FİZ433 FİZİKTE BİLGİSAYAR UYGULAMALARI

(DERS NOTLARI)

Hazırlayan:

Prof.Dr. Orhan ÇAKIR

Ankara Üniversitesi, Fen Fakültesi, Fizik Bölümü

Ankara, 2017

İÇİNDEKİLER

KONU 8

SAYISAL İNTEGRAL HESAPLARI II

İntegral Sınırlarının Dönüştürülmesi

Gauss quadrature formülleri genellikle -1 ile +1 sınırları arasındaki integraller için kullanılmıştır. Bu formülleri başka sınırlar (örneğin, [*a*,*b*]) içeren integrallere uygularken değişken değiştirmesi yapılır. Eğer [*a*,*b*] sınırlı bir integral verilmişse

$$x = \frac{2y - (a+b)}{b-a} \qquad \text{veya} \qquad y = \frac{(b-a)x + (a+b)}{2}$$

yazarak integralin sınırları -1 ile +1 ‘e dönüştürülür. Burada

$$dy = \frac{(b-a)dx}{2}$$

olur. Böylece integral

$$\int_a^b f(y)dy = \left(\frac{b-a}{2}\right)\int_{-1}^{1} f\left[\frac{(b-a)x + (a+b)}{2}\right]dx$$

yazılabilir. Gauss quadrature formülünün uygulanmasıyla [*a*,*b*] sınırlı integral

$$\int_a^b f(y)dy = \left(\frac{b-a}{2}\right)\sum_{i=0}^{n} w_i f\left[\frac{(b-a)x_i + (a+b)}{2}\right]$$

şeklinde hesaplanabilir. Gauss-Legendre quadrature yöntemini kullanarak integral hesabı yapan bir FORTRAN programı aşağıda verilmiştir.

- **FORTRAN program**

```
implicit double precision(a-h,o-z)

write(*,*)’Gauss-Legendre Quadrature’

write(*,*)'Alt sinir:'

read(*,*)a

write(*,*)'Ust sinir:'

read(*,*)b

write(*,*)'--------------------'

call gauss_quadrature(a,b,top)

write(*,*)'Integral sonucu:',top

write(*,*)'--------------------'

end


subroutine gauss_quadrature(a,b,top)

implicit double precision(a-h,o-z)

dimension x(10),w(10)

X(1)=0.148874338981631d0

X(2)=-x(1)

X(3)=0.433395394129247d0

X(4)=-x(3)

X(5)=0.679409568299024d0

X(6)=-x(5)

X(7)=0.865063366688985d0

X(8)=-x(7)
```

```
      X(9)=0.973906528517172d0
      X(10)=-x(9)
      W(1)=0.295524224714753d0
      W(2)=w(1)
      W(3)=0.269266719309996d0
      W(4)=w(3)
      W(5)=0.219086362515982d0
      W(6)=w(5)
      W(7)=0.149451349150581d0
      W(8)=w(7)
      W(9)=0.066671344308688d0
      W(10)=w(9)
      Top=0.d0
      Do 10 i=1,10
      y=((b-a)*x(i)+(a+b))/2.d0
      call f(y,fy)
      Top=Top+w(i)*fy
10    continue
      Top=Top*(b-a)/2.d0
      return
      end

      subroutine f(y,fy)
      implicit double precision(a-h,o-z)
      fy=y*y*y*y-y*y*y-3.d0*y*y-4.d0*y-3.d0
      return
      end
```

Bu programın çıktısı aşağıdaki gibi olacaktır.

Gauss-Legendre Quadrature

Alt sinir:

0.

Ust sinir:

10.

Integral sonucu: 16270.

Çok Boyutlu Integrallerin Hesaplanması

Fen ve mühendislikte bazı problemlerde çok katlı integrallerin sayısal hesaplanması gerekmektedir. 1≤n≤6 olmak üzere, n-boyutlu bir integral aşağıdaki gibi gösterilebilir.

$$I_n = \int_{a_n}^{b_n} dx_n f_n(x_n) \int_{a_{n-1}(x_n)}^{b_{n-1}(x_n)} dx_{n-1} f_{n-1}(x_{n-1}, x_n) \cdots \int_{a_1(x_2,\cdots,x_n)}^{b_1(x_2,\cdots,x_n)} dx_1 f_1(x_1, \cdots, x_n)$$

Burada I_n integrali herbiri bir altprogramda hesaplanan n adet örgüsel yapı içermektedir. Altprogramların genel yapısı Gauss quadrature yöntemine uygundur. İsimlendirme ise GQIk(FAPk,Ak,Bk,NIk,NGk,x) şeklindedir. Burada aşağıdaki tanımlar kullanılmıştır:

FAPk : Kullanıcı tarafından tanımlanan subroutine tipli altprogramdır, çağıran programda external deyimi ile bildirilmelidir. Genel yapısı SUBROUTINE FAPk(m,Uk,Fk,x) şeklindedir. Burada m, GQIk tarafından belirlenen ve m≤64 olan bir tamsayıdır. Uk ve Fk değişkenlerinin içeriği, sırasıyla GQIk ve FAPk tarafından verilen bir boyutlu dizilerdir.

Ak, Bk : x_k değişkeni için integral sınırlarını gösterir.

NIk : [Ak:Bk] aralığının kaç eşit bölmeye ayrılacağını gösterir.

NGk : Her bir NIk bölmesinde kullanılan Gauss quadrature nokta sayısını gösterir.

x : n-elemanlı bir boyutlu diziyi gösterir.

Her bir değişkene göre integral hesabı, eşit alt aralıklarda 6 veya 8 noktalı Gauss quadrature formullerinin uygulanması ile gerçekleştirilir. Burada n-boyutlu bir integral hesabı için geçen süre $\prod_{k=1}^{n} \mathrm{NGk} \cdot \mathrm{NIk}$ ile orantılıdır.

Örnek: 6-noktalı Gauss quadrature yöntemini kullanarak

$$Q_2 = \int_0^1 dx_2 \sqrt{x_2}\, e^{x_2} \int_0^{\sqrt{x_2}} dx_1 x_1 \sqrt{x_1^2 + x_2}$$

integralini hesaplayalım, çift duyarlı gerçel veri tipi kullanalım. Integral sınırlarını n_1=4 ve n_2=3 alt bölme olacak şekilde alırsak FORTRAN programı aşağıdaki gibi olacaktır:

- **FORTRAN programı**

```
program gqi

implicit double precision(a-h,o-z)

external fap2

dimension x(2)

q2=gqi2(fap2,0.d0,1.d0,4,6,x)

gdeger=1.d0/3.d0*(2.d0*sqrt(2.d0)-1.d0)*(exp(1.d0)-2.d0)

write(*,*)"gq: ",q2," gd: ",gdeger

end
```

```
      subroutine fap2(m,u2,f2,x)
      implicit double precision(a-h,o-z)
      external fap1
      dimension u2(*),f2(*),x(2)
      do 1 i=1,m
      x(2)=u2(i)
      r=sqrt(x(2))
1     f2(i)=r*exp(x(2))*gqi1(fap1,0.d0,r,3,6,x)
      return
      end

      subroutine fap1(m,u1,f1,x)
      implicit double precision(a-h,o-z)
      dimension u1(*),f1(*),x(2)
      do 2 j=1,m
      x(1)=u1(j)
2     f1(j)=x(1)*sqrt(x(1)**2+x(2))
      return
      end

      double precision function gqi1(fap1,a,b,ni,ng,x)
      implicit double precision(a-h,o-z)
      parameter (z1 = 1.d0, half = z1/2.d0)
       dimension x(6),w(14),t(14),v(64),u(64),f(64)
       data (t(i),w(i),i=1,14)
     1/-0.93246 95142 03152 028d0, 0.17132 44923 79170 345d0,
```

```
2 -0.66120 93864 66264 514d0, 0.36076 15730 48138 608d0,
3 -0.23861 91860 83196 909d0, 0.46791 39345 72691 047d0,
4  0.23861 91860 83196 909d0, 0.46791 39345 72691 047d0,
5  0.66120 93864 66264 514d0, 0.36076 15730 48138 608d0,
6  0.93246 95142 03152 028d0, 0.17132 44923 79170 345d0,
7 -0.96028 98564 97536 232d0, 0.10122 85362 90376 259d0,
8 -0.79666 64774 13626 740d0, 0.22238 10344 53374 471d0,
9 -0.52553 24099 16328 986d0, 0.31370 66458 77887 287d0,
a -0.18343 46424 95649 805d0, 0.36268 37833 78361 983d0,
b  0.18343 46424 95649 805d0, 0.36268 37833 78361 983d0,
c  0.52553 24099 16328 986d0, 0.31370 66458 77887 287d0,
d  0.79666 64774 13626 740d0, 0.22238 10344 53374 471d0,
e  0.96028 98564 97536 232d0, 0.10122 85362 90376 259d0/
 m0=ng
 if(m0 .ne. 8) m0=6
 i0=0
 if(m0 .eq. 8) i0=6
 d=(b-a)/ni
 r=half*d
 ra=r+a
 mv=mod(m0*ni-1,64)+1
 s=0.d0
 j=0
 do 1 i = 1+i0,m0+i0
 rta=r*t(i)+ra
 do 2 k = 1,ni
 j=j+1
```

```
     v(j)=w(i)
     u(j)=rta+(k-1)*d
     if(j .eq. mv) then
      call fap1(mv,u,f,x)
      do 3 j = 1,mv
   3  s=s+v(j)*f(j)
      mv=64
      j=0
     end if
   2  continue
   1  continue
     gqi1=r*s
     return
 101 format('n2 = ',i4,' <= 0')
     end

     double precision function gqi2(fap2,a,b,ni,ng,x)
     implicit double precision(a-h,o-z)
     parameter (z1 = 1.d0, half = z1/2.d0)
      dimension x(6),w(14),t(14),v(64),u(64),f(64)
      data (t(i),w(i),i=1,14)
    1/-0.93246 95142 03152 028d0, 0.17132 44923 79170 345d0,
    2 -0.66120 93864 66264 514d0, 0.36076 15730 48138 608d0,
    3 -0.23861 91860 83196 909d0, 0.46791 39345 72691 047d0,
    4  0.23861 91860 83196 909d0, 0.46791 39345 72691 047d0,
    5  0.66120 93864 66264 514d0, 0.36076 15730 48138 608d0,
    6  0.93246 95142 03152 028d0, 0.17132 44923 79170 345d0,
```

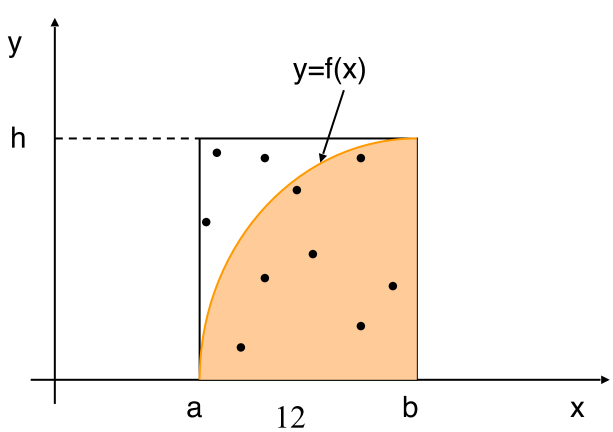
```
7 -0.96028 98564 97536 232d0, 0.10122 85362 90376 259d0,
8 -0.79666 64774 13626 740d0, 0.22238 10344 53374 471d0,
9 -0.52553 24099 16328 986d0, 0.31370 66458 77887 287d0,
a -0.18343 46424 95649 805d0, 0.36268 37833 78361 983d0,
b  0.18343 46424 95649 805d0, 0.36268 37833 78361 983d0,
c  0.52553 24099 16328 986d0, 0.31370 66458 77887 287d0,
d  0.79666 64774 13626 740d0, 0.22238 10344 53374 471d0,
e  0.96028 98564 97536 232d0, 0.10122 85362 90376 259d0/
  m0=ng
  if(m0 .ne. 8) m0=6
  i0=0
  if(m0 .eq. 8) i0=6
  d=(b-a)/ni
  r=half*d
  ra=r+a
  mv=mod(m0*ni-1,64)+1
  s=0.d0
  j=0
  do 1 i = 1+i0,m0+i0
  rta=r*t(i)+ra
  do 2 k = 1,ni
  j=j+1
  v(j)=w(i)
  u(j)=rta+(k-1)*d
  if(j .eq. mv) then
   call fap2(mv,u,f,x)
   do 3 j = 1,mv
```

```
  3   s=s+v(j)*f(j)
      mv=64
      j=0
     end if
  2  continue
  1  continue
     gqi2=r*s
     return
101 format('n2 = ',i4,' <= 0')
     end
```

Program çalıştırıldığında Q_2=0.437775326 değeri bulunmaktadır. Bu ise verilen iki katlı integralin gerçek değeri ile aynıdır. Bu sonuç Gauss quadrature yönteminin çok katlı integrallerde de oldukça iyi sonuç verdiğini göstermektedir.

Monte Carlo Yöntemi ile Integrallerin Sayısal Hesaplanması

Bir simulasyon uygulaması olarak Monte Carlo integral hesabında rastgele sayı üreticiler kullanılır. Çok karmaşık fonksiyonlar içeren integrallerin hesaplanmasında diğer yöntemlerle yeterli duyarlılık elde edilemeyebilir. Böyle durumda bile Monte Carlo teknikleri kullanarak integral yaklaşık olarak hesaplanabilir. Yöntemi kolay anlamak için $[a,b]$ aralığında $f(x)$ fonksiyonunun integralini düşünelim. Bu yöntemin uygulanışı basit olarak Şekil 4.5'de gösterilmiştir.



Şekil 4.5. Monte Carlo İntegrali

Karşılıklı köşeleri [*a,0*] ve [*b,h*] noktalarında olan bir dikdörtgen düşünelim. Burada, [*a,b*] aralığındaki bütün *x* değerleri için dikdörtgenin yüksekliği $h \geq f(x)$ bağıntısını sağlar. $f(x)$ fonksiyonu dikdörtgeni iki bölgeye ayırır; (1) integrali hesaplamak istediğimiz ve fonksiyonun altında kalan bölge, (2) fonksiyonun üstünde kalan bölge. Bir uniform dağılımdan rastgele koordinat çifti üreterek dikdörtgenin içine yerleştirilir. Bunun için noktaların *x* koordinatları [*a,b*] aralığında ve *y* koordinatları da [*0,h*] aralığında kalmaya zorlanır. Herbir koordinat çifti için seçilen noktanın eğrinin altına mı yoksa üstüne mi düştüğü belirlenir. Fonksiyonun altında kalan noktaların sayısının (N_x) toplam nokta sayısına (N_T) oranı, bu fonksiyonun altında kalan alanın (A_x) dikdörtgenin alanına (A_T) oranının bir yaklaşık ifadesini verir.

$$\frac{N_x}{N_T} \cong \frac{A_x}{A_T}$$

Dikdörtgenin alanını bildiğimiz için, $A_T=(b-a)\cdot h$, fonksiyonun altında kalan alanı, eğrinin altına düşen noktaların kesri ile dikdörtgen alanın çarpılmasından kolaylıkla hesaplayabiliriz, $A_x \cong A_T N_x / N_T$. Bu yöntem için basit bir algoritma aşağıda verilmiştir. Burada üretilen rastgele sayılar ne kadar çok olursa hesaplamanın sonucu da o kadar iyi olacaktır. Bu işlemin algoritması aşağıdaki gibi özetlenebilir:

- [a:b] aralığında rastgele x değerleri al
- [0:h] aralığında rastgele y değerleri al
- y>f(x) ise sayı=sayı+1 yap
- yukarıdaki işlemi n kez tekrarla
- (b-a)*h*sayı/n sonucunu hesapla

Örnek: $f(x)=x^2$ fonksiyonunun [0,2] aralığında integralini Monte Carlo yöntemi ile hesaplayan bir FORTRAN programı yazınız, olay sayısını n=10, 100 ve 1000 alarak sonucu yorumlayınız.

- **FORTRAN program**

```
      program omc
      write(*,*)'***f(x) fonksiyonunun MC integrali'
      write(*,*)'alt sinir'
      read(*,*)a
      write(*,*)'ust sinir'
      read(*,*)b
      write(*,*)'yukseklik'
      read(*,*)h
      write(*,*)'olay sayisi'
      read(*,*)n
      ncount=0
      do 1 i=1,n
      x=rand()*abs(b-a)+a
C ..................................................................
      fx=x*x ! integrali alinacak fonksiyon
C ..................................................................
      y=rand()*h
1     if(y<fx) ncount=ncount+1
      xr=abs(b-a)*h*ncount/n
      write(*,*)'sonuc=',xr
      end
```

Program çalıştırıldığında n=100 olay için I=2.72, n=1000 olay için I=2.624, n=10000 olay için I=2.6864 ve n=100000 olay için I=2.66016 elde edilmiştir. Buradaki sonuçlar yöntemin özelliği gereği olay sayısı (n) arttıkça integralin gerçek değeri 2.666...'ya yakınsamaktadır.

Fiziksel Uygulamalar

Örnek Problem 1: Zamanla sinüsel olarak değişen bir akım, bir peryot üzerinden ortalaması sıfır olduğu halde, elektrik devresi elemanları üzerine iş yapabilir, güç aktarabilir ve ısıya neden olabilir. Böyle durumlarda akımın kare ortalama karekök KOK (RMS) değerinden bahsedilir.

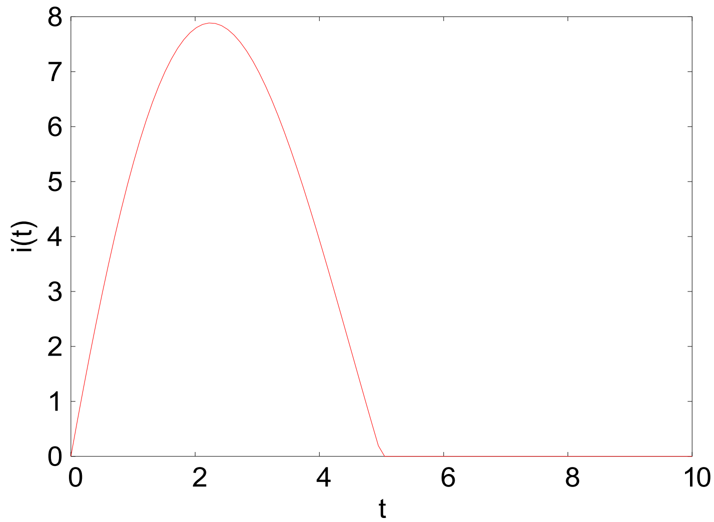
$$I_{KOK} = \sqrt{\frac{1}{T}\int_0^T i^2(t)dt}$$

Burada $i(t)$ anlık akımdır. Verilen herhangi bir akım fonksiyonu için bu KOK değer sayısal integralleme ile hesaplanabilir. Burada $i(t)$ akımını aşağıda verildiği gibi alarak

$$i(t) = \begin{cases} 10e^{-t/T}\sin\left(\dfrac{2\pi t}{T}\right) & , \quad 0 \le t \le T/2 \\ 0 & , \quad T/2 < t \le T \end{cases}$$

I_{KOK} değerini, trapez kuralı ile hesaplayıp sonucu yorumlayınız. Bu fonksiyonun grafiği (periyot T=10 s alınmıştır) Bölüm 1'de anlatılan gnuplot programı ile aşağıdaki komutlarla çizilebilir, Şekil 4.6.

```
gnuplot> set xlabel "t"
gnuplot> set ylabel "i(t)"
gnuplot> set nokey
gnuplot> f(t)=(t<5)
gnuplot> plot [t=0:10] 10.*exp(-t/10.)*sin(2.*pi*t/10.)*f(t)
```



Şekil 4.6 $i(t)$ akım fonksiyonunun t'ye bağlı değişimi

Çözüm 1: Verilen akım fonksiyonun t[0:5] aralığında integrale katkıda bulunan kısmı için I_{KOK} değeri aşağıdaki ifadeye göre hesaplanabilir.

$$I_{KOK} = \sqrt{10\int_0^5 e^{-t/5} \sin^2(\pi t/5)dt}$$

Trapez kuralına göre hesap yapan program aşağıda verilmiştir.

- **FORTRAN programı**

```
program trapakim

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

a=0.d0

b=5.d0

takim=trapf(128,a,b)

write(*,*)sqrt(takim)

end
```

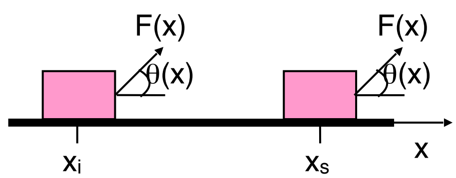
```
Function trapf(n,a,b)
implicit real*8 (a-h,o-z)
h=(b-a)/n
x=a
top=f(x)
do i=1,n-1
x=x+h
top=top+2.d0*f(x)
enddo
top=top+f(b)
trapf=(b-a)*top/(2.d0*n)
return
end

function f(t)
implicit real*8 (a-h,o-z)
pi=3.141593d0
f=10.d0*exp(-t/5.d0)*(sin(pi*t/5.d0))**2
return
end
```

Program çalıştırıldığında n=2 için I_{KOK}=3.894 elde edilir, burada hata ε=0.8% dir. Aralık sayısını artırdığımızda n=32 için I_{KOK}=3.92589 elde edilir, bu da gerçek değer ile aynıdır.

Örnek Problem 2: Birçok fen ve mühendislik probleminde

$$W = \int_{x_i}^{x_s} \mathbf{F}(x) \cdot d\mathbf{x}$$

şeklindeki integrallerin sayısal hesaplanması gerekir. Fizikte bir $F(x)$ kuvveti uygulayarak bir cismin konumu dx kadar değiştirilirse dW kadar iş yapılır. F(x) kuvveti vektörel bir nicelik olduğu için iş hesaplanırken hareketin yönündeki bileşeni hesaplanır, Şekil 4.7.



Şekil 4.7 Bir cisim üzerine etki eden kuvvetin ve yatayla yaptığı açının konuma bağlı olduğu durum.

Kuvvet ile hareket doğrultusu arasındaki açı da konumla değişebilir, böyle durumlarda $F(x)$ ve $\theta(x)$ bilinen fonksiyonlar ise integral analitik olarak da hesaplanabilir. Ancak bu niceliklerin değerleri konumun bir fonksiyonu olarak Çizelge 4.3'deki gibi verilirse sayısal integral yöntemlerinden birini uygulayarak, integrali hesaplayınız.

Çizelge 4.3 Cisim üzerine etki eden $F(x)$ kuvvetinin ve $\theta(x)$ açısının konuma bağlı olarak değişimi.

x, m	$F(x)$, N	$\theta(x)$, radyan
0.00	0.0	0.50
0.10	9.0	1.50
0.15	13.0	0.80
0.25	15.0	1.48
0.30	14.0	1.50
0.40	10.0	1.40
0.42	5.0	1.20

Çözüm 2: Eşit bölmelenmemiş verilerin sayısal integrallenmesinde Trapez kuralı uygulanabilir. Konum metre cinsinden, kuvvet de newton cinsinden verildiğinde yapılan işi Joule olarak hesaplarız. Aşağıdaki program bu integrali (fiziksel iş *W*) hesaplamaktadır.

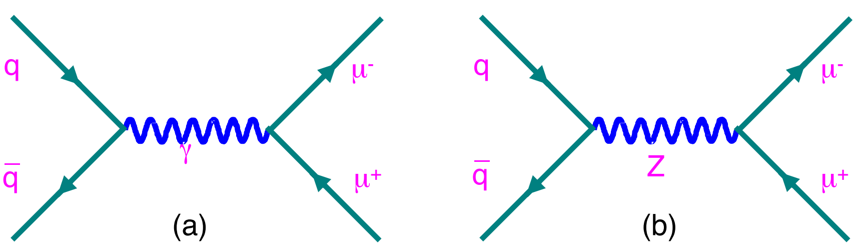
- **FORTRAN altprogramı**

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

Parameter (n=7)

Dimension x(n),y(n),xy(n),teta(n)

Data x/0.d0,0.1d0,0.15d0,0.25d0,0.3d0,0.4d0,0.42d0/

Data teta/0.5d0,1.4d0,0.8d0,1.48d0,1.5d0,1.3d0,1.2d0/

Data y/0.d0,9.d0,13.d0,15.d0,14.d0,10.d0,5.d0/

do i=1,n

xy(i)=y(i)*cos(teta(i))

enddo

Trapint=trapez(x,xy,n)

Write(*,*)Trapint

end


Function trapez(x,y,n)

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

Dimension x(n),y(n)

Top=0.d0

Do i=2,n

Top=top+(x(i)-x(i-1))*(y(i)+y(i-1))/2.d0

Enddo

Trapez=top

return

end
```

Program çalıştırıldığında integral sonucu W=1.14891 J bulunur, bu da gerçek değere oldukça yakındır.

Örnek Problem 3: 25 GeV kütle merkezi enerjisinde çarpışan iki proton demeti için Drell-Yan sürecinin, $d^2\sigma/dx_1dx_2$ diferensiyel tesir kesitini Monte Carlo kabul-red yöntemi ile hesaplayınız.

Çözüm 3: Yüksek enerji fiziğinde Drell-Yan sürecini inceleyelim [Drell and Yan 1970]. Bir nükleon olarak proton (veya nötron), partonlardan oluşmuştur. Partonlar ise madde parçacıkları valans kuarklar (u, d) ve deniz kuarklar ($\bar{u}$, $\bar{d}$, c, $\bar{c}$, s, $\bar{s}$, b, $\bar{b}$) ile bunlar arasında güçlü etkileşmeyi sağlayan gluonlardır (g). Parçacık fiziğinin Standart Modeline (SM) göre 6 çeşit kuark (yükleri 2/3 olan "yukarı" kuarklar: u, c, t ve yükleri -1/3 olan "aşağı" kuarklar d, s, b) ve bunların anti-kuarkları vardır. Düşük enerjilerde protonun (uud), baskın olarak iki u valans kuarkından ve bir d valans kuarkından oluştuğu kabul edilebilir. Drell-Yan sürecinde, protonlardan gelen bir kuark ve anti-kuark yok olarak sanal foton veya Z-bozon aracılığıyla muon ve anti-muon oluşumu gerçekleşebilir. Bu sürecin Feynman diyagramları Şekil 4.8'de gösterilmiştir.



Şekil 4.8 Kuark ve anti-kuark yokolması sürecinde muon ve anti-muon üretimi, (a) foton alışverişi ile elektromagnetik etkileşme diyagramı, (b) Z-bozonu alışverişi ile zayıf etkileşme diyagramı

Bu süreci çalışmak için iki proton demetinin en az 110 MeV kütle merkezi enerjisinde çarpıştırılması gerekir. Burada $pp\rightarrow\mu^+\mu^-X$ sürecinin diferensiyel tesir kesiti aşağıdaki gibi yazılabilir.

$$\frac{d^2\sigma}{dx_1dx_2} = \frac{C_{DY}}{M^2}\sum_i Q_i^2\left[f_{i/A}(x_1)\bar{f}_{i/B}(x_2) + \bar{f}_{i/A}(x_1)f_{i/B}(x_2)\right]$$

Burada C_{DY} bir sabittir, Q_i gelen kuarkın yüküdür (i=1, 2, 3; sırasıyla u_v, d_v ve deniz kuarkları temsil eder). Bjorken x değişkeni quarkın momentumunun protonun momentumuna oranı olarak tanımlanır ($x_i=p_i/P$). Kuarklar protonun x, 0 (durgun) ile 1 (hepsi) arasındaki, momentum kesrini taşırlar. Bu alanda yapılan deneylerden elde edilen sonuçları içeren parametrizasyona göre proton içindeki valans kuark (u_v, d_v) ve deniz kuark (q) dağılımları

$$u_v(x) = 2.13\sqrt{x}(1-x)^{2.8}$$
$$d_v(x) = 1.26\sqrt{x}(1-x)^{3.8}$$
$$q(x) = 0.27(1-x)^{8.1}$$

ile verilir. Yukarıdaki tesir kesiti formulunde f_i(x) valans kuark dağılımı fonksiyonuna, $\bar{f}_i(x)$ deniz kuark dağılımı fonksiyonuna karşı gelir. Yukarıdaki tesir kesiti formulunde iki lepton değişmez (invaryant) kütlesinin karesi M^2 değişkeni ile gösterilmiştir. Toplam kütle merkezi enerjisi ve kuarkların momentum kesirleri ile orantılıdır, $M^2=Sx_1x_2$.

Aşağıda verilen program iki çarpışan proton demetini kullanarak tesir kesitini hesaplar. Basitlik için yapı fonksiyonları momentum aktarımı Q dan bağımsız alınmıştır. Program kuark ve anti-kuarkların katkılarını toplamak için MC kabul-red yöntemini kullanmaktadır.

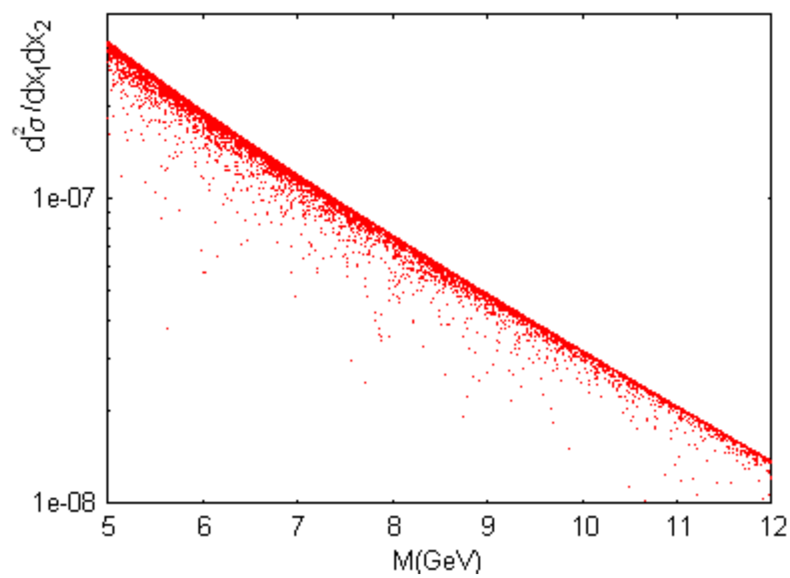
- FORTRAN program

```
program kab_red

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

open(1,file='kab_red.txt')

pi=3.1415d0

CDY=5.3279343d-5

edemet=25.d0

Ecm=2.d0*edemet

imax=5000000

do i=0,imax

xa=rand()

xb=rand()
```

```
ya=rand()
xm2=Ecm**2*xa*xb
if((uq(xa).gt.ya).and.(dq(xa).gt.ya).and.(qq(xb).gt.ya)
..and.(uq(xb).gt.ya).and.(dq(xb).gt.ya).and.(qq(xa).gt.ya)
..and.(xm2.gt.25.d0).and.(xm2.lt.144.d0)) then
x1=xa
x2=xb
xm=sqrt(xm2)
fq1=8.d0/9.d0*(uq(xa)*qq(xb)+qq(xa)*uq(xb))
fq2=1.d0/9.d0*(dq(xa)*qq(xb)+qq(xa)*dq(xb))
ds=CDY*4.d0*pi/9.d0/xm2*(fq1+fq2)
xf=abs(xa-xb)
write(*,*)xf,xm,ds
write(1,*)xf,xm,ds
endif
enddo
end

function uq(x)
implicit real*8 (a-h,o-z)
uq=2.13d0*sqrt(x)*(1.d0-x)**(2.8d0)
return
end

function dq(x)
implicit real*8 (a-h,o-z)
dq=1.26d0*sqrt(x)*(1.d0-x)**(3.8d0)
```

```
return

end


function qq(x)

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

qq=0.27d0*(1.d0-x)**(8.1d0)

return

end
```

Program çalıştırıldığında elde edilen verilere göre değişmez kütle dağılımı Şekil 4.9'de gösterilmiştir.



Şekil 4.9 Drell-Yan sürecinin diferensiyel tesir kesitinin değişmez kütleye bağlılığı, kabul-red yöntemi kullanılmıştır.

ÖZET

Integral: $f(x)$ eğrisinin altında kalan alanın bulunması, integralin grafiksel anlatımı aşağıdaki şekilde verilmiştir.