

Algunos resultados sobre valores y vectores propios de una matriz usando el Método QR

Victor S. Riveros, Ras Patnaik

*División de Postgrado, Facultad de Ingeniería
Universidad del Zulia, Maracaibo, Venezuela*

Resumen

El presente trabajo, tuvo por objeto formular, implementar y aplicar el Método QR, a la resolución del problema algebraico $AX = X$. En donde se analizó y se determinó el contexto teórico, numérico y computacional. Se establecen algunos resultados importantes sobre: sus expresiones iterativas, la convergencia y se implementa el programa HQR en lenguaje Fortran, que permite establecer la efectividad computacional del mismo.

Palabras claves: Iterativo, convergencia, valores propios.

Some results on eigenvalues and eigenvectors of a matrix using the QR method

Abstract

The present research had, as a goal, to formulate, implement and apply QR method to solve the algebraic problem $AX = X$, approach in the theoretical, numerical and computational context. Some important results on the iterative method and computational discussion in Fortran language are established for an efficient computer solution of HQR.

Key words: Iterative, Convergence, Eigenvalues.

Introducción

El cálculo del número de valores propios deseados influye en la escogencia del método a seguir, dichos valores usualmente son calculados primero, y luego usados en el cálculo de los vectores propios.

El procedimiento usual para el cálculo de los valores propios se hace en dos etapas primero transformaciones semejantes usadas para reducir la matriz inicial a una forma sencilla (Tridiagonal o Hessenberg) y segundo la matriz reducida para calcular los valores propios.

La idea de hacer transformaciones matriciales, es una consecuencia directa del resultado de que los valores propios son invariantes bajo

tales transformaciones, las cuales tienden a ser numéricamente estable [1].

Debido a esta situación surge el Método QR como una alternativa eficiente para el cálculo de todos los valores propios de una matriz.

Método QR para una matriz

La base de este método es la sucesiva factorización de una sucesión de matrices $\{A_m\}$ donde éstas mantienen la misma forma de la matriz original (Tridiagonal o Hessenberg) [7].

El procedimiento del método se puede expresar:

Dada una matriz A, cuyos valores propios son deseados (A no singular), existe una factori-

zación $A=QR$ (Q ortogonal y R triangular superior).

Si se examina los efectos acumulativos de m pasos se tiene:

Sea $A_1=A$, se define una sucesión de matrices A_m , Q_m y R_m por:

$$A_m=Q_m R_m ; \quad A_{m+1}=R_m Q_m \quad (1)$$

La sucesión A_m convergerá a una matriz triangular con los valores propios de A sobre su diagonal o una casi triangular cuyos valores pueden ser fácilmente calculados.

Se retoma (1) y se llega:

$$A_{m+1}=Q_m^T A_m Q_m \quad (2)$$

A_{m+1} es ortogonalmente semejante a A_m y así por inducción a $A_1=A$, esto indica que en cada etapa se usan transformaciones ortogonales semejantes.

Las sucesivas iteraciones satisfacen relaciones semejantes que se pueden expresar así:

En toda descomposición $A_m = Q_m R_m$, existe entonces:

a) A_{m+1} es semejante a A_m :

$$A_{m+1} = Q_m^T A_m Q_m \quad m=1,2,\dots$$

b) $A_{m+1} = (Q_1 Q_2 \dots Q_m)^T A_1 (Q_1 Q_2 \dots Q_m)$
 $m=1,2,\dots$

c) Para la matriz ortogonal $P_m=Q_1 Q_2 \dots Q_m$ y la matriz triangular superior $U_m=R_m \dots R_1$ uno tiene $A^m=A^m_1=P_m U_m \quad m=1,2,\dots$

Observación:

El método QR puede ser relativamente costoso, ya que la factorización QR consume tiempo cuando se repite muchas veces. Para decrecer el costo la matriz se prepara para el método reduciéndola a una forma sencilla. En tal sentido se debe reducir la matriz A a una forma (Tridagonal o Hessenberg) usando los procedimientos de Givens u Householder [6].

El método QR con desplazamiento del origen

El método QR, es generalmente aplicado con desplazamiento del origen para el cálculo de

los valores propios en cuanto incrementa la velocidad de convergencia [7].

El uso de dichos desplazamientos reducen drásticamente, el número total de pasos requeridos por el método.

El desplazamiento del origen puede ser con o sin restitución. El proceso con restitución está definido así:

Para una sucesión de constantes

$$K_m, \quad A_1=A \text{ y } A_m - K_m I = Q_m R_m ; \\ R_m Q_m + K_m I = A_{m+1}, \quad m=1,2,\dots \quad (3)$$

Las matrices A_m son semejantes a A_1 , ya que:

$$R_m = Q_m^T (A_m - K_m I); \\ A_{m+1} = K_m I + Q_m^T (A_m - K_m I) \\ Q_m = Q_m^T A_m Q_m, \quad m \geq 1 \quad (4)$$

En la escogencia del factor de desplazamiento es necesario considerar cuando la matriz es real y tiene valores propios reales o tiene algunos valores propios complejos conjugados o cuando la matriz es compleja.

Para matrices reales con valores propios reales se propone escoger K_m calculando los valores propios de la matriz 2×2 de la esquina inferior. Si estos son complejos e iguales a $p_m \pm i q_m$ entonces se toma $k_m = p_m$.

Si ellos son reales e iguales a p_m y q_m se toma k_m igual a p_m o q_m siempre que $|p_m - a_{mm}(m)|$ sea menor o mayor que $|q_m - a_{mm}(m)|$.

Si la matriz original es compleja entonces en general la matriz relevante tendrá dos valores propios complejos p_m y q_m de nuevo la decisión se hace al comparar como en la parte anterior [8].

Se puede considerar el método QR doble este sirve para determinar Q y A_3 sin calcular Q_1 , R_1 , Q_2 , o A_2 de las ecuaciones:

$$A_1 - KI = Q_1 R_1, \quad R_1 Q_1 + KI = A_2 \quad (5)$$

$$A_2 - KI = Q_2 R_2, \quad R_2 Q_2 + KI = A_3 \quad (6)$$

$$A_3 = (Q_1 Q_2)^* A (Q_1 Q_2) = Q^* A Q \quad (7)$$

$$(Q_1 Q_2) (R_2 R_1) = (A_1 - KI) (A_1 - KI) = B(K) \quad (8)$$

Las matrices Q_1, Q_2, R_1, R_2 serán complejas pero Q y $R=R_2R_1$ son reales, ya que ellas corresponden a la factorización de la matriz real $B(K)$, así que A_3 también es real.

$B(K)$ es el producto de 2 matrices Hessenberg superiores, solo los primeros elementos de sus primeras columnas son no nulos. Ellos están dados por:

$$\begin{aligned} b_{11} &= a_{11}^2 - a_{11}(K+K) + KK + a_{12}a_{21} \\ b_{21} &= a_{21}[a_{11} + a_{22} - (K+K)] \\ b_{31} &= a_{32}a_{21} \end{aligned} \tag{9}$$

Al calcularse b_{11}, b_{21} y b_{31} de los valores de A_1 y los desplazamientos K , se calcula P_1 por transformaciones de Householder donde:

$$w = [b_{11} + \text{sgn}(b_{11})(b_{11}^2 + b_{21}^2 + b_{31}^2)^{1/2}, b_{21}, b_{31}, 0, \dots, 0]^T \tag{10}$$

Se considera ahora $P_1A_1P_1$ esta puede reducirse a una forma Hessenberg, por una serie de transformaciones ortogonales, usando Householder.

Un paso completo del algoritmo QR doble es como sigue:

- 1) Calcular $K_1^{(k)} + K_2^{(k)}, K_1^{(k)}K_2^{(k)}$, basado en los valores propios de T_k .
- 2) Calcular b_{11}, b_{21}, b_{31} , usando (10) con $K_1^{(k)}, K_2^{(k)}$ en lugar de k_1, k_2 determinado la matriz P_1 y calcular $P_1A_kP_1$.
- 3) Reducir $P_1A_kP_1$ a una forma Hessenberg usando transformaciones semejantes de Householder obteniendo P_{k+1} [4].

Cabe señalar la existencia de otros tipos de desplazamientos entre los cuales se pueden mencionar. Cociente de Rayleigh, Wilkinson, desplazamiento de Newton, desplazamiento de SAAD [3]. Haciendo una breve descripción de los dos últimos mencionados se puede decir:

El desplazamiento de Newton produce los valores propios de las matrices tridiagonales en forma monótona sin pérdida de la convergencia de segundo orden. Inicialmente la matriz debería ser definida (positiva o negativa). La convergencia puede ser un poco lenta en casos difíciles y la causa radica en que dicho desplazamiento es solo necesario para la estabilidad de la implementación. Mientras que el desplazamiento de

SAAD sólo se propone usar el Método de Newton para calcular un valor propio a la precisión deseada y solo así ejecutar una transformación QL usando el valor propio como desplazamiento. Tal técnica conduce el número total de iteraciones QL cerca de n y permite el cálculo de $P = \prod_{k=1}^n Q_k$ con una reducción del 40% del costo [3].

Descripción del programa HQR

La subrutina HQR tiene como propósito calcular los valores propios de una matriz real Hessenberg superior usando el Método QR. El uso del mismo plantea en su formulación una secuencia empleada (A,N,NP,WR,WI), cuya descripción se puede expresar así:

NP: es un conjunto de variables de entradas enteras igual a la dimensión de la fila del arreglo bidimensional A como se especifica en la formulación de la dimensión para A en el programa.

N: es un conjunto de variables de entrada enteras igual al orden de la matriz A, N no puede ser mayor que NP.

WR, WI: son variables de salida de trabajo con precisión real unidimensional, de dimensiones menores que N, contienen las partes real e imaginarias respectivamente de los valores propios de la matriz Hessenberg.

Si más de 30 iteraciones son requeridas para determinar un valor propio, esta subrutina termina.

Su aplicabilidad radica en determinar los valores propios de una matriz real general, HQR debería ser precedida por la subrutina ELMHES que permite producir una matriz Hessenberg superior para HQR. Para determinar algunos de los vectores propios de una matriz real general, HQR debería ser precedida por ELMHES o ORTHES y seguida por INVIT y ELMBAK; se recomienda en general usar BALANC.

Discutiendo algo del método se tiene: los valores propios son determinados por el Método QR, la esencia de este es un proceso por el cual una sucesión de matrices Hessenberg superior son unitariamente semejante a la matriz Hessenberg original. La razón de convergencia de esta sucesión es mejorada por desplazamientos del origen en cada iteración. La aritmética completa

del proceso es mantenida como real, por combinación de dos iteraciones en una, usando dos desplazamientos o un par de desplazamientos complejos conjugados, la última forma Hessenberg es chequeada para una posible partición de dos submatrices. Si una partición ocurre solo submatrices inferiores participan en la próxima iteración. Los desplazamientos del origen en cada iteración son los valores propios de la matriz de la esquina inferior derecha. Cuando las submatrices principales 1×1 , 2×2 , finalmente se particionan del resto de la matriz, los valores propios de estas son tomados como los valores propios de la matriz original y el algoritmo procede con el resto de las submatrices.

Este proceso se continúa, hasta que la matriz esté particionada completamente en submatrices de orden 1 o 2. La tolerancia en las pruebas de partición son proporcionales a la precisión relativa de la máquina. Algunos de los valores propios puede ser aislados en la diagonal por la subrutina BALANC. Esta información es transmitida a HQR e inmediatamente extrae los valores propios y se aplica el procedimiento QR a la submatriz situada en las filas y columnas de los límites de la matriz balanceada.

La subrutina HQR es numéricamente estable, ya que cada valor propio calculado es exacto para una matriz cercana a la matriz original Hessenberg superior [2,5]

Conclusiones

Con base en el desarrollo de la investigación realizada se llegaron a las siguientes conclusiones.

1) La experiencia obtenida en el análisis del Método QR ha demostrado que el método investigado constituye una alternativa relevante para calcular los valores propios de una matriz.

2) El análisis computacional realizado permitió establecer que la implementación de la subrutina HQR determina que los resultados muestran la efectividad, velocidad del mismo.

Referencias Bibliográficas

1. Burden, R., Faires, J.D.; Análisis Numérico. Iberoamericana, 1.985.
2. Press, W., Flannery, B., Teukokly, S., Vetterling, W., Numerical Recipes. Cambridge, 1.988.
3. Parlett, Beresford N.: The Symmetric Eigenvalue Problem. Prentice Hall, Inc. 1980.
4. Ralston, A., Rabinowitz, P.: A First Course in Numerical Analysis. McGraw-Hill. New York, 1.978.
5. Smith, B., Boyle, J., Dongarra, J., Garbow, B., Ikebe, Y., Klema, V., Moler, C.: Matrix Eigensystem Routine. Eispack Guide. Springer-Verlag, 1.976.
6. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. Springer-Verlag, 1.983.
7. Watkins, David S.: Understanding the QR algorithm. SIAM No. 4, 1.982.
8. Wilkinson, J.: The Algebraic Eigenvalue Problem. Clarendon Press Oxford, 1.965.

Recibido el 03 de Diciembre de 1.990

En forma revisada el 31 de Mayo de 1.993

```

PROGRAM VICTOR
C*****
***
C PROGRAMA DE CALCULO DE VALORES PROPIOS APOYADO EN LA RUTINA
"HQR"
C DESARROLLADO POR
:PRESS,W.,FLANNERY,B.,TEUKOLSKY,S.,VETTERLING,W
C IMPLEMENTADO PARA EL COMPUTADOR/NOVIEMBRE 1990.
C POR EL: LIC. VICTOR S. RIVEROS V.
C
C ***** 29/11/90 *****
C PROGRAMA PRINCIPAL PARA LA RUTINA "HQR"
C*****
***
PARAMETER(NP=5)
DIMENSION A(NP,NP),WR(NP),WI(NP)
DATA
A/1.0,-2.0,3.0,-4.0,-5.0,2.0,3.0,4.0,5.0,6.0,0.0,0.0,50.0,
*-60.0,-70.0,0.0,0.0,0.0,7.0,8.0,0.0,0.0,0.0,0.0,-9.0/
OPEN(3,FILE='TABLAS2.TXT')
WRITE(3,'(/1X,A)') 'MATRIZ:'
DO 11 I=1,NP
WRITE(3,'(1X,5F12.2)') (A(I,J),J=1,NP)
11 CONTINUE
CALL BALANC (A,NP,NP)
CALL ELMHES (A,NP,NP)
CALL HQR (A,NP,NP,WR,WI)
WRITE(3,'(/1X,A)') 'VALORES PROPIOS:'
WRITE(3,'(/1X,T9,A,T24,A/)') 'REAL', 'IMAG.'
DO 12 I=1,NP
WRITE(3,'(1X,I1,2E15.6)') I,WR(I),WI(I)
12 CONTINUE
END
C*****
*****
C LA SUBROUTINA BALANC,BALANCEA UNA MATRIZ REAL/GENERAL Y AISLA
VALORES
C PROPIOS CUANDO SEA POSIBLE
C*****
*****
SUBROUTINE BALANC (A,N,NP)
PARAMETER (RADIX=2,SQRDX=RADIX**2)
DIMENSION A(NP,NP)
1 CONTINUE
LAST=1
DO 14 I=1,N
C=0.
R=0.
DO 11 J=1,N
IF (J.NE.I) THEN
C = C + AFS(A(J,I))
R = R + ABS(A(I,J))
ENDIF
11 CONTINUE
IF (C.NE.0..AND.R.NE.0.)THEN

```

```

      G = R/RADIX
      F = 1
      S = C + R
2     IF (C .LT. G) THEN
          F = F * RADIX
          C = C * SQRDY
      GO TO 2
      ENDIF
      G = R*RADIX
3     IF (C .GT. G) THEN
          F = F/RADIX
          C = C/SQRDY
      GO TO 3
      ENDIF
      IF ((C+R)/F .LT. 0.95 * S) THEN
          LAST = 0
          G = 1./F
          DO 12 J = 1,N
              A(I,J) = A(I,J) * G
12         CONTINUE
          DO 13 J = 1,N
              A(J,I) = A(J,I) * F
13         CONTINUE
          ENDIF
      ENDIF
14     CONTINUE
      IF (LAST .EQ. 0) GO TO 1
      RETURN
      END

```

```

C*****
C****
C LA SUBROUTINA ELMHES, REDUCE UNA MATRIZ REAL GENERAL A UNA FORMA
C HESSENBERG SUPERIOR USANDO TRANSFORMACIONES ELEMENTALES
SEMEJANTES
C*****
C****

```

```

      SUBROUTINE ELMHES (A,N,NP)
      DIMENSION A(NP,NP)
      IF (N .GT. 2) THEN
          DO 17 M=2,N-1
              X = 0.
              I = M
              DO 11 J = M,N
                  IF (ABS(A(J,M-1)) .GT. ABS(X)) THEN
                      X = A(J,M-1)
                      I = J
                  ENDIF
11             CONTINUE
              IF (I .NE. M) THEN
                  DO 12 J = M-1,N
                      Y = A(I,J)
                      A(I,J) = A(M,J)
                      A(M,J) = Y
12             CONTINUE

```

```

DO 13 J = 1,N
    Y = A(J,I)
    A(J,I) = A(J,M)
    A(J,M) = Y
13    CONTINUE
    ENDIF
    IF(X .NE. 0.) THEN
    DO 16 I = M+1,N
        Y = A(I,M-1)
        IF(Y .NE. 0.) THEN
            Y = Y/X
            A(I,M-1) = Y
14            DO 14 J = M,N
                A(I,J) = A(I,J) - Y*A(M,J)
                CONTINUE
            DO 15 J = 1,N
                A(J,M) = A(J,M) + Y*A(J,I)
15            CONTINUE
        ENDIF
16        CONTINUE
    ENDIF
17    CONTINUE
    ENDIF
    RETURN
    END
C*****
****
C LA SUBROUTINA "HQR", CALCULA LOS VALORES PROPIOS DE UNA MATRIZ
REAL
C HESSENBERG SUPERIOR USANDO EL METODO "QR"
C*****
****
    SUBROUTINE HQR (A,N,NP,WR,WI)
    DIMENSION A(NP,NP),WR(NP),WI(NP)
    ANORM=ABS(A(1,1))
    DO 12 I=2,N
        DO 11 J=I-1,N
            ANORM=ANORM+ABS(A(I,J))
11        CONTINUE
    DO 12 CONTINUE
    NN=N
    T=0.
    1 IF(NN .GE. 1) THEN
        ITS=0
        2 DO 13 L=NN,2,-1
            S=ABS(A(L-1,L-1))+ABS(A(L,L))
            IF(S .EQ. 0.) S=ANORM
            IF(ABS(A(L,L-1))+S .EQ. S) GO TO 3
13        CONTINUE
            L=1
            3 X=A(NN,NN)
            IF(L .EQ. NN) THEN
                WR(NN)=X+T
                WI(NN)=0.

```

```

      NN=NN-1
ELSE
  Y=A(NN-1,NN-1)
  W=A(NN,NN-1)*A(NN-1,NN)
  IF(L .EQ. NN-1) THEN
    P=0.5*(Y-X)
    Q=P**2+W
    Z=SQRT(ABS(Q))
    X=X+T
    IF(Q .GE. 0.) THEN
      Z=P*SIGN(Z,P)
      WR(NN)=X+Z
      WR(NN-1)=WR(NN)
      IF(Z .NE. 0.) WR(NN)=X-W/Z
      WI(NN)=0.
      WI(NN-1)=0.
    ELSE
      WR(NN)=X+P
      WR(NN-1)=WR(NN)
      WI(NN)=Z
      WI(NN-1)=-Z
    ENDIF
  NN = NN-2
ELSE
  IF(ITS .EQ. 30) PAUSE 'DEMASIADAS ITERACIONES'
  IF(ITS .EQ. 10 .OR. ITS .EQ. 20) THEN
    T=T+X
    DO 14 I=1,NN
      A(I,I)=A(I,I)-X
    CONTINUE
    S=ABS(A(NN,NN-1))+ABS(A(NN-1,NN-2))
    X=0.75*S
    Y=X
    W=-0.4375*S**2
  ENDIF
  ITS = ITS+1
  DO 15 M=NN-2,L,-1
    Z=A(M,M)
    R=X-Z
    S=Y-Z
    P=(R*S-W)/A(M+1,M)+A(M,M+1)
    Q=A(M+1,M+1)-Z-R-S
    R=A(M+2,M+1)
    S=ABS(P)+ABS(Q)+ABS(R)
    P=P/S
    Q=Q/S
    R=R/S
    IF(M .EQ. L) GO TO 4
    U=ABS(A(M,M-1))*(ABS(Q)+ABS(R))
    V=ABS(P)*(ABS(A(M-1,M-1))+ABS(Z)+ABS(A(M+1,M+1)))
    IF(U+V .EQ. V) GO TO 4
  CONTINUE
  DO 16 I=M+2,NN
    A(I,I-2)=0.

```



```

16      IF(I .NE. M+2) A(I,I-3)=0.
        CONTINUE
DO 19 K=M,NN-1
  IF(K .NE. M) THEN
    P=A(K,K-1)
    Q=A(K+1,K-1)
    R=0.
    IF(K .NE. NN-1) R=A(K+2,K-1)
    X=ABS(P)+ABS(Q)+ABS(R)
    IF(X .NE. 0.) THEN
      P=P/X
      Q=Q/X
      R=R/X
    ENDIF
  ENDIF
  S=SIGN(SQRT(P**2+Q**2+R**2),P)
  IF(S .NE. 0.) THEN
    IF(K .EQ. M) THEN
      IF(L .NE. M) A(K,K-1)=-A(K,K-1)
    ELSE
      A(K,K-1)=-S*X
    ENDIF
    P=P+S
    X=P/S
    Y=Q/S
    Z=R/S
    Q=Q/P
    R=R/P
    DO 17 J=K,NN
      P=A(K,J)+Q*A(K+1,J)
      IF(K .NE. NN-1) THEN
        P=P+R*A(K+2,J)
        A(K+2,J)=A(K+2,J)-P*Z
      ENDIF
      A(K+1,J)=A(K+1,J)-P*Y
      A(K,J)=A(K,J)-P*X
    CONTINUE
17    DO 18 I=L,MIN(NN,K+3)
      P=X*A(I,K)+Y*A(I,K+1)
      IF(K .NE. NN-1) THEN
        P=P+Z*A(I,K+2)
        A(I,K+2)=A(I,K+2)-P*R
      ENDIF
      A(I,K+1)=A(I,K+1)-P*Q
      A(I,K)=A(I,K)-P
18    CONTINUE
19  CONTINUE
    GO TO 2
  ENDIF
ENDIF
GO TO 1
ENDIF
RETURN

```

END

Tiempo de compilación 20.49 seg

Tiempo de ejecución 02.58 seg

MATRIZ:

1.00	2.00	0.00	0.00	0.00
-2.00	3.00	0.00	0.00	0.00
3.00	4.00	50.00	0.00	0.00
-4.00	5.00	-60.00	7.00	0.00
-5.00	6.00	-70.00	8.00	-9.00

VALORES PROPIOS:

	REAL	IMAG.
1	0.500000E+02	0.000000E+00
2	0.200000E+01	-0.173205E+01
3	0.200000E+01	0.173205E+01
4	0.709996E+02	0.000000E+00
5	0.699994E+01	0.000000E+00