



**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
PCG.BAS:PCG
DEFINT I-N
DEFDBL A-H, O-Z
SUB PCG (n, a(), b(), x(), iTer, iHata)
-----
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, ESKİŞEHİR, 2006
' Ax=b doğrusal denklem sistemi PCG (Preconditioned Conjugate Gradient)
' iterasyon metodu ile çözülür, x bilinmeyenler vektörü hesaplanır.
'
' Veriler:
' n: Denklem sayısı
' a(n,n): simetrik ve pozitif tanımlı katsayılar matrisi. a nın
' 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
'
' d(n), r(n), s(n) ve aSi(n) ara değer vektörleridir
'
' çağrılan programlar: yok
-----
DIM d(n)
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
```

# 23

**PROGRAMLAR:**

**İterasyon ile denklem sistemi çözümü  
PCG (Preconditioned Conjugate Gradient) metodu**

## 23. PROGRAMLAR: iterasyon ile denklem sistemi çözümü - PCG (Preconditioned Conjugate Gradient) metodu

Ekte verilen program simetrik katsayılı  $\underline{A} \underline{x} = \underline{b}$  lineer cebirsel denklem sistemini PCG iterasyon yöntemi ile çözer. Çözüm öncesi  $\underline{A}$  matrisinin kondisyonu iyileştirilir. İyileştirme  $\underline{A}$  nın diyagonal elemanlarından oluşan  $\underline{D}$  diyagonal matrisi yardımıyla yapılır:

$$\underline{A} \rightarrow \text{Kondisyon iyileştirmesi} \rightarrow \underline{D}^{-1/2} \underline{A} \underline{D}^{-1/2} \underline{D}^{1/2} \underline{x} = \underline{D}^{-1/2} \underline{b}$$

Burada  $\underline{D}^{-1/2}$  matrisine  $\underline{D}$  nin karekökü denir ve

$$\underline{D} = \begin{bmatrix} a_{11} & & & & \\ & a_{22} & & & \\ & & \dots & & \\ & & & & a_{nn} \end{bmatrix} \rightarrow \underline{D}^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{a_{11}}} & & & & \\ & \frac{1}{\sqrt{a_{22}}} & & & \\ & & \dots & & \\ & & & & \frac{1}{\sqrt{a_{nn}}} \end{bmatrix}$$

dir.

$\underline{A}$  nın simetrik-pozitif tanımlı matrisi ve  $\underline{b}_n$  vektörü çağırılan programda depolanmış olmalı ve ayrıca  $\underline{x}_n$  vektörü de boyutlandırılmış olmalıdır.

PCG metodu sadece simetrik ve pozitif tanımlı matrisler içindir. Bu koşulu sağlamayan denklem 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 PCG alt programı iHata  $\neq 0$  ile döner.

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

### Örnek:

$$\begin{bmatrix} 33 & 0 & -10 & 0 & 0 \\ & 43 & 0 & 14 & 0 \\ & & 28 & 0 & 1 \\ \text{Sim} & & & 55 & 0 \\ & & & & 67 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 13 \\ 76 \\ 206 \\ 137 \end{bmatrix}, \underline{x} = ?$$

PCG alt programının sonucu

```
C:\Basic\QBASIC.EXE
Denklem sisteminin çözümü(PCG):
1 -1 3 4 2
İterasyon sayısı= 5
```

$$\underline{x} = \begin{bmatrix} 1 \\ -1 \\ 3 \\ 4 \\ 2 \end{bmatrix}$$

```

'----- Ana program PCG(Preconditioned Conjugate Gradient Method)-----
' Dr. Ahmet TOPÇU, Osmangazi Üniversitesi, ESKİŞEHİR, 2006
'  $a(n,n)*x(n)=b(n)$  doğrusal denklem sistemi PCG (Preconditioned Conjugate
' Gradient) iterasyon metodu ile çözülür,  $x(n)$  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: PCG
-----
DATA 5
' a matrisi
DATA 33,0,-10,0,0
DATA 43,0,14,0
DATA 28,0,1
DATA 55,0
DATA 67
' b matrisi
DATA 3,13,76,206,137

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

CLS
READ n
DIM a(n, n), x(n), b(n)
' a nın üst üçgen kısmını oku
FOR i = 1 TO n
  FOR j = i TO n
    READ a(i, j)
  NEXT j
NEXT i
' b yi oku
FOR i = 1 TO n
  READ b(i)
NEXT i

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

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

END ' PCG ana sonu

```

PCG ana programı

A'nın boyutu

A'nın üst üçgen kısmının satırları

Karşı taraf vektörü b

Denklem sayısı okunuyor

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

A'nın üst üçgen kısmı okunuyor

b okunuyor

PCG alt programı çağırılıyor

Çözüm yazdırılıyor

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

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

```
' Veriler:
```

```
' n: Denklemler sayısı
' a(n,n): simetrik ve pozitif tanımlı katsayılar matrisi. a'nın
' 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 tekdirdir veya pozitif tanımlı değildir)
' = 2: n+2 iterasyon yapıldı fakat çözüm yakınsamadı
```

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

```
' çağrılan programlar: yok
```

```
'-----
DIM d(n)
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
```

```
' Kondisyonun iyileştirme(preconditioning)
FOR i = 1 TO n
  d(i) = 1 / SQR(a(i, i))
NEXT i
FOR i = 1 TO n
  FOR j = i TO n
    a(i, j) = d(i) * a(i, j) * d(j)
  NEXT j
  b(i) = b(i) * d(i)
NEXT i
```

```
' 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)
```

PCG alt programı

```
' İterasyona başla
iTer = 1
DO
' Hesapla: Alfa = [s(transpoz)*r]/[s(transpoz)*(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

' 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
    FOR i = 1 TO n
      x(i) = d(i) * x(i)
    NEXT i
    EXIT SUB
  END IF

' 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 ' PCG sonu
```