Implementación del esquema de deflación para ecuaciones algebraicas

Pablo Santamaría

v0.1 (Julio 2007)

Introducción

Consideremos el problema de determinar *todos* los ceros de una *ecuación algebraica*

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m, \quad a_0 \neq 0.$$

De acuerdo al *teorema fundamental de la aritmética*, esta ecuación tiene, en el campo complejo, exactamente m raíces, $\alpha_1, \alpha_2, \ldots, \alpha_n$ y $p(x)$ se factoriza como

$$p(x) = a_0(x - \alpha_1)(x - \alpha_2)\cdots(x - \alpha_m)$$

Si los coeficientes $a_0, \ldots, a_m$ son todos reales, entonces las posibles raices complejas aparecerán en pares conjugados, y el polinomio $p(x)$ se factorizará, en el campo real, como el producto de factores lineales y polinomios cuadráticos irreducibles.

En un apunte anterior[1] vimos que para ecuaciones algebraicas el método de Newton puede ser eficientemente implementado si la evaluación del polinomio (y su derivada) es realizada por el método iterativo de Horner. El procedimiento resultante es conocido como *método de Birge–Vieta* y su implementación como una subrutina de FORTRAN se encuentra en dicho apunte.

Ahora bien, el método de Horner nos da también un camino para factorizar el polinomio y hallar, así, sus raíces. En efecto, recordemos que la evaluación de $p(x)$ en un punto x_n por la regla de Horner procede computando

$$\begin{cases} b_m = a_m \\ b_k = a_k + b_{k+1} x_n \quad k = m-1, \ldots, 0 \end{cases}$$

siendo, entonces

$$b_0 = p(x_n).$$

Pero además, si construimos el polinomio de grado $m-1$

$$q(x) = b_1 + b_2 x^2 + \cdots + b_{m-1} x^{m-2} + b_m x^{m-1}$$

entonces

$$p(x) = (x - x_n) q(x) + b_0$$

Ahora bien, supongamos que hemos encontrado por el método de Bierge–Vieta una raíz $x_n = \alpha$ de $p(x)$, entonces $b_0 = 0$ y las restantes raíces son también raíces del polinomio $q(x)$. Para calcular una de estas raíces aplicamos el método de Bierge–Vieta ahora a $q(x)$, obteniendo la raíz y un factor polinómico de un grado menor cuyas raíces son también raíces de $p(x)$. El proceso se repite tantas veces como raíces reales tenga el polinomio, obteniendo en cada paso polinomios de grado

[1] Ver *Subrutinas para la resolución de ecuaciones de una variable.*

cada vez menor. Si al final del proceso obtenemos un polinomio de grado mayor que 1, éste será irreductible en el campo real y sus raíces complejas deben ser determinadas con algún método apropiado[2]. El procedimiento descrito se conoce como *deflación*.

Nótese que en la discusión anterior no hemos tenido en cuenta el hecho de que x_n no es exactamente una raíz de $p(x)$, y hemos ignorado cualquier error de redondeo introducido en el cómputo de los coeficientes $b_0, b_1, \ldots, b_m$. Estos errores pueden producir que las raíces que calculamos de los sucesivos polinomios factorizados se alejen cada vez más de las raíces del polinomio original $p(x)$. Puede mostrarse que los errores cometidos son despreciables si: (i) las raíces son determinadas ordenándolas por su *incremento* en valor absoluto (es decir, primero las que tienen valores absolutos más chicos) y (ii) cada raíz es determinada a su máxima precisión alcanzable. En todo caso, una forma simple de solventar el problema consiste en utilizar cada raíz determinada por el procedimiento de deflación como aproximación inicial para la ecuación original $p(x) = 0$ y efectuar al menos una última iteración.

Implementación del esquema de deflación

Para implementar el esquema de deflación tenemos que modificar primero nuestra subrutina original del método de Bierge–Vieta de manera tal que ahora devuelva también los coeficientes b_k para construir el polinomio factorizado $q(x)$ requerido en cada paso. A continuación presentamos la subrutina incorporando tal modificación.

```
C       ------------------------------------------------
        SUBROUTINE BIRGE_VIETA (A,B,M,X0,TOL,NMAX,RAIZ,ICLAVE)
C       ------------------------------------------------
C       METODO DE BIRGE-VIETA para resolver ECUACIONES
C       ALGEBRAICAS:
C
C                       P (x) = 0
C
C       donde P es un polinomio de grado n de coeficientes
C       reales; basado en el método de Newton-Raphson
C       implementando el esquema de Horner para la evalua-
C       ción del polinomio y su derivada.
C       ------------------------------------------------
C       Bloque de identificación de argumentos
C       ------------------------------------------------
C       A       = Arreglo unidimensional de 0 a M que
C                 contiene los coeficientes del polinomio
C       B       = Arreglo unidimensional de 0 a M que
C                 contiene los coeficientes del polinomio
C                 factor resultante de la division sintetica
C                 (Nota: B(0) = P(RAIZ)=~ 0)
C       X0      = Aproximación inicial
C       M       = Grado del polinomio
C       NMAX    = Número máximo de iteraciones
C       TOL     = Tolerancia para el error relativo
C       RAIZ    = Estimación de la solución
C       ICLAVE  = Clave de éxito
C                 0 = éxito
C                 1 = iteraciones excedidas
```

[2]Si el polinomio es de grado 2 la determinación de las raíces, claramente, se reduce a utilizar la fórmula usual para ecuaciones cuadráticas.

```
C                2 = división por cero
C       ------------------------------------------------
C       Bloque de declaración de tipo
C       ------------------------------------------------
        IMPLICIT NONE
        INTEGER M, NMAX
        DOUBLE PRECISION A(0:M), B(0:M)
        DOUBLE PRECISION X0,TOL,RAIZ
        INTEGER ICLAVE
C       ------------------------------------------------
        INTEGER I, J
        DOUBLE PRECISION C, ERROR
        DOUBLE PRECISION TINY
        PARAMETER (TINY=1.0D-25)
C       ------------------------------------------------
        DO I=1,NMAX
C           -------------------------
C           Esquema de Horner
C           -------------------------
            B(M) = A(M)
            C    = A(M)
            DO J=M-1,1,-1
                B(J) = B(J+1)*X0+A(J)
                C    = B(J)+X0*C
            ENDDO
            B(0) = B(1)*X0+A(0)
            IF (ABS(C).LE.TINY) THEN
                ICLAVE = 2
                RETURN
            ENDIF
C           -------------------------
C           Método de Newton
C           -------------------------
            RAIZ  = X0 - B(0)/C
            ERROR = ABS((RAIZ-X0)/RAIZ)
            IF (ERROR.LT.TOL) RETURN
            X0 = RAIZ
        END DO
        ICLAVE = 1
        RETURN
        END
```

Con la subrutina anterior podemos construir fácilmente un programa para implementar el método de deflación. Inicialmente el usuario ingresa el grado del polinomio, sus coeficientes y aproximaciones iniciales, suficientemente buenas, para cada una de las raíces reales. Entonces el programa obtiene mejores aproximaciones de las raíces (dentro de una tolerancia prefijada en el programa) y, paso a paso, factoriza el polinomio.

```
        PROGRAM DEFLACION
C       ------------------------------------------
        IMPLICIT NONE
C       ------------------------------------------
        INTEGER MMAX            ! Grado máximo del polinomio
        PARAMETER (MMAX=50)
```

```
      INTEGER NMAX            ! Número máximo de iteraciones
      PARAMETER (NMAX=100)
      DOUBLE PRECISION TOL    ! Precisión para las raíces
      PARAMETER (TOL=1.0D-20)
C     -----------------------------------------
      INTEGER I, J, NRAIZ, M, ICLAVE
      DOUBLE PRECISION RAIZ
      DOUBLE PRECISION A(0:MMAX), B(0:MMAX), X(MMAX)
C     -----------------------------------------
C     Leer coeficientes del polinomio
C     -----------------------------------------
      WRITE(*,'(A,$)') ' Grado del polinomio: '
      READ(*,*) M
      WRITE(*,'(A,$)') ' Ingrese coeficientes: '
      READ(*,*) (A(I), I=0,M)
C     -----------------------------------------
C     Leer las aproximaciones de las raices
C     -----------------------------------------
      WRITE(*,'(A,$)') ' Número de raíces reales a considerar: '
      READ(*,*) NRAIZ
      IF (NRAIZ.GT.M) THEN
          WRITE(*,*) ' El número de raíces no puede exceder al grado'
          STOP
      ENDIF
      WRITE(*,'(A,$)') ' Ingrese aproximaciones iniciales: '
      READ(*,*) (X(I), I=1,NRAIZ)
C     -----------------------------------------
C     Imprimir polinomio original
C     -----------------------------------------
      WRITE(*,*) '--------------------------'
      WRITE(*,*) 'Grado del polinomio = ', M
      WRITE(*,*) 'Coeficientes:'
      WRITE(*,*) (A(J),J=0,M)
C     -----------------------------------------
C     Proceder con la factorización
C     -----------------------------------------
      DO I=1,NRAIZ
C         ---------------------------------------
C         Efectuar el método de Birge Vieta
C         ---------------------------------------
          RAIZ   = 0.0D0
          ICLAVE = 0
          CALL BIRGE_VIETA(A,B,M,X(I),TOL,NMAX,RAIZ,ICLAVE)
          IF (ICLAVE.NE.0) THEN
              WRITE(*,*) 'Error. No se puede continuar'
              STOP
          ENDIF
C         ---------------------------------------
C         Imprimir resultados
C         ---------------------------------------
          WRITE(*,*) '-------------------'
          WRITE(*,*) 'Factorizacion ', I
          WRITE(*,*) '-------------------'
          WRITE(*,*) 'Raiz = ', RAIZ
```

```
            WRITE(*,*) 'Grado del factor = ', M-1
            WRITE(*,*) 'Coeficientes:'
            WRITE(*,*) (B(J),J=1,M)
C           ---------------------------------------
C           Tomar nuevo polinomio
C           ---------------------------------------
            M = M-1
            DO J = 0,M
                A(J) = B(J+1)
            ENDDO
        ENDDO
        STOP
        END
```

Un ejemplo

Como ejemplo, consideremos la ecuación algebraica

$$p(x) = -2 - x - x^2 - x^3 + x^4 = 0$$

El gráfico del polinomio nos muestra que hay dos raíces reales, y que $\tilde{x}_1 = -1{,}5$ y $\tilde{x}_2 = 2{,}5$ son, respectivamente, buenas aproximaciones para ellas[3]. Al ejecutar nuestro programa obtenemos lo siguiente.

```
 Grado del polinomio:  4
 Ingrese coeficientes: -2.0 -1.0 -1.0 -1.0 1.0
 Número de raíces reales a considerar:  2
 Ingrese aproximaciones iniciales:  -1.5 2.5
 --------------------------
 Grado del polinomio =  4
 Coeficientes:
 -2. -1. -1. -1.  1.
 --------------------
 Factorizacion  1
 --------------------
 Raiz =  -1.
 Grado del factor =  3
 Coeficientes:
 -2.  1. -2.  1.
 --------------------
 Factorizacion  2
 --------------------
 Raiz =   2.
 Grado del factor =  2
 Coeficientes:
  1.  0.  1.
```

De este modo, la factorización de nuestro polinomio es

$$p(x) = (x+1)(x-2)(1+x^2)$$

siendo $x_1 = -1$ y $x_2 = 2$ las dos raíces reales. Puesto que $1 + x^2 = (x+i)(x-i)$, resulta que las dos restantes raíces (complejas) son $x_3 = i, x_4 = -i$.

[3]Los ceros *exactos* son (no se lo digan a nadie) $x_1 = -1, x_2 = 2, x_3 = i, x_4 = -i$.