



**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

Mühendislik Mimarlık Fakültesi

İnşaat Mühendisliği Bölümü

E-Posta: [ogu.ahmet.topcu@gmail.com](mailto:ogu.ahmet.topcu@gmail.com)

Web: <http://mmf2.ogu.edu.tr/atopcu>

## Bilgisayar Destekli

# Nümerik Analiz

*Ders notları 2014*

**Ahmet TOPÇU**

```
C:\Basic\QBASIC.EXE
File Edit View Search Run Debug Calls Utility Options Help
SKYLINE.BAS:CholeskySkyline
DEFINT I-N
DEFDBL A-H, O-Z
SUB CholeskySkyline (n, a(), b(), kH(), iHata)
'-----
' Lineer denklem sistemi çözümü
' CHOLESKY Skyline(Ufuk çizgisi) çözüm tekniği
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, Eskişehir, 1998
' A(n,n)*x(n)=b(n) denklem sistemini ufuk çizgisi çözüm
' tekniği ile çözer. A simetrik ve pozitif tanımlı matristir.
' A'nın kolonlarındaki diyagonal ile ufuk çizgisi
' arasındaki elemanları sıra ile A tek boyutlu matrisinde
' depolanmış olmalıdır.
' b(n) sağ tarafı b tek boyutlu matrisinde depolanmış olmalıdır.
' A'nın diyagonal-ufuk çizgisi yüksekliği=kolonların eleman sayısı
' kH(n) vektöründe depolanmış olmalıdır.
' n: denklem sayısı
' iHata: hata değişkeni. Matris pozitif tanımlı ise iHata=0 olarak
' aksi halde iHata<>0 olarak döner. iHata<>0 durumunda çözüm yoktur
' ve çözümün kesildiği denklem numarasını belirtir.
' FORTRAN kodu BATHE, k.j. Page 448 den alınmıştır
'-----
  DIM MaxA(n + 1): ' Diyagonal elemanların adresleri
  iHata = 0
  EpsMach
  EpsMach = 1
  DO
    EpsMach = EpsMach / 2
    s = 1 + EpsMach
    LOOP UNTIL s <= 1
    EpsMach = 2 * EpsMach
    Zero = EpsMach
  ' diyagonal elemanların adresleri
  MaxA(1) = 1
  Dmax = ABS(a(1))
  IF n > 1 THEN
    FOR i = 2 TO n + 1
      MaxA(i) = MaxA(i - 1) + kH(i - 1)
    NEXT i
  END IF

  FOR i = 1 TO n
    kn = MaxA(i)
```

# 17

**PROGRAMLAR:**

CHOLESKY Skyline(ufuk çizgisi) tekniği

## 17. PROGRAMLAR: CHOLESKY Skyline(ufuk çizgisi) tekniği

Mühendislikte karşılaşılan problemlerin çoğunda  $\underline{A} \underline{x} = \underline{b}$  denklem sisteminin  $\underline{A}_{n \times n}$  katsayılar matrisi simetrik ( $\underline{A}^T = \underline{A}$ ), pozitif tanımlı ( $\underline{x}^T \underline{A} \underline{x} > 0$ ) ve bant şeklindedir. Ayrıca bant içindeki kolonların yüksekliği (=kolonun diyagonal elemanı ile aynı kolonun en yukarıdaki sıfırdan farklı son elemanı arasındaki eleman sayısı) azdır. Kolonların en yukarıdaki sıfırdan farklı elemanlarının yatay çizgilerle birleştirilmesi sonucu oluşan hatta ufuk çizgisi(skyline) denilmektedir. Denklem sistemi

$$\underline{A}_{n \times n} \underline{x}_{n \times 1} = \underline{b}_{n \times 1} \rightarrow \begin{array}{cccccccc} x & x & x & & & & & \\ & x & x & x & & & & \\ & & x & x & & & & \\ & & & x & x & & & \\ & & & & x & x & & \\ & & & & & x & x & \\ & & & & & & x & x \\ & & & & & & & x \\ & & & & & & & \cdot \\ & & & & & & & \cdot \\ & & & & & & & x \end{array} \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \cdot \\ x_n \end{array} = \begin{array}{c} b_1 \\ b_2 \\ b_3 \\ \cdot \\ b_n \end{array}$$

olsun. Bilgisayar belleğinden olabildiğince tasarruf sağlamak ve sıfırlar ile dört işlem yapmamak için özel bir depolama tekniği kullanılır:  $\underline{A}$  nın üst üçgen matrisinin sadece ufuk çizgisi altındaki sayıları depolanır. Her kolonun, diyagonal elemandan başlanarak kolonun ufuk çizgisine kadar olan elemanları tek boyutlu A matrisinde satır olarak depolanır.

$$\underline{A}_{n \times n} = \begin{array}{cccccccc} x & x & x & & & & & \\ & x & x & x & & & & \\ & & x & x & & & & \\ & & & x & x & & & \\ & & & & x & x & & \\ & & & & & x & x & \\ & & & & & & x & x \\ & & & & & & & x \\ & & & & & & & \cdot \\ & & & & & & & \cdot \\ & & & & & & & x \end{array}$$

Simetrik

Ufuk çizgisi(skyline)

$$\underline{A} \rightarrow [x | x \ x | x \ x | x \ x \ x | x \ x \ x \ x | x \ x \ x \ x \ x | x \ x \cdot x \ x \ x \ x]$$

1. kolon    2. kolon    3. kolon    4. kolon    n. kolon

A nın kolonlarının tek boyutlu matriste depolanması

Çözümün yapılabilmesi için kolon yüksekliklerinin (kolonun ufuk çizgisi altındaki eleman sayısı) bilinmesi gerekir. Programdaki  $k_H$  vektörü bu amaca yöneliktir.  $k_H$  vektöründe kolonların yükseklikleri aşağıdaki gibi depolanır. Yukarıdaki  $\underline{A}$  matrisi için örnek:

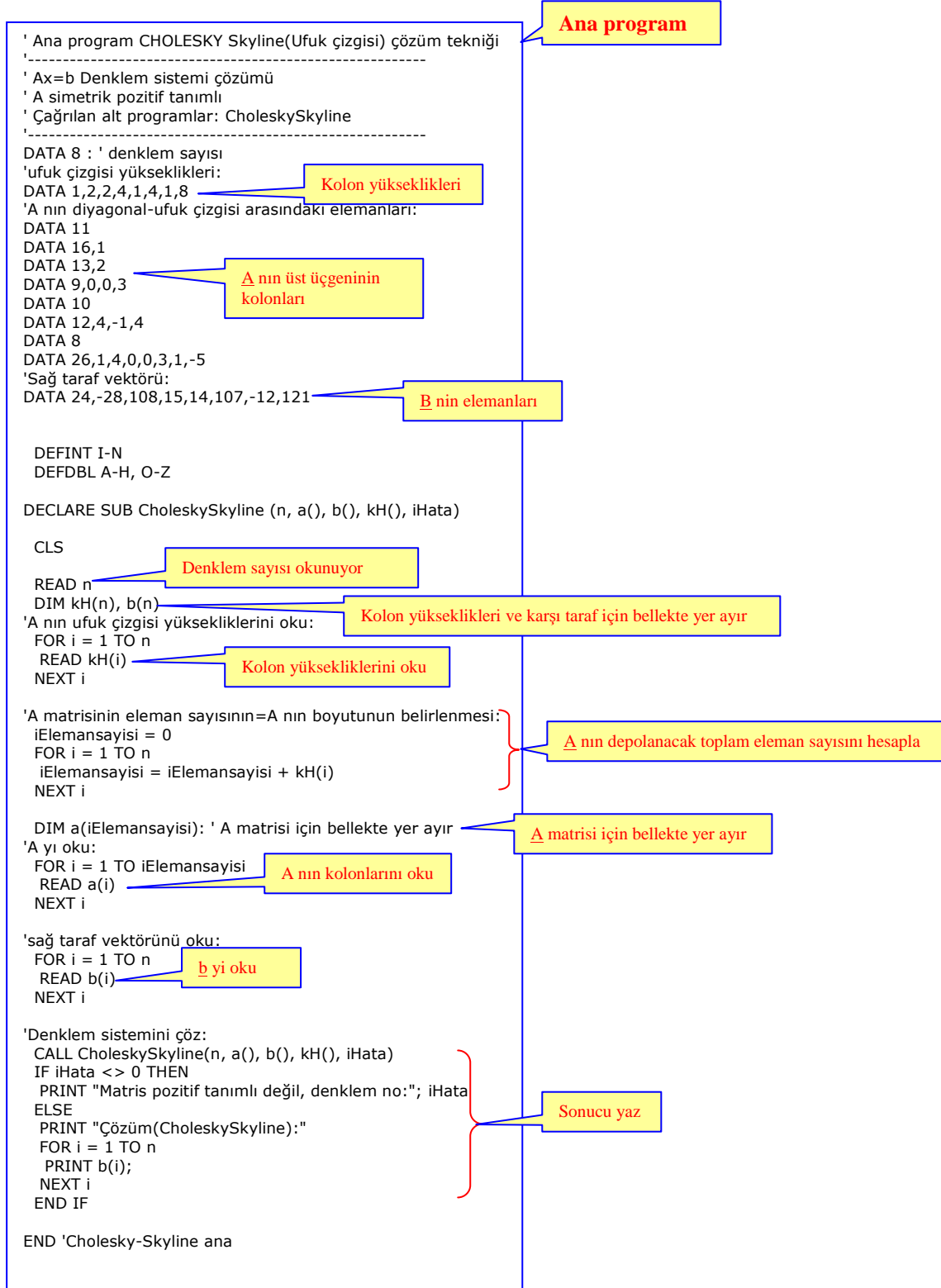
$$k_H \rightarrow [1 \ 2 \ 2 \ 4 \ 2 \ 2 \ 6 \ \dots]$$

Kolon yükseklikleri

$$\text{Karşı taraf vektörünün depolanması: } \underline{b}_n \rightarrow [b_1 \ b_2 \ \dots \ b_n]$$

Çözüm için CHOLESKY metodu kullanılır. Aşağıda QBASIC kodu verilen CholeskySkyline alt programı ufuk çizgisi tekniği ile depolanmış katsayılar matrisi  $\underline{A}$  yı, kolon yükseklikleri vektörü  $k_H$  yı, karşı taraf vektörü  $\underline{b}$  yi ve denklemin sayısı  $n$  yi parametre olarak alır,  $\underline{x}$  bilinmeyenler vektörünü hesaplar.  $\underline{x}$  çözümü  $\underline{b}$  vektöründe depolanır.





```
SUB CholeskySkyline (n, a(), b(), kH(), iHata)
```

```
' Lineer denklem sistemi çözümü
' CHOLESKY Skyline(Ufuk çizgisi) çözüm tekniği
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, Eskişehir, 1998
'  $A(n,n) \cdot x(n) = b(n)$  denklem sistemini ufuk çizgisi çözüm
' tekniği ile çözer. A simetrik ve pozitif tanımlı matristir.
' A'nın kolonlarındaki diyagonal ile ufuk çizgisi
' arasındaki elemanları sıra ile A tek boyutlu matrisinde
' depolanmış olmalıdır.
' b(n) sağ tarafı b tek boyutlu matrisinde depolanmış olmalıdır.
' A'nın diyagonal-ufuk çizgisi yüksekliği=kolonların eleman sayısı
' kH(n) vektöründe depolanmış olmalıdır.
' n: denklem sayısı
' iHata: hata değişkeni. Matris pozitif tanımlı ise iHata=0 olarak
' aksi halde iHata<>0 olarak döner. iHata<>0 durumunda çözüm yoktur
' ve çözümün kesildiği denklem numarasını belirtir.
' FORTRAN kodu BATHE, k.j, Page 448 den alınmıştır
```

```
-----
DIM MaxA(n + 1): ' Diyagonal elemanların adresleri
iHata = 0
EpsMach
EpsMach = 1
DO
EpsMach = EpsMach / 2
s = 1 + EpsMach
LOOP UNTIL s <= 1
EpsMach = 2 * EpsMach
Zero = EpsMach
```

```
' diyagonal elemanların adresleri
```

```
MaxA(1) = 1
Dmax = ABS(a(1))
IF n > 1 THEN
FOR i = 2 TO n + 1
MaxA(i) = MaxA(i - 1) + kH(i - 1)
NEXT i
END IF
```

```
FOR i = 1 TO n
kn = MaxA(i)
kl = kn + 1
ku = MaxA(i + 1) - 1
kH = ku - kl
IF kH < 0 GOTO 110
```

```
IF kH > 0 THEN
k = i - kH
ic = 0
KLT = ku
FOR j = 1 TO kH
ic = ic + 1
KLT = KLT - 1
ki = MaxA(k)
nd = MaxA(k + 1) - ki - 1
IF nd > 0 THEN
IF ic < nd THEN kk = ic ELSE kk = nd
T = 0
FOR L = 1 TO kk
T = T + a(ki + L) * a(KLT + L)
NEXT L
a(KLT) = a(KLT) - T
END IF
k = k + 1
NEXT j
END IF
```

```
k = i
b = 0
FOR kk = kl TO ku
k = k - 1
ki = MaxA(k)
T = a(kk) / a(ki)
b = b + T * a(kk)
a(kk) = T
NEXT kk
a(kn) = a(kn) - b
```

```
110 IF a(kn) <= Zero THEN iHata = i: EXIT SUB
NEXT i
```

**CholeskySkyline  
alt programı**

```
' Aşağı doğru hesap
FOR i = 1 TO n
kl = MaxA(i) + 1
ku = MaxA(i + 1) - 1
IF ku - kl >= 0 THEN
k = i
T = 0
FOR kk = kl TO ku
k = k - 1
T = T + a(kk) * b(k)
NEXT kk
b(i) = b(i) - T
END IF
NEXT i
' Yukarı doğru hesap
FOR i = 1 TO n
k = MaxA(i)
b(i) = b(i) / a(k)
NEXT i
IF n = 1 THEN EXIT SUB
i = n
FOR L = 2 TO n
kl = MaxA(i) + 1
ku = MaxA(i + 1) - 1
IF ku - kl >= 0 THEN
k = i
FOR kk = kl TO ku
k = k - 1
b(k) = b(k) - a(kk) * b(i)
NEXT kk
END IF
i = i - 1
NEXT L
END SUB ' CholeskySkyline
```