Instituto Tecnológico Autónomo de México

Métodos Numéricos y Optimización

Trabajo Final

Solución de Mínimos Cuadrados con Factorización QR Vía Transformaciones de Householder

Alumnas:
Mónica Ballesteros 124960
Fabiola Cerón 36027
Ariana López 160281

Profesor:
Erick Palacios

30 de mayo de 2017

Índice

1. Objetivo

Investigar la solución al problema de mínimos cuadrados con factorización QR vía transformaciones de Householder y desarrollar un algoritmo secuencial en el lenguaje C que implemente la solución.

2. Marco Teórico

2.1. Mínimos Cuadrados

El problema del ajuste de datos consiste en encontrar una función que explique de la mejor forma posible el comportamiento de un fenómeno medible del cual conocemos algunos datos.
Lo anterior, en el contexto del álgebra lineal consiste en resolver:

$$Ax = b \tag{1}$$

Donde:
$A \in \mathbb{R}^{mxn}$
$x \in \mathbb{R}^n$
$b \in \mathbb{R}^m$.

A y b contienen la información observada que se quiere ajustar.

Sobre el sistema lineal:

- Cuando $m > n$ (más ecuaciones que incógnitas) se trata de un sistema sobredeterminado y no existe una solución exacta.
  - Como no existe esta solución, el problema se transforma en resolver $\min_{x \in \mathbb{R}^n} ||Ax - b||$ (Problema de optimización matemática).
  - Si la $||.||$ es euclidiana el problema se llama mínimos cuadrados.
- Cuando $m < n$ (menos ecuaciones que incógnitas) el sistema se llama indeterminado y tiene infinidad o ninguna solución dependiendo del rank(A).

- Cuando m=n es un sistema cuadrado que puede ser singular(no hay solución o infinidad de soluciones) o No singular (existe solución única).

Este proyecto se enfoca en resolver el problema de ***mínimos cuadrados***:

Existen diversos métodos de solución de mínimos cuadrados:

- Cálculo de polinomio de interpolación
- Ecuaciones normales.
- Métodos de Factorización de A:
  - Factorización LU
  - Factorización QR
  - Cholesky
  - SVD

Se eligieron los métodos basados en *factorización de A* ya que definen sistemas de ecuaciones lineales equivalentes que son más fáciles de resolver. En particular se opta por las factorizaciones QR, las cuales se pueden realizar con los siguientes algoritmos:

- QR vía Gram-Schmidt (Ortonormalización)
- QR vía Reflexiones de Householder (Transformación Ortogonal)
- QR vía Rotación Givens (Transformación Ortogonal)

El algoritmo de Gram-Schmidt es inestable numéricamente respecto al redondeo. La factorización QR vía Reflexiones de Householder en estable numéricamente respecto al redondeo y calcula la factorización QR por completo, además de que haciendo uso de los reflectores de Householder para transformar el sistema original a un sistema triangular es posible construir ceros en todas las posiciones por debajo de una entrada en un vector, mientras que en Givens la transformación a ceros no se hace de manera abrupta, sino que se elige la entrada en la que se desea hacer ceros.

Por lo anterior, este proyecto se enfoca en calcular la descomposición QR vía reflexiones de Householder para solucionar mínimos cuadrados

2.2. Factorización QR vía Reflexiones Householder

La descomposición QR de una matriz es la factorización de una matriz A en un producto:

$$A = QR$$

Donde:

$A \in \mathbb{R}^{mxn}$ con $m >= n$

$Q \in \mathbb{R}^{mxm}$ es ortogonal

$R \in \mathbb{R}^{mxn}$ es triangular superior

El algoritmo calcula matrices de Householder $H_1, ..., H_n$ tales que si:

$$Q = H_1 H_2 ... H_n \tag{2}$$

Entonces:

$$Q^T A = R \tag{3}$$

Con $Q^T = H_n H_{n-1} ... H_1$

La matriz H es un reflector de Householder, también llamado transformación de Householder que al ser multiplicado por un vector, genera la reflexión del mismo a través del hiperplano $span\{v\}^{\perp}$.

Las transformaciones de Householder fueron introducidas en 1958 por Alston Scott Householder, un matemático estadounidense de la Universidad de Chicago.

Sea $v \in \mathbb{R}^m$ no nulo, se define H^{mxm} reflexión de Householder como:

$$H = I_m - \beta v v^T \text{ con } \beta = \frac{2}{v^T v}$$

A v se le llama vector de Householder (Ver algoritmos 1 y 2, sección 3).

Para determinar la matriz de Householder, es necesario calcular el vector de Householder.

Propiedades de la matriz H de Householder:

- Simétrica.

- Ortogonal (Estas propiedades coadyuvan a la estabilidad numérica de este algoritmo).
  - Las columnas y renglones de H forman un conjunto ortonormal (Ortogonal y sus vectores tienen norma 1).
  - $H^T H = H H^T = H^2 = I_m$
  - $H^{-1} = H^T$
  - H es una isometría
  - El producto de $H's$ es ortogonal

- $H^{-1} = H$
- $det(H) = -1$
- Se usa para hacer ceros debajo de la entrada de un vector cuando $v = x - ||x||_2 e_1$, sin embargo en la primera entrada de v se puede presentar un problema de cancelación en la aritmética de punto flotante. Para evitar esto existen dos alternativas:
  1. Usar $v = x + signo(x_1)||x||_2 e_1$
  2. $v_1 = x_1 - ||x||_2$ (seleccionada en este trabajo con base en el algoritmo de Parlett 1971).
  - Cuando $x_1 <= 0 \Rightarrow v_1 = x_1 - ||x||_2$
  - Cuando $x_1 > 0 \Rightarrow v_1 = x_1 - ||x||_2 = \dfrac{x_1^2 - ||x||_2^2}{x_1 - ||x||_2} = \dfrac{-(x_2^2 + ... + x_n^2)}{x_1 - ||x||_2}$

.

Al aplicar una serie de transformaciones de Householder a la matriz A (ecuaciones 2 y 3), se sobrescribe la matriz A (Matriz A actualizada), con la forma explícita de la matriz R (triangular superior). Además obtenemos $v^{(j)}$ vectores Householder, con $v[1] = 1$ y cuya parte esencial ($v[2:m]$) será almacenada por debajo de la diagonal de la matriz A actualizada, para generar el factor *form representation* de Q, por lo que no es necesario calcular la forma explícita de ésta última. [1]

2.3. Uso de Factorización QR a mínimos cuadrados

Regresando al problema de mínimos cuadrados:

$$Ax = b \Rightarrow Q^T Ax = Q^T b$$

Utilizando (3) obtenemos el nuevo sistema a resolver:

$$Rx = Q^T b$$

Para realizar el cálculo $Q^T b$ es posible usar el factor *form representation* de Q utilizando la parte esencial de los vectores de Householder $v^{(j)}$ para actualizar b de la siguiente manera:

$$b = b - \beta(v^T v)v \tag{4}$$

con: $\beta = 2/v^T v$ (ver algoritmo 3, sección 3 Algoritmos)

Se define como $\hat{b} \in \mathbb{R}^n$ las primeras n entradas del vector b y $\hat{R} \in \mathbb{R}^{nxn}$ como las nxn entradas superiores izquierdas de R.

Por último, se resuelve $\hat{R}x = \hat{b}$

Al ser $\hat{R}$ triangular superior, se observa que el sistema resulta mucho más sencillo de resolver que el sistema original, para lo cual se utiliza el método de sustitución hacia atrás. (ver algoritmo 4, sección 3 Algoritmos)

3. Algoritmos

A continuación se describen los algoritmos que fueron utilizados como base:

3.1. Algoritmo para calcular QR con reflexiones de Householder

Aplicación de la matriz de Householder $H = I - \beta vv^T$ a una matriz A. En una situación típica, house es aplicada a una subcolumna o subfila de una matriz y βvv^T es aplicada a una submatriz obteniendo una serie de matrices H.

Algorithm 1 Algoritmo para calcular QR con reflexiones de Householder

for $j = 1 : n$ **do**
  $(v, \beta) = house(A(j : m, j))$
  $A(j : m, j : n) = (I - \beta vv^T)A(j : m, j : n)$
  if $j < m$ **then**
    $A(j + 1 : m, j) = v(2 : m - j + 1)$
  end if
end for

El algoritmo que encuentra las matrices de Householder $H_1...H_n$ requiere $2n^2(m - n/3)$ flops Si además se requiere calcular $Q = H_1...H_n$ la acumulación requiere $4(m^2n - mn^2 + n^3/3$ flops.

3.2. Algoritmo para calcular el Vector de Householder:

El proceso de cálculo del vector se resume en el siguiente algoritmo, donde considerando $x \in \mathbb{R}^m$, esta función calcula $v \in \mathbb{R}^m$ con $v(1) = 1$ y $\beta \in \mathbb{R}$ tal que $H = I_m - \beta vv^T$ es ortogonal y $Hx = ||x||_2 e_1$

Algorithm 2 Algoritmo Vector Householder

procedure FUNCTION(v, β)= house(x) ▷ Definición
  $m = length(x), \sigma = x(2:m)^T x(2:m), v = \begin{pmatrix} 1 \\ x(2:m) \end{pmatrix}$
  if $\sigma = 0$ and $x(1) >= 0$ **then**
    $\beta = 0$
  else if $\sigma = 0$ and $x(1) < 0$ **then**
    $\beta = -2$
  else
    $\mu = \sqrt{x(1)^2 + \sigma}$
    if $x(1) <= 0$ **then**
      $v(1) = x(1) - \mu$
    else
      $v(1) = -\sigma/(x(1) + \mu)$
    end if
    $\beta = 2v(1)^2/(\sigma + v(1)^2)$
    $v = v/v(1)$
  end if
end procedure

El algoritmo para calcular el vector de Householder requiere 3m flops.

3.3. Algoritmo para calcular Mínimos Cuadrados con reflexiones de Householder

Si $A \in \mathbb{R}^{mxn}$ tiene un rango de columna completo y $b \in \mathbb{R}^m$ entonces el siguiente algoritmo calcula un vector $x_{LS} \in \mathbb{R}^n$ tal que $||Ax_{LS} - b||_2$ es mínimo.

Algorithm 3 Algoritmo para calcular Mínimos Cuadrados con reflexiones de Householder

for $j = 1 : n$ **do**
  $v = \begin{bmatrix} 1 \\ A(j+1:m,j) \end{bmatrix}$
  $\beta = 2/v^T v$
  $b(j:m) = b(j:m) - \beta(v^T b(j:m))v$
end for
Solve $R(1:n, 1:n) \cdot x_{LS} = b(1:n)$

Este método para solucionar mínimos cuadrados requiere $2n^2(m - n/3)$ flops. Los $O(mn)$ flops asociados con la actualización de b y los $O(n^2)$ flops asociados con la

sustitución hacia atrás, no son significativos comparados con el trabajo requerido para factorizar A.

3.4. Algoritmo de sustitución hacia atrás (Row oriented)

Si $U \in \mathbb{R}^{nxn}$ es triangular superior y $b \in \mathbb{R}^n$, entonces el siguiente algoritmo sobrescribe b con la solución a $Ux = b$, U es considerada no singular.

Algorithm 4 Algoritmo de sustitución hacia atrás

$b(n) = b(n)/U(n,n)$
for $i = n-1 : -1 : 1$ **do**
    $b(i) = (b(i) - U(i, i+1:n) \cdot b(i+1:n))/U(i,i)$
end for

Este algoritmo requiere n^2 flops y accesa a U por fila.

4. Implementación y Resultados

Se implementaron en C los algoritmos anteriores en la siguiente manera:

Esquema general de la implementación

Algorithm 5 Código secuencial

Algoritmo para calcular QR con reflexiones de Householder

Para cada columna j se obtienen las entradas de la diagonal hacia abajo de la matriz A:

Paso 1. Algoritmo para calcular el Vector de Householder:

Paso 1.1 Obtener mu y sigma de la columna j (Funcion: mcol)

Paso 1.2 Calcular el vector de Househoulder y β (Funcion: vectorHH)

Paso 2. Calcular la Matriz H (Función: I Bvvt)

Paso 3. Calcular $R = HA$ y actualizar A (Funcion: matrix mul)

Paso 4. Almacenar parte esencial del vector de Householder(Funcion: factor form Q)

Algoritmo para calcular Mínimos Cuadrados con reflexiones de Householder

Para cada renglón i del vector b:

Paso 1. Actualización de b (Función MC)

Algoritmo de sustitución hacia atrás (Función sust a RO)

Se detalla el procedimiento y resultados obtenidos en la implementación de la solución a siguiente problema:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1{,}5 \\ 3 \\ 6 \end{bmatrix}$$

Algoritmo para calcular QR con reflexiones de Householder:

$$MatrizA: \begin{bmatrix} matriz[0][0] = 1{,}00000 & matriz[0][1] = 1{,}00000 & matriz[0][2] = 1{,}00000 \\ matriz[1][0] = 1{,}00000 & matriz[1][1] = 2{,}00000 & matriz[1][2] = 4{,}00000 \\ matriz[2][0] = 1{,}00000 & matriz[2][1] = 3{,}00000 & matriz[2][2] = 9{,}00000 \\ matriz[3][0] = 1{,}00000 & matriz[3][1] = 4{,}00000 & matriz[3][2] = 16{,}00000 \end{bmatrix}$$

Para cada columna j se obtienen las entradas de la diagonal hacia abajo de la matriz A. *Esto provoca que en cada iteración el vector columna se reduzca en una entrada:*

$$Vectorcolumna0: \begin{bmatrix} Vector[0] = 1{,}00000 \\ Vector[1] = 1{,}00000 \\ Vector[2] = 1{,}00000 \\ Vector[3] = 1{,}00000 \end{bmatrix}$$

Paso 1. Algoritmo para calcular el Vector de Householder:

Paso 1.1 Obtener mu y sigma de la columna j (Función: mcol)

Descripción: Calcula la norma euclidiana (mu) del vector columna considerando todas sus entradas y el producto punto (sigma) del vector sin incluir la primera entrada $[v(2:m)]$

$$MU: 2{,}000$$
$$SIGMA: 3{,}000$$

Paso 1.2 Calcular el vector de Househoulder y β (Función: vectorHH)

Descripción: Calcula el vector de Householder realizando lo siguiente:

1. Verificar si x es múltiplo de e_1.
2. Evaluar el signo de x_1.
3. Calcular β.
4. Normalizar el vector $v[1] = 1$ lo cual permitirá almacenar posteriormente la parte esencial del vector de Householer $v(2:m)$ en las entradas 0 debajo de la diagonal obtenidas en A.

$$VectordeHouseholder \begin{bmatrix} Vector[0] = 1{,}00000 \\ Vector[1] = -1{,}00000 \\ Vector[2] = -1{,}00000 \\ Vector[3] = -1{,}00000 \end{bmatrix}$$

BETA: 0.500000
$V(1)$: -1.00000

Paso 2. Calcular la Matriz H (Función: $I - \beta vv^T$)

Descripción: Calcula la matriz $H = I_j - \beta vv^T$. Es importante destacar que, tal como se menciona en la sección 2.2 esta matriz es cuadrada y simétrica. Además, en cada

iteración se reduce su tamaño puesto que la entrada a esta función es el vector de Householder de cada iteración.

Cálculo de H(i)=(I- βvv^T):

$$\begin{bmatrix} H[0][0] = 0{,}50000 & H[0][1] = 0{,}50000 & H[0][2] = 0{,}50000 & H[0][3] = 0{,}50000 \\ H[1][0] = 0{,}50000 & H[1][1] = 0{,}50000 & H[1][2] = -0{,}50000 & H[1][3] = -0{,}50000 \\ H[2][0] = 0{,}50000 & H[2][1] = -0{,}50000 & H[2][2] = 0{,}50000 & H[2][3] = -0{,}50000 \\ H[3][0] = 0{,}50000 & H[3][1] = -0{,}50000 & H[3][2] = -0{,}50000 & H[3][3] = 0{,}50000 \end{bmatrix}$$

Paso 3. Calcular $R = HA$ y actualizar A (Función: matrix mul)

Descripción: Realiza la multiplicación de la matriz H por A y el resultado lo actualiza en A. Es importante mencionar que el orden de los ciclos sigue el orden *jik*, por lo que el acceso a los datos de A es por renglón y β por columna. Se observa que toda la matriz se actualiza y las entradas debajo de la diagonal de la columna en la que se está iterando se actualizan con 0's. Al final de las iteraciones se obtiene una matriz triangular superior.

Cálculo de R=(I-β vv^T)*A:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = 0{,}00000 & matriz[1][1] = -2{,}00000 & matriz[1][2] = -10{,}00000 \\ matriz[2][0] = 0{,}00000 & matriz[2][1] = -1{,}00000 & matriz[2][2] = -5{,}00000 \\ matriz[3][0] = 0{,}00000 & matriz[3][1] = 0{,}00000 & matriz[3][2] = 2{,}00000 \end{bmatrix}$$

Paso 4. Almacenar parte esencial del vector de Householder (Función: factor form Q)

Descripción: El resultado de la función anterior conforma columnas con 0's debajo de la diagonal y esto es aprovechado para almacenar la parte esencial del vector de Householder $v[2:m]$. Es en este momento que se observa la ventaja de haber construido el vector de Householder con 1 en la primera entrada (ya que no es necesario almacenarla).

Factor Form Q y R:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = -2{,}00000 & matriz[1][2] = -10{,}00000 \\ matriz[2][0] = -1{,}00000 & matriz[2][1] = -1{,}00000 & matriz[2][2] = -5{,}00000 \\ matriz[3][0] = -1{,}00000 & matriz[3][1] = 0{,}00000 & matriz[3][2] = 2{,}00000 \end{bmatrix}$$

En las siguientes iteraciones se observa el proceso de reducción del vector columna, así como el proceso de actualización del matriz A.

Submatriz A:

$$\begin{bmatrix} matriz[0][0] = -2{,}00000 & matriz[0][1] = -10{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = -5{,}00000 \\ matriz[2][0] = 0{,}00000 & matriz[2][1] = 2{,}00000 \end{bmatrix}$$

Cálculo del Vector de Householder, MU, SIGMA y V(1) (Evaluación del signo):

$$Vectorcolumna1 : \begin{bmatrix} Vector[0] = -2{,}00000 \\ Vector[1] = -1{,}00000 \\ Vector[2] = 0{,}00000 \end{bmatrix}$$

MU: 2.236,
SIGMA: 1.000

$$VectordeHouseholder : \begin{bmatrix} Vector[0] : 1{,}00000 \\ Vector[1] : 0{,}23607 \\ Vector[2] : -0{,}00000 \end{bmatrix}$$

BETA: 1.894427
$V(1)$: -4.236068

Cálculo de H(i)=(I-βvv^T):

$$\begin{bmatrix} I - Bvvt[0][0] = -0{,}89443 & I - Bvvt[0][1] = -0{,}44721 & I - Bvvt[0][2] = 0{,}00000 \\ I - Bvvt[1][0] = -0{,}44721 & I - Bvvt[1][1] = 0{,}89443 & I - Bvvt[1][2] = 0{,}00000 \\ I - Bvvt[2][0] = 0{,}00000 & I - Bvvt[2][1] = 0{,}00000 & I - Bvvt[2][2] = 1{,}00000 \end{bmatrix}$$

Cálculo de R=(I-βvv^T)*A:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = 2{,}23607 & matriz[1][2] = 11{,}18034 \\ matriz[2][0] = -1{,}00000 & matriz[2][1] = 0{,}00000 & matriz[2][2] = 0{,}00000 \\ matriz[3][0] = -1{,}00000 & matriz[3][1] = 0{,}00000 & matriz[3][2] = 2{,}00000 \end{bmatrix}$$

Factor Form Q y R:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = 2{,}23607 & matriz[1][2] = 11{,}18034 \\ matriz[2][0] = -1{,}00000 & matriz[2][1] = 0{,}23607 & matriz[2][2] = 0{,}00000 \\ matriz[3][0] = -1{,}00000 & matriz[3][1] = 0{,}00000 & matriz[3][2] = 2{,}00000 \end{bmatrix}$$

Submatriz A:

$$\begin{bmatrix} matriz[0][0] = 0{,}00000 \\ matriz[1][0] = 2{,}00000 \end{bmatrix}$$

Cálculo del Vector de Householder, MU, SIGMA y V(1) (Evaluación del signo):

$$Vectorcolumna2 : \begin{bmatrix} Vector[0] = 0{,}00000 \\ Vector[1] = 2{,}00000 \end{bmatrix}$$

MU: 2.000
SIGMA: 4.000

$$VectordeHouseholder : \begin{bmatrix} Vector[0] = 1{,}00000 \\ Vector[1] = -1{,}00000 \end{bmatrix}$$

BETA: 1.000000
$V(1)$: -2.000000

Cálculo de H(i)=(I-βvv^T):

$$\begin{bmatrix} I - Bvvt[0][0] = 0{,}00000 & I - Bvvt[0][1] = 1{,}00000 \\ I - Bvvt[1][0] = 1{,}00000 & I - Bvvt[1][1] = 0{,}00000 \end{bmatrix}$$

Cálculo de R=(I-βvv^T)*A:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = 2{,}23607 & matriz[1][2] = 11{,}18034 \\ matriz[2][0] = -1{,}00000 & matriz[2][1] = 0{,}23607 & matriz[2][2] = 2{,}00000 \\ matriz[3][0] = -1{,}00000 & matriz[3][1] = -0{,}00000 & matriz[3][2] = 0{,}00000 \end{bmatrix}$$

Factor Form Q y R:

$$\begin{bmatrix} matriz[0][0] = 2{,}00000 & matriz[0][1] = 5{,}00000 & matriz[0][2] = 15{,}00000 \\ matriz[1][0] = -1{,}00000 & matriz[1][1] = 2{,}23607 & matriz[1][2] = 11{,}18034 \\ matriz[2][0] = -1{,}00000 & matriz[2][1] = 0{,}23607 & matriz[2][2] = 2{,}00000 \\ matriz[3][0] = -1{,}00000 & matriz[3][1] = -0{,}00000 & matriz[3][2] = -1{,}00000 \end{bmatrix}$$

Algoritmo para calcular Mínimos Cuadrados con reflexiones de Householder

$$Vectorb: \begin{bmatrix} vector[0] = 1{,}00000 \\ vector[1] = 1{,}50000 \\ vector[2] = 3{,}00000 \\ vector[3] = 6{,}00000 \end{bmatrix}$$

Para cada renglón i del vector b:

Paso 1. Actualización de b (Función MC)

Descripción: Se actualiza b ($Q^T b$) haciendo uso de la parte esencial del vector de Householder almacenada previamente. En esta parte del proceso es donde se hace evidente una de las ventajas de las propiedades de los reflectores elementales (ya que no es necesario calcular la forma explícita de Q)

Actualización de la b

β=0.50000

$$\begin{bmatrix} Vectorv[0] = 5{,}75000 \\ Vectorv[1] = -3{,}25000 \\ Vectorv[2] = -1{,}75000 \\ vector[3] = 6{,}00000 \end{bmatrix}$$

β=1.89443

En las siguientes iteraciones se observa el proceso de actualización de b:

$$\begin{bmatrix} Vectorv[1] = 3{,}68951 \\ Vectorv[2] = -0{,}11180 \\ Vectorv[3] = 1{,}25000 \end{bmatrix}$$

β=1.00000

$$\begin{bmatrix} Vectorv[2] = 1{,}25000 \\ Vectorv[3] = -0{,}11180 \end{bmatrix}$$

Vector b actualizado final:

$$\begin{bmatrix} vector[0] = 5{,}75000 \\ vector[1] = 3{,}68951 \\ vector[2] = 1{,}25000 \end{bmatrix}$$

Algoritmo de sustitución hacia atrás (Función sust a RO)

Descripción: Se utiliza el algoritmo de sustitución hacia atrás considerando en su versión Row Oriented para resolver $Rx = b$. Es importante destacar que sólo se utilizan los primeros n renglones de R y las primeras n entradas de b.

Solución $Rx = b$ (Sustitución Hacia Atrás Row Oriented):
$b[2]$ actualizado : $b[2])/R[2][2] = 0{,}62500$

$$\begin{bmatrix} b[1]actualizado = -1{,}47500 \\ b[0]actualizado = 1{,}87500 \end{bmatrix}$$

Solución Mínimos Cuadrados x:

$$\begin{bmatrix} vector[0] = 1{,}87500 \\ vector[1] = -1{,}47500 \\ vector[2] = 0{,}62500 \end{bmatrix}$$

5. Conclusiones

Se demostró a través del algoritmo implementado en este trabajo, la eficiencia de las reflexiones de Householder para la factorización QR de una matriz, así como solucionar mínimos cuadrados, comprobando que la teoría investigada va de acuerdo a los resultados obtenidos. Se pudo verificar la utilidad que representa la estrategia de la sobrescritura de la matriz A (con R y vectores de Householder) y se hizo evidente el beneficio de no almacenar Q en cada iteración.

La estructura del lenguaje de programación de C permitió una mejor definición de las variables, objetos y funciones a utilizar lo cual facilitó la traducción del problema teórico a un algoritmo secuencial además de facilitar el acceso a los datos.

En este trabajo se eligió el método de Householder con el fin de profundizar en las propiedades y ventajas de su uso en la factorización QR para solucionar mínimos cuadrados; sin embargo, existen otros métodos que podrían ser mas eficientes para este tipo de problemas, por ejemplo, el método de SVD que sería de gran interés implementar en un futuro.

Referencias

[1] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.