FİZ433 FİZİKTE BİLGİSAYAR UYGULAMALARI

(DERS NOTLARI)

Hazırlayan:

Prof.Dr. Orhan ÇAKIR

Ankara Üniversitesi, Fen Fakültesi, Fizik Bölümü

Ankara, 2017

İÇİNDEKİLER

KONU 4

LİNEER DENKLEM SİSTEMLERİNİN ÇÖZÜLMESİ II

Gauss-Seidel Yöntemi

Bu yöntem Jacobi yönteminin biraz değiştirilmesi ile elde edilmiş tekrarlamalı bir yöntemdir. Lineer denklem sisteminin çözümleri için ilk tahminler aynen uygulanır, fakat ikinci ve diğer tahminler en son hesaplanan tahminler kullanılarak uygulanır.

$$x_i^{(k+1)} = \frac{b_i \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}$$

Yukarıdaki bağıntıdan da görüleceği gibi i=1 olduğunda sağ-tarafta sadece (k) üs indisi gelmektedir ve i=n olduğunda ise sadece (k+1) üs indisi gelir. Burada i'nin 1 ile n arasındaki ara değerlerinde hem (k), hem de (k+1) üs indisli terimler gelmektedir. Gauss-Seidel yöntemi Jacobi yöntemine göre daha hızlı yakınsamaktadır. Yöntemi uygulayan alt program aşağıdaki gibidir.

- **FORTRAN altprogram**

```
Subroutine GS(a,b,n,x,iterm,tol)

implicit real*8 (a-h,o-z)

Dimension a(n,n),b(n),x(n)

Logical fl, tolex

iter=0

tolex=.true.

do i=1,n
```

```
top=0.
Do j=1,n
if(i.ne.j) then
top=top+abs(a(i,j))
endif
enddo
if(abs(a(i,i)).lt.top)  then
fl=.true.
print*,i,". diagonal baskin degil !"
endif
enddo
if(fl.eqv..true.) then
do i=1,n
do k=i,n
if(abs(a(k,i)).gt.abs(a(i,i))) then
do l=1,n
gec=a(k,l)
a(k,l)=a(i,l)
a(i,l)=gec
enddo
gec=b(k)
b(k)=b(i)
b(i)=gec
endif
enddo
enddo
endif
```

```
 do i=1,n
 x(i)=b(i)/a(i,i)
enddo
 do while(tolex.eqv..true..and.iter.lt.iterm)
 do i=1,n
 xo=x(i)
 tolex=.false.
 top=b(i)
 do j=1,n
 if(i.ne.j) then
 top=top-a(i,j)*x(j)
 endif
 enddo
 x(i)=top/a(i,i)
 if(abs(x(i)-xo).gt.abs(xo*tol)) then
 tolex=.true.
endif
 enddo
 iter=iter+1
enddo
return
end
```

Bu alt program Ax=B şeklindeki denklem sistemini çözer. Girdi parametreleri: n denklem sayısı, a(n,n) katsayılar matrisi, b(n) eşitliğin sağındaki vektör, tol istenen tolerans, iterm maksimum iterasyon sayısıdır. Çıktı parametresi ise x(n) dir.

LU Ayrıştırma Yöntemi

Bir A matrisi aşağı üçgen matris (L) ile yukarı üçgen matrisin (U) çarpımı şeklinde yazılabilir. Bu durumda Lineer denklem sistemi

$$
\begin{aligned}
&Ax = B \\
&(LU)x = B \\
&L(Ux) = B
\end{aligned}
$$

şeklinde yazılabilir. Burada önce y için Ly=B denklemini, sonra x çözümleri için de Ux=y denklemini çözmeliyiz. Bu işlemler için çarpımları A matrisini verecek L ve U matrislerini bulmalıyız. A matrisinin LU ayrışması yapıldıktan sonra çözümler geri yerine koymakla bulunabilir. LU algoritmasına göre N^2+N bilinmeyen içeren N^2 denklem yazılabilir:

$$\sum_{k=1}^{N} l_{ik} u_{kj} = a_{ij}$$

Burada kolay çözüm bulmak için keyfi olarak u_{ii}=1 veya l_{ii}=1 alınabilir.

$$
\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}
$$

Böylece diğer elemanlar aşağıdaki ifadelerle belirlenebilir, burada l_{ii}=1 alınmıştır.

$$u_{11} = a_{11}\ , \quad u_{12} = a_{12}\ , \quad u_{13} = a_{13}\ , \quad u_{14} = a_{14}$$
$$u_{22} = a_{22} - l_{21}u_{12}$$
$$u_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23}$$
$$u_{44} = a_{44} - l_{41}u_{14} - l_{42}u_{24} - l_{43}u_{34}$$
$$u_{23} = a_{23} - l_{21}u_{13}$$
$$u_{24} = a_{24} - l_{21}u_{14}$$
$$u_{34} = a_{34} - l_{31}u_{14} - l_{32}u_{24}$$
$$l_{21} = a_{21} / u_{11}\ , \quad l_{31} = a_{31} / u_{11}\ , \quad l_{41} = a_{41} / u_{11}$$
$$l_{32} = (a_{32} - l_{31}u_{12}) / u_{22}$$
$$l_{42} = (a_{42} - l_{41}u_{12}) / u_{22}$$
$$l_{43} = (a_{43} - l_{41}u_{13} - l_{42}u_{23}) / u_{33}$$

Bu işlemler genelleştirilerek aşağıdaki formulleri buluruz.

$$u_{1i} = a_{1i}\ , \quad i = 1,2,\cdots,n$$
$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\ , \quad j = i, i+1, \cdots, n$$
$$l_{i1} = \frac{a_{i1}}{u_{11}}, \quad i = 2,3,\cdots,n$$
$$l_{jk} = \frac{a_{jk} - \sum_{i=1}^{k-1} l_{ji}u_{ik}}{u_{kk}}, \quad j = k+1, k+2, \cdots, n$$

Benzer şekilde bu işlemler u_{ii} =1 (Crout ayrıştırması) alınarak da genelleştirilebilir.

$$l_{i1} = a_{i1}\ , \quad i = 1,2,\cdots,n$$

$$u_{1j} = \frac{a_{1j}}{l_{11}}, \quad j = 2,3,\cdots,n$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}\ , \quad i = j, j+1,\cdots,n$$

$$u_{jk} = \frac{a_{jk} - \sum_{i=1}^{j-1} l_{ji} u_{ik}}{l_{jj}}, \quad k = j+1, j+2,\cdots,n$$

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn}$$

Bu durmda LU ayrıştırması yönteminin uygulandığı FORTRAN alt programı aşağıda verilmiştir.

- **FORTRAN alt programları**

```
Subroutine LU(a,b,n,tol,x,nhat)
implicit real*8 (a-h,o-z)
Dimension o(n),s(n),x(n)
nhat=0
Call AY(a,n,tol,o,s,nhat)
If(nhat.gt.-1.or.nhat.lt.-1) then
Call YK(a,o,n,b,x)
Endif
Return
End

Subroutine AY(a,n,tol,o,s,nhat)
implicit real*8 (a-h,o-z)
Dimension a(n,n),o(n),s(n)
```

```
      Do i=1,n
      O(i)=i
      S(i)=abs(a(i,1))
      Do j=2,n
      If(abs(a(i,j)).gt.s(i)) then
      S(i)=abs(a(i,j))
      Endif
      Enddo
      Enddo
      Do 10 k=1,n-1
      Call DY(a,o,s,n,k)
      If(abs(a(k,k))/s(k).lt.tol) then
      nhat=-1
      print*,a(k,k)/s(k)
      go to 10
      endif
      do i=k+1,n
      faktor=a(i,k)/a(k,k)
      a(i,k)=faktor
      do j=k+1,n
      a(i,j)=a(i,j)-faktor*a(k,j)
      enddo
      enddo
10    continue
      if(abs(a(k,k)/s(k)).lt.tol) then
      nhat=-1
      print*,a(k,k)/s(k)
```

```
endif
return
end

subroutine DY(a,o,s,n,k)
implicit real*8 (a-h,o-z)
Dimension a(n,n),o(n),s(n)
      np=k
      buyuk=abs(a(k,k)/s(k))
      do ii=k+1,n
      yedek=abs(a(ii,k)/s(ii))
      if(yedek.gt.buyuk) then
      buyuk=yedek
      np=ii
      endif
      enddo
      yedek=o(np)
      o(np)=o(k)
      o(k)=yedek
      return
      end

subroutine YK(a,o,n,b,x)
implicit real*8 (a-h,o-z)
Dimension a(n,n),b(n),x(n)
do i=2,n
top=b(i)
```

```
do j=1,i-1
top=top-a(i,j)*b(j)
enddo
b(i)=top
enddo
x(n)=b(n)/a(n,n)
do i=n-1,1,-1
top=0.
do j=i+1,n
top=top+a(i,j)*x(j)
enddo
x(i)=(b(i)-top)/a(i,i)
enddo
return
end
```

Bu yöntemde A matrisinin boyutu ve elemanları dışarıdan girilebilir. Bunun için bir girdi dosyası bu programın okuyacağı şekilde hazırlanabilir. Ana programda bu dosya açılıp dosyadan uygun okuma yapılabilir.

Karmaşık Sayı İçeren Matrislerle İşlemler

Karmaşık denklem sisteminin (Cz=W) doğrudan çözümü için değişkenler veya matrisler karmaşık sayı olarak tanımlanmalıdır. FORTRAN'da karmaşık sayı değişkenler complex deyimiyle tanımlanabilir, kullanımda ise z=cmplx(x,y) kullanılır. Burada x ve y gerçel değişkenlerdir. Karmaşık değişkenler z=x+iy şeklinde, matrisler C=a+ib ve W=u+iv şeklinde tanımlanır. Mutlak değeri ise abs fonksiyonu ile $|z| = \sqrt{zz^*}$ olacak şekilde hesaplanır.

Karmaşık sayılarla işlem yapmak yerine, gerçel sayılarla işlem yapılmak istenirse, n karmaşık denklem sistemi, 2n gerçel denklem sistemine dönüştürülebilir. Bu durumda

Ax-By=U

Bx+Ay=V

denklem sisteminin çözümü yapılmalıdır.

Fizikte Uygulamalar

Örnek Problem 1: 12 voltluk bir doğru akım kaynağından beslenen üç-halkalı bir dirençli elektrik devresinde Kirchoff kurallarına göre halka akımları için aşağıdaki denklem sistemi verilmiştir.

$$\begin{aligned} 7I_1 - 2I_2 - 4I_3 &= 12 \\ -2I_1 + 13I_2 - 6I_3 &= 0 \\ -4I_1 - 6I_2 + 13I_3 &= 0 \end{aligned}$$

Halka akımlarını Gauss eleme yöntemine göre ve Gauss-Seidel yöntemine göre hesaplayınız ve sonuçlarınızı yorumlayınız.

Çözüm: Problemin denklem sistemini matris formunda yazabiliriz, RI=V. Burada R matrisi dirençleri verir. V matrisi de kaynak gerilimleridir. I matrisi de halka akımlarını gösterir. Gauss eleme yöntemi ile problemi çözmek için aşağıdaki ana programı yazabiliriz.

- **FORTRAN ana programı**

```
 Program Elk
implicit real*8 (a-h,o-z)
dimension r(3,3),v(3),x(3)
data r/7.,-2.,-4.,-2.,13.,-6.,-4.,-6.,13./
```

```
data v/12.,0.,0./
tol=1e-6
call gauss(r,v,3,x,tol,nhat)
write(*,*) "I= ",x
end
```

Ana programda r(3,4) matrisinin elemanları girilmiştir. Program gauss alt programı ile birlikte derlenerek çalıştırıldığında i_1=2.775 A, i_2=1.043 A, i_3=1.335 A olarak bulunur. Elektrik devresindeki kaynak gerilimi 12 Volttur. Dirençlerin değerleri küçük olduğundan halkalardaki akımlar amper civarında olacaktır. Bu akımlar büyükten küçüğe doğu $I_1>I_3>I_2$ şeklinde sıralanır.

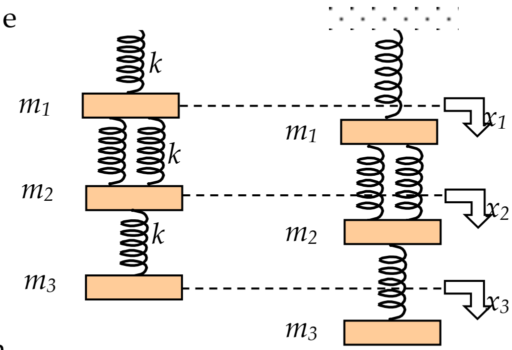
Gauss-Seidel yöntemine göre problemin çözümü aşağıdaki programla yapılabilir.

- **FORTRAN ana programı**

```
 implicit real*8 (a-h,o-z)
Dimension r(3,3),v(3),x(3)
data r/7.,-2.,-4.,-2.,13.,-6.,-4.,-6.,13./
data v/12.,0.,0./
iterm=10
tol=1e-6
call GS(r,v,3,x,iterm,tol)
print*,"x= ",x
end
```

Program GS alt programı ile birlikte derlenerek çalıştırıldığında I1=2.770 A, I2=1.040 A ve I3=1.332 A elde edilir. Burada 10 iterasyon sonucu yazılmıştır. Eğer iterasyon sayısı 20 alınırsa o zaman önceki programda Gauss eleme yönteminden bulunan sonuç ile aynı sonucu elde ederiz.

Örnek Problem 2: Üç kütle, Şekil 2.2'de gösterilen yay sitemine düşey olarak hareket edecek şekilde bağlanmıştır. Kütleler m_1=1 kg, m_2=1.5 kg, m_3=1.25 kg ve yay sabiti k=5 kg/s². Sistem serbest bırakıldığında kütleler düşey yönde hareket etmektedir. LU ayrıştırma yöntemini uygulayarak yerdeğiştirmeleri (x_1, x_2, x_3) çözünüz.



Şekil 2.2. Kütle yay sistemi

Çözüm: Herbir kütle için hareket denklemleri yazılırsa

$$m_1 \frac{d^2 x_1}{dt^2} = 2k(x_2 - x_1) - kx_1 + m_1 g$$
$$m_2 \frac{d^2 x_2}{dt^2} = k(x_3 - x_2) - 2k(x_2 - x_1) + m_2 g$$
$$m_3 \frac{d^2 x_3}{dt^2} = -k(x_3 - x_2) + m_3 g$$

elde edilir. Sistem durgun hale geldiğinde kütlelerin hızları ve ivmeleri sıfır olur. Bu durumda lineer denklem sistemi

$$\begin{aligned} 3kx_1 - 2kx_2 \quad &= m_1 g \\ -2kx_1 + 3kx_2 - kx_3 &= m_2 g \\ -kx_2 + kx_3 &= m_3 g \end{aligned}$$

şeklinde veya matris notasyonunda Kx=W yazılabilir. Burada K ve W matrislerinin elemanlarının sayısal değerleri

$$K = \begin{pmatrix} 15 & -10 & 0 \\ -10 & 15 & -5 \\ 0 & -5 & 5 \end{pmatrix} \quad , \quad W = \begin{pmatrix} 9.8 \\ 14.7 \\ 12.25 \end{pmatrix}$$

şeklindedir. LU ayrıştırma yönteminde K=LU yazılarak L ve U matrislerinin elemanları bulunur. Bir ana programdan önce LU yöntemi çağrılır, sonra LU yöntemi içinde ayrıştırma, döndürme ve yerine koyma işlemleri için ilgili altprogramlar çağrılır. Bu alt programlar önceki sayfalarda verilmiştir.

- **FORTRAN programı**

```
program KYS
 implicit real*8 (a-h,o-z)
 dimension a(3,3),b(3),x(3)
 data a/15.,-10.,0.,-10.,15.,-5.,0.,-5.,5./
 data b/0.98,1.47,1.225/
 tol=1e-6
 call LU(a,b,3,tol,x,nhat)
 print*,"x= ",x
 end
```

Bu program önceki alt programlarla birlikte çalıştırıldığında x_1=0.7350 m, x_2=1.0045 m ve x_3=1.2495 m elde edilir.

ÖZET

Lineer cebirsel denklemlerde verilen a ve c değerleri için x'ler çözülür. Bu uygulamanın grafiksel gösterimi şekilde verilmiştir.

$$a_{11}x_1+a_{12}x_2=c_1$$

$$a_{21}x_1+a_{22}x_2=c_2$$