



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

```
C:\Basic\QBASIC.EXE
File Edit View Search Run Debug Calls Utility Options Help
CG.BAS:CG
DEFINT I-N
DEFDBL A-H, O-Z
SUB CG (n, a(), b(), x(), iTer, iHata)
'-----
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, ESKİŞEHİR, 2006
' Ax=b doğrusal denklem sistemi CG (Conjugate Gradient) iterasyon
' metodu ile çözülür, x bilinmeyenler vektörü hesaplanır.
'-----
' Veriler:
' n: Denklem sayısı
' a(n,n): simetrik ve pozitif tanımlı katsayılar matrisi. a n'ın
' sadece üst üçgen kısmı kullanılır
' b(n): karşıtaraf vektörü
'-----
' Çıktılar:
' x(n): Bilinmeyenler vektörü
' iTer: iterasyon sayısı
' iHata:
' = 0: Çözüm bulundu
' = 1: Çözüm bulunamadı(a tekildir veya pozitif tanımlı değildir)
' = 2: Çözüm yakınsamadı
'-----
' r(n), s(n) ve aSi(n) ara değer vektörleridir
' Orijinal adı: cg Verfahren
'-----
DIM r(n)
DIM s(n)
DIM aSi(n) ' a * s çarpımı depolanır
' EpsMach: Kullanılan bilgisayarda sıfır varsayılan en küçük sayı
EpsMach = 1
DO
EpsMach = EpsMach / 2
s = 1 + EpsMach
LOOP UNTIL s <= 1
EpsMach = 2 * EpsMach
Epsilon = EpsMach
iHata = 0
iTer = 0
n=1 durumu
```

22

PROGRAMLAR:

**İterasyon ile denklem sistemi çözümü
CG (Conjugate Gradient) metodu**

22. PROGRAMLAR: İterasyon ile denklemler sistemi çözümü-CG (Conjugate Gradient) metodu¹

Ekte verilen program $\underline{A} \underline{x} = \underline{b}$ lineer cebirsel denklemler sistemini CG iterasyon yöntemi ile çözer. $\underline{A}_{n \times n}$ simetrik-pozitif tanımlı matrisi ve \underline{b}_n vektörü çağırarak programda depolanmış olmalı ve ayrıca \underline{x}_n vektörü de boyutlandırılmış olmalıdır.

CG metodu sadece simetrik ve pozitif tanımlı matrisler içindir. Bu koşulu sağlamayan denklemler sistemlerinde kullanılmaz! Maksimum iterasyon sayısı, teorik olarak, n kadardır. Ancak, yuvarlama hataları nedeniyle fazladan 1-2 iterasyon daha gerekebilir.

İterasyon sırasında \underline{A} nın pozitif tanımlı olmadığı anlaşılırsa veya n+2 iterasyon sonunda çözüm yakınsamazsa CG alt programı iHata $\neq 0$ ile döner.

\underline{A} simetrik ve pozitif tanımlı ise, CG metodu en uygun iterasyon metodudur. İterasyon sırasında \underline{A} nın elemanları değişmediğinden, bellekte çözümü mümkün olmayan çok büyük denklemler sistemlerinin çözümünde öncelikle tercih edilir.

Örnek:

$$\begin{bmatrix} 4 & 1 & 1 & 0 \\ 1 & 4 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ 0 & 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}, \underline{x} = ?$$

GG alt programının sonucu

```

C:\Basic\QBASIC.EXE
Denklemler sisteminin çözümü(CG):
.1666666666666667 .4166666666666667 -8.33333333333333D-02 .1666666666666667
İterasyon sayısı= 2
  
```

$$\underline{x} = \begin{bmatrix} 0.1667 \\ 0.4167 \\ -0.0833 \\ 0.1667 \end{bmatrix}$$

¹ Teori ve sayısal örnekler için bak: bölüm 7

```
'----- Ana program CG(Conjugate Gradient Metodu)-----
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, ESKİŞEHİR, 2006
' a(n,n)*x(n)=b(n) doğrusal denklem sistemi CG (Conjugate Gradient)
' iterasyon metodu ile çözülerek x(n) vektörü hesaplanır.
' a ve b matrisleri çağırılan programda depolanmış ve x
' vektörü boyutlandırılmış olmalıdır.
' a(n,n): simetrik pozitif tanımlı matris(sadece üst üçgen gerekir)
' b(n): karşı taraf vektörü
' x(n): bilinmeyenler vektörü
' n: Denklem sayısı
' Çağrılan alt programlar: CG
'-----
```

Ana program

```
DATA 4
' a matrisi:
DATA 4,1,1,0
DATA 4,0,1
DATA 4,1
DATA 4
```

A'nın üst üçgen kısmının satırları

```
' b karşı taraf vektörü:
DATA 1,2,0,1
```

Karşı taraf vektörü b

```
DEFINT I-N
DEFDBL A-H, O-Z
DECLARE SUB CG (n, a(), b(), x(), iTer, iHata)
```

```
CLS
```

```
READ n
```

Denklem sayısı okunuyor

```
DIM a(n, n), x(n), b(n)
```

A matrisi, b vektörü ve x çözüm vektörü için bellekte yer ayır

```
' a'nın üst üçgen kısmını oku
FOR i = 1 TO n
  FOR j = i TO n
    READ a(i, j)
  NEXT j
NEXT i
```

A'nın üst üçgen kısmı okunuyor

```
' b yi oku
FOR i = 1 TO n
  READ b(i)
NEXT i
```

b okunuyor

CG alt programı çağırılıyor

```
CALL CG(n, a(), b(), x(), iTer, iHata)
IF iHata <> 0 THEN PRINT "HATA(CG)= "; iHata: END
```

```
PRINT "Denklem sisteminin çözümü(CG):"
FOR i = 1 TO n
  PRINT x(i);
NEXT i
PRINT
PRINT "İterasyon sayısı="; iTer
```

Çözüm yazdırılıyor

```
END ' CG ana sonu
```

```
SUB CG (n, a(), b(), x(), iTer, iHata)
```

```
'-----
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, ESKİŞEHİR, 2006
' Ax=b doğrusal denklem sistemi CG (Conjugate Gradient)
' iterasyon metodu ile çözülür, x bilinmeyenler vektörü hesaplanır.
```

```
' Veriler:
```

```
' n: Denklem sayısı
' a(n,n): simetrik ve pozitif tanımlı katsayılar matrisi. a n'ın
' sadece üst üçgen kısmı kullanılır
' b(n): karşıtaraf vektörü
```

```
' Çıktılar:
```

```
' x(n): Bilinmeyenler vektörü
' iTer: iterasyon sayısı
' iHata:
' = 0: çözüm bulundu
' = 1: çözüm bulunamadı(a tekildir veya pozitif tanımlı değildir)
' = 2: n+2 iterasyon yapıldı fakat çözüm yakınsamadı
```

```
' r(n), s(n) ve aSi(n) ara değer vektörleridir
```

```
'-----
DIM r(n)
DIM s(n)
DIM aSi(n) ' a * s çarpımı depolanır
```

```
' EpsMach: Kullanılan bilgisayarda sıfır varsayılan en küçük sayı
```

```
EpsMach = 1
DO
  EpsMach = EpsMach / 2
  s = 1 + EpsMach
LOOP UNTIL s <= 1
EpsMach = 2 * EpsMach
Epsilon = EpsMach
```

```
iHata = 0
iTer = 0
n=1 durumu
IF n = 1 THEN
  IF a(1, 1) = 0 THEN
    iHata = 1
  ELSE
    x(1) = b(1) / a(1, 1)
  EXIT SUB
END IF
END IF
```

```
' başlangıç vektörleri x, r, s hazırla
FOR i = 1 TO n
  x(i) = 0
  r(i) = b(i)
  s(i) = r(i)
NEXT i
```

```
' B nin öklid normu
bNorm = 0
FOR i = 1 TO n
  bNorm = bNorm + b(i) * b(i)
NEXT i
IF bNorm = 0 THEN EXIT SUB
bNorm = SQR(bNorm)
```

```
' İterasyona başla
```

```
iTer = 1
DO
' Hesapla: Alfa = [s(transpoz)*r]/[s(transp)*(a*s)]
Pay = 0
Payda = 0
FOR i = 1 TO n
  Pay = Pay + s(i) * r(i)
  T = 0
  FOR j = i TO n
    T = T + a(i, j) * s(j)
  NEXT j
  FOR j = 1 TO i - 1
    T = T + a(j, i) * s(j)
  NEXT j

  aSi(i) = T
  Payda = Payda + s(i) * T
NEXT i
```

CG alt programı

```
' a matrisi tekil mi? Veya pozitif tanımlı değil mi? Kontrol et
IF Payda <= 0 THEN
  iHata = 1
  EXIT SUB
END IF
Alfa = Pay / Payda
```

```
' Hesapla: x=x+Alfa*s ve r=r-Alfa*aSi
' Hesapla: rNorm=r nin öklid normu
rNorm = 0
FOR i = 1 TO n
  x(i) = x(i) + Alfa * s(i)
  r(i) = r(i) - Alfa * aSi(i)
  rNorm = rNorm + r(i) * r(i)
NEXT i
rNorm = SQR(rNorm) / bNorm
```

```
' Yakınsamayı kontrol et (en az iki iterasyondan sonra)
IF iTer >= 2 AND rNorm <= Epsilon THEN EXIT SUB
```

```
' Hesapla: Beta = -[r(transpoz)*(a*s)]/[s(transpoz)*(a*s)]
Pay = 0
FOR i = 1 TO n
  Pay = Pay + r(i) * aSi(i)
NEXT i
Beta = -Pay / Payda
```

```
' Hesapla: s = r + Beta * s
FOR i = 1 TO n
  s(i) = r(i) + Beta * s(i)
NEXT i
```

```
iTer = iTer + 1
LOOP UNTIL iTer > n + 2
```

```
iHata = 2
```

```
END SUB ' CG sonu
```