistema
% Elaborado por: Enrique Flores, 2014.

C=zeros(N+4,N+1);



for i=3:N+2
C(i,1)=0;
end

for j=2:N+1
for i=1:2
                if i==1
C(i,j)=1;
                else
                C(i,j)=1;
                end
end
end



for j=3:N+1
for i=N+3:N+4
C(i,j)=0;
end
end
% FIN cond_inic_bord_orden_2.m
```

## Código en Octave para la generación de la matriz de transfomación *wavelet* a partir de la matriz de coeficientes de un sistema de ecuaciones

```
function [W] = genera_matriz_wavelet(A,s,t)
[n,mm] =size(A);
% Genera la matriz W, matriz de transformacion a la base wavelet
% s numero de coeficientes para la famila wavelet: 4,6,8,12.
% t familia wavelet: Daubechies Symlets, Biorthogonal, Reverse Biorthogonal
% Elaborado por: Enrique Flores, 2014.

if s==1
m=4;
end;


if s==2
m=6;
end;


if s==3
m=8;
end;


if s==4
m=12;
end;


switch m
   case 4 % cuatro coeficientes
   if t == 1 % daubechies 2
   h=[-0.12940952255092145 0.22414386804185735
   0.83651630373746899 0.48296291314469025]
```

```
./norm([-0.12940952255092145 0.22414386804185735
0.83651630373746899 0.48296291314469025]);
end
if t == 2 % symlets 2
h=[-0.12940952255092145 0.22414386804185735
0.83651630373746899 0.48296291314469025]
./norm([-0.12940952255092145 0.22414386804185735
0.83651630373746899 0.48296291314469025]);
end

if t == 3 % biorth 3.1
h=[-0.35355339059327379 1.0606601717798214
1.0606601717798214 -0.35355339059327379]
./norm([-0.35355339059327379 1.0606601717798214
1.0606601717798214 -0.35355339059327379]);
end

if t == 4 % reverse biorth 3.1
h=[0.17677669529663689 0.53033008588991071
0.53033008588991071 0.17677669529663689]
./norm([0.17677669529663689 0.53033008588991071
0.53033008588991071 0.17677669529663689]);
end

case 6 % seis coeficientes
if t == 1 % daubechies 3
h=[0.035226291882100656 -0.085441273882241486
-0.13501102001039084 0.45987750211933132
0.80689150931333875 0.33267055295095688]
./norm([0.035226291882100656 -0.085441273882241486
-0.13501102001039084 0.45987750211933132
0.80689150931333875 0.33267055295095688]);
end
if t == 2 % symlets 3
h=[0.035226291882100656 -0.085441273882241486
-0.13501102001039084 0.45987750211933132
```

```
0.80689150931333875 0.33267055295095688]
./norm([0.035226291882100656 -0.085441273882241486
-0.13501102001039084 0.45987750211933132
0.80689150931333875 0.33267055295095688]);
end

if t == 3 % biorth 2.2
h=[0.0 -0.17677669529663689
0.35355339059327379 1.0606601717798214
0.35355339059327379 -0.17677669529663689]
./norm([0.0 -0.17677669529663689
0.35355339059327379 1.0606601717798214
0.35355339059327379 -0.17677669529663689]);
end

if t == 4 % reverse biorth 2.2
h=[0.0 0.0
0.35355339059327379 0.70710678118654757
0.35355339059327379 0.0]
./norm([0.0 0.0
0.35355339059327379 0.70710678118654757
0.35355339059327379 0.0]);
end

case 8 % ocho coeficientes
if t == 1 % daubechies 4
h=[-0.010597401784997278 0.032883011666982945
0.030841381835986965 -0.18703481171888114
-0.027983769416983849 0.63088076792959036
0.71484657055254153 0.23037781330885523]
./norm([-0.010597401784997278 0.032883011666982945
0.030841381835986965 -0.18703481171888114
-0.027983769416983849 0.63088076792959036
0.71484657055254153 0.23037781330885523]);
end
if t == 2 % symlets 4
```

```
    h=[-0.075765714789273325 -0.02963552764599851
    0.49761866763201545 0.80373875180591614
    0.29785779560527736 -0.099219543576847216
    -0.012603967262037833 0.032223100604042702]
    ./norm([-0.075765714789273325 -0.02963552764599851
    0.49761866763201545 0.80373875180591614
    0.29785779560527736 -0.099219543576847216
    -0.012603967262037833 0.032223100604042702]);
    end

    if t == 3 % biorth 3.3
    h=[0.066291260736238838 -0.19887378220871652
    -0.15467960838455727 0.99436891104358249
    0.99436891104358249 -0.15467960838455727
    -0.19887378220871652 0.066291260736238838]
    ./norm([0.066291260736238838 -0.19887378220871652
    -0.15467960838455727 0.99436891104358249
    0.99436891104358249 -0.15467960838455727
    -0.19887378220871652 0.066291260736238838]);
    end

    if t == 4 % reverse biorth 3.3
    h=[0.0 0.0
    0.17677669529663689 0.53033008588991071
    0.53033008588991071 0.17677669529663689
    0.0 0.0]
    ./norm(h=[0.0 0.0
    0.17677669529663689 0.53033008588991071
    0.53033008588991071 0.17677669529663689
    0.0 0.0]);
    end

case 12 % doce coeficientes
    if t == 1 % daubechies 6
    h=[-0.0010773010849955799 0.0047772575110106514
    0.0005538422009938016 -0.031582039318031156
```

```
0.027522865530016288 0.097501605587079362
-0.12976686756709563 -0.22626469396516913
0.3152503517092432 0.75113390802157753
0.49462389039838539 0.11154074335008017]
./norm([-0.0010773010849955799 0.0047772575110106514
0.0005538422009938016 -0.031582039318031156
0.027522865530016288 0.097501605587079362
-0.12976686756709563 -0.22626469396516913
0.3152503517092432 0.75113390802157753
0.49462389039838539 0.11154074335008017]);
end
if t == 2 % symlets 6
h=[0.015404109327027373 0.0034907120842174702
-0.11799011114819057 -0.048311742585632998
 0.49105594192674662 0.787641141030194
 0.3379294217276218 -0.072637522786462516
 -0.021060292512300564 0.044724901770665779
 0.0017677118642428036 -0.007800708325034148]
 ./norm([0.015404109327027373 0.0034907120842174702
 -0.11799011114819057 -0.048311742585632998
 0.49105594192674662 0.787641141030194
 0.3379294217276218 -0.072637522786462516
 -0.021060292512300564 0.044724901770665779
 0.0017677118642428036 -0.007800708325034148]);

end

if t == 3 % biorth 5.5
h=[0.0 0.0
0.03968708834740544 0.0079481086372403219
-0.054463788468236907 0.34560528195603346
0.73666018142821055 0.34560528195603346
-0.054463788468236907 0.0079481086372403219
0.03968708834740544 0.0]
./norm([0.0 0.0
0.03968708834740544 0.0079481086372403219
```

```
   -0.054463788468236907 0.34560528195603346
   0.73666018142821055 0.34560528195603346
   -0.054463788468236907 0.0079481086372403219
   0.03968708834740544 0.0]);
   end

   if t == 4 % reverse biorth 5.5
   h=[0.0 0.013456709459118716
   -0.0026949668801115071 -0.13670658466432914
   -0.093504697400938863 0.47680326579848425
   0.89950610974864842 0.47680326579848425
   -0.093504697400938863 -0.13670658466432914
   -0.0026949668801115071 0.013456709459118716]
   ./norm([0.0 0.013456709459118716
   -0.0026949668801115071 -0.13670658466432914
   -0.093504697400938863 0.47680326579848425
   0.89950610974864842 0.47680326579848425
   -0.093504697400938863 -0.13670658466432914
   -0.0026949668801115071 0.013456709459118716]);
   end


end

w1=zeros(n/2,n);


r=0;
for i=1:n/2
   for k=1:m
      if(k+r<=n)
         w1(i,k+r)=h(k);
      k1=k;
      end
   end
   for q=k1+1:m
```

```
      w1(i,q-k1)=h(q);

   end
r=r+2;
end




g=h(m:-1:1);

for i=2:2:m
g(i)=-1*g(i);
end
g;

w2=zeros(n/2,n);

r=0;
for i=1:n/2
   for k=1:m
      if(k+r<=n)
         w2(i,k+r)=g(k);
      k1=k;
      end
   end
   for q=k1+1:m
      w2(i,q-k1)=g(q);

   end
r=r+2;
end

W=[w1;w2];
% FIN genera_matriz_wavelet.m
```

## Código en Octave para la factorización incompleta ILU(0)

```
function [L,U]=ilu_0(A);
% Genera la descomposicion incompleta ILU(0) de la matriz A
% M = LU sirve de precondicionador de un sistema de ecuaciones lineales
% cuya matriz de coeficientes es A
% Elaborado por: Enrique Flores, 2014.

n = size(A,1); B = A;

for i=2:n
for k=1:(i-1)
if abs(A(i,k))>0
B(i,k) = B(i,k) / B(k,k);
     for j=(k+1):n
     if abs(A(i,j))>0
     B(i,j) = B(i,j) - B(i,k)*B(k,j);
      end
      end
   end
   end

end
  L = tril(B,-1) + speye(n);
  U = triu(B);
% FIN ilu_0.m
```

## Código en Octave del método iterativo BiCG

```
function [x, error, iter, flag] = bicg(A, x, b, M, max_it, tol)

%  -- Iterative template routine --
%     Univ. of Tennessee and Oak Ridge National Laboratory
%     October 1, 1993
%     Details of this algorithm are described in "Templates for the
%     Solution of Linear Systems: Building Blocks for Iterative
%     Methods", Barrett, Berry, Chan, Demmel, Donato, Dongarra,
%     Eijkhout, Pozo, Romine, and van der Vorst, SIAM Publications,
%     1993. (ftp netlib2.cs.utk.edu; cd linalg; get templates.ps).
%
%  [x, error, iter, flag] = bicg(A, x, b, M, max_it, tol)
%
% bicg.m solves the linear system Ax=b using the
% BiConjugate Gradient Method with preconditioning.
%
% input   A        REAL matrix
%         M        REAL preconditioner matrix
%         x        REAL initial guess vector
%         b        REAL right hand side vector
%         max_it   INTEGER maximum number of iterations
%         tol      REAL error tolerance
%
% output  x        REAL solution vector
%         error    REAL error norm
%         iter     INTEGER number of iterations performed
%         flag     INTEGER: 0 = solution found to tolerance
%                           1 = no convergence given max_it
%                          -1 = breakdown

  iter = 0;                              % initialization
  flag = 0;
```

```
bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end

r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

r_tld  = r;

for iter = 1:max_it                     % begin iteration

   z = M \ r;
   z_tld = M' \ r_tld;
   rho   = ( z'*r_tld );
   if ( rho == 0.0 ),
      break
   end

   if ( iter > 1 ),                     % direction vectors
      beta = rho / rho_1;
      p   = z  + beta*p;
      p_tld = z_tld + beta*p_tld;
   else
      p = z;
      p_tld = z_tld;
   end

   q = A*p;                             % compute residual pair
   q_tld = A'*p_tld;
   alpha = rho / (p_tld'*q );

   x = x + alpha*p;                     % update approximation
   r = r - alpha*q;
   r_tld = r_tld - alpha*q_tld;

   error = norm( r ) / bnrm2;           % check convergence
```

```
      if ( error <= tol ), break, end

      rho_1 = rho;

   end

   if ( error <= tol ),                    % converged
      flag =  0;
   elseif ( rho == 0.0 ),                  % breakdown
      flag = -1;
   else
      flag = 1;                            % no convergence
   end

% END bicg.m
```

## Código en Octave del método iterativo BiCGStab

```
function [x, error, iter, flag] = bicgstab(A, x, b, M, max_it, tol)

%  -- Iterative template routine --
%     Univ. of Tennessee and Oak Ridge National Laboratory
%     October 1, 1993
%     Details of this algorithm are described in "Templates for the
%     Solution of Linear Systems: Building Blocks for Iterative
%     Methods", Barrett, Berry, Chan, Demmel, Donato, Dongarra,
%     Eijkhout, Pozo, Romine, and van der Vorst, SIAM Publications,
%     1993. (ftp netlib2.cs.utk.edu; cd linalg; get templates.ps).
%
%  [x, error, iter, flag] = bicgstab(A, x, b, M, max_it, tol)
%
% bicgstab.m solves the linear system Ax=b using the
% BiConjugate Gradient Stabilized Method with preconditioning.
%
% input   A        REAL matrix
%         x        REAL initial guess vector
%         b        REAL right hand side vector
%         M        REAL preconditioner matrix
%         max_it   INTEGER maximum number of iterations
%         tol      REAL error tolerance
%
% output  x        REAL solution vector
%         error    REAL error norm
%         iter     INTEGER number of iterations performed
%         flag     INTEGER: 0 = solution found to tolerance
%                           1 = no convergence given max_it
%                          -1 = breakdown: rho = 0
%                          -2 = breakdown: omega = 0

  iter = 0;                                           % initialization
  flag = 0;
```

```
bnrm2 = norm( b );
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end

r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

omega  = 1.0;
r_tld = r;

for iter = 1:max_it,                                % begin iteration

   rho   = ( r_tld'*r );                            % direction vector
   if ( rho == 0.0 ) break, end

   if ( iter > 1 ),
      beta  = ( rho/rho_1 )*( alpha/omega );
      p = r + beta*( p - omega*v );
   else
      p = r;
   end

   p_hat = M \ p;
   v = A*p_hat;
   alpha = rho / ( r_tld'*v );
   s = r - alpha*v;
   if ( norm(s) < tol ),                            % early convergence check
      x = x + alpha*p_hat;
      resid = norm( s ) / bnrm2;
      break;
   end

   s_hat = M \ s;                                   % stabilizer
   t = A*s_hat;
   omega = ( t'*s) / ( t'*t );
```

```
      x = x + alpha*p_hat + omega*s_hat;             % update approximation

      r = s - omega*t;
      error = norm( r ) / bnrm2;                     % check convergence
      if ( error <= tol ), break, end

      if ( omega == 0.0 ), break, end
      rho_1 = rho;

   end

   if ( error <= tol | s <= tol ),                   % converged
      if ( s <= tol ),
         error = norm(s) / bnrm2;
      end
      flag =  0;
   elseif ( omega == 0.0 ),                          % breakdown
      flag = -2;
   elseif ( rho == 0.0 ),
      flag = -1;
   else                                              % no convergence
      flag = 1;
   end

% END bicgstab.m
```

## Código en Octave del método iterativo CGS

```
function [x, error, iter, flag] = cgs(A, x, b, M, max_it, tol)

%  -- Iterative template routine --
%     Univ. of Tennessee and Oak Ridge National Laboratory
%     October 1, 1993
%     Details of this algorithm are described in "Templates for the
%     Solution of Linear Systems: Building Blocks for Iterative
%     Methods", Barrett, Berry, Chan, Demmel, Donato, Dongarra,
%     Eijkhout, Pozo, Romine, and van der Vorst, SIAM Publications,
%     1993. (ftp netlib2.cs.utk.edu; cd linalg; get templates.ps).
%
%  [x, error, iter, flag] = cgs(A, x, b, M, max_it, tol)
%
% cgs.m solves the linear system Ax=b using the
% Conjugate Gradient Squared Method with preconditioning.
%
% input   A        REAL matrix
%         x        REAL initial guess vector
%         b        REAL right hand side vector
%         M        REAL preconditioner
%         max_it   INTEGER maximum number of iterations
%         tol      REAL error tolerance
%
% output  x        REAL solution vector
%         error    REAL error norm
%         iter     INTEGER number of iterations performed
%         flag     INTEGER: 0 = solution found to tolerance
%                           1 = no convergence given max_it

  iter = 0;                                % initialization
  flag = 0;

  bnrm2 = norm( b );
```

```
if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end

r = b - A*x;
error = norm( r ) / bnrm2;
if ( error < tol ) return, end

r_tld = r;

for iter = 1:max_it,                    % begin iteration

   rho = (r_tld'*r );
   if (rho == 0.0), break, end

   if ( iter > 1 ),                     % direction vectors
      beta = rho / rho_1;
      u = r + beta*q;
      p = u + beta*( q + beta*p );
   else
      u = r;
      p = u;
   end

   p_hat = M \ p;
   v_hat = A*p_hat;                     % adjusting scalars
   alpha = rho / ( r_tld'*v_hat );
   q = u - alpha*v_hat;
   u_hat = M \ (u+q);

   x = x + alpha*u_hat;                 % update approximation

   r = r - alpha*A*u_hat;
   error = norm( r ) / bnrm2;           % check convergence
   if ( error <= tol ), break, end

   rho_1 = rho;
```

```
  end

  if (error <= tol),                    % converged
     flag =  0;
  elseif ( rho == 0.0 ),                % breakdown
     flag = -1;
  else                                  % no convergence
     flag = 1;
  end

% END cgs.m
```

## Código en Octave del método iterativo QMR

```
function [x, error, iter, flag] = qmr( A, x, b, M, max_it, tol )

%  -- Iterative template routine --
%     Univ. of Tennessee and Oak Ridge National Laboratory
%     October 1, 1993
%     Details of this algorithm are described in "Templates for the
%     Solution of Linear Systems: Building Blocks for Iterative
%     Methods", Barrett, Berry, Chan, Demmel, Donato, Dongarra,
%     Eijkhout, Pozo, Romine, and van der Vorst, SIAM Publications,
%     1993. (ftp netlib2.cs.utk.edu; cd linalg; get templates.ps).
%
%  [x, error, iter, flag] = qmr( A, x, b, M, max_it, tol )
%
% qmr.m solves the linear system Ax=b using the
% Quasi Minimal Residual Method with preconditioning.
%
% input   A        REAL matrix
%         x        REAL initial guess vector
%         b        REAL right hand side vector
%         M        REAL preconditioner
%         max_it   INTEGER maximum number of iterations
%         tol      REAL error tolerance
%
% output  x        REAL solution vector
%         error    REAL error norm
%         iter     INTEGER number of iterations performed
%         flag     INTEGER: 0: solution found to tolerance
%                           1: no convergence given max_it
%                     breakdown:
%                          -1: rho
%                          -2: beta
%                          -3: gamma
%                          -4: delta
```

```
%                          -5: ep
%                          -6: xi

   iter = 0;                                      % initialization
   flag = 0;

   bnrm2 = norm( b );
   if  ( bnrm2 == 0.0 ), bnrm2 = 1.0; end

   r = b - A*x;
   error = norm( r ) / bnrm2;
   if ( error < tol ) return, end

   [M1,M2] = lu( M );

   v_tld = r;
   y = M1 \ v_tld;
   rho = norm( y );

   w_tld = r;
   z = M2' \ w_tld;
   xi = norm( z );

   gamma =  1.0;
   eta = -1.0;
   theta =  0.0;

   for iter = 1:max_it,                        % begin iteration

      if ( rho == 0.0 | xi == 0.0 ), break, end

      v = v_tld / rho;
      y = y / rho;

      w = w_tld / xi;
      z = z / xi;
```

```
delta = z'*y;
if ( delta == 0.0 ), break, end

y_tld = M2 \ y;
z_tld = M1'\ z;

if ( iter > 1 ),                        % direction vector
   p = y_tld - ( xi*delta / ep )*p;
   q = z_tld - ( rho*delta / ep )*q;
      else
   p = y_tld;
   q = z_tld;
end

p_tld = A*p;
ep = q'*p_tld;
if ( ep == 0.0 ), break, end

beta = ep / delta;
if ( beta == 0.0 ), break, end

v_tld = p_tld - beta*v;
y =  M1 \ v_tld;

rho_1 = rho;
rho = norm( y );
w_tld = ( A'*q ) - ( beta*w );
z =  M2' \ w_tld;

xi = norm( z );

gamma_1 = gamma;
theta_1 = theta;

theta = rho / ( gamma_1*beta );
```

```
   gamma = 1.0 / sqrt( 1.0 + (theta^2) );
   if ( gamma == 0.0 ), break, end

   eta = -eta*rho_1*(gamma^2) / ( beta*(gamma_1^2) );

   if ( iter > 1 ),                          % compute adjustment
      d = eta*p + (( theta_1*gamma )^2)*d;
      s = eta*p_tld + (( theta_1*gamma )^2)*s;
   else
      d = eta*p;
      s = eta*p_tld;
   end

   x = x + d;                                % update approximation

   r = r - s;                                % update residual
   error = norm( r ) / bnrm2;                % check convergence
   if ( error <= tol ), break, end

end

if ( error <= tol ),                         % converged
   flag =  0;
elseif ( rho == 0.0 ),                       % breakdown
   flag = -1;
elseif ( beta == 0.0 ),
   flag = -2;
elseif ( gamma == 0.0 ),
   flag = -3;
elseif ( delta == 0.0 ),
   flag = -4;
elseif ( ep == 0.0 ),
   flag = -5;
elseif ( xi == 0.0 ),
   flag = -6;
else                                         % no convergence
```

```
      flag = 1;
   end

%  END qmr.m
```