FİZ433 FİZİKTE BİLGİSAYAR UYGULAMALARI

(DERS NOTLARI)

Hazırlayan:

Prof.Dr. Orhan ÇAKIR

Ankara Üniversitesi, Fen Fakültesi, Fizik Bölümü

Ankara, 2017

İÇİNDEKİLER

1. LİNEER OLMAYAN DENKLEMLERİN KÖKLERİNİN BULUNMASI I/II

2. LİNEER DENKLEM SİSTEMLERİNİN ÇÖZÜLMESİ I/II

3. UYGUN EĞRİNİN BULUNMASI VE INTERPOLASYON I/II

4. SAYISAL İNTEGRAL HESAPLARI I/II

5. DİFERENSİYEL DENKLEMLERİN SAYISAL ÇÖZÜMLERİ I/II

6. BENZETİM I/II

7. FİZİKTE SEMBOLIK HESAPLAMA I/II

EKLER

KAYNAKLAR

KONU 6

UYGUN EĞRİNİN BULUNMASI ve İNTERPOLASYON II

Lagrange Polinomları

Genelleştirme yapılırsa n. mertebeden Lagrange interpolasyon polinomu için

$$f(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + \cdots + L_n(x)f(x_n) = \sum_{i=0}^{n} L_i(x)f(x_i)$$

yazılabilir. $L_i(x)$ fonksiyonu

$$L_i(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} = \prod_{\substack{j=0\\ j\neq i}}^{n}\left(\frac{x-x_i}{x_i-x_j}\right)$$

ile verilir. Özel değerlerde bu fonksiyonun özellikleri kullanılarak, $x=x_i$ için $L_i(x)f(x_i)=f(x_i)$ ve $x\neq x_i$ için $L_i(x)=0$ yazılabilir. Bu ise $f(x)$ fonksiyonunun her bir n+1 veri noktasından geçmesi anlamına gelir. Lagrange interpolasyon polinomu kullanmanın bir çok avantajı vardır. Birincisi, bilgisayarda hesaplamada iterasyon döngüleri yaparak toplam ve çarpım işlemleri yapılabilir. İkincisi, veri noktalarının eşit aralıklı olmasını gerektirmemesidir. Diğer önemli bir özelliği ise, lineer denklem sistemlerinin çözümüne gerek kalmadan, interpolasyon fonksiyonunun doğrudan belirlenmesine imkan vermesidir.

Örnek: Çizelge 3.4'deki veriler için dördüncü mertebeden Lagrange interpolasyon polinomu kullanarak x=3'deki $f(x)$ değerini bulunuz.

Çizelge 3.4 Veriler, x'e karşı gelen $f(x)$ değerleridir.

i	0	1	2	3	4
x_i	0	1	2	4	5
$f(x_i)$	4	6	10	48	94

Çözüm: 4. mertebeden Lagrange polinomu

$$f(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3) + L_4(x)f(x_4)$$

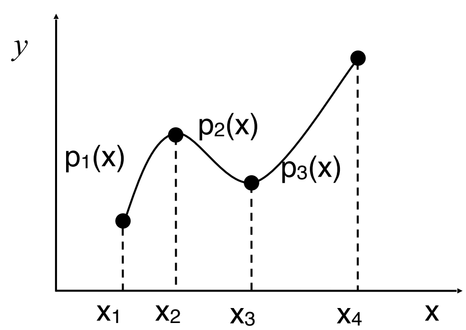
yazılabilir. Burada $L_0(3)=0.1$, $L_1(3)=-0.5$, $L_2(3)=1.0$, $L_3(3)=0.5$, ve $L_4(3)=-0.1$ olarak bulunur. Bu değerler $f(x_i)$'ler ile çarpılarak, toplamları yapılır ve $f(3)=22.0$ bulunur.

- **FORTRAN altprogramı**

```
Function lagrng(x,y,n,xd)
Dimension x(n), y(n)
Top=0.
Do i=0,n
Carp=y(i)
Do j=0,n
if(i.ne.j) then
carp=carp*(xd-x(j))/(x(i)-x(j))
endif
enddo
top=top+carp
enddo
lagrng=top
return
end
```

Kübik Şeritleme Interpolasyonu

N tane veri noktasından geçen en uygun eğrinin bulunması için, kübik şeritleme (spline) yönteminde, başlangıçta iki noktadan geçen kübik bir polinom alınır ve bu iki noktada fonksiyonun birinci ve ikinci türevinin sürekliliğine bakılarak bilinmeyen katsayılar hesaplanır.



Şekil 3.7 Kübik şeritleme interpolasyonunda kullanılan kübik polinomlar

Bu yöntemde $[x_i,x_{i+1}]$ aralığında tanımlı bir kübik fonksiyon (üçüncü dereceden bir polinom),

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

yazılır. İlk dört veri noktasına üç polinomun yerleştirilmesi Şekil 3.7'de gösterilmiştir. Burada $p_i(x)$ fonksiyonu x_i noktasında tam tanımlıdır, $p_i(x_i)=y_i=a_i$. Sonraki x_{i+1} noktasında

$p_i(x_{i+1})$ fonksiyonu ile $p_{i+1}(x_{i+1})$ fonksiyonu aynı değerli olmalıdır. Bu sınır koşulları kullanılırsa

$$p_i(x_{i+1}) = y_{i+1}$$
$$p_{i+1}(x_{i+1}) = y_{i+1}$$

elde edilir. Burada katsayılar arasında aşağıdaki bağıntılar elde edilir.

$$a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = y_{i+1}$$
$$a_{i+1} + b_{i+1}(x_{i+1} - x_i) + c_{i+1}(x_{i+1} - x_i)^2 + d_{i+1}(x_{i+1} - x_i)^3 = y_{i+1}$$

Kübik fonksiyonların birinci türevlerinin de sürekliliği koşulları uygulanırsa

$$p_i'(x_{i+1}) = b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2$$
$$p_{i+1}'(x_{i+1}) = b_{i+1} + 2c_{i+1}(x_{i+1} - x_i) + 3d_{i+1}(x_{i+1} - x_i)^2$$

elde edilir. İkinci türevin de sürekliliği koşulu uygulanırsa aşağıdaki denklemler bulunur.

$$p_i''(x_{i+1}) = 2c_i + 6d_i(x_{i+1} - x_i)$$
$$p_{i+1}''(x_{i+1}) = 2c_{i+1} + 6d_{i+1}(x_{i+1} - x_i)$$

Buradaki sınır koşullarından toplam altı bağıntı elde etmiş olduk. İki uçta türevler belli olmadığına göre doğal olarak,

$$p_1''(x_1) = 0$$
$$p_{N-1}''(x_N) = 0$$

seçilir. N-1 sayıdaki aralıkta, N-1 polinom ve 4(N-1) bilinmeyen vardır. Aralıklar arasındaki sınırlarda ise 4(N-2) bağıntı yazılabilir. Katsayıları veren bağıntılar aşağıda gösterilmiştir.

$$a_i = y_i$$
$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1})$$
$$h_i c_{i+1} + h_{i-1} c_{i-1} = 3\frac{y_{i+1} - y_i}{h_i} - 3\frac{y_i - y_{i-1}}{h_{i-1}} - 2(h_i + h_{i-1})$$
$$d_i = \frac{c_{i+1} - c_i}{3h_i}$$

Burada h_i=x_{i+1}-x_i adım uzunluğu her aralık için aynı olmayabilir. Katsayılardan c_i lerin bulunması için bir tekrarlama bağıntısı kullanılır. Bu yöntemi uygulayan bir altprogram (x_i,y_i) noktalarını kullanarak (a_i,b_i,c_i,d_i) katsayılarını hesaplayabilir. Burada c_i katsayılarının hesabı için önceki bölümdeki lineer denklem sistemlerinin çözüm yöntemlerinden biri kullanılabilir. Başka bir altprogramla da bu katsayıları kullanarak istenen x noktası için interpolasyon yapılabilir.

- **FORTRAN programı**

```
Subroutine spline3(n,x,y,a,b,c,d)
implicit real*8 (a-h,o-z)
Dimension x(n),y(n),a(n),b(n),c(n),d(n)
Dimension h(n-1),t(n-2,n-1),v(n-1)
Do i=1,n-1
h(i)=x(i+1)-x(i)
v(i)=(y(i+1)-y(i))/h(i)
a(i)=y(i)
enddo
do i=1,n-2
do j=1,n-2
t(i,j)=0.
Enddo
T(i,i)=2.*(h(i)+h(i+1))
enddo
if(n.gt.3) then
do i=2,n-2
t(i,i-1)=h(i)
t(i-1,i)=h(i)
enddo
endif
do i=1,n-2
t(i,n-1)=3.*(v(i+1)-v(i))
enddo
m=1
np=100
tol=1.e-6
call gauss(t,n-2,m,np,tol)
do i=2,n-1
c(i)=t(i-1,n-1)
enddo
c(1)=0.
c(n)=0.
Do i=1,n-1
B(i)=v(i)-h(i)*(2.*c(i)+c(i+1))/3.
d(i)=(c(i+1)-c(i))/(3.*h(i))
enddo
return
end
```

```
subroutine interpol3x(n,xi,a,b,c,d,x,y)
implicit real*8 (a-h,o-z)
dimension xi(n),a(n),b(n),c(n),d(n)
i=2
do while(x.gt.xi(i))
i=i+1
enddo
i=i-1
x1=x-xi(i)
y=a(i)+x1*(b(i)+x1*(c(i)+d(i)*x1))
return
end
```

Fizikte Uygulamalar:

Problem 1: Bir radyoaktif maddenin bozunumu

$$N(t) = N_0 e^{-\lambda t}$$

denklemi ile verilir. Burada $N(t)$, t zamanında kalan bu madde miktarını; N_0 ise t=0 da başlangıçtaki madde miktarını vermektedir. Bozunmanın yarı-ömrü $\tau = \ln 2/\lambda$ olarak tanımlanır, burada λ bozunma sabitidir. Çizelge 3.5 de zamana (gün) göre kalan radyoaktif madde miktarının (mg) ölçümü yer almaktadır. Bu maddenin başlangıçtaki miktarını ve yarı ömrünü regresyonla hesaplayınız.

Çizelge 3.5 Radyoaktif maddenin zamana göre değişimi

t(gün)	10	20	30	60	90	120	150	180
$N(t)$, mg	19.0	18.1	17.2	14.9	12.8	11.0	9.5	8.2

Çözüm: Lineer regresyon analizi yapabilmek için öncelikle verilen denklemin lineerleştirilmesi gerekir. Her iki tarafın logaritmasını alınırsa

$$\ln N = \ln N_0 - \lambda t \quad \Rightarrow \quad y' = a_0' + a_1' x$$

elde edilir. Buradaki yeni tanımlar $y'=\ln N$, $a_0'=\ln N_0$ ve $a_1'=-\lambda$ olarak alınmıştır. Verilere bu doğruyu fit etmek için aşağıdaki fortran programını kullanabiliriz.

- **FORTRAN program**

```
Dimension x(8),y(8)
Data x/10.,20.,30.,60.,90.,120.,150.,180./
Data y/2.9444,2.8959,2.8449,2.7014,2.5494,2.3979,2.2513,2.1041/
```

```
      Data tx,ty,txk,tyk,txy/5*0./,n/8/
      do 15 i=1,8
      tx=tx+x(i)
      ty=ty+y(i)
      txk=txk+x(i)**2
      tyk=tyk+y(i)**2
15    txy=txy+x(i)*y(i)
      orx=tx/n
      ory=ty/n
      xork=orx**2
      york=ory**2
      tsxk=txk-n*xork
      tsyk=tyk-n*york
      tsxy=txy-n*orx*ory
      b1=tsxy/tsxk
      b0=ory-b1*orx
      corr=tsxy/((tsxk**0.5)*(tsyk**0.5))
30    format(//,3x,'Regresyon denklemi:
     + y=',f7.4,'',f7.4,'x',
     + //3x,'Korelasyon katsayisi=',f7.4)
      write(*,30)b0,b1,corr
      write(*,*)"N0= ",exp(b0)," lambda= ",-b1,
     + " yari-omur= ",log(2.)/(-b1)
      End
```

Program çalıştırıldığı zaman regresyon denklemi y=2.9948-0.0050x elde edilir. Korelasyon katsayısı r=-1 dir. Başlangıçtaki madde miktarı N_0=19.982 mg, bozunma sabiti λ=0.0050 gün^{-1} ve yarı ömür τ=139.9 gün bulunur.

Problem 2: Millikan'ın yağ damlası deneyinde yüklü yağ damlaları, bir kondansatörün levhaları arasına uygulanan elektrik alanı ile yerçekimi ve sürtünme kuvvetlerinin dengelenmesi sonucu havada asılı gibi kalıyorlar. Millikan bu elektrik yüklerini hesaplayıp küçükten büyüğe doğru sıraladığında, temel bir yük biriminin katları olduğunu gösterdi ve buradan birim elektrik yükünü $|e|=1.65\times10^{-19}$ C olarak belirledi. Bu ölçüm sonuçlarının bir kısmı aşağıdaki Çizelge 3.6 de verilmiştir. Burada deneysel hata σ=3% olarak verilmiştir.

Çizelge 3.6 Millikan'ın gözlemlerinden hesapladığı yük miktarları

N	Q_n (10^{-19} C)	N	Q_n (10^{-19} C)
4	6.563	11	18.08
5	8.204	12	19.71
6	9.880	13	21.32
7	11.50	14	22.89

8	13.13	15	24.60
9	14.82	16	26.13
10	16.48	17	27.88

Bu verilerden geçen en iyi fonksiyonun bulunması için en küçük kareler yöntemini uygulayınız.

Çözüm: Deney verilerine uygun model için

$Q_n = a + bn$ veya $y = a + bx$

kabul edilirse belirsiz katsayılardan a'nın sıfıra yakın ve b'nin de |e| yükünün değerine yakın olması beklenir. Problemin çözümü için yazılacak program en küçük kareler yöntemi ile a ve b katsayılarını bulur standart sapmayı hesaplar ve lineer modelini uyum yüzdesini verir.

- **FORTRAN programı**

```
      Program Millikan
      implicit real*8 (a-h,o-z)
      dimension x(14),y(14)
      data x/4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17./
      data y/6.563,8.204,9.880,11.50,13.13,14.82,16.48,18.08,
     +   19.71,21.32,22.89,24.60,26.13,27.88/
      Call ekk(14,x,y,a,b,r)
      Write(*,*)"y= ",a,"+",b,"x"
      write(*,*)"r= ",r
      end

      subroutine ekk(n,x,y,a,b,r)
      implicit real*8 (a-h,o-z)
      dimension x(n),y(n)
      data topx,topxy,topy,topx2,st,sr /6*0./
      do i=1,n
      topx=topx+x(i)
      topy=topy+y(i)
      topxy=topxy+x(i)*y(i)
      topx2=topx2+x(i)*x(i)
      enddo
      xm=topx/n
      ym=topy/n
      b=(n*topxy-topx*topy)/(n*topx2-topx*topx)
      a=ym-b*xm
      do i=1,n
      st=st+(y(i)-ym)**2
      sr=sr+(y(i)-b*x(i)-a)**2
```

```
enddo
r2=(st-sr)/st
r=sqrt(r2)
end
```

Program çalıştırıldığında a=0.064 ve b=1.634 bulunur. Korelasyon katsayısı r=0.999 olduğundan çok iyi bir fit yapıldığını gösterir.

Problem 3: Bessel fonksiyonları genellikle ileri mühendislik analizlerinde (örneğin elektrik alanlarını çalışırken) kullanılır. Bu fonksiyonların bazı değerleri standart matematik tablolarında verilir. Bunlardan bazı değerler Çizelge 3.7'da verilmiştir.

Çizelge 3.7 $J_0(x)$ fonksiyonunun bazı değerleri

X	1.8	2.0	2.2	2.4	2.6
$J_0(x)$	0.34	0.2239	0.1104	0.0024	0.0968

Kübik şeritleme kullanarak $J_0(2.1)$ değerini belirleyiniz.

Çözüm 3: Kübik şeritleme yöntemini kullanarak problemin çözümünü yapan ana program aşağıda verilmiştir.

- **FORTRAN programı**

```
program kubik_spline
implicit real*8 (a-h,o-z)
parameter (n=5)
dimension xi(n),yi(n),a(n),b(n),c(n),d(n)
data xi/1.8,2.0,2.2,2.4,2.6/
data yi/0.34,0.2239,0.1104,0.0024,0.0968/
call spline3(n,xi,yi,a,b,c,d)
x=2.1
call interpol3x(n,xi,a,b,c,d,x,y)
write(*,*)x,y
end

subroutine spline3(n,x,y,a,b,c,d)
implicit real*8 (a-h,o-z)
dimension x(n),y(n),a(n),b(n),c(n),d(n)
dimension h(n-1),t(n-2,n-1),v(n-1)
do i=1,n-1
h(i)=x(i+1)-x(i)
v(i)=(y(i+1)-y(i))/h(i)
a(i)=y(i)
enddo
do i=1,n-2
do j=1,n-2
```

```
t(i,j)=0.
enddo
t(i,i)=2.*(h(i)+h(i+1))
enddo
if(n.gt.3) then
do i=2,n-2
t(i,i-1)=h(i)
t(i-1,i)=h(i)
enddo
endif
do i=1,n-2
t(i,n-1)=3.*(v(i+1)-v(i))
enddo
m=1
np=10
tol=1.e-6
call gauss2(t,n-2,m,np,tol)
do i=2,n-1
c(i)=t(i-1,n-1)
enddo
c(1)=0.
c(n)=0.
do i=1,n-1
b(i)=v(i)-h(i)*(2.*c(i)+c(i+1))/3.
d(i)=(c(i+1)-c(i))/(3.*h(i))
enddo
return
end

subroutine interpol3x(n,xi,a,b,c,d,x,y)
implicit real*8 (a-h,o-z)
dimension xi(n),a(n),b(n),c(n),d(n)
i=2
do while(x.gt.xi(i))
i=i+1
enddo
i=i-1
x1=x-xi(i)
y=a(i)+x1*(b(i)+x1*(c(i)+d(i)*x1))
return
end

subroutine gauss2(a,n,m,np,tol)
implicit real*8 (a-h,o-z)
dimension a(np,np+m)
if(n.gt.1) then
do k=1,n-1
```

```
amax=abs(a(k,k))
k1=k+1
in=k
do i=k1,n
if(abs(a(i,k)).gt.amax) then
amax=abs(a(i,k))
in=i
endif
enddo
if(k.ne.in) then
do j=k,m+n
x=a(k,j)
a(k,j)=a(in,j)
a(in,j)=x
enddo
endif
if(amax.lt.tol) then
goto 100
endif
do i=k1,n
do j=k1,m+n
a(i,j)=a(i,j)-a(i,k)*a(k,j)/a(k,k)
enddo
enddo
enddo
if(abs(a(n,n)).lt.tol) then
go to 100
endif
do k=1,m
a(n,k+n)=a(n,k+n)/a(n,n)
do l=1,n-1
i=n-l
i1=i+1
do j=i1,n
a(i,k+n)=a(i,k+n)-a(j,k+n)*a(i,j)
enddo
a(i,k+n)=a(i,k+n)/a(i,i)
enddo
enddo
goto 110
else if(abs(a(1,1)).lt.tol) then
go to 100
endif
do j=1,m
a(1,n+j)=a(1,n+j)/a(1,1)
enddo
goto 110
```

```
100     print*,"Cozum !"
110     return
        end
```

Program çalıştırıldığında $J_0(2.1)$ değeri 0.1666 olarak bulunur.

ÖZET

Eğriye fit, regresyon ve interpolasyon uygulamalarının grafiksel anlatımı şekilde gösterilmiştir.