

Un algoritmo tipo Newton globalizado para resolver la ecuación cuadrática matricial

MAURICIO MACÍAS^{a*}, HÉCTOR J. MARTÍNEZ^b, ROSANA PÉREZ^a

^a Universidad del Cauca, Departamento de Matemáticas, Popayán, Colombia.

^b Universidad del Valle, Departamento de Matemáticas, Cali, Colombia.

Resumen. En este artículo se presenta una globalización del algoritmo cuasi-Newton local propuesto en [16] para resolver la ecuación cuadrática matricial. Se demuestra que la estrategia de globalización usada no interfiere en la tasa de convergencia del algoritmo cuasi-Newton. Pruebas numéricas muestran un buen desempeño del algoritmo global propuesto.

Palabras clave: Función matricial, método de Newton, ecuaciones matriciales no lineales, convergencia.

MSC2010: 15A24, 39B42, 49M15, 65H20, 90C53.

A globalized Newton type algorithm to solve the matrix quadratic equation

Abstract. In this paper we propose a globalized quasi-Newton algorithm to solve the quadratic matrix equation. It is shown that the globalization strategy used does not interfere in the convergence of the local quasi-Newton algorithm. We present some numerical tests that show the good performance of the global quasi-Newton algorithm.

Keywords: Matrix function, secant method, nonlinear matrix equations, superlinear convergence.

1. Introducción

En [16] los autores proponen un algoritmo cuasi-Newton local y hasta cuadráticamente convergente, para resolver la ecuación cuadrática matricial

$$Q(\mathbf{X}) = \mathbf{A}\mathbf{X}^2 + \mathbf{B}\mathbf{X} + \mathbf{C} = \mathbf{0}, \quad (1)$$

*E-mail: mauromac@unicauca.edu.co

Recibido: 4 de mayo de 2018, Aceptado: 27 de agosto de 2018.

Para citar este artículo: M. Macías, H.J. Martínez, R. Pérez, Un algoritmo tipo Newton globalizado para resolver la ecuación cuadrática matricial, *Rev. Integr. temas mat.* 36 (2018), No. 2, 117–132. doi: 10.18273/revint.v36n2-2018004.

donde $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{C}^{n \times n}$, $\mathbf{0}$ es la matriz cero en $\mathbb{C}^{n \times n}$ y Q es una función cuadrática matricial, el cual reduce el costo computacional del método de *Newton-Schur* tradicionalmente usado para resolver dicha ecuación ([10], [15]).

Las numerosas aplicaciones en las cuales surge la ecuación matricial (1) han motivado un gran interés en su estudio y el desarrollo de algoritmos para su solución. Áreas en las cuales surge la ecuación cuadrática matricial son el análisis dinámico de mecánica estructural y sistemas acústicos ([1], [22]), simulación de circuitos eléctricos [14], mecánica de fluidos [21], procesamiento de señales ([4], [11]), problemas de vibración y modelamiento de sistemas mecánicos microelectrónicos ([3], [2], [13]).

El algoritmo propuesto en [16], por ser un algoritmo local, es ventajoso siempre que la matriz inicial esté suficientemente cerca de la solución, pero pierde su efectividad en caso contrario. Una alternativa a esta situación es introducir en este algoritmo una estrategia de globalización, como por ejemplo una búsqueda lineal ([6], [20]).

En la literatura sobre algoritmos tipo *Newton* globalizados para resolver la ecuación (1), se destaca en primer lugar, por su amplio uso, el algoritmo de *Newton-Schur* globalizado propuesto en [10], donde se introduce una estrategia de globalización análoga a la búsqueda lineal exacta usada en algoritmos de optimización de funciones vectoriales [6]. En segundo lugar, se menciona el algoritmo propuesto en [15], que modifica el algoritmo propuesto en [10] controlando la cercanía entre el paso de *Newton* y el solvente o solución de la ecuación matricial. Para ello, se compara el tamaño del valor de la función en el paso de *Newton* (en alguna norma) con cero, ya que en este caso no es necesario hacer una búsqueda lineal, y como consecuencia el algoritmo resultante es menos costoso computacionalmente.

Teniendo en cuenta las ventajas de un algoritmo global y las bondades del algoritmo cuasi-*Newton* local propuesto en [16], en el presente artículo se propone una globalización de este algoritmo cuasi-Newton para resolver la ecuación cuadrática matricial (1). Para ello, se incorpora al algoritmo cuasi-Newton local una estrategia de *búsqueda lineal exacta* análoga a la usada en [10] y una modificación de la alternativa usada en [15] para evitar la búsqueda lineal cerca de la solución. Adicionalmente, se demuestra que la estrategia de búsqueda lineal exacta no afecta la convergencia hasta de orden cuadrático obtenida en [16]. También se analiza el desempeño numérico del algoritmo propuesto con el algoritmo de *Newton-Schur* frecuentemente usado para resolver la ecuación cuadrática. Además, para apreciar la ventaja de tener un algoritmo global frente a uno local, se compara su desempeño con el algoritmo propuesto en [16].

Organizamos este artículo en la siguiente forma. En la Sección 2 se presenta en forma descriptiva la estrategia de globalización de *búsqueda lineal exacta* en el contexto de la función cuadrática matricial. En la Sección 3 se propone un algoritmo cuasi-Newton global para resolver la ecuación (1). En la Sección 4, bajo ciertas hipótesis, se demuestra que la búsqueda lineal no afecta la tasa de convergencia del algoritmo cuasi-Newton global. En la Sección 5 se analiza numéricamente el comportamiento global del algoritmo propuesto. Finalmente, en la Sección 6 se presentan algunos comentarios finales y propuestas de trabajos futuros sobre el tema.

2. Una estrategia de búsqueda lineal exacta

En el caso de un campo vectorial $H: \mathbb{R}^n \rightarrow \mathbb{R}^n$, una forma de globalizar el método de *Newton* para resolver un sistema de ecuaciones no lineales, $H(\mathbf{x}) = 0$, consiste en combinarlo con una estrategia global para optimización irrestricta ([6], [20]). En efecto, independientemente de que el vector \mathbf{x} esté o no próximo a la solución \mathbf{x}_* del sistema, una forma razonable de aceptar la iteración siguiente \mathbf{x}_{k+1} es exigir que $\|H(\mathbf{x}_{k+1})\|$ sea suficientemente menor que $\|H(\mathbf{x}_k)\|$ para alguna norma conveniente. Exigencia que también es hecha cuando se trata de minimizar $\|H(\mathbf{x})\|$, con lo cual se llega a un problema de minimización.

Extendiendo estas ideas al caso de la función matricial $Q: \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$, donde $Q(\mathbf{X}) = \mathbf{A}\mathbf{X}^2 + \mathbf{B}\mathbf{X} + \mathbf{C}$, se tiene que, si la matriz inicial \mathbf{X}_0 no está próxima de la solución \mathbf{X}_* del sistema matricial $Q(\mathbf{X}) = \mathbf{0}$ y \mathbf{S}_k es el paso de Newton, entonces en el algoritmo global se busca una aproximación siguiente de la forma $\mathbf{X}_{k+1} = \mathbf{X}_k + t\mathbf{S}_k$ para algún valor de t convenientemente elegido, de manera que \mathbf{X}_{k+1} sea una aproximación aceptable; en este sentido, buscamos un valor de t tal que $\|Q(\mathbf{X}_{k+1})\|$ sea menor que $\|Q(\mathbf{X}_k)\|$ para alguna norma, lo cual conduce a minimizar la función $\|Q(\mathbf{X}_k + t\mathbf{S}_k)\|$ para alguna norma conveniente. Este problema es equivalente a minimizar $g(t) = \|Q(\mathbf{X}_k + t\mathbf{S}_k)\|_F^2$.

Por conveniencia para el problema matricial, en [10] se considera la norma de Frobenius en la definición de la función g , ya que $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^* \mathbf{A})$, para cualquier $\mathbf{A} \in \mathbb{C}^{n \times n}$. En efecto, $g(t) = \|Q(\mathbf{X}_k + t\mathbf{S}_k)\|_F^2$ es el polinomio de cuarto grado en la variable t

$$g(t) = \gamma t^4 - \beta t^3 + (\alpha + \beta)t^2 - 2\alpha t + \alpha, \quad (2)$$

donde

$$\gamma = \|\mathbf{A}\mathbf{S}_k^2\|_F^2, \quad \alpha = \|Q(\mathbf{X}_k)\|_F^2, \quad \beta = \text{tr}(Q(\mathbf{X}_k)^* \mathbf{A}\mathbf{S}_k^2 + (\mathbf{A}\mathbf{S}_k^2)^* Q(\mathbf{X}_k)).$$

Analicemos el polinomio $g(t)$ dado por la ecuación (2)

1. Si $\gamma = 0$ entonces $\mathbf{A}\mathbf{S}_k^2 = \mathbf{0}$ y $\beta = 0$; en consecuencia, $g(t) = \alpha(t - 1)^2$. En este caso, $g(t)$ alcanza su mínimo global en $t = 1$, con lo cual \mathbf{X}_{k+1} es el paso de Newton estándar.
2. Si $\alpha = 0$ entonces \mathbf{X}_k es un solvente de $Q(\mathbf{X}) = \mathbf{0}$.

Se puede suponer que $\gamma > 0$ y $\alpha > 0$, con lo cual $g(t)$ es un polinomio de cuarto grado que tiene a lo más dos mínimos, de los cuales uno es el global. Las raíces de $g'(t)$ que corresponden a los puntos críticos de g son las de la ecuación

$$g'(t) = 4\gamma t^3 - 3\beta t^2 + 2(\alpha + \beta)t - 2\alpha = 0. \quad (3)$$

Para determinar el número de posibles raíces reales positivas y negativas de $g'(t)$, se analizan, siguiendo la *Regla de signos de Descartes*, los cambios de signo de los coeficientes del polinomio $g'(t)$ y los cambios de signo de los coeficientes del polinomio $g'(-t)$, respectivamente.

1. Si $\beta > 0$ entonces los coeficientes del polinomio $g'(t)$ tienen 3 cambios de signo, es decir, $g'(t)$ tiene una o tres raíces reales positivas. Por otro lado, los coeficientes del polinomio $g'(-t)$ no tienen cambios de signo, es decir, el polinomio $g'(t)$ no tiene raíces reales negativas. Por lo tanto, los puntos críticos del polinomio $g(t)$ son positivos (Figura 1).

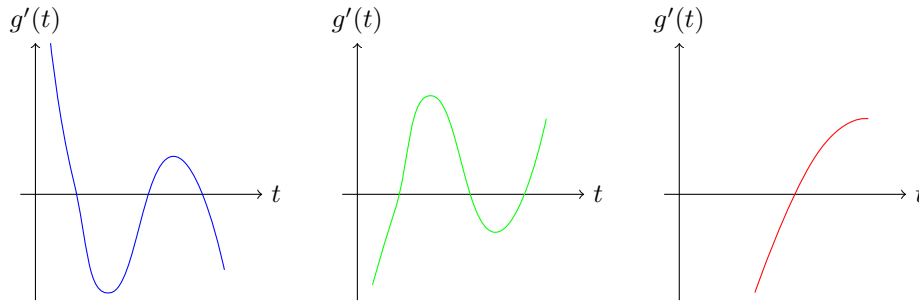


Figura 1. Polinomio $g'(t)$ para $\beta > 0$.

2. Si $\beta = 0$ entonces los coeficientes del polinomio $g'(t)$ tienen un cambio de signo, es decir, $g'(t)$ tiene una raíz real positiva. Por otro lado, los coeficientes del polinomio $g'(-t)$ no tienen cambios de signo, es decir, el polinomio $g'(t)$ no tiene raíces reales negativas. Por lo tanto, el polinomio $g(t)$ tiene un único punto crítico real positivo (Figura 2).

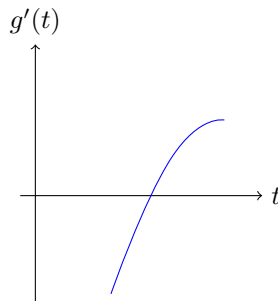


Figura 2. Polinomio $g'(t)$ para $\beta = 0$.

3. Si $\beta < 0$, debemos tener en cuenta el signo de $\alpha + \beta$. Cálculos sencillos permiten verificar que independientemente del signo de $\alpha + \beta$, los coeficientes del polinomio $g'(t)$ tienen un cambio de signo (una raíz real positiva), y los del polinomio $g'(-t)$ tienen 2 cambios de signo (2 raíces reales negativas o ninguna). (Ver Figura 3).

Por otro lado, evaluando g' en 0 y en 2, se tiene

$$g'(0) = -2\alpha < 0, \quad (4)$$

$$g'(2) = 2(16\gamma - 4\beta + \alpha) = 2 \operatorname{tr} [(Q(\mathbf{X}_k) - 4\mathbf{A}\mathbf{S}_k^2)^*(Q(\mathbf{X}_k) - 4\mathbf{A}\mathbf{S}_k^2)] \geq 0. \quad (5)$$

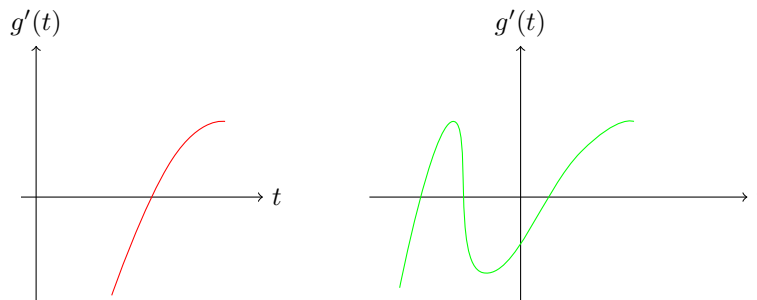


Figura 3. Polinomio $g'(t)$ para $\beta < 0$.

De ahí que $g'(t)$ tiene al menos un cero en $(0, 2]$. Para verificar que en efecto este cero es un minimizador de g , se consideran los siguientes casos para el polinomio $g'(t)$.

1. El polinomio $g'(t)$ tiene un cero real y un par de ceros complejos conjugados; en este caso, tal cero pertenece al intervalo $(0, 2]$ y corresponde al minimizador global del polinomio $g(t)$. En la Figura 4, se ilustra este caso.

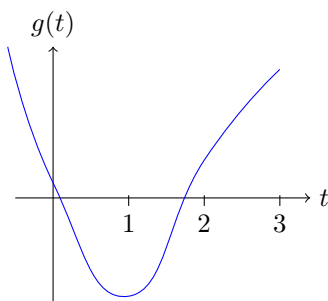


Figura 4. Polinomio $g(t)$ cuando $g'(t)$ tiene un cero real y un par de ceros complejos conjugados.

2. El polinomio $g'(t)$ tiene 3 ceros reales y, por el cálculo diferencial tendríamos que dos de ellos corresponden a minimizadores de g y el otro es un maximizador. Se debe verificar que por lo menos uno de los minimizadores pertenezca al intervalo $(0, 2]$. Una vez hallados y ordenados los ceros de $g'(t)$, se concluye a partir de (4) y (5), que el primero y el último corresponden a los minimizadores de $g(t)$; si estos minimizadores pertenecen al intervalo $(0, 2]$ (ver Figura 5), entonces el minimizador buscado será aquel donde g alcance su menor valor. En caso contrario, es decir, si el tercer cero es mayor que 2 (ver Figura 6), entonces se tiene que el minimizador buscado será el correspondiente al primer cero de $g'(t)$, ya que este pertenece al intervalo $(0, 2]$.

Por lo anterior, el valor óptimo de t en la búsqueda lineal corresponde al minimizador de la función g , y su cálculo implica resolver el siguiente problema de minimización exacta:

$$\min_{t \in (0, 2]} g(t) \tag{6}$$

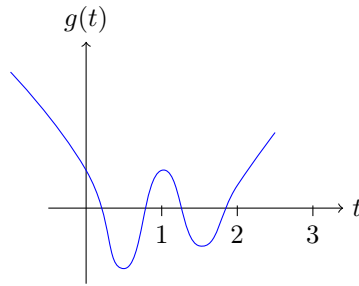


Figura 5. Polinomio $g(t)$ cuando $g'(t)$ tiene tres ceros reales en $(0, 2]$.

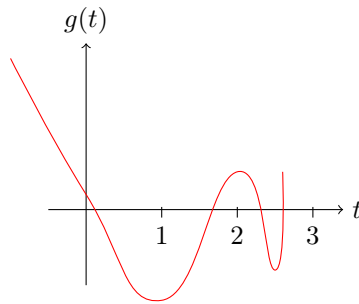


Figura 6. Polinomio $g(t)$ cuando $g'(t)$ tiene tres ceros reales en $(0, 2]$.

Observación: es importante notar que este tipo de búsqueda lineal exacta requeriría, en el peor de los casos, de 3 evaluaciones de la función g , lo cual marca una gran diferencia con la estrategia de búsqueda lineal inexacta [6] que requeriría de muchas más evaluaciones para encontrar un valor de t aceptable. Esto justifica el por qué usar una búsqueda lineal exacta en lugar de una inexacta.

3. Propuesta algorítmica

Como alternativa al alto costo computacional del método *Newton-Schur*, tradicionalmente usado para resolver la ecuación (1), se propone en [16] un algoritmo cuasi-Newton local cuya iteración básica, a partir de una matriz inicial \mathbf{X}_0 , es la siguiente:

$$\begin{aligned} (2\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k &= -Q(\mathbf{X}_k), \\ \mathbf{X}_{k+1} &= \mathbf{X}_k + \mathbf{S}_k. \end{aligned} \quad (7)$$

Además, en [16], se demuestra que este algoritmo es local y hasta cuadráticamente convergente. Para mayor claridad en la lectura de este documento, en particular del Teorema 4.1, listamos a continuación dichas hipótesis:

H1. $Q : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ es Fréchet diferenciable en un conjunto $D \subset \mathbb{C}^{n \times n}$ abierto y convexo.

H2. Existe $\mathbf{X}_* \in D$ tal que $Q(\mathbf{X}_*) = \mathbf{0}$.

H3. La matriz $2\mathbf{A}\mathbf{X}_* + \mathbf{B}$ es no singular y $\bar{\beta}$ es la norma de su inversa.

H4. Existe $\theta \in [0, 1]$ tal que

$$\lim_{\|\mathbf{S}\| \rightarrow 0} \frac{\|L_{\mathbf{X}_*}(\mathbf{S}) - (2\mathbf{A}\mathbf{X}_* + \mathbf{B})\mathbf{S}\|}{\|\mathbf{S}\|^{1+\theta}} = 0.$$

En esta sección se presenta una globalización del algoritmo anterior. Para ello, se incorpora a dicho algoritmo una estrategia de búsqueda lineal exacta análoga a la usada en [10], la cual se describe detalladamente en la sección anterior. Además, se hace una modificación de la alternativa usada en [15] para evitar la búsqueda lineal cerca de la solución.

Algoritmo 1 Método cuasi-Newton global

Entrada: $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{X}_0, \epsilon, \epsilon_1 = \sqrt{\epsilon}, N_{max}, y k = 0$.

Salida: Matriz aproximación a \mathbf{X}_* .

- 1: **mientras** $k \leq N_{max}, Res(\mathbf{X}_k) \geq \epsilon$, **hacer**
- 2: Resolver, para \mathbf{S}_k , el sistema matricial

$$(2\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k = -Q(\mathbf{X}_k).$$

- 3: **si** $Res(\mathbf{X}_k) < \epsilon_1$, **entonces**
- 4: $t = 1$,
- 5: **si no**,
- 6: Búsqueda lineal exacta: encontrar t tal que

$$\|Q(\mathbf{X}_k + t\mathbf{S}_k)\|_F^2 = \min_{p \in (0,2)} \|Q(\mathbf{X}_k + p\mathbf{S}_k)\|_F^2;$$

- 7: **fin si**:
 - 8: Actualizar $\mathbf{X}_{k+1} = \mathbf{X}_k + t\mathbf{S}_k$ y $k = k + 1$.
 - 9: **fin mientras**:
 - 10: **Salida** X_*
-

En el **Algoritmo 1** la expresión $Res(\mathbf{X}_k)$ es una medida relativa del error en $\|Q(\mathbf{X}_k)\|_F$, y es definida como en [10] por

$$Res(\mathbf{X}_k) = \frac{\|Q(\mathbf{X}_k)\|_F}{\|\mathbf{A}\|_F \|\mathbf{X}_k\|_F^2 + \|\mathbf{B}\|_F \|\mathbf{X}_k\|_F + \|\mathbf{C}\|_F}. \tag{8}$$

La condición en el Paso 3 es análoga a la usada en [15], y permite decidir en qué caso no es necesario hacer búsqueda lineal.

4. Análisis de convergencia

En esta sección se demuestra que la estrategia de búsqueda lineal exacta no afecta la tasa de convergencia del método *cuasi-Newton* propuesto en [16].

Teorema 4.1 (Convergencia). *Si $Q: \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ es una función cuadrática matricial que satisface las hipótesis **H1** a **H4**, y $\{\mathbf{X}_k\}$ es la sucesión generada por el método *cuasi-Newton* local (7), entonces la sucesión generada por*

$$\begin{aligned} (2\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k &= -Q(\mathbf{X}_k), \\ \overline{\mathbf{X}_{k+1}} &= \mathbf{X}_k + t\mathbf{S}_k, \end{aligned} \quad (9)$$

también converge a \mathbf{X}_* con orden $1 + \theta$, $\theta \in [0, 1]$,

Demostración. Por hipótesis y por Teorema 3.2 de [16], para $\mu > 0$ y $\theta \in [0, 1]$, se satisface que $\|\mathbf{X}_{k+1} - \mathbf{X}_*\| < \mu \|\mathbf{X}_k - \mathbf{X}_*\|^{1+\theta}$. Equivalentemente,

$$\|\Delta_{k+1}\| = O\left(\|\Delta_k\|^{1+\theta}\right), \quad (10)$$

donde $\Delta_k = \mathbf{X}_k - \mathbf{X}_*$. Debido a que \mathbf{X}_k está en la región de convergencia de \mathbf{X}_* entonces se satisface que

$$\|\Delta_k\|^{1+\theta} < \|\Delta_k\| \quad (11)$$

para cualquier $\theta \in (0, 1]$. Además, $\mathbf{S}_k = \mathbf{X}_{k+1} - \mathbf{X}_k = \Delta_{k+1} - \Delta_k$. A partir de esta igualdad, de (10) y (11) se tiene

$$\|\mathbf{S}_k\| = O(\|\Delta_k\|), \quad (12)$$

$Q(\mathbf{X} + \mathbf{S}) = Q(\mathbf{X}) + L_{\mathbf{X}}(\mathbf{S}) + \mathbf{A}\mathbf{S}^2$, con lo cual,

$$\begin{aligned} Q(\mathbf{X}_{k+1}) &= Q(\mathbf{X}_* + \Delta_{k+1}) = Q(\mathbf{X}_*) + L_{\mathbf{X}_*}(\Delta_{k+1}) + O\left(\|\Delta_{k+1}\|^2\right) \\ &= L_{\mathbf{X}_*}(\Delta_{k+1}) + O\left(\|\Delta_{k+1}\|^2\right). \end{aligned} \quad (13)$$

De $L_{\mathbf{X}_*}(\Delta_{k+1}) = \mathbf{A}\Delta_{k+1}\mathbf{X}_* + (\mathbf{A}\mathbf{X}_* + \mathbf{B})\Delta_{k+1}$ se tiene $\|L_{\mathbf{X}_*}(\Delta_{k+1})\| = O(\|\Delta_{k+1}\|)$, y por (13),

$$\|Q(\mathbf{X}_{k+1})\| = O(\|\Delta_{k+1}\|). \quad (14)$$

En forma análoga,

$$\|Q(\mathbf{X}_k)\| = O(\|\Delta_k\|). \quad (15)$$

Por otro lado, se sabe que en la búsqueda lineal exacta el tamaño de paso t es el minimizador global del polinomio

$$p(t) = \|Q(\mathbf{X}_k + t\mathbf{S}_k)\|^2 = \|(1-t)Q(\mathbf{X}_k) + t^2\mathbf{A}\mathbf{S}_k^2\|^2; \quad (16)$$

entonces,

$$\|Q(\mathbf{X}_k + t\mathbf{S}_k)\| \leq \|Q(\mathbf{X}_k + \mathbf{S}_k)\| = \|Q(\mathbf{X}_{k+1})\|. \quad (17)$$

Usando (10), (13), (14) y (17) en (16), se tiene

$$\|(1-t)Q(\mathbf{X}_k) + t^2\mathbf{A}\mathbf{S}_k^2\| = \|Q(\mathbf{X}_k + t\mathbf{S}_k)\| \leq \|Q(\mathbf{X}_{k+1})\| = O\left(\|\Delta_k\|^{1+\theta}\right).$$

Por lo tanto,

$$\|(1-t)Q(\mathbf{X}_k) + t^2\mathbf{A}\mathbf{S}_k^2\| = O\left(\|\Delta_k\|^{1+\theta}\right). \quad (18)$$

Usando (18), la definición de O grande, (12) y propiedades de la norma matricial, se tiene que existen constantes positivas γ_1 , γ_2 , y $\gamma_3 = t^2\gamma_2\|\mathbf{A}\|$, tales que

$$\begin{aligned} |1-t|\|Q(\mathbf{X}_k)\| - t^2\|\mathbf{A}\mathbf{S}_k^2\| &\leq \gamma_1\|\Delta_k\|^{1+\theta} \\ |1-t|\|Q(\mathbf{X}_k)\| &\leq \gamma_1\|\Delta_k\|^{1+\theta} + t^2\|\mathbf{A}\|\gamma_2\|\Delta_k\|^2 \\ &\leq \gamma_1\|\Delta_k\|^{1+\theta} + \gamma_3\|\Delta_k\|^2 \\ &\leq \gamma\|\Delta_k\|^{1+\theta}, \end{aligned} \quad (19)$$

donde γ es una constante positiva. De (15),

$$\|Q(\mathbf{X}_k)\| \leq \rho\|\Delta_k\|, \quad (20)$$

para alguna constante ρ .

Como la derivada de Fréchet de Q es no singular en la solución \mathbf{X}_* , entonces $\|Q(\mathbf{X}_k)\| \neq 0$. Tomando recíprocos en (20), se obtiene

$$\frac{1}{\|Q(\mathbf{X}_k)\|} \geq \frac{1}{\rho\|\Delta_k\|}.$$

Multiplicando (19) por $\frac{1}{\rho\|\Delta_k\|}$, se tiene

$$|1-t|\beta \leq \frac{\gamma}{\rho}\|\Delta_k\|^\theta,$$

donde $\beta = \frac{\|Q(\mathbf{X}_k)\|}{\rho\|\Delta_k\|} \leq 1$. Luego $|1-t| \leq \gamma_4\left(\|\Delta_k\|^\theta\right)$, con $\gamma_4 = \frac{\gamma}{\beta\rho}$.

Por lo tanto,

$$|1-t| = O\left(\|\Delta_k\|^\theta\right). \quad (21)$$

Luego, usando las actualizaciones (7) y (9), se tiene

$$\overline{\mathbf{X}}_{k+1} - \mathbf{X}_* = \overline{\mathbf{X}}_{k+1} - \mathbf{X}_{k+1} + \mathbf{X}_{k+1} - \mathbf{X}_* = (1-t)\mathbf{S}_k + \Delta_{k+1}.$$

Aplicando la desigualdad triangular y usando (10), (12) y (21), se obtiene

$$\|\overline{\mathbf{X}}_{k+1} - \mathbf{X}_*\| \leq |1-t|\|\mathbf{S}_k\| + \|\Delta_{k+1}\| = O\left(\|\Delta_k\|^{1+\theta}\right).$$

Por lo tanto, la sucesión $\{\overline{\mathbf{X}}_k\}$ converge con orden $1 + \theta$ a \mathbf{X}_* . \square

5. Pruebas numéricas

En esta sección se analiza numéricamente el comportamiento del Algoritmo 1 (Método cuasi-Newton global) propuesto en la Sección 3. Para ello se consideran tres problemas, entre los cuales uno es una aplicación a un sistema masa-resorte amortiguador ([9], [16]), y otro es un problema de matrices de Hilbert [15].

Con el propósito de comparar el desempeño global de nuestra propuesta algorítmica, se implementa también el algoritmo global de *Newton-Schur* propuesto en [10]. Además, para apreciar la ventaja de tener un algoritmo global frente a uno local, se implementa el algoritmo local cuasi-Newton [16]. Para mayor claridad en la lectura del documento, se presentan a continuación estos algoritmos.

Algoritmo 2 Método de *Newton-Schur* global [10].

Entrada: $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{X}_0, N_{max}, \epsilon > 0$ y $k = 0$.

Salida: Aproximación a \mathbf{X}_* .

- 1: **mientras** $k \leq N_{max}$, $Res(\mathbf{X}_k) \geq \epsilon$, **hacer**
- 2: Resolver, para \mathbf{S}_k , la ecuación de Sylvester generalizada
- 3:

$$\mathbf{A}\mathbf{S}_k\mathbf{X}_k + (\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k = -Q(\mathbf{X}_k).$$

- 4: **si** $Res(\mathbf{X}_k) \leq \sqrt{\epsilon}$, **entonces**
- 5: $t = 1$,
- 6: **si no**,
- 7: Búsqueda lineal exacta: encontrar t tal que
- 8:

$$\|Q(\mathbf{X}_k + t\mathbf{S}_k)\|_F^2 = \min_{p \in (0,2]} \|Q(\mathbf{X}_k + p\mathbf{S}_k)\|_F^2;$$

- 9: **fin si**:
 - 10: Actualizar $\mathbf{X}_{k+1} = \mathbf{X}_k + t\mathbf{S}_k$ y $k = k + 1$.
 - 11: **fin mientras**:
 - 12: **Salida** X_*
-

Algoritmo 3 Método cuasi-Newton local [16].

Entrada: $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{X}_0, N_{max}, \epsilon$ y $k = 0$.

Salida: Matriz aproximación a \mathbf{X}_* .

- 1: **mientras** $k \leq N_{max}$, $Res(\mathbf{X}_k) \geq \epsilon$, **hacer**
- 2: Resolver, para \mathbf{S}_k , el sistema matricial

$$(2\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k = -Q(\mathbf{X}_k).$$

- 3: Actualizar $\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{S}_k$ y $k = k + 1$.
 - 4: **fin mientras**
 - 5: **Salida** X_*
-

El Algoritmo 3 encuentra \mathbf{S}_k en cada iteración mediante la solución de la ecuación matricial $(2\mathbf{A}\mathbf{X}_k + \mathbf{B})\mathbf{S}_k = -Q(\mathbf{X}_k)$. Con esta estrategia, se obtuvo una reducción en el costo computacional respecto al Algoritmo de *Newton-Schur* [16].

Se escriben los códigos de los algoritmos y de las funciones que definen los problemas en MATLAB® y se realizan los experimentos numéricos en un computador Intel (R) Core (TM) i5-3450 de 2,8 GHz.

Las matrices iniciales son de la forma $\mathbf{X}_0 = 10^p \mathbf{I}_n$, con $p \in \mathbb{Z}$ y $b \mathbf{I}_n$, donde \mathbf{I}_n denota la matriz identidad de orden n y $b = \frac{\|\mathbf{B}\|_F + \sqrt{\|\mathbf{B}\|_F^2 + 4\|\mathbf{A}\|_F\|\mathbf{C}\|_F}}{2\|\mathbf{A}\|_F}$ es la constante dada en [5]. Además, para el Problema 1 se consideran matrices iniciales $\mathbf{X}_0 = r \mathbf{K}$, con $r \in \mathbb{R}$ y

$$\mathbf{K} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Para el Problema 2 se consideran matrices iniciales $\mathbf{X}_0 = \mathbf{C}^m$, con $m = 3, 4, 5, 6$, donde \mathbf{C} es la matriz constante de la ecuación cuadrática a resolver.

Se declara convergencia en los algoritmos cuando $Res(\mathbf{X}_k) < \epsilon$, donde $\epsilon = 10^{-10}$, y divergencia si el número de iteraciones es mayor que $N_{max} = 100$.

A continuación, se describe cada problema y se presentan los resultados en sendas tablas que tienen la siguiente información: matriz inicial utilizada (\mathbf{X}_0); número de iteraciones (N); valor de Res definido por (8) y el tiempo de ejecución medido en segundos (*tiempo*). Para medir el tiempo y quitar el efecto aleatorio de los computadores multitareas se realiza el promedio de tiempo obtenido al ejecutar 100 veces el algoritmo con la misma matriz inicial.

Problema 1. Calcular un solvente de $Q(\mathbf{X}) = \mathbf{A}\mathbf{X}^2 + \mathbf{B}\mathbf{X} + \mathbf{C} = \mathbf{0}$ [10], donde

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix} \quad \text{y} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

La ecuación tiene dos solventes reales $\mathbf{X}_*^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ y $\mathbf{X}_*^2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

Método cuasi-Newton global				Método de <i>Newton-Schur</i> global		
\mathbf{X}_0	N	Res	$tiempo$	N	Res	$tiempo$
$b \mathbf{I}_2$	5	2.6584e-012	0.0018	5	6.7913e-014	0.0022
$10 \mathbf{I}_2$	6	1.9102e-012	0.0022	6	0	0.0024
$10^5 \mathbf{I}_2$	7	2.5206e-013	0.0025	7	1.8064e-013	0.0028
$10 \mathbf{K}$	7	3.8787e-013	0.0025	nc	–	–
$-10^2 \mathbf{K}$	7	6.4750e-015	0.0031	nc	–	–
$-10^3 \mathbf{K}$	7	4.8777e-011	0.0031	nc	–	–
$-10^4 \mathbf{K}$	7	8.1500e-014	0.0033	nc	–	–
$-10^5 \mathbf{K}$	7	5.4798e-013	0.0030	nc	–	–

Método cuasi-Newton local			
\mathbf{X}_0	N	Res	$tiempo$
bI_2	5	6.6128e-013	9.2915e-004
$10I_2$	8	2.1623e-013	0.0014
$10^5 I_2$	22	0	0.0037
$10 K$	8	2.1623e-013	0.0019
$-10^2 K$	12	1.5701e-017	0.0028
$-10^3 K$	15	2.4132e-015	0.0035
$-10^4 K$	18	1.6536e-012	0.0056
$-10^5 K$	22	0	0.0086

Tabla 1. Resultados para el Problema 1.

Para este problema se observa el excelente desempeño del método cuasi-Newton global frente al método de *Newton-Schur* global. Además, se nota la ventaja del algoritmo global propuesto frente al algoritmo local en cuanto al número de iteraciones. En cuanto al tiempo de ejecución de los algoritmos, se observa un comportamiento similar en los casos donde se obtiene convergencia. Los métodos utilizados que convergen lo hacen al solvente \mathbf{X}_*^1 .

Problema 2. Sistema masa-resorte amortiguador conectado ([9], [16], [18], [24]).

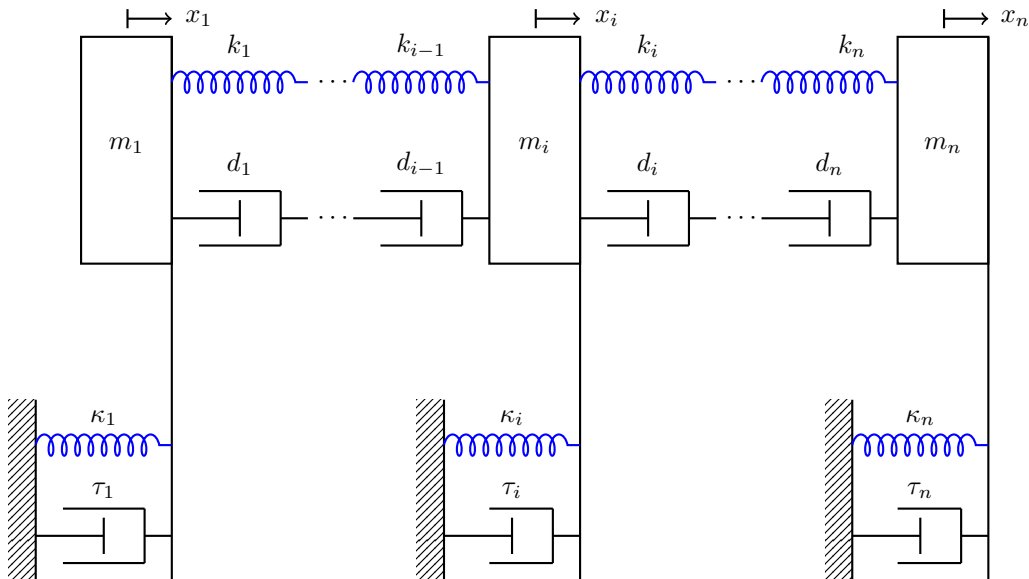


Figura 7. Sistema masa-resorte amortiguador conectado.

En este problema se considera un *sistema masa-resorte amortiguador* [25], donde la masa i -ésima de peso m_i está conectada a su $(i + 1)$ vecino por un resorte y un amortiguador con constantes κ_i y d_i , respectivamente. La masa i -ésima también está conectada a

tierra por un resorte y un amortiguador con constantes κ_i y τ_i , respectivamente (ver Figura 7). La vibración de este sistema está descrita por una ecuación diferencial de segundo orden

$$\mathbf{A} \frac{d^2x}{dt^2} + \mathbf{B} \frac{dx}{dt} + \mathbf{C}x = 0,$$

donde $\mathbf{A} = \text{diag}(m_1, \dots, m_n)$ es la *matriz de las masas*, $\mathbf{B} = \text{tridiag}(-\tau_i, 3\tau_i, -\tau_i)$ es la *matriz de amortiguación* y $\mathbf{C} = \text{tridiag}(-\kappa_i, 3\kappa_i, -\kappa_i)$ es la *matriz de rigidez*, respectivamente.

En las pruebas numéricas, se toma $n = 150$, $m_i = 1$, para todo i , los *resortes* y los *amortiguadores* con un mismo valor constante; es decir, $\kappa_i = \kappa = 5$ y $\tau_i = \tau = 10$, para todo i , excepto $B(1, 1) = B(n, n) = 2\tau = 20$.

Método cuasi-Newton global				Método de <i>Newton-Schur</i> global		
\mathbf{X}_0	N	<i>Res</i>	<i>tiempo</i>	N	<i>Res</i>	<i>tiempo</i>
$b\mathbf{I}_{150}$	8	1.2081e-011	0.1093	6	9.0172e-012	3.1442
$10^4\mathbf{I}_{150}$	9	5.4594e-012	0.1646	7	4.6101e-012	3.9574
$10^5\mathbf{I}_{150}$	9	8.2761e-012	0.1672	7	3.4881e-012	4.1703
\mathbf{C}^3	10	1.0179e-011	0.1454	nc	–	–
\mathbf{C}^4	13	4.5956e-012	0.1924	nc	–	–
\mathbf{C}^5	15	7.1044e-011	0.2353	nc	–	–
\mathbf{C}^6	18	2.6335e-011	0.2695	16	6.4935e-013	8.0279
$-\mathbf{C}^9$	27	9.9914e-12	0.2610	nc	–	–

Método cuasi-Newton local			
\mathbf{X}_0	N	<i>Res</i>	<i>tiempo</i>
$b\mathbf{I}_{150}3$	9	2.9414e-011	0.0842
$10^4\mathbf{I}_{150}$	17	3.4120e-011	0.2422
$10^5\mathbf{I}_{150}$	21	3.4785e-012	0.3241
\mathbf{C}^3	18	3.8361e-012	0.1805
\mathbf{C}^4	23	1.8099e-011	0.2312
\mathbf{C}^5	28	7.0755e-011	0.2869
\mathbf{C}^6	34	6.9429e-012	0.3397
$-\mathbf{C}^9$	nc	–	–

Tabla 2. Resultados para el **Problema 2**.

Para las diferentes matrices iniciales, se obtiene convergencia del algoritmo propuesto. En cuanto al tiempo de ejecución, tenemos que en el caso de convergencia en los métodos globales el algoritmo propuesto emplea aproximadamente el 5% del tiempo que usa el *Newton-Schur* global, mostrando un excelente desempeño. Además, se aprecia la ventaja de un algoritmo global frente a uno local en cuanto a convergencia se refiere, pues en la mayoría de los casos el número de iteraciones de la propuesta global es cercano a la mitad de las iteraciones empleadas por el algoritmo *cuasi-Newton* local, y hay un caso en el que el algoritmo local no converge.

Problema 3 (Matrices de Hilbert)

Calcular un solvente de $Q(\mathbf{X}) = \mathbf{A}\mathbf{X}^2 + \mathbf{B}\mathbf{X} + \mathbf{C} = \mathbf{0}$ ([12], [15]), donde $\mathbf{A} = \mathbf{I}_{100}$, $\mathbf{B} = \mathbf{I}_{100}$, $\mathbf{C} = -(\mathbf{H}^2 + \mathbf{H})$, y $\mathbf{H} = (h_{ij})$, con $h_{ij} = \frac{1}{i+j-1}$.

Método cuasi-Newton global				Método de <i>Newton-Schur</i> global		
\mathbf{X}_0	N	Res	$tiempo$	N	Res	$tiempo$
$b\mathbf{I}_{100}$	4	1.8537e-011	0.0307	4	1.8537e-011	0.5584
$10^2\mathbf{I}_{100}$	5	8.5338e-013	0.0345	5	8.5336e-013	0.6902
$10^{18}\mathbf{I}_{100}$	8	4.1196e-013	0.0595	10	2.1252e-014	1.4126
$10^{20}\mathbf{I}_{100}$	8	4.1196e-013	0.0595	19	3.5556e-017	2.6004
$10^{21}\mathbf{I}_{100}$	8	4.1196e-013	0.0595	26	4.4448e-011	3.5661
$10^{23}\mathbf{I}_{100}$	8	4.1196e-013	0.0595	37	6.8201e-013	5.0639
$10^{29}\mathbf{I}_{100}$	10	2.032e-010	0.0728	nc	–	–
$10^{39}\mathbf{I}_{100}$	10	9.7085e-011	0.0752	nc	–	–
$10^{40}\mathbf{I}_{100}$	13	6.7891e-014	0.0901	nc	–	–
$10^{42}\mathbf{I}_{100}$	13	2.8279e-012	0.0967	nc	–	–
$10^{45}\mathbf{I}_{100}$	14	4.8537e-013	0.1001	nc	–	–
$10^{50}\mathbf{I}_{100}$	15	6.2980e-013	0.1076	nc	–	–

Método de cuasi-Newton local			
\mathbf{X}_0	N	Res	$tiempo$
$b\mathbf{I}_{100}$	6	5.9214e-016	0.0270
$10^2\mathbf{I}_{100}$	12	2.6857e-016	0.0498
$10^{18}\mathbf{I}_{100}$	65	2.9741e-016	0.2869
$10^{20}\mathbf{I}_{100}$	71	6.3579e-012	0.3050
$10^{21}\mathbf{I}_{100}$	75	2.7692e-016	0.3094
$10^{23}\mathbf{I}_{100}$	81	3.6072e-012	0.3388
$10^{29}\mathbf{I}_{100}$	nc	–	–
$10^{39}\mathbf{I}_{100}$	nc	–	–
$10^{40}\mathbf{I}_{100}$	nc	–	–
$10^{42}\mathbf{I}_{100}$	nc	–	–
$10^{45}\mathbf{I}_{100}$	nc	–	–
$10^{50}\mathbf{I}_{100}$	nc	–	–

Tabla 3. Resultados para el **Problema 3**.

En este problema es en donde mejor se aprecia el impacto en convergencia del método cuasi-*Newton* global frente al método cuasi-*Newton* local, así como el excelente desempeño frente al método de *Newton Schur* global. También se observa el mejor comportamiento del método cuasi-*Newton* global frente al *Newton Schur* global en cuanto a tiempo de ejecución, pues la nueva propuesta algorítmica usa aproximadamente un 5 % del tiempo del método *Newton Schur* global para matrices iniciales donde hay convergencia. Además, la propuesta algorítmica converge en todos los casos considerados.

6. Comentarios finales

Motivados por las bondades de disponer de un algoritmo global frente a uno local [6], y por el buen desempeño local del algoritmo cuasi-*Newton* que se propone en [16] para resolver la ecuación cuadrática matricial frente al tradicional método de *Newton-Schur* [10], en este artículo, se presentó una globalización del algoritmo cuasi-*Newton* basada en la incorporación de una estrategia de búsqueda lineal exacta análoga a la usada en [10] para el método de *Newton-Schur*.

Se demostró que la estrategia de globalización no interfiere en la tasa de convergencia obtenida para el algoritmo cuasi-*Newton*, y pruebas numéricas preliminares muestran el buen desempeño del algoritmo global propuesto frente al algoritmo global de *Newton-Schur* y al algoritmo cuasi-*Newton* local.

Finalmente, en búsqueda de ampliar el espectro de trabajo con respecto a la ecuación cuadrática matricial y la aplicación del algoritmo global propuesto, pensamos que vale la pena usarlo en la solución de problemas prácticos, ampliando de esta manera el alcance numérico del algoritmo. Por otro lado, sería muy útil usar el algoritmo globalizado para resolver el problema de los valores propios cuadrático [23], que surge en muchos problemas de ingeniería, y que dio origen a la mayoría de desarrollos teóricos de la ecuación cuadrática matricial.

Agradecimientos

Agradecemos a la Universidad del Cauca por el tiempo concedido para esta investigación, mediante el Proyecto de investigación, del grupo de Optimización VRI ID 4440, y a los evaluadores de este artículo por sus acertadas observaciones.

Referencias

- [1] Bermúdez A., Durán R.G., Rodríguez R. and Salomin J., “Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics”, *SIAM J. Numer. Anal.* 38 (2000), No. 1, 267–291.
- [2] Clark J.V., Zhou N. and Pister K.S.J., “Nodal simulation for MEMS design using SUGAR v0.5”, International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, Santa Clara, CA. 1988, 308–313.
- [3] Clark J.V., Zhou N. and Pister K.S.J., “Modified nodal analysis for MEMS with multi-energydomains”, International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Diego, CA. March, 2000.
- [4] Davila C.E., “A subspace approach to estimation of autoregressive parameters from noisy measurements”, *IEEE Trans. Signal Process.* 46 (1988), No. 2, 531–534.
- [5] Davis J.G., “Numerical solution of a quadratic matrix equation”, *SIAM J. Sci. Statist. Comput.* 2 (1981), No. 2, 164–175.
- [6] Dennis J.E. and Schnabel R.B., *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, Philadelphia, 1996.

- [7] Golub H.G. and Van Loan C.F., *Matrix Computations*, The Johns Hopkins University Press, Johns Hopkins University Press, Baltimore, MD, 1996.
- [8] Higham N.J., *Functions of matrices theory and computations*, SIAM, Philadelphia, PA, 2008.
- [9] Higham N.J. and Kim H., “Numerical analysis of a quadratic matrix equation”, *IMA J. Numer. Anal.* 20 (2000), No. 4, 499–519.
- [10] Higham N.J. and Kim H., “Solving a quadratic matrix equation by Newton’s methods with exact line searches”, *SIAM J. Anal. Appl.* 23 (2001), No. 2, 303–316.
- [11] Kay S.M., “Noise compensation for autoregressive spectral estimates”, *IEEE Trans. Acoust. Speech Signal Process.* 28 (1980), No. 3, 292–303.
- [12] Kim H.M., “Numerical methods for solving a quadratic matrix equation,” Thesis (Ph.D.), Manchester University, 2000, 59–60.
- [13] Lancaster P., *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford-New York-Paris, 1966.
- [14] Laub A.J., “Efficient multivariable frequency response computations”, 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, Albuquerque, USA, 39–41, December, 1980.
- [15] Long J.H., Xi-Yan and Zhang L., “Improved Newton’s method with exact line searches to solve quadratic matrix equation”, *J. Comput. Appl. Math.* 222 (2008), No. 2, 645–654.
- [16] Macías M., Martínez H.J. y Pérez R., “Un algoritmo cuasi-Newton para resolver la ecuación cuadrática matricial”, *Rev. Integr. Temas Mat.* 34 (2016), No. 2, 187–206.
- [17] Macías M., Martínez H.J. y Pérez R., “Sobre la convergencia de un método secante para ecuaciones matriciales no lineales”, *Rev. Integr. Temas Mat.* 32 (2014), No. 2, 181–197.
- [18] Monsalve M. and Raydan M., “Newton’s method and secant methods: A long-standing relationship from vectors to matrices”, *Port. Math.* 68 (2011), No. 4, 431–475.
- [19] Monsalve M. and Raydan M., “A secant method for nonlinear matrix problems”, *Numerical linear algebra in signals, systems and control*, Springer 80 (2011), 387–402.
- [20] Nocedal J. and Wright J.G., *Numerical Optimizations*, Springer-Verlag, New York, 1999.
- [21] Reddy S.C., Schmid P.J. and Henningson D.S., “Pseudospectra of the Orr-Sommerfeld operator”, *SIAM J. App. Math.* 53 (1993), No. 1, 15–47.
- [22] Smith H.A., Singh R.K. and Sorensen D.C., “Formulation and solution of the nonlinear, damped eigenvalue problem for skeletal systems”, *Internat. J. Numer. Methods Engrg.* 38 (1995), No. 18, 3071–3085.
- [23] Tisseur F. and Meerbergen K., “The quadratic eigenvalue problem”, *SIAM Rev.* 43 (2001), No. 2, 235–286.
- [24] Tisseur F., “Backward error and condition of polynomial eigenvalue problems”, *Linear Algebra Appl.* 309 (2000), No. 1-3, 339–361.
- [25] Thomson W.T., *Teoría de vibraciones con aplicaciones*, CRC Press, 1996.