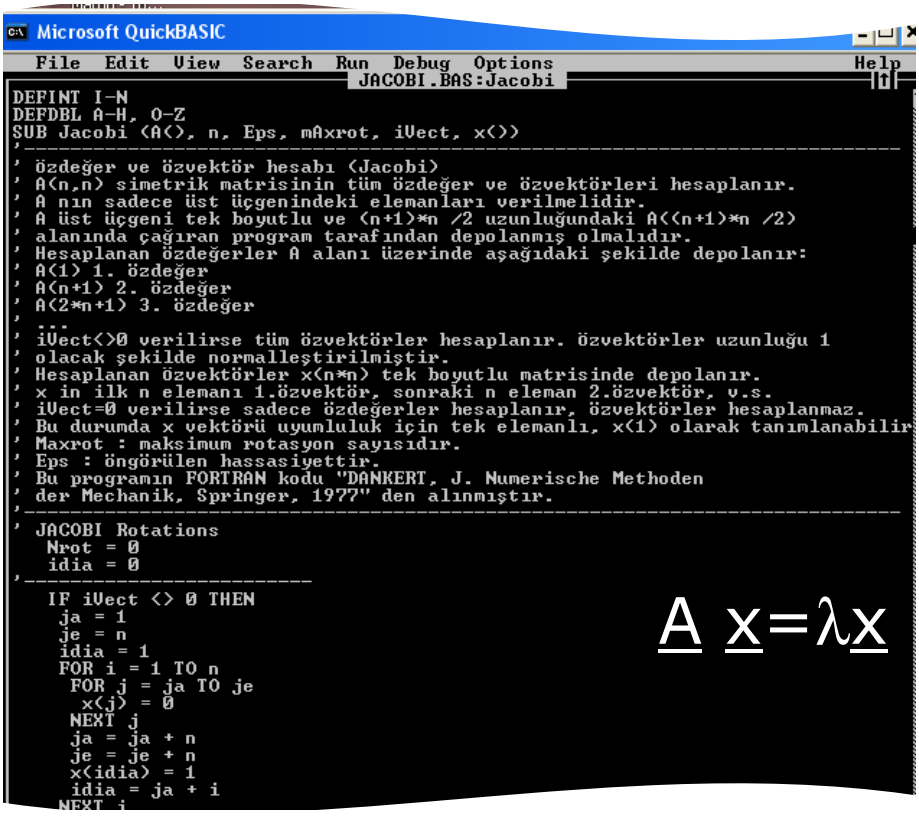
ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
Mühendislik Mimarlık Fakültesi
İnşaat Mühendisliği Bölümü
E-Posta: ogu.ahmet.topcu@gmail.com
Web: http://mmf2.ogu.edu.tr/atopcu

Bilgisayar Destekli
Nümerik Analiz

Ders notları 2014

Ahmet TOPÇU



30

PROGRAMLAR: Özdeğer ve özvektör hesabı

- Tüm özdeğerler ve özvektörler - Jacobi

30. PROGRAMLAR: Tüm özdeğer ve özvektörlerin hesabı[1] - Jacobi

Jacobi alt programı

$$\underline{A}\underline{x} = \lambda \underline{x}$$

standart özdeğer probleminin tüm özdeğerini ve bunlara ait normalleştirilmiş özvektörlerini Jacobi rotasyon yöntemi ile hesaplar. $\underline{A}_{nxn}$ simetrik olmak zorundadır.

$\underline{A}$ nın alt üçgen kısmının satırları tek boyutlu A((n+1)·n/2) alanında, çağıran programda aşağıdaki gibi depolanmış olmalıdır:

$$\underline{A} = \begin{bmatrix} a_{11} & & & & \\ a_{21} & a_{22} & & \text{simetrik} & \\ a_{31} & a_{32} & a_{33} & & \\ . & & & & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \rightarrow [a_{11}\ a_{21}\ a_{22}\ a_{31}\ a_{31}\ a_{33}\ \dots\ a_{n1}\ a_{n2}\ a_{n3}\ \dots a_{nn}]$$

1.satır, 2.satır, 3.satır, n.satır

Özvektörler için x(n·n) tek boyutlu alanı çağıran programda boyutlandırılmış olmalıdır. Hesap sonrası özdeğerler A alanı, özvektörler x alanında depolanır. A(1) 1.özdeğer, a(n+1)·2. özdeğer, a(2n+1)·3. özdeğer, … v.s. dir. x alanının ilk n elemanı 1. özvektör, sonraki n elemanı 2. özvektör, … v.s dir.

Maxrot izin verilen maksimum rotasyon sayısı, Eps özdeğerler için ulaşılmak istenen hassasiyettir. ivect parametresi=0 verilirse sadece özdeğerler, ivect≠0 verilirse özdeğer ve özvektörler hesaplanır.

Jacobi her özvektörü uzunluğu 1 olacak şekilde normalleştirir. Özvektörün en büyük elemanı 1 olacak şekilde normalleştirilmesi istenirse JNormalize alt programının Jacobi alt programının hemen arkasından çağrılması gerekir.

Örnek:

$$\underline{A} = \begin{bmatrix} 9 & & & & \\ 2 & 10 & & \text{simetrik} & \\ 7 & 4 & 7 & & \\ 3 & 1 & 5 & 8 & \\ 4 & 2 & 1 & 3 & 6 \end{bmatrix} \rightarrow [9\ 2\ 10\ 7\ 4\ 7\ 3\ 1\ 5\ 8\ 4\ 2\ 13\ 6]$$

Bellekte depolanışı

Jacobi alt programının sonucu

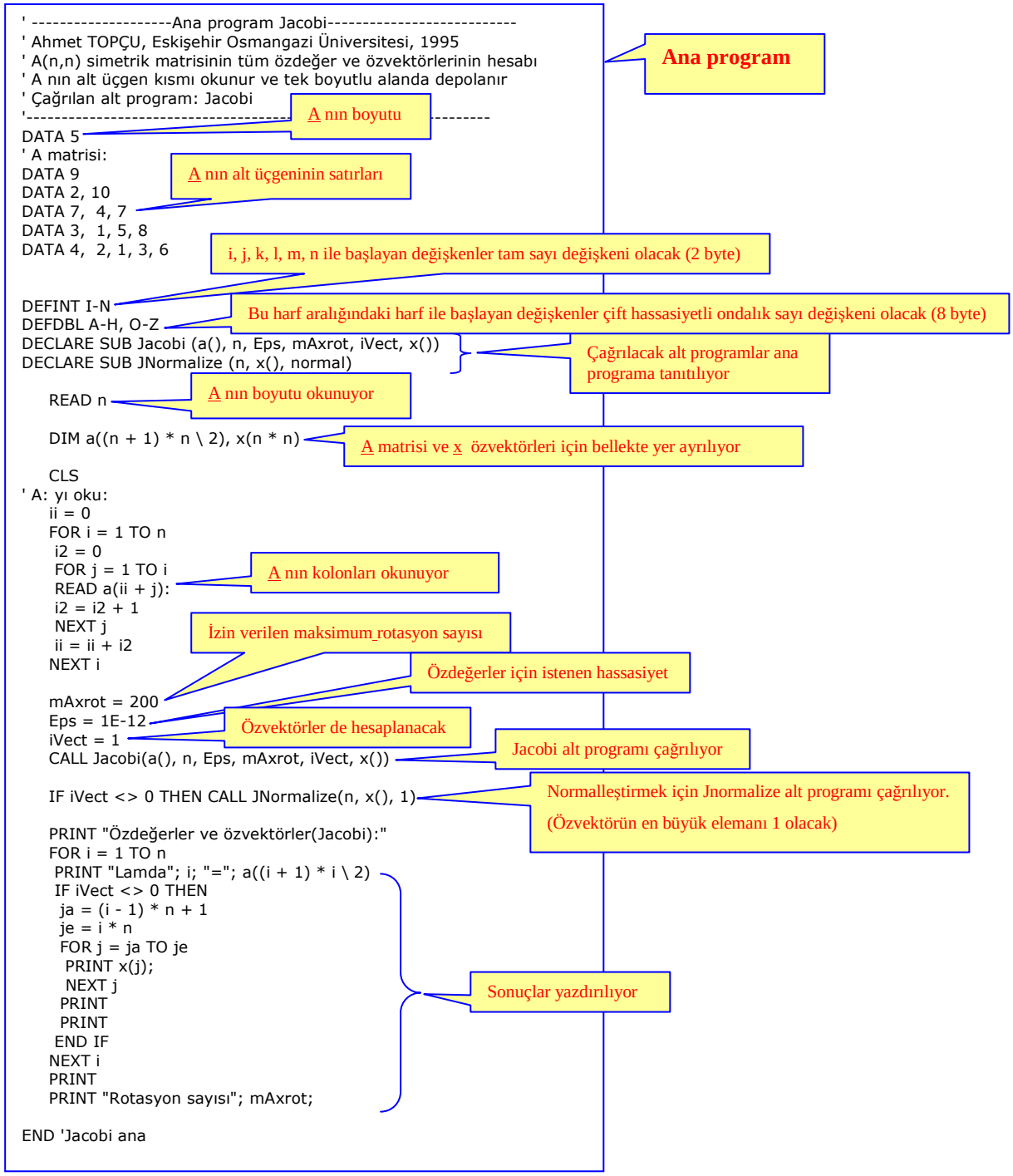
```
C:\WINDOWS\system32\cmd.exe - qbx
özdeğerler ve özvektörler(Jacobi):
Lamda 1 = 21.5209263072947
 1   .668230691722249   .962039355865446   .749231168506461   .550623981818188

Lamda 2 = 8.70489838600009
-.264061165045631   1 -.027113138148636 -.415229694331677 -.121648518491214

Lamda 3 =-1.00201832414679
-.723056643884942 -.290257575027028   1 -.466523942291979   .553028194615816

Lamda 4 = 5.44834488041317
-.860801671822246   .262011037823883 -.475209665138297   1   .714929061249172

Lamda 5 = 5.32784875043879
 .424644084954822 -2.75718425725509D-02 -.531438286300854 -.594715205978181   1

Rotasyon sayısı 35
```

En büyük elemanı 1 olacak şekilde normalleştirilmiş özvektörler

Özdeğerler

$$\underline{\Lambda} = \begin{bmatrix} 21.5209 & & & & \\ & 8.7049 & & & \\ & & -1.0020 & & \\ & & & 5.4483 & \\ & & & & 5.3278 \end{bmatrix}, \quad \underline{X} = \begin{bmatrix} 1 & -0.2641 & -0.7231 & -0.8608 & 0.4246 \\ 0.6682 & 1 & -0.2903 & 0.2620 & -0.0276 \\ 0.9620 & -0.0271 & 1 & -0.4752 & -0.5314 \\ 0.7492 & -04152 & -0.4665 & 1 & -0.5947 \\ 0.5506 & -0.1216 & 0.5530 & 0.7149 & 1 \end{bmatrix}$$

[1] Teori ve örnekler için bak: bölüm 25


' --------------------Ana program Jacobi--------------------------
' Ahmet TOPÇU, Eskişehir Osmangazi Üniversitesi, 1995
' A(n,n) simetrik matrisinin tüm özdeğer ve özvektörlerinin hesabı
' A nın alt üçgen kısmı okunur ve tek boyutlu alanda depolanır
' Çağrılan alt program: Jacobi
'---
DATA 5
' A matrisi:
DATA 9
DATA 2, 10
DATA 7, 4, 7
DATA 3, 1, 5, 8
DATA 4, 2, 1, 3, 6
DEFINT I-N
DEFDBL A-H, O-Z
DECLARE SUB Jacobi (a(), n, Eps, mAxrot, iVect, x())
DECLARE SUB JNormalize (n, x(), normal)
READ n
DIM a((n + 1) * n \ 2), x(n * n)
CLS
' A: yı oku:
ii = 0
FOR i = 1 TO n
i2 = 0
FOR j = 1 TO i
READ a(ii + j):
i2 = i2 + 1
NEXT j
ii = ii + i2
NEXT i
mAxrot = 200
Eps = 1E-12
iVect = 1
CALL Jacobi(a(), n, Eps, mAxrot, iVect, x())
IF iVect <> 0 THEN CALL JNormalize(n, x(), 1)
PRINT "Özdeğerler ve özvektörler(Jacobi):"
FOR i = 1 TO n
PRINT "Lamda"; i; "="; a((i + 1) * i \ 2)
IF iVect <> 0 THEN
ja = (i - 1) * n + 1
je = i * n
FOR j = ja TO je
PRINT x(j);
NEXT j
PRINT
PRINT
END IF
NEXT i
PRINT
PRINT "Rotasyon sayısı"; mAxrot;
END 'Jacobi ana
Ana program
A nın boyutu
A nın alt üçgeninin satırları
i, j, k, l, m, n ile başlayan değişkenler tam sayı değişkeni olacak (2 byte)
Bu harf aralığındaki harf ile başlayan değişkenler çift hassasiyetli ondalık sayı değişkeni olacak (8 byte)
Çağrılacak alt programlar ana programa tanıtılıyor
A nın boyutu okunuyor
A matrisi ve x özvektörleri için bellekte yer ayrılıyor
A nın kolonları okunuyor
İzin verilen maksimum rotasyon sayısı
Özdeğerler için istenen hassasiyet
Özvektörler de hesaplanacak
Jacobi alt programı çağrılıyor
Normalleştirmek için Jnormalize alt programı çağrılıyor.
(Özvektörün en büyük elemanı 1 olacak)
Sonuçlar yazdırılıyor

```
SUB Jacobi (a(), n, Eps, mAxrot, iVect, x())
'-------------------------------------------------------------------------------
' Özdeğer ve özvektör hesabı (Jacobi)
' A(n,n) simetrik matrisinin tüm özdeğer ve özvektörleri hesaplanır.
' A nın sadece alt üçgenindeki satırları verilmelidir.
' A alt üçgeni tek boyutlu ve (n+1)*n /2 uzunluğundaki A((n+1)*n /2)
' alanında çağıran program tarafından depolanmış olmalıdır.
' Hesaplanan özdeğerler A alanı üzerinde aşağıdaki şekilde depolanır:
' A(1) 1. özdeğer
' A(n+1) 2. özdeğer
' A(2*n+1) 3. özdeğer
' ...
' iVect<>0 verilirse tüm özvektörler hesaplanır. Özvektörler uzunluğu 1
' olacak şekilde normalleştirilmiştir.
' Hesaplanan Özvektörler x(n*n) tek boyutlu matrisinde depolanır.
' x in ilk n elemanı 1.özvektör, sonraki n eleman 2.özvektör, v.s.
' iVect=0 verilirse sadece özdeğerler hesaplanır, özvektörler hesaplanmaz.
' Bu durumda x vektörü uyumluluk için tek elemanlı, x(1) olarak tanımlanabilir.
' Maxrot : maksimum rotasyon sayısıdır.
' Eps : öngörülen hassasiyettir.
' Bu programın FORTRAN kodu "DANKERT, J. Numerische Methoden
' der Mechanik, Springer, 1977" den alınmıştır.
'-------------------------------------------------------------------------------
' JACOBI Rotations
   Nrot = 0
   indi = 0
'-----------------------------
   IF iVect <> 0 THEN
    ja = 1
    je = n
    idia = 1
    FOR i = 1 TO n
     FOR j = ja TO je
      x(j) = 0
     NEXT j
     ja = ja + n
     je = je + n
     x(idia) = 1
     idia = ja + i
    NEXT i
   END IF
'------------------------
71 s = 0
   ja = 2
   je = 0
   FOR i = 2 TO n
    je = je + i
    FOR j = ja TO je
     s = s + a(j) ^ 2
    NEXT j
    ja = ja + i
   NEXT i
   s = SQR(2 * s)
   IF indi = 1 THEN EXIT SUB
80 s = s / n
90 ja = 2
   je = 0
   ind = 0
   FOR i = 2 TO n
    je = je + i
    jj = 1
    FOR j = ja TO je
     apq = a(j)
     IF ABS(apq) < s GOTO 10
     Nrot = Nrot + 1
     ind = 1
     ipp = (jj + 1) * jj / 2
     app = a(ipp): aqq = a(je + 1)
     tht = (aqq - app) * .5 / apq
     IF tht = 0 THEN
      T = 1
     ELSE
      T = 1 / (SQR(1 + tht * tht) * SGN(tht) + tht): ' Sign
     END IF
     co = 1 / SQR(T * T + 1)
     si = co * T
     ii = ipp - jj + 1
     ke = j - 1
     IF ke >= ja THEN
     FOR k = ja TO ke
      aip = a(ii)
      aiq = a(k)
      a(ii) = aip * co - aiq * si
      a(k) = aip * si + aiq * co
      ii = ii + 1
     NEXT k
     END IF
```

Jacobi alt programı

Sonraki sayfada devam ediyor

Jacobi devamı

```
    ke = j + 1
    IF je >= ke THEN
     ii = ipp
     iErh = jj
     FOR k = ke TO je
      ii = ii + iErh
      aip = a(ii)
      aiq = a(k)
      a(ii) = aip * co - aiq * si
      a(k) = aip * si + aiq * co
      iErh = iErh + 1
     NEXT k
    END IF
    IF i <= n THEN
     kk = je + 1
     ii = j
     iErh = i
     ke = i + 1
     FOR k = ke TO n
      kk = kk + iErh
      ii = ii + iErh
      aip = a(ii)
      aiq = a(kk)
      a(ii) = aip * co - aiq * si
      a(kk) = aip * si + aiq * co
      iErh = iErh + 1
     NEXT k
    END IF
'---------------------------------
' Eigenvectors
    IF iVect <> 0 THEN
     ii = (i - 1) * n + 1
     ka = (jj - 1) * n
     ke = ka + n
     ka = ka + 1
     FOR k = ka TO ke
      aip = x(k)
      aiq = x(ii)
      x(k) = aip * co - aiq * si
      x(ii) = aip * si + aiq * co
      ii = ii + 1
     NEXT k
    END IF
'---------------------------------
    T = a(j) * si * co * 2
    co = co * co
    si = si * si
    aip = a(ipp)
    aiq = a(je + 1)
    a(ipp) = aip * co + aiq * si - T
    a(je + 1) = aip * si + aiq * co + T
    a(j) = 0
10   jj = jj + 1
   NEXT j
   ja = ja + i
  NEXT i
  IF ind = 1 GOTO 90
  IF s * n < Eps THEN mAxrot = Nrot: indi = 1: GOTO 71
  IF Nrot < mAxrot GOTO 80
  mAxrot = -Nrot
  indi = 1
  GOTO 71

END SUB ' Jacobi
```

```
SUB JNormalize (n, x(), normal)
' X(n*n) alanında depolanmış özvektörler normalleştirilir.
' Normal<>0: özvektürün en büyük eleman 1 olacak şekilde normalleştir.
' Normal=0: hiç bir şey yapma, çık
  IF normal = 0 THEN EXIT SUB
  FOR i = 1 TO n
   ja = (i - 1) * n + 1
   je = i * n
   Dmaxx = 0
   FOR j = ja TO je
    IF ABS(x(j)) > ABS(Dmaxx) THEN Dmaxx = x(j)
   NEXT j
   Dmaxx = 1 / Dmaxx
   FOR j = ja TO je
    x(j) = x(j) * Dmaxx
   NEXT j
  NEXT i

END SUB ' iNormalize
```

JNormalize Alt programı