

ELECCIÓN EFICIENTE DE UNA BASE PARA
EL ESPACIO NULO DE UNA MATRIZ

AN EFFICIENT CHOICE OF A BASIS FOR THE NULL SPACE OF A MATRIX

María G. Eberle, Victor M. Ferracutti, María C. Maciel,
Departamento de Matemática, Universidad Nacional del Sur, Av. L. N. Alem 1253,
8000 Bahía Blanca, Argentina.

Resumen

Se considera el sistema lineal indeterminado $Ax = b$ y el problema de hallar una base del núcleo de A . $\mathcal{N}(A)$. Este problema surge en una gran variedad de algoritmos que resuelven problemas de optimización con restricciones de igualdad, utilizando entre otros el Método de Programación Cuadrática Sucesiva, (PCS). Para que el algoritmo que resuelve el problema cuadrático en PCS sea eficiente, es indispensable que la base del espacio tangente de las restricciones sea elegida con el menor costo computacional posible, evitando factorizaciones de la matriz. En consecuencia, el objetivo de esta contribución es presentar un método eficiente que en cada iteración resuelva un problema de cuadrados mínimos con restricciones: hallar una matriz con estructura triangular en bloques, con bloques no singulares en la diagonal, más próxima en la norma de Frobenius a una submatriz cuadrada de la matriz dada A .

La estrategia usada se basa en el Método de Proyecciones Alternadas de Dykstra para conos convexos y cerrados. Modificando este método de acuerdo a las restricciones del problema de cuadrados mínimos que surgen de la estructura de la matriz buscada, se deduce un algoritmo para el cual se establecen propiedades de convergencia. Se presentan experimentos numéricos preliminares que muestran como trabaja esta técnica, y se compara con otras ya existentes.

Abstract

The undetermined linear system $Ax = b$, and the problem of finding a basis of the null space of A are considered. This problem appears in many algorithms based on Successive Quadratic Programming (SQP) Method, for solving optimization problems with equality constraints. The efficiency of the algorithm for solving the quadratic subproblem depends strongly on the choice of the basis of the tangent space of the constraints. Avoiding factorization of the matrix, the computational cost is reduced. Therefore, the objective of this contribution is to present an efficient method such that at each iteration it solves a least squares problem with constraints: find a matrix with block triangular structure with no singular diagonal blocks nearest, in the Frobenius norm, to a square submatrix of the given matrix A .

The strategy used is based on the Alternate Projection Method for closed convex cones suggested by Dykstra. An algorithm is deduced modifying Dykstra's method according to the constraints of the least squares problem, which appear from the structure of the matrix. Convergence properties are stated. Preliminaries numerical experiments showing how this technique works are presented.

1 Introducción.

Dado el sistema lineal indeterminado $Ax = b$, $A \in \mathbb{R}^{m \times n}$, $m < n$, $rank(A) = m$, interesa resolver el problema de hallar, de manera eficiente, una base del núcleo de A , en particular en el caso en que el sistema sea de gran porte. Si se considera la partición de A

$$A = [B \mid N]$$

dónde $B \in \mathbb{R}^{m \times m}$ es una submatriz no singular de A y $N \in \mathbb{R}^{m \times (n-m)}$, entonces la matriz $\mathcal{W} \in \mathbb{R}^{n \times (n-m)}$:

$$\mathcal{W} = \begin{pmatrix} -B^{-1}N \\ I_{n-m} \end{pmatrix};$$

es tal que sus columnas forman una base de $\mathcal{N}(A)$. Resulta claro que la forma en que \mathcal{W} sea encontrada depende de B . El objetivo es entonces diseñar un algoritmo destinado a obtener B de manera eficiente.

En una gran variedad de algoritmos que resuelven problemas de programación no lineal surge la necesidad de hallar una base para el espacio nulo de alguna matriz; en particular esa matriz suele ser el gradiente de las restricciones del problema:

Sea

$$(mri) = \begin{cases} \min & f(x) \\ s.a. & C(x) = 0 \end{cases}$$

con $f: \mathbb{R}^n \rightarrow \mathbb{R}$, y $C: \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Una manera popular de abordarlo es aplicar el método de programación cuadrática sucesiva, en el cual, en cada iteración se debe resolver un problema cuadrático

$$(Q_c) = \begin{cases} \min & q(s) = \frac{1}{2}s^T H_c s + h_c^T s \\ s.a. & \nabla C_c^T s + C_c = 0. \end{cases}$$

dónde:

$H_c = \nabla_{xx}^2 \mathcal{L}(x_c, \lambda_c)$ ó bien una aproximación del Hessiano del Lagrangiano,

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T C(x);$$

$h_c = \nabla \mathcal{L}_c$ es el gradiente del Lagrangiano en x_c

∇C_c es el Jacobiano de las restricciones en x_c , siendo x_c el iterado actual.

El Algoritmo de Maciel [4], se basa en un algoritmo de programación cuadrática sucesiva que usa como punto factible inicial un paso en la dirección normal del espacio tangente de las restricciones, y resuelve el problema cuadrático usando direcciones conjugadas, restringidas al espacio tangente. La reducción a $\mathcal{N}(\nabla C^T)$ se efectúa como sigue: conocido x_c , debemos hallar s_c como solución aproximada de (Q_c) . Sin pérdida de generalidad, supondremos que $C_c = 0$ pues de no ser así sería posible pasar a ese caso por medio de traslaciones. Eliminemos los subíndices para facilitar la presentación. Supongamos que $\nabla C^T \in \mathbb{R}^{m \times n}$, con $m < n$, m y n grandes; H simétrica y definida positiva en $\mathcal{N}(\nabla C^T)$ y $\text{rank}(\nabla C^T) = m$.

La idea es transformar a (Q_c) en un problema sin restricciones y de menor dimensión, para que el problema de gran porte sea más fácil de tratar.

Sea la siguiente partición de la matriz ∇C^T :

$$\nabla C^T = [B \quad | \quad N]$$

$B \in \mathbb{R}^{m \times m}$ es una submatriz no singular de ∇C^T y $N \in \mathbb{R}^{m \times (n-m)}$.

Realizando una partición análoga para s :

$$s = (s_B, s_N)^T,$$

resulta que:

$$\nabla C^T s = B s_B + N s_N = 0$$

por lo tanto

$$s_B = -B^{-1} N s_N.$$

Con una adecuada permutación de filas y columnas de H se practica la siguiente partición al Hessiano:

$$H = \left[\begin{array}{c|c} H_{11} & H_{12} \\ \hline H_{21} & H_{22} \end{array} \right]$$

y para el gradiente:

$$h^T = (h_B \quad h_N)$$

De modo que es posible escribir a $q(x)$ en términos de la componente tangencial, es decir en $\mathcal{N}(\nabla C^T)$. La forma cuadrática $q(s)$ queda expresada:

$$\bar{q}(s) = \frac{1}{2} s_N^T \bar{H} s_N + \bar{h}^T s_N$$

Siendo:

$$\begin{aligned} \bar{H} &= (-B^{-1}N)^T H_{11} (B^{-1}N) - H_{21} (B^{-1}N) - H_{21} (B^{-1}N) \\ &\quad - (B^{-1}N)^T H_{12} + H_{22}. \end{aligned}$$

$$\bar{h} = (h_B^T (-B^{-1}N) + h_N^T)$$

Dado que:

$$s = \begin{pmatrix} s_B \\ s_N \end{pmatrix} = \begin{pmatrix} -B^{-1}N s_N \\ s_N \end{pmatrix} = \begin{pmatrix} -B^{-1}N \\ I_{n-m} \end{pmatrix} s_N.$$

Si definimos $\mathcal{W} \in \mathbb{R}^{n \times (n-m)}$:

$$\mathcal{W} = \begin{pmatrix} -B^{-1}N \\ I_{n-m} \end{pmatrix};$$

entonces

$$s = \mathcal{W} s_N.$$

luego:

$$q(s_N) = \frac{1}{2} s_N^T \bar{H} s_N + \bar{h}^T s_N.$$

y si $s_N \equiv \bar{s}$: resulta:

$$\bar{q}(\bar{s}) = \frac{1}{2} \bar{s}^T \bar{H} \bar{s} + \bar{h}^T \bar{s}.$$

Es claro que las columnas de \mathcal{W} forman una base de $\mathcal{N}(\nabla C^T)$. De modo que es posible definir una transformación lineal de \mathbb{R}^n en $\mathcal{N}(\nabla C^T)$, que a cada $s \in \mathbb{R}^n$ lo transforme en $(\mathcal{W}^T s) \in \mathcal{N}(\nabla C^T)$, con lo cual el problema: (Q_c) se reduce a:

$$(\bar{Q}) = \begin{cases} \min & \bar{q}(\bar{s}) = \frac{1}{2} \bar{s}^T \bar{H} \bar{s} + \bar{h}^T \bar{s} \\ s.a. & \bar{s} \in \mathcal{N}(\nabla C^T). \end{cases}$$

Este último problema puede ser resuelto aplicando cualesquiera de los algoritmos disponibles para la resolución de problemas cuadráticos. Debe notarse que la eficiencia del algoritmo, y en sí la transformación definida, dependen de la elección de la matriz \mathcal{W} , y por lo tanto de B .

En la Sección 2 se establecen los conceptos preliminares, en la Sección 3 es presentado el nuevo algoritmo acompañado por los resultados teóricos más importantes. Finalmente en la Sección 4 se presentan resultados numéricos preliminares, y en la Sección 5 las conclusiones.

2 Preliminares.

Se incluyen en esta sección dos algoritmos que son motivación para el nuevo algoritmo: el método de proyecciones alternadas de Dykstra [2], y una aplicación del mismo sugerida por Escalante-Raydán [3].

2.1 Método de proyecciones alternadas de Dykstra.

Dado el problema

$$\begin{cases} \min & \|g - x\| \\ \text{s.a.} & x \in \bigcap K_i, \quad i = 1, \dots, r. \end{cases} \quad (1)$$

donde K_1, K_2, \dots, K_r son conos convexos cerrados de \mathbb{R}^n , $g \in \mathbb{R}^n$ y la norma es la norma asociada al producto interno:

$$(x, y) = \sum_{i=1}^n x_i y_i w_i,$$

para $w \in \mathbb{R}^n$, es decir:

$$\|x\| = (x, x)^{1/2} = \left(\sum_{i=1}^n x_i^2 w_i \right)^{1/2}.$$

Asumiendo que es posible resolver el problema de hallar un vector en K_i tal que sea solución de

$$\begin{cases} \min & \|f - x\| \\ \text{s.a.} & x \in K_i \end{cases} \quad (2)$$

para toda f y para todo i , la solución del problema (1) se obtiene resolviendo una sucesión de problemas de tipo (2).

Teniendo en cuenta la siguiente notación :

- a) $\mathcal{P}|_{K_i}(f)$ es la proyección de f sobre K_i , solución de (2).
- b) Para todo entero positivo n se define $m \bmod r = i$, si $n = kr + i$, para todo k, i enteros, $1 \leq i \leq r$,

se describe a continuación el algoritmo:

Algoritmo 2.1 (*Método de Dykstra*)

$g_0 = g$, $I_i \equiv 0$, $i = 1, \dots, r$, $n = 1$

- 1-) Sea $g_n = \mathcal{P}|_{K_{n \bmod r}}((g_{n-1} - I_{n \bmod r}))$,
Actualizar $I_{n \bmod r} = g_n - (g_{n-1} - I_{n \bmod r})$.

- 2-) Hacer $n = n + 1$ e ir a 1-).

El teorema de convergencia para el algoritmo anterior:

Teorema 2.1 *Los vectores g_n generados por el Algoritmo 2.1, convergen a una solución de (1), es decir $g_n \rightarrow g^*$ cuando $n \rightarrow \infty$.*

La demostración puede ser consultada en [2].

2.2 Algoritmo de Escalante-Raydán

Se considera el problema de cuadrados mínimos con restricciones:

$$\left\{ \begin{array}{l} \min \quad \|X - A\|_F^2 \\ \text{s.a.} \\ L \leq X \leq U, \\ X^T = X, \quad \lambda_{\min}(X) \geq \epsilon > 0, \\ X \in \mathcal{P}. \end{array} \right. \quad (3)$$

dónde

$A, L, U \in \mathbb{R}^{n \times n}$, X matriz cuadrada simétrica.

$\lambda_{\min}(X)$ es el menor autovalor de X , y ϵ es una constante dada.

$\mathcal{P} \subseteq \mathbb{R}^{n \times n}$, contiene matrices con alguna estructura especial.

La región de factibilidad del problema esta dada por la intersección de los siguientes conjuntos:

$$B = \{X \in \mathbb{R}^{n \times n} : L \leq X \leq U\},$$

$$\text{epd} = \{X \in \mathbb{R}^{n \times n} : X^T = X, \lambda_{\min}(X) \geq \epsilon > 0\},$$

$$\mathcal{P} = \{X \in \mathbb{R}^{n \times n} : X = \sum_{i=1}^m \alpha_i G_i, \alpha_i \in \mathbb{R}, 1 \leq i \leq m\}.$$

Las matrices G_1, G_2, \dots, G_m , son matrices no nulas, simétricas, en $\mathbb{R}^{n \times n}$, cuyas entradas valen cero ó uno de acuerdo a la siguiente propiedad: para cada elemento ubicado en la posición st , $1 \leq s, t \leq n$ existe un y sólo un k tal que $1 \leq k \leq m$, y $(G_k)_{s,t} = 1$.

De modo que (3) puede ser expresado como sigue:

$$\min\{\|X - A\|_F^2 : X \in B \cap \text{epd} \cap \mathcal{P}\}. \quad (4)$$

Luego la región de factibilidad para el problema (4) está dada por la intersección de conjuntos cerrados y convexos en el espacio producto interno $\mathbb{R}^{n \times n}$. Debe notarse que \mathcal{P} es un subespacio del subespacio de matrices simétricas y que el conjunto de matrices $\{G_1, G_2, \dots, G_m\}$ constituye una base para \mathcal{P} .

Se aplicará el Método de Proyecciones Alternadas de Dykstra para resolver el problema (4). La idea fundamental es proyectar sobre cada uno de los subconjuntos cerrados y convexos que forman la región de factibilidad, por lo cual es necesario caracterizar las proyecciones sobre los mismos. En [3] pueden consultarse los tres teoremas, con sus respectivas demostraciones, que cumplen el objetivo de caracterizar las proyecciones sobre cada convexo.

Algoritmo 2.2 (*Método de Escalante-Raydán*)

Dada $A \in \mathbb{R}^{n \times n}$, sea $A_0 = A$, $I_{\text{epd}}^0 = I_B^0 = 0$

para $i = 1, 2, \dots$,

$$A_i = P_{\mathcal{P}}(A_i) - I_{\text{epd}}^i,$$

$$I_{\text{epd}}^{i+1} = P_{\text{epd}}(A_i) - A_i,$$

$$A_i = P_{\text{epd}}(A_i) - I_B^i,$$

$$I_B^{i+1} = P_B(A_i) - A_i,$$

$$A_{i+1} = P_B(A_i).$$

El siguiente resultado constituye el teorema de convergencia para el Algoritmo (2.2):

Teorema 2.2 *Si el conjunto cerrado y convexo $B \cap \text{epd} \cap \mathcal{P}$, es no vacío, entonces para alguna matriz $A \in \mathbb{R}^{n \times n}$ la sucesiones $\{P_{\mathcal{P}}(A_i)\}$, $\{P_B(A_i)\}$, y $\{P_{\text{epd}}(A_i)\}$ generadas por el Algoritmo (2.2) convergen en norma de Frobenius a una solución de (4).*

La demostración se sigue de un resultado de convergencia establecido por Boyle y Dykstra [1]. Los autores han presentado una versión mejorada del algoritmo, en la cual se proyecta primero sobre epd y luego sobre la intersección de los otros dos convexos. Más detalles pueden consultarse en [3] Finalmente, en cuanto a la experiencia computacional presentada por los autores, son claros los beneficios de aplicar el algoritmo mejorado, no sólo se impone al primero en cuanto a tiempo, sino que es significativa la diferencia en número de iteraciones que resulta de aplicar uno u otro. Debe destacarse lo ingenioso, simple y práctico de este método, lo cual ha permitido explotar estas ideas para encontrar la solución al problema presentado en la Sección 1.

3 El nuevo algoritmo.

Sea $Ax = b$ un sistema lineal indeterminado. En esta Sección se considerará el problema de aproximar de manera eficiente y con bajo costo computacional la submatriz no singular B de A . Se presenta un método que en cada iteración resuelve un problema de cuadrados mínimos con restricciones. La propuesta consiste en proyectar una submatriz cuadrada de la matriz dada A , sobre cierto conjunto \mathcal{P} , de matrices con determinada estructura. Para ser más precisos la idea es hallar una matriz triangular con elementos no nulos sobre la diagonal, más próxima en la norma de Frobenius a la submatriz de la matriz dada A .

3.1 Presentación del problema

Sea el siguiente problema de minimización:

$$\begin{cases} \min & \|\bar{X} - A\|_F^2 \\ \text{s.a.} & \bar{X} \in \bar{\mathcal{P}}. \end{cases} \quad (5)$$

dónde

$$A = [B \mid N] \in \mathbb{R}^{m \times n}$$

$$\bar{\mathcal{P}} = \{ \bar{X} \in \mathbb{R}^{m \times n}, \bar{X} = [X \mid N] \},$$

con:

$$\bar{X} = \begin{cases} X_{ij}, & 1 \leq i, j \leq m \\ A_{ij} & \text{en otro caso.} \end{cases}$$

para

$$X = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \cdots & \cdots & \alpha_m \\ 0 & \alpha_1 & \alpha_2 & \cdots & \cdots & \cdots & \alpha_{m-1} \\ 0 & 0 & \alpha_1 & \alpha_2 & \cdots & \cdots & \alpha_{m-2} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & & & \cdots & \cdots & \cdots & \vdots \\ 0 & & & & \cdots & \alpha_1 & \alpha_2 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & \alpha_1 \end{bmatrix}.$$

esto es

$$X = \sum_{p=1}^m \alpha_p G_p,$$

dónde en general G_p está definidas como sigue:

$$G_p = \begin{cases} 1 & \text{si } j = i + p - 1 \\ 0 & \text{en otro caso.} \end{cases} \quad (6)$$

Con las consideraciones anteriores, el problema (5) es equivalente a:

$$\begin{cases} \min & \|X - B\|_F^2 \\ \text{s.a.} & X \in \mathcal{P}. \end{cases} \quad (7)$$

siendo $\mathcal{P} \subseteq \mathbb{R}^{m \times m}$, tal que:

$$\mathcal{P} = \left\{ X \in \mathbb{R}^{m \times m}, X = \sum_{p=1}^m \alpha_p G_p \right\},$$

con las matrices G_p definidas como en (6). Como ya se ha señalado, la matriz cuadrada solución debe tener columnas linealmente independientes, es preciso aclarar que la estructura triangular no es suficiente para asegurar la independencia lineal de las columnas de la solución, a menos que se imponga a los elementos de la diagonal de X la condición de ser no nulos, esto es resolver uno de los problemas siguientes:

$$\begin{cases} \min & \|X - B\|_F^2 \\ \text{s.a.} & X \in \mathcal{P}, \\ & \alpha_1 \geq \epsilon, \epsilon > 0 \end{cases} \quad (8)$$

$$\begin{cases} \min & \|X - B\|_F^2 \\ \text{s.a.} & X \in \mathcal{P}, \\ & \alpha_1 \leq -\epsilon, \epsilon > 0 \end{cases} \quad (9)$$

De modo que la región de factibilidad para el problema a resolver es el conjunto de matrices triangular superior con elementos no nulos sobre la diagonal, y la solución es la proyección de B sobre ese conjunto. a quien se notará $\mathcal{P}(B)$. El siguiente resultado da una caracterización para las soluciones de (8) y (9):

Lema 3.1 Sean $B \in \mathbb{R}^{m \times m}$, y $\epsilon > 0$, entonces la única solución del problema (8) está dada por:

$$X^* = \mathcal{P}(B) = \sum_{p=1}^m \bar{\alpha}_p G_p,$$

con

$$\begin{cases} \bar{\alpha}_p = \frac{\sum_{i,j=1}^m B_{ij}(G_p)_{ij}}{m-p+1}, & 2 \leq p \leq m, \\ \bar{\alpha}_1 = \epsilon; \end{cases}$$

De manera análoga, la única solución de (9) es:

$$X^{**} = \mathcal{P}(B) = \sum_{p=1}^m \bar{\alpha}_p G_p,$$

con

$$\begin{cases} \bar{\alpha}_p = \frac{\sum_{i,j=1}^m B_{ij}(G_p)_{ij}}{m-p+1}, & 2 \leq p \leq m, \\ \bar{\alpha}_1 = -\epsilon. \end{cases}$$

En lo que sigue la expresión “problema a resolver y su solución”, significará “problema (8) y X^* ”.

3.2 El Algoritmo

Se presenta a continuación un algoritmo que resuelve el problema (8) mediante una estrategia basada en el método de proyecciones alternadas de Dykstra. En cada iteración nuestro algoritmo resuelve el problema de minimización:

$$\begin{cases} \min & \|X - B_k\|_F^2 \\ \text{s.a.} & X \in \mathcal{P}, \quad \alpha_1^{k+1} \geq \epsilon_{k+1}, \end{cases} \quad (10)$$

donde los α_1^{k+1} son los elementos de la diagonal de la solución y ϵ_{k+1} un valor positivo, tal que $\epsilon_{k+1} > \sum_{i=1}^m B_{ii}/m$, que se actualiza en cada iteración.

Una primera versión del algoritmo es la siguiente:

Algoritmo 3.1 (Versión 1)

Dados ϵ_0, B_0 :

Para $k = 0, 1, 2, \dots$, hasta convergencia

- Actualizar ϵ_k ,

- Obtener B_{k+1} .

El algoritmo termina cuando dos proyecciones consecutivas son muy próximas, es decir cuando:

$$\|B_{k+1} - B_k\|_F < Tol,$$

siendo Tol un valor fijado por el usuario.

Algoritmo 3.2 (*Versión 2*)*Dados*

$$Tol > 0, B_0 = B,$$

$$\epsilon_0 = \frac{\sum_{i=1}^m B_{ii}}{m} + \frac{(|\sum_{i=1}^m B_{ii}| + 1)}{m(m-1)},$$

$$a = \left(\frac{(|\sum_{i=1}^m B_{ii}| + 1)}{m^2} \right),$$

para $k = 0, 1, 2, \dots$, hasta $\|B_{k+1} - B_k\|_F < Tol$.

$$\epsilon_{k+1} = \epsilon_k - \frac{a}{m^k},$$

$$B_{k+1} = \mathcal{P}(B_k),$$

Escalante-Raydán [3], plantean como posibles valores de la tolerancia en los problemas resueltos utilizando su algoritmo, cifras que oscilan entre 10^{-2} y 10^{-10} , según sean las dimensiones del problema. Debe notarse que en la definición de ϵ_0 el denominador $(m-1)$ no presenta problemas puesto que se supone $m > 1$.

El siguiente teorema muestra que el Algoritmo 3.2 genera una sucesión de matrices que converge a una solución de (8).

Teorema 3.1 *Para toda $A \in \mathbb{R}^{m \times n}$, sea B una submatriz $m \times m$ de A , la sucesión de matrices generada por el Algoritmo 3.2 converge en norma de Frobenius a una solución de (8).*

4 Resultados numéricos.

La experiencia numérica fue llevada a cabo con el objetivo de medir la eficiencia del algoritmo en el sentido de analizar la relación entre número de iteraciones, dimensión de la matriz A , y tiempo de ejecución. En todos los casos el criterio de parada utilizado es la proximidad de dos proyecciones consecutivas, para distintos valores de tolerancia. El programa fue diseñado utilizando el Microsoft FORTRAN PowerStation, Version 1.0a, 1993, y ejecutado en una computadora PC compatible con procesador 486, 66 Mhz de velocidad y 8MB de memoria principal.

Ejemplo 1: En este caso se han generado, por medio de un sencillo programa, matrices rectangulares en las cuales algunas columnas son columnas de una matriz de Hilbert, y las restantes formadas por números aleatorios. Hemos obtenido los siguientes resultados:

m	n	Tolerancia	Iteraciones	Tiempo
10	20	10^{-8}	7	$< 1 \times 10^{-2} \text{seg.}$
10	20	10^{-16}	12	$< 1 \times 10^{-2} \text{seg.}$
100	200	10^{-8}	3	$1 \times 10^{-2} \text{seg.}$
100	200	10^{-16}	6	$1 \times 10^{-2} \text{seg.}$
800	810	10^{-8}	2	$61 \times 10^{-2} \text{seg.}$
800	810	10^{-16}	5	$55 \times 10^{-2} \text{seg.}$

Ejemplo 2: En este caso las matrices han sido generadas utilizando números aleatorios para todas las entradas. Se ha observado:

m	n	Tolerancia	Iteraciones	Tiempo
10	20	10^{-8}	12	$< 1 \times 10^{-2} \text{seg.}$
10	20	10^{-16}	12	$< 1 \times 10^{-2} \text{seg.}$
100	200	10^{-8}	6	$1 \times 10^{-2} \text{seg.}$
100	200	10^{-16}	6	$1 \times 10^{-2} \text{seg.}$
800	810	10^{-8}	5	$55 \times 10^{-2} \text{seg.}$
800	810	10^{-16}	5	$61 \times 10^{-2} \text{seg.}$

En ambos casos se observa que para valores crecientes de m , el número de iteraciones decrece, a modo de conclusión podemos afirmar que el nuevo algoritmo es adecuado especialmente para problemas de gran porte.

5 Conclusiones

Utilizando una estrategia sencilla basada en el método de proyecciones alternadas de Dykstra, se ha desarrollado un algoritmo que resuelve el problema de aproximar de manera eficiente y con bajo costo computacional la submatriz no singular B de A . Este algoritmo forma parte de un proyecto para encontrar una base del espacio nulo de una matriz de rango completo $A \in \mathbb{R}^{m \times n}$, $m < n$. Los resultados obtenidos permiten concluir que el algoritmo propuesto es competitivo con aquellos que efectúan factorizaciones de A .

Este algoritmo se puede aplicar a la resolución de problemas de programación no lineal con restricciones de igualdad basados en programación cuadrática sucesiva.

Referencias

- [1] J. P. BOYLE and R. L. DYKSTRA. A method for finding projections onto the intersections of convex sets in hilbert spaces. *Lecture Notes in Statistics*, 37:28–47, 1986.
- [2] R.L. DYKSTRA. Restricted least-squares regression. *J. Amer. Stat. Assoc.*, 78:837–842, 1983.
- [3] R. ESCALANTE and MARCOS RAYDAN. Dykstra's algorithm for a constrained least-squares matrix problem. *Numerical Linear Algebra with applications*, 9(6):459–471, 1996.
- [4] M.C. MACIEL. *A global convergence theory for a general class of trust region algorithms for equality constrained optimization*. PhD thesis, Department of Mathematical Science, Rice University, Houston, Texas, 1992.