T. C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LAGRANGE İNTERPOLASYONU VE BAZI UYGULAMALARI

Mevlana CANTÜRK

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

KONYA
2009

T. C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

LAGRANGE İNTERPOLASYONU VE BAZI UYGULAMALARI

Mevlana CANTÜRK

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

Bu tez 18/09/2009 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Öğr. Gör. Dr. Ferhat YILDIRIM	Prof. Dr. Durmuş BOZKURT
(Danışman)	(Üye)

Yrd. Doç. Dr. Hasan KÖSE

(Üye)

ÖZET

Yüksek Lisans Tezi

LAGRANGE İNTERPOLASYONU VE BAZI UYGULAMALARI

Mevlana CANTÜRK
Selçuk Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman: Öğr. Gör. Dr. Ferhat YILDIRIM

Bu çalışmada, M.T. Rashed'in 2004 yılında, Lagrange interpolasyon yöntemi ile diferansiyel, integral ve integro diferansiyel denklemlerin nümerik çözümleri için geliştirdiği iki metot, örnekleri ile birlikte incelenecektir. Bunun için ikinci kısımda çalışmamıza temel teşkil eden; Lagrange interpolasyonu, Chebyshev polinomları, Volterra integral denklemleri ve Fredholm integral denklemleri ile ilgili genel bilgiler verilmiştir. Üçüncü kısımda Volterra integral denklemleri, Fredholm integral denklemleri ve lineer integro-diferansiyel denklemleri için geliştirilen iki metot tanımlanmış ve örnekler üzerinde uygulanmıştır. Ayrıca metotlar için yazılabilecek algoritmanın bilgisayar ortamına kolaylıkla aktarılması da yöntemin olumlu yanlarından birini teşkil ettiğini gösterir. Son kısımda çalışmalarımızın değerlendirilmesi yapılmıştır.

Anahtar Kelimeler: Lagrange interpolasyonu, İntegral denklemleri, Volterra integral denklemleri, Fredholm integral denklemleri, İntegro-diferansiyel denklemleri

ABSTRACT

MS Thesis

LAGRANGE INTERPOLATION

AND SOME APPLICATIONS

Mevlana CANTÜRK
Selcuk University
Graduate School of Natural and Applied Sciences
Department of Mathematics

Supervisor: Öğr. Gör. Dr. Ferhat YILDIRIM

In this work, the Lagrange interpolation and two methods to compute numerical solutions of differential, integral and integro-differential equations using Lagrange interpolation with examples are studied. For this study; in second chapter, the general informations about Lagrange interpolation, Chebyshev polynomials, Volterra integral equations and Fredholm integral equations are given. In third chapter, two methods for Volterra integral equations ve Fredholm integral equations, integro-differential equations are defined and these methods are applied on examples. Another positive aspect of the proposed methods is the ease by which the algorithm can be transferred to computation media. In the last chapter, evaluations of our studies are done.

Keywords: Lagrange interpolation, Integral equations, Volterra integral equations, Fredholm integral equations, Integro-differential equations

ÖNSÖZ

Yüksek lisans tez çalışması çeşitli kaynaklardan yararlanılarak derleme biçiminde hazırlanmıştır. Bu tez çalışmasında yardımlarını benden esirgemeyen Selçuk Üniversitesi Fen Fakültesi Matematik Bölümü öğretim üyesi ve değerli danışman hocam Öğr. Gör. Dr. FERHAT YILDIRIM'a en içten teşekkürlerimi sunarım. Çalışmalarım boyunca değerli yardımlarıyla beni yönlendiren ve fikirleriyle yol gösteren değerli hocalarımdan; Prof Dr. Durmuş BOZKURT'a, Doç. Dr. Galip OTURANÇ'a, Yrd. Doç. Dr. Hasan KÖSE'ye ve Arş. Gör. Yıldıray KESKİN'e teşekkürlerimi bir borç bilirim. Ayrıca çalışmam süresince bana destek veren eşime ve aileme bu vesileyle teşekkür ederim.

Ağustos, 2009 **Mevlana CANTÜRK**

İÇİNDEKİLER

ÖZET.....i
ABSTRACT.....ii
ÖNSÖZ.....iii
İÇİNDEKİLER..... iv

1.GİRİŞ VE AMAÇ.....1

2. GENEL BİLGİLER.....2
2.1. İnterpolasyon….....2
2.2. Lagrange interpolasyon polinomu..... 2
2.3. Chebyshev polinomları.....24
2.4. İntegral denklemleri.....29
2.4.1.Volterra integral denklemleri.....29
2.4.2. Fredholm integral denklemleri.....30
2.5. Gauss – Lengendre İntegrasyonu.....31
2.6. Chebyshev Formülleri.....33

3. DİFERANSİYEL, İNTEGRAL VE İNTEGRO-DİFERANSİYEL DENKLEMLERİNİN NÜMERİK ÇÖZÜMLERİ İÇİN LAGRANGE İNTERPOLASYON YÖNTEMİ.....35
3.1. Birinci Metot.....37
3.2. İkinci Metot.....39
3.3. Lineer İntegro-Diferansiyel Denklemler.....40
3.4. Nümerik Örnekler.....42

4. DEĞERLENDİRMELER VE SONUÇLAR.....48

1. GİRİŞ VE AMAÇ

Matematikte önemli bir yere sahip olan nümerik analizdeki yöntemlerle; tam olarak hesap edilemeyen değerler için yaklaşık değerler çok küçük hatalar ile elde edilebilmektedir. Bundan dolayı nümerik analizde kullanılan interpolasyon metotlarından önemli bir yere sahip olan Lagrange interpolasyon yöntemi ile verilen bir fonksiyonun bilinen değerlerinden yola çıkarak, istenen değere de ulaşılacağı gibi, aynı yöntem ile türev ve integral hesaplamaları, diferansiyel denklem çözümleri için çeşitli metotlar geliştirilmiştir. Uygulamalı bilim dallarında karşılaştığımız problemlerde birden fazla denklemler çıkabilir. Bu denklemlerde çok bilinmeyen içeren diferansiyel, integral veya integro-diferansiyel denklemlerin çözümleri güncelliğini sürekli koruyan bir araştırma konusudur. Bu tür denklemler ve denklem sistemleri; statik, mekanik, fizik, akışkanlar mekaniği, kuantum mekaniği, elektromanyetik, dalga kırınımları gibi önemli alanlarda karşılaşılan birçok diferansiyel denklem, integral denklem şeklinde kullanılabilmektedir. Bu çalışmada fizik ve mühendislik uygulamalarında önemli bir yere sahip olan fonksiyonel diferansiyel, integral ve integro-diferansiyel denklemleri için, M.T. Rashed'in 2004'te, elde ettiği nümerik çözüm yöntemleri incelenmiştir. Bu çalışmalar incelenerek, M.T. Rashed'in 2004 yılında, Volterra integral denklemlerinin, Fredholm integral denklemlerinin ve lineer integro-diferansiyel denklemlerinin nümerik çözümleri için geliştirilen iki metot ele alınmıştır. Metotlar örnekler üzerinde uygulanarak hesaplanan hata değişim değerleri tablolar halinde verilmiştir. Bu incelemeler sonucunda, metotların olumlu ve olumsuz yönleri ile birlikte değerlendirmeler yapılmıştır.

2. GENEL BİLGİLER

Bu bölüm tez çalışmasının sonunda verilen kaynaklardan yararlanılarak hazırlanmıştır.

2.1. İnterpolasyon (Ara Değer Bulma)

Basit anlamda interpolasyon işlemi tablo halinde değerleri verilen bir değişkenin tabloda bulunmayan bir değerini bulma işlemi olarak tanımlanabilir. Genel anlamda interpolasyon ise, bilinmeyen bir $y = f(x)$ fonksiyonunun $x_0, x_1, ..., x_n$ gibi ayrık noktalarda verilen $f_0, f_1, f_2, ..., f_n$ değerlerini kullanarak, bu fonksiyonun daha basit ve bilinen $Y = F(x)$ fonksiyonu ile ifade edilmesi işlemidir. Bulunan $F(x)$ fonksiyonuna 'interpolasyon fonksiyonu' denir.

2.2. Lagrange interpolasyon polinomu

Nümerik matematikte önemli bir yere sahip olan Lagrange interpolasyon yöntemini, Hildebrand, (1956) da detaylı olarak incelemiştir. Bu yöntem çalışmamıza temel teşkil ettiğinden, tanımı ile birlikte örnekler üzerinde inceleyelim.

$x_0, x_1 \ldots, x_n$ birbirlerinden farklı reel veya kompleks noktalar olmak üzere, $y = f(x)$ fonksiyonu için bu noktalara karşılık gelen fonksiyon değerleri $y_0, y_1 \ldots, y_n$ olsun. Bu durumda interpolasyon polinomu da *n*. dereceden en genel polinom olan,

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \qquad (2.2.1.a)$$

olarak tanımlanırsa. $p(x_i) = y_i;\ i = 0,1,...,n$ için,

$$\begin{aligned} y_0 &= a_0 + a_1 x_0 + a_2 x_0^{\,2} + ... + a_n x_0^{\,n} \\ y_1 &= a_1 + a_1 x_1 + a_2 x_1^{\,2} + ... + a_n x_1^{\,n} \\ &\vdots \\ y_n &= a_n + a_1 x_n + a_2 x_n^{\,2} + ... + a_n x_n^{\,n} \end{aligned} \qquad (2.2.1.b)$$

olacak biçimde $(n+1)$ bilinmeyenli ve $(n+1)$ denklemden oluşan (2.2.1.b) sistemi elde edilsin. (2.2.1.b) sisteminin katsayılar matrisini V , bilinmeyenler matrisini A ve sabitlerin matrisini Y olarak alırsak;

$$VA = Y \tag{2.2.2}$$

olarak yazılabilir. Buradaki katsayılar matrisi olan V matrisine özel olarak Vandermonde matrisi denir. (2.2.1.b) sisteminin çözümü ile derecesi n' den büyük olmayan ve $f(x)$ ile $(n+1)$ noktada uyuşan $p(x)$ interpolasyon polinomunun reel katsayılarını, dolayısıyla $p(x)$ interpolasyon polinomu tek türlü olarak elde edilebilir. Bu polinomun tekliği, aşağıdaki teorem ile verilmiştir. (Oturanç ve ark., 2008)

Teorem: $(n+1)$ farklı nokta için, bu noktalara karşılık gelen değerlere interpole edilen, yani bu karşılık gelen değerlerden geçen en fazla *n.* derece olan bir polinom vardır ve bu polinom *n.* dereceden polinomların oluşturduğu kümede tektir. (Oturanç ve ark., 2008)

İspat: Birbirinden farklı $x_0, x_1 \ldots, x_n$ değerleri ve $y = f(x)$ fonksiyonu için bu noktalara karşılık gelen fonksiyon değerleri $y_0, y_1 \ldots, y_n$ olsun. Bu durum için *n.* dereceden interpolasyon polinomu, (2.2.1.a) denklemindeki gibi olsun. Bu durumda elde edilecek olan (2.2.1.b) sisteminin katsayılar matrisinin determinantı

$$\det(V) = \begin{bmatrix} 1 & x_0 & x_0^2 \ldots & x_0^n \\ 1 & x_1 & x_1^2 \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \ldots & x_n^n \end{bmatrix}$$

olur. Burada determinanta elemanter satır ve sütun işlemleri uygulandığında,

$$\det(V) = \prod_{i>j} \left(x_i - x_j \right) \quad i, j = 0,1,...,n$$

bulunur. Bütün x_i, $i = 0,1,...,n$ noktaları birbirlerinden farklı olduklarından $\det(V) \neq 0$ dır. Yani denklemin bir tek çözümü vardır.

Elde edilen ifadelerden yola çıkarak, $p(x)$ interpolasyon polinomunu farklı bir biçimde

$$\begin{aligned} p(x) &= y_0 L_0(x) + y_1 L_1(x) + \ldots + y_n L_n(x) \\ &= \sum_{k=0}^{n} y_k L_k(x) \end{aligned} \tag{2.2.3}$$

ile ifade edelim. Ayrıca polinom verilen noktalardan geçeceğinden

$$\begin{aligned} p(x_0) &= y_0 L_0(x_0) + y_1 L_1(x_0) + \ldots + y_n L_n(x_0) = y_0 \\ p(x_1) &= y_0 L_0(x_1) + y_1 L_1(x_1) + \ldots + y_n L_n(x_1) = y_1 \\ &\vdots \\ p(x_n) &= y_0 L_0(x_n) + y_1 L_1(x_n) + \ldots + y_n L_n(x_n) = y_n \end{aligned} \tag{2.2.4}$$

olmalıdır. Burada; birinci denklemde $L_0(x_0) = 1,$ diğerleri sıfır; ikinci denklemde $L_1(x_1) = 1,$ diğerleri sıfır ve benzer şekilde diğer denklemler için;

$$\begin{aligned} P_n(x_0) &= f(x_0) \Rightarrow L_0(x_0) = 1, L_1(x_0) = 0, L_2(x_0) = 0, \ldots, L_n(x_0) = 0 \\ P_n(x_1) &= f(x_1) \Rightarrow L_0(x_1) = 0, L_1(x_1) = 1, L_2(x_1) = 0, \ldots, L_n(x_1) = 0 \\ &\vdots \\ P_n(x_n) &= f(x_n) \Rightarrow L_0(x_2) = 0, L_1(x_2) = 0, L_2(x_2) = 0, \ldots, L_{n-1}(x_2) = 0, L_n(x_2) = 1 \end{aligned}$$

elde edilir. Bir başka ifadeyle

$$\delta_{ij} = \sum_{k=0}^{n} y_k L_k(x_j) = \sum_{k=0}^{n} \delta_{ki} L_k(x_j) = L_i(x_j) \tag{2.2.5}$$

yazabiliriz. Burada δ Kronecker Deltası olup,

$$\delta_{ki} = \begin{cases} 1, & k = i \\ 0, & k \neq i \end{cases}$$

şeklindedir. Şimdi $\mathrm{L}_0(x)$ polinomunu göz önüne alalım. Kendi kökleri $x_1, x_2, \ldots, x_n$ cinsinden $\mathrm{L}_0(x)$ polinomu,

$$L_0(x) = c(x - x_1)(x - x_2)\ldots(x - x_n) \tag{2.2.6}$$

şeklinde yazılabilir. Ayrıca, $L_0(x_0)=1$ olduğundan,

$$c=\frac{1}{(x_0-x_1)(x_0-x_2)...(x_0-x_n)}$$

ve

$$L_0(x)=\frac{(x-x_1)(x-x_2)...(x-x_n)}{(x_0-x_1)(x_0-x_2)...(x_0-x_n)} \tag{2.2.7}$$

dır.

$\mathrm{L}_1(x)$ polinomu, kendi kökleri $x_0, x_2, \ldots, x_n$ cinsinden

$$L_1(x)=c_1(x-x_0)(x-x_2)...(x-x_n) \tag{2.2.8}$$

şeklinde yazılabilir. Ayrıca, $L_1(x_1)=1$ olduğundan,

$$c_1=\frac{1}{(x_1-x_0)(x_1-x_2)...(x_1-x_n)}$$

ve

$$L_1(x)=\frac{(x-x_0)(x-x_2)...(x-x_n)}{(x_1-x_0)(x_1-x_2)...(x_1-x_n)} \tag{2.2.9}$$

dır. Benzer şekilde işlemler yapılarak, $L_{\mathrm{i}}(x)$ polinomları için

$$\begin{aligned} L_{\mathrm{i}}(x) &= \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)} \\ &= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} \end{aligned} \tag{2.2.10}$$

genel formülü elde edilir. Böylece Lagrange interpolasyon polinomunu,

$$P_n(x)=\sum_{i=0}^{n} y_i L_i(x)=\sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} \tag{2.2.11}$$

şeklinde elde ederiz.

$n=2$ için, $(x_0,y_0),(x_1,y_1)$ ve (x_2,y_2) noktalarından geçen Lagrange kuadratik interpolasyon polinomu,

$$P_2(x)=y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}+y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \qquad (2.2.12)$$

olarak yazılabilir. Benzer biçimde Lagrange kübik interpolasyon polinomu da,

$$\begin{aligned}P_3(x)=&y_0\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}+y_1\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}\\&+y_2\frac{(x-x_1)(x-x_0)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}+y_3\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}\end{aligned} \qquad (2.2.13)$$

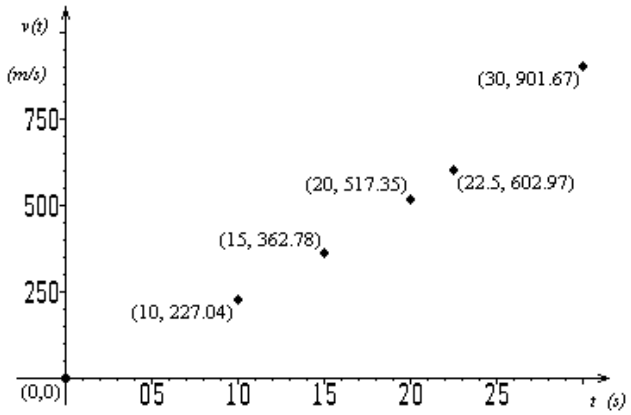
olarak tanımlanır.

Şimdi verdiğimiz tanımlar için aşağıdaki, http://numericalmethods.eng.usf.edu/mtl/gen/05inp/ de verilen örnekleri inceleyelim.

Örnek 1: Uzaya gönderilen bir roketin zamana bağlı hız değerleri Tablo 2.1'de veriliyor.

t (zaman) (s)	0	10	15	20	22.5	30
v(t) (Hız) (m/s)	0	227.04	362.78	517.35	602.97	901.67

Tablo 2.1: Hız ve zaman değerleri



Şekil 2.1: Hız ve zaman grafiği verilen roket

Roketin 16 saniyedeki hız değerini birinci dereceden polinomu kullanarak hesaplayalım.

Çözüm: Birinci dereceden bir polinom için, (Lineer interpolasyon), hızı;

$$\begin{aligned} v(t) &= \sum_{i=0}^{1} L_i(t) v(t_i) \\ &= L_0(t) v(t_0) + L_1(t) v(t_1) \end{aligned}$$

olarak seçebiliriz. t=16 saniyedeki hızını bulmak istediğimizden, t=16 ya yakın iki nokta seçeceğiz ki bu iki nokta arasında t=16 olsun. Bu noktalar t_0=15 ve t_1=20 dir.

$$t_0 = 15, v(t_0) = 362.78$$

$$t_1 = 20, v(t_1) = 517.35$$

$$\begin{aligned} L_0(t) &= \prod_{\substack{j=0 \\ j\neq 0}}^{1} \frac{t - t_j}{t_0 - t_j} \\ &= \frac{t - t_1}{t_0 - t_1} \end{aligned}$$

$$\begin{aligned} L_1(t) &= \prod_{\substack{j=0 \\ j\neq 1}}^{1} \frac{t - t_j}{t_1 - t_j} \\ &= \frac{t - t_0}{t_1 - t_0} \end{aligned}$$

olup,

$$\begin{aligned} v(t) &= \frac{t - t_1}{t_0 - t_1} v(t_0) + \frac{t - t_0}{t_1 - t_0} v(t_1) \\ &= \frac{t - 20}{15 - 20}(362.78) + \frac{t - 15}{20 - 15}(517.35) \end{aligned}$$

$$v(16) = \frac{16-20}{15-20}(362.78) + \frac{16-15}{20-15}(517.35)$$
$$= 0.8(362.78) + 0.2(517.35)$$
$$= 393.7\,\text{m/s}$$

olarak elde edilir. Şimdi Örnek 1'i Maple programı yardımıyla çözelim:

Roketin hız verilerini *y* değerleri olarak ve zaman verilerini de *x* değerleri olarak yazalım:

```
> restart;
with(LinearAlgebra):
with(linalg):
>
xy:=[[10,227.04],[0,0],[20,517.35],[15,362.78],[30,901.67
],[22.5,602.97]]:
```

Roketin 16 saniyesindeki değeri istenmişti. Buna göre,

```
> xdesired:=16:
```

olarak yazalım.

Aşağıda tanımlayacağımız fonksiyon *x* verisini ele alır. Bu fonksiyon *x* in istenen değerini eşit kabul eden en yakın iki veri noktasını seçer. İlk olarak istenen *x* değerinin yerine en yakın veri noktasını alır. Eğer azsa daha sonra istenen değerden, ayrıca en yakın değerden daha büyük veri noktasını seçer ve bunun birde tam tersini yapar.

```
> n:=rowdim(xy):
> for i from 1 to n do
x[i]:=xy[i,1];
y[i]:=xy[i,2];
end do:
```

x değerleri ve istenen *x* değeri arasındaki mutlak farkı bulur:

```
> for i from 1 to n do
  co[i]:=abs(x[i]-xdesired);
end do:
```

En küçük mutlak farkla *x* değerini tespit eder:

```
c:=co[1]:
for i from 2 to n do
  if c > co[i] then
    c:=co[i];
    ci:=i;
  end if:
end do:
```

Eğer en küçük mutlak fark değeri istenen değerden küçükse, istenen değeri eşit kabul etmek için istenen değerden daha büyük en yakın veri noktasını seçer:

```
if x[ci] < xdesired then
  q:=1;
  for i from 1 to n do
    if x[i] > xdesired then
      nex[q]:=x[i];
      q:=q+1;
    end if;
  end do;

  b:=nex[1]:
  for i from 2 to q-1 do
    if b > nex[i] then
      b:=nex[i];
    end if:
  end do:

  for i from 1 to n do
    if x[i]=b then bi:=i end if;
  end do;
end if:
```

Eğer en küçük mutlak fark değeri istenen değerden büyükse; istenen değeri eşit kabul etmek için istenen değerden daha küçük en yakın veri noktasını seçer.

```
if x[ci] > xdesired then
  q:=1;
```

```
  for i from 1 to n do
    if x[i] < xdesired then
      nex[q]:=x[i];
      q:=q+1;
    end if;
  end do;

  b:=nex[1]:
  for i from 2 to q-1 do
    if b < nex[i] then
      b:=nex[i];
    end if:
  end do:

  for i from 1 to n do
    if x[i]=b then bi:=i end if;
  end do;
end if:

firsttwo:=<ci,bi>:
```

İki değerden daha fazla değer gerekiyorsa, istenen değeri eşit kabul eden iki veri noktasını seçmek için yukarıdaki aynı işlem takip edilir. Ayrıca, aşağıdaki fonksiyon istenen değere en yakın takip eden değerleri seçer ve orijinal veri derecesini koruyarak tüm değerleri dizeye koyar.

```
> for i from 1 to n do
  A[i,2]:=i;
  A[i,1]:=co[i];
end do:

for passnum from 1 to n-1 do
  for i from 1 to n-passnum do
    if A[i,1]>A[i+1,1] then
      temp1:=A[i,1];
```

```
      temp2:=A[i,2];
      A[i,1]:=A[i+1,1];
      A[i,2]:=A[i+1,2];
      A[i+1,1]:=temp1;
      A[i+1,2]:=temp2;
    end if:
  end do:
end do:

for i from 1 to n do
  A[i,3]:=i;
end do:

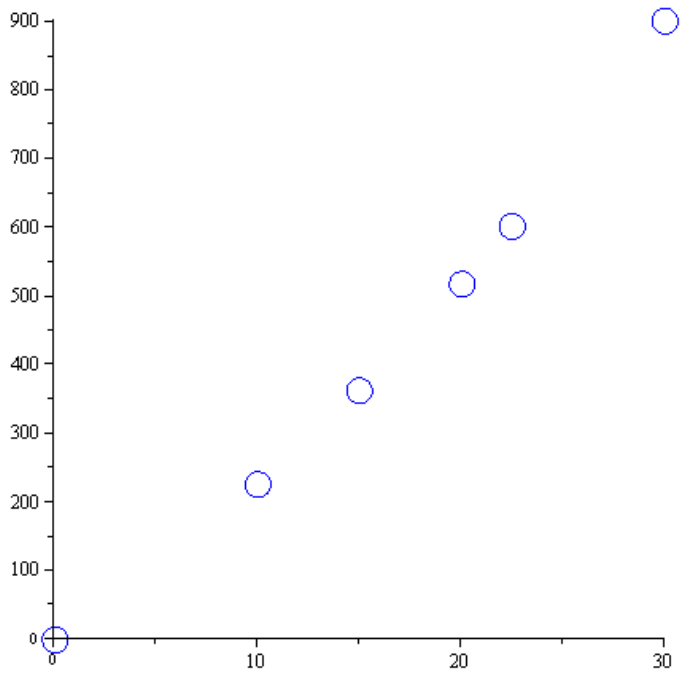
for passnum from 1 to n-1 do
  for i from 1 to n-passnum do
    if A[i,2]>A[i+1,2] then
      temp1:=A[i,1];
      temp2:=A[i,2];
      temp3:=A[i,3];
      A[i,1]:=A[i+1,1];
      A[i,2]:=A[i+1,2];
      A[i,3]:=A[i+1,3];
      A[i+1,1]:=temp1;
      A[i+1,2]:=temp2;
      A[i+1,3]:=temp3;
    end if:
  end do:
end do:

for i from 1 to n do
  d[i]:=A[i,3];
end do:
```

```
if d[firsttwo[2]]<>2 then
  temp:=d[firsttwo[2]];
  d[firsttwo[2]]:=1;
  for i from 1 to n do
    if i <> firsttwo[1] and i <> firsttwo[2] and d[i] <=
temp then
       d[i]:=d[i]+1;
    end if:
  end do;
  d[firsttwo[1]]:=1;
end if:
> ranger:=proc(ar,n)
local i,xmin,xmax,xrange;
xmin:=ar[1]:
xmax:=ar[1]:
for i from 1 to n do
if ar[i] > xmax then xmax:=ar[i] end if;
if ar[i] < xmin then xmin:=ar[i] end if;
end do;
xrange:=xmin..xmax;
end proc:
```

x ve *y* nin verilen değerlerinin tablosu çizersek:

```
>
plot(xy,ranger(x,n),style=POINT,color=BLUE,symbol=CIRCLE,
symbolsize=30);
```



Şekil 2.2: Roketin hız-zaman grafiği

Lineer interpolasyonu oluşturmak ve istenen değer için en yakın iki veri noktası seçilir.

```
> datapoints:=2:
> p:=1:
for i from 1 to n do
if d[i] <= datapoints then
xdata[p]:=x[i];
ydata[p]:=y[i];
p:=p+1;
end if
end do:
```

$$[20],[15]$$

```
> entries(ydata);
```

$$[517.35],[362.78]$$

Yukarda bahsedilen formül kullanılarak ağırlık fonksiyonları hesaplanmıştır. Bu fonksiyonlar seçilen *x* değerlerine karşılık gelen *y* değerleri ağırlıklarını verir. Ağırlıkların toplamı daima bire eşittir.

```
> L0(z):=(z-xdata[2])/(xdata[1]-xdata[2]):
> L1(z):=(z-xdata[1])/(xdata[2]-xdata[1]):
```

Lagrange interpolasyonu düzenlenir.

```
>
flinear(z):=eval(L0(t),t=z)*ydata[1]+eval(L1(t),t=z)*ydat
a[2];
```

$$flinear(z) := 3091400000z - 100.93$$

z için istenen *x* değeri yukarıdaki eşitlikte yerine konularak karşılık gelen *y* değeri bulunur.

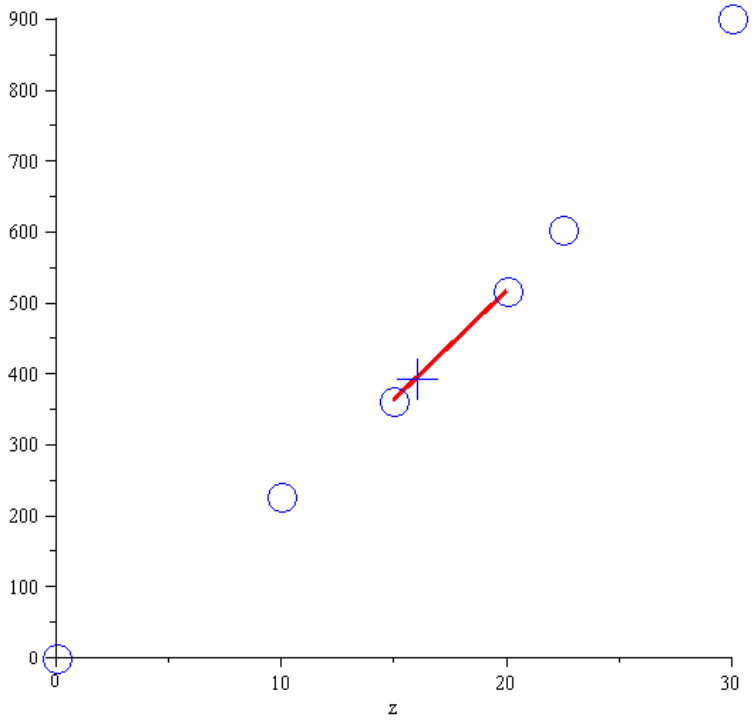
```
> eval(flinear(z),z=xdesired);
```

$$[393.6940000]$$

```
> fprev:=%:
```

Lineer interpolasyonu ve istenen *x* için *y* değerinin tablosu çizilir.

```
>
plot([[t,eval(flinear(z),z=t),t=ranger(xdata,datapoints)]
,xy,[[xdesired,eval(flinear(z),z=xdesired)]]],z=ranger(x,
n),style=[LINE,POINT,POINT],color=[RED,BLUE,BLUE],symbol=
[CROSS,CIRCLE],symbolsize=[40,30],thickness=3,title="Line
ar interpolation");
```



Şekil 2.3: Lineer İnterpolasyon

Örnek 2: Uzaya gönderilen bir roketin zamana bağlı artan hız değerleri Tablo 2.2 de

t	0	10	15	20	22.5	30
v(t)	0	227.04	362.78	517.35	602.97	901.67

Tablo 2.2: Hız ve zaman değerleri

olarak verilsin. Roketin 16 saniyedeki hız değerini ikinci dereceden bir polinom kullanarak hesaplayalım. Ayrıca hata seviyesini ikinci dereceden polinom için bulalım.

Çözüm: t =16 saniyedeki hızını bulmak için, t =16 noktasının içinde bulunduğu üç noktayı t_0=10, t_1=15, t_2=20 olarak seçebiliriz**.**

$$t_o = 10,\ \ v(t_o) = 227.04$$

$$t_1 = 15,\ \ v(t_1) = 362.78$$

$$t_2 = 20,\ \ v(t_2) = 517.35$$

için

$$L_0(t) = \prod_{\substack{j=1 \\ j\neq 0}}^{2} \frac{t-t_j}{t_0-t_j} = \left(\frac{t-t_1}{t_0-t_1}\right)\left(\frac{t-t_2}{t_0-t_2}\right)$$

$$L_1(t) = \prod_{\substack{j=0 \\ j\neq 1}}^{2} \frac{t-t_j}{t_1-t_j} = \left(\frac{t-t_0}{t_1-t_0}\right)\left(\frac{t-t_2}{t_1-t_2}\right)$$

$$L_2(t) = \prod_{\substack{j=0 \\ j\neq 2}}^{2} \frac{t-t_j}{t_2-t_j} = \left(\frac{t-t_0}{t_2-t_0}\right)\left(\frac{t-t_1}{t_2-t_1}\right)$$

olup,

$$v(t) = \left(\frac{t-t_1}{t_0-t_1}\right)\left(\frac{t-t_2}{t_0-t_2}\right)v(t_0) + \left(\frac{t-t_0}{t_1-t_0}\right)\left(\frac{t-t_2}{t_1-t_2}\right)v(t_1) + \left(\frac{t-t_0}{t_2-t_0}\right)\left(\frac{t-t_1}{t_2-t_1}\right)v(t_2)$$

$$\begin{aligned} v(16) &= \frac{(16-15)(16-20)}{(10-15)(10-20)}(227.04) + \frac{(16-10)(16-20)}{(15-10)(15-20)}(362.78) \\ &\qquad + \frac{(16-10)(16-15)}{(20-10)(20-15)}(517.35) \\ &= (-0.08)(227.04) + (0.96)(362.78) + (0.12)(517.35) \\ &= 392.19\,\text{m/s} \end{aligned}$$

elde edilir.

Bir önceki örnekle bu örnek arasındaki hesaplama değerinin göreceli hata yüzdesi,

$$|\in_a| = \left|\frac{392.19 - 393.70}{392.19}\right| \times 100$$

$$|\in_a| = 0.38502\%$$

dir.

Şimdi Örnek 2'yi Maple programı yardımıyla çözelim:

İkinci dereceden (kuadratik) interpolasyon polinomu oluşturmak ve istenen değeri bulmak için en yakın 3 veri noktası seçilir.

```
> datapoints:=3:
> p:=1:
for i from 1 to n do
if d[i] <= datapoints then
xdata[p]:=x[i];
ydata[p]:=y[i];
p:=p+1;
end if
end do:

> entries(xdata);
```

$$[10],[20],[15]$$

```
> entries(ydata);
```

[227.04], [517.35], [362.78]

Yukarda bahsedilen formülü kullanarak ağırlık fonksiyonları elde edilmiştir. Bu fonksiyonlar seçilen *x* değerlerine karşılık *y* değerlerlerine ağırlık verir.

```
>L0(z):=((z-xdata[2])*(z-xdata[3]))/(xdata[1]-
xdata[2])/(xdata[1]-xdata[3]):
>L1(z):=((z-xdata[1])*(z-xdata[3]))/(xdata[2]-
xdata[1])/(xdata[2]-xdata[3]):
>L2(z):=((z-xdata[1])*(z-xdata[2]))/(xdata[3]-
xdata[1])/(xdata[3]-xdata[2]):
```

Lagrange polinomunu düzenlersek,

```
>fquad(z):=eval(L0(t),t=z)*ydata[1]+eval(L1(t),t=z)*ydata
[2]+eval(L2(t),t=z)*ydata[3];
```

$$fquad(z) := 4.540800000(z-20)(z-15)+10.34700000(z-10)(z-15)$$
$$-14.51120000(z-10)(z-20)$$

z için istenen *x* değeri yukarıdaki eşitlikte yerine konularak karşılık gelen *y* değeri bulunur.

```
> eval(fquad(z),z=xdesired);
```

392.1876000

```
> fnew:=%:
```

Birinci derece ve ikinci derece interpolasyon arasındaki mutlak yüzde göreceli yaklaşık hata şudur;

```
> epsilon:=abs((fnew-fprev)/fnew)*100;
```

ε:=0.3841018941

Çözüme ulaşmak için virgülden sonraki basamak sayısı;

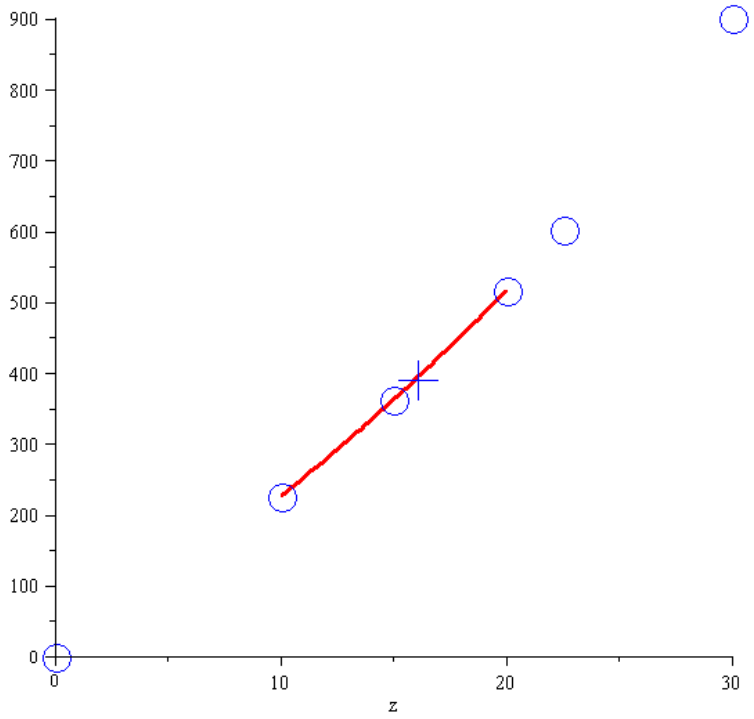
```
> sigdig:=floor(2-log10(epsilon/0.5));
```

$$sigdig := 2$$

```
> fprev:=fnew:
```

İkinci dereceden interpolasyon ve istenen *x* için *y* değeri tabloya dökülür;

```
>
plot([[t,eval(fquad(z),z=t),t=ranger(xdata,datapoints)],x
y,[[xdesired,eval(fquad(z),z=xdesired)]],z=ranger(x,n),s
tyle=[LINE,POINT,POINT],color=[RED,BLUE,BLUE],symbol=[CRO
SS,CIRCLE],symbolsize=[40,30],thickness=3,title="Quadrati
c interpolation");
```



Şekil 2.4: Kuadratik İnterpolasyon

Örnek 3: Uzaya gönderilen bir roketin zamana bağlı artan hız değerleri Tablo 2.3 de;

t	0	10	15	20	22.5	30
v(t)	0	227.04	362.78	517.35	602.97	901.67

Tablo 2.3: Hız ve zaman değerleri

şeklinde verilsin. Lagrange interpolasyon yöntemiyle, roketin 16. saniyedeki hız değerini üçüncü dereceden bir polinom kullanarak hesaplayalım. Ayrıca hata seviyesini üçüncü dereceden polinom için bulalım.

Çözüm: Üçüncü dereceden bir polinom için (kübik interpolasyon olarak da adlandırılabilir), hızı şu şekilde seçebiliriz;

$$v(t) = \sum_{i=0}^{3} L_i(t) v(t_i)$$
$$= L_0(t)v(t_0) + L_1(t)v(t_1) + L_2(t)v(t_2) + L_3(t)v(t_3)$$

Üçüncü dereceden bir polinomu kullanarak, t=16 saniyesinde hızını bulmak istediğimizden, t=16 noktasının içinde bulunduğu dört noktayı t_0=10, t_1=15, t_2=20 ve t_3 =22.5 olarak seçebiliriz**.**

$$t_o = 10,\ \ v(t_o) = 227.04$$

$$t_1 = 15,\ \ v(t_1) = 362.78$$

$$t_2 = 20,\ \ v(t_2) = 517.35$$

$$t_3 = 22.5,\ \ v(t_3) = 602.97$$

böylece

$$L_0(t) = \prod_{\substack{j=1 \\ j\neq 0}}^{3} \frac{t - t_j}{t_0 - t_j}$$
$$= \left(\frac{t - t_1}{t_0 - t_1}\right)\left(\frac{t - t_2}{t_0 - t_2}\right)\left(\frac{t - t_3}{t_0 - t_3}\right)$$

$$L_1(t) = \prod_{\substack{j=0 \\ j\neq 1}}^{3} \frac{t - t_j}{t_1 - t_j}$$
$$= \left(\frac{t - t_0}{t_1 - t_0}\right)\left(\frac{t - t_2}{t_1 - t_2}\right)\left(\frac{t - t_3}{t_1 - t_3}\right)$$

$$L_2(t) = \prod_{\substack{j=0 \\ j\neq 2}}^{3} \frac{t - t_j}{t_2 - t_j}$$
$$= \left(\frac{t - t_0}{t_2 - t_0}\right)\left(\frac{t - t_1}{t_2 - t_1}\right)\left(\frac{t - t_3}{t_2 - t_3}\right)$$

$$L_3(t) = \prod_{\substack{j=0 \\ j\neq 3}}^{3} \frac{t - t_j}{t_3 - t_j}$$
$$= \left(\frac{t - t_0}{t_3 - t_0}\right)\left(\frac{t - t_1}{t_3 - t_1}\right)\left(\frac{t - t_2}{t_3 - t_2}\right)$$

olup,

$$v(t)=\left(\frac{t-t_1}{t_0-t_1}\right)\left(\frac{t-t_2}{t_0-t_2}\right)\left(\frac{t-t_3}{t_0-t_3}\right)v(t_0)+\left(\frac{t-t_0}{t_1-t_0}\right)\left(\frac{t-t_2}{t_1-t_2}\right)\left(\frac{t-t_3}{t_1-t_3}\right)v(t_1)$$
$$+\left(\frac{t-t_0}{t_2-t_0}\right)\left(\frac{t-t_1}{t_2-t_1}\right)\left(\frac{t-t_3}{t_2-t_3}\right)v(t_2)+\left(\frac{t-t_0}{t_3-t_0}\right)\left(\frac{t-t_1}{t_3-t_1}\right)\left(\frac{t-t_2}{t_3-t_2}\right)v(t_3)$$

$$v(16)=\frac{(16-15)(16-20)(16-22.5)}{(10-15)(10-20)(10-22.5)}(227.04)+\frac{(16-10)(16-20)(16-22.5)}{(15-10)(15-20)(15-22.5)}(362.78)$$
$$+\frac{(16-10)(16-15)(16-22.5)}{(20-10)(20-15)(20-22.5)}(517.35)+\frac{(16-10)(16-15)(16-20)}{(22.5-10)(22.5-15)(22.5-20)}(602.97)$$
$$=(-0.0416)(227.04)+(0.832)(362.78)+(0.312)(517.35)+(-0.1024)(602.97)$$
$$=392.06\,\text{m/s}$$

bulunur. Bir önceki örnekle bu örnek arasındaki hesaplama değerindeki göreceli hata yüzdesi,

$$|\in_a|=\left|\frac{392.06-392.19}{392.06}\right|\times 100$$
$$=0.033427\,\%$$

dir.

Şimdi Örnek 3'ü Maple programı yardımıyla çözelim:

Üçüncü dereceden interpolasyon polinomunu oluşturmak ve istenen değeri bulmak için en yakın dört veri noktası seçilir.

```
> datapoints:=4:
> p:=1:
for i from 1 to n do
if d[i] <= datapoints then
xdata[p]:=x[i];
ydata[p]:=y[i];
p:=p+1;
end if
end do:

> entries(xdata);
```

[10], [20], [15], [22.5]

```
> entries(ydata);
```

[227.04], [517.35], [362.78], [602.97]

Yukarda bahsedilen formül kullanılarak ağırlık fonksiyonları elde edilmiştir. Bu fonksiyonlar seçilen *x* değerlerine karşılık gelen *y* değerlerine ağırlık verir.

```
>L0(z):=((z-xdata[2])*(z-xdata[3])*(z-
xdata[4]))/(xdata[1]-xdata[2])/(xdata[1]-
xdata[3])/(xdata[1]-xdata[4]):
>L1(z):=((z-xdata[1])*(z-xdata[3])*(z-
xdata[4]))/(xdata[2]-xdata[1])/(xdata[2]-
xdata[3])/(xdata[2]-xdata[4]):
>L2(z):=((z-xdata[1])*(z-xdata[2])*(z-
xdata[4]))/(xdata[3]-xdata[1])/(xdata[3]-
xdata[2])/(xdata[3]-xdata[4]):
>L3(z):=((z-xdata[1])*(z-xdata[2])*(z-
xdata[3]))/(xdata[4]-xdata[1])/(xdata[4]-
xdata[2])/(xdata[4]-xdata[3]):
```

Lagrange polinomunu düzenlersek,

```
>
fcubic(z):=eval(L0(t),t=z)*ydata[1]+eval(L1(t),t=z)*ydata
[2]+eval(L2(t),t=z)*ydata[3]+eval(L3(t),t=z)*ydata[4];
```

$$fcubic(z) := -.3632640000(z-20)(z-15)(z-22.5)-4.138800000(z-10)(z-15)(z-22.5) +1.934826667(z-10)(z-20)(z-22.5)+2.572672000(z-10)(z-20)(z-15)$$

z için istenen *x* değeri yukarıdaki eşitlikte yerine konularak karşılık gelen *y* değeri bulunur;

```
> eval(fcubic(z),z=xdesired);
```

392.0571681

```
> fnew:=%:
```

İkinci derece ve üçüncü derece interpolasyon arasındaki mutlak yüzde göreceli yaklaşık hata;

```
> epsilon:=abs((fnew-fprev)/fnew)*100;
```

ε:=0.03326859209

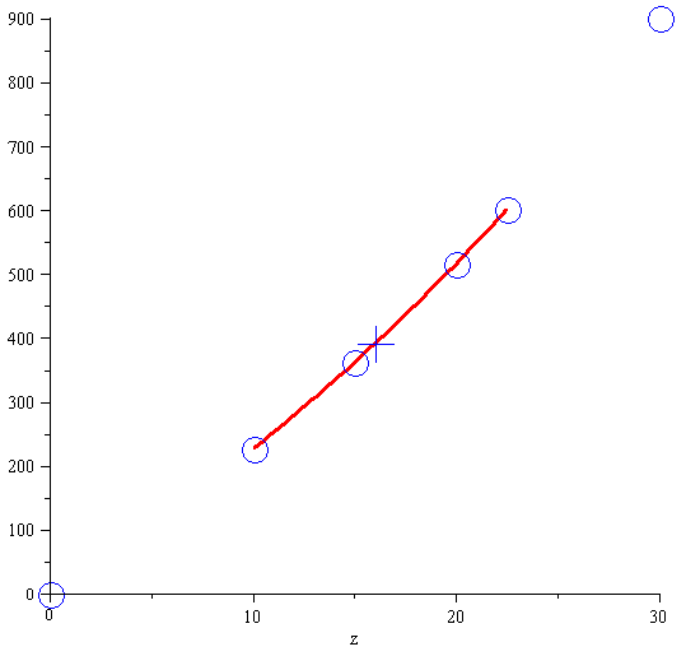
Çözüme ulaşmak için virgülden sonraki basamak sayısı;

```
> sigdig:=floor(2-log10(epsilon/0.5));
```

sigdig:=3

Üçüncü derece interpolasyonunun ve istenen *x* için *y* değerinin tablosu çizilir.

```
>
plot([[t,eval(fcubic(z),z=t),t=ranger(xdata,datapoints)],
xy,[[xdesired,eval(fcubic(z),z=xdesired)]],z=ranger(x,n)
,style=[LINE,POINT,POINT],color=[RED,BLUE,BLUE],symbol=[C
ROSS,CIRCLE],symbolsize=[40,30],thickness=3,title="Cubic
interpolation");
```



Şekil 2.5: Kübik İnterpolasyon

2.3. Chebyshev polinomları

İntegrallerin yaklaşık değerlerinin hesaplanmasında Chebyshev polinomları ve serisinden yararlanacağız. Davis ve ark. (1965), tarafından verilen polinomun tanımı ve genel yapısı ile integrasyonuna dair bilgiler bu kısımda incelenmiştir.

$$T_n(x) = \cos(n \arccos x), \quad (n = 0,1,2,...) \tag{2.3.1}$$

biçiminde tanımlanan $T_n(x)$ polinomuna n'inci 1.tür dereceden Chebyshev polinomu denir. $\arccos x = \theta$ alınırsa;

$$T_n(\cos\theta) = \cos n\theta \tag{2.3.2}$$

olur.

$$T_{n+1}(x) = \cos(n+1)\theta = \cos n\theta \cos\theta - \sin n\theta \sin\theta$$
$$T_{n-1}(x) = \cos(n-1)\theta = \cos n\theta \cos\theta + \sin n\theta \sin\theta$$

ise

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos n\theta \cos\theta = 2T_n(x).x$$

olur. Düzenlersek

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

elde edilir. Buradan,

$$T_0(x) = \cos 0 = 1$$
$$T_1(x) = \cos(\arccos x) = x$$
$$T_2(x) = 2x.x - 1 = 2x^2 - 1$$
$$T_3(x) = 2x.(2x^2 - 1) - x = 4x^3 - 3x$$

benzer biçimde

$$T_4(x) = 8x^4 - 8x^2 + 1$$

elde edilebilir. Böylece diğer Chebyshev polinomlarını da bulunabilir. Davis ve ark. (1965) de, $f(x)$ fonksiyonu için Chebyshev serisini;

$$f(x)=\frac{1}{2}a_0T_0(x)+a_1T_1(x)+a_2T_2(x)+...$$

biçiminde ifade etmişlerdir. Bu ifadeyi sonlu bir toplam olarak;

$$f(x)=\frac{1}{2}a_0T_0(x)+a_1T_1(x)+a_2T_2(x)+...+a_NT_N(x)=\sum_{k=0}^{N'}a_kT_k(x) \quad (2.3.3)$$

şeklinde yazabiliriz. Chebyshev katsayıları da;

$$a_k=\frac{2}{N}\sum_{k=0}^{N''}f\left(\cos\frac{\pi j}{N}\right)\cos\frac{\pi kj}{N}$$

ile belirtilmiştir. Burada N' ifadesi $k=0$ için bulunan terimin 0.5 ile çarpılacağını, N'' ifadesi de $j=0$ ve $j=N$ için bulunan terimin 0.5 ile çarpılacağını belirtir.

Chebyshev polinomları aşağıdaki şekilde integre edilebilir:

$$\int T_0(x)dx=\int dx=x+c=T_1(x)+c$$
$$\int T_1(x)dx=\int xdx=\frac{x^2}{2}+c=\frac{1}{4}T_2(x)+c$$
$$\vdots$$
$$\int T_n(x)dx=\int\cos(n\arccos x)dx=$$

integralinde $\arccos x=\theta$ dönüşümü uygulanırsa; $x=\cos\theta$ ve $dx=-\sin\theta\,d\theta$ olup,

$$\int\cos(n\arccos x)dx=\int\cos n\theta\left(-\sin\theta\right)d\theta$$

elde edilir. Burada ters dönüşüm formüllerini kullanarak,

$$\left(\cos a.\sin b=\frac{1}{2}\left[\sin(a+b)-\sin(a-b)\right]\right)$$

$$\int\cos n\theta\left(-\sin\theta\right)d\theta=-\int\frac{1}{2}\left[\sin(n+1)\theta-\sin(n-1)\theta\right]d\theta$$

elde edilir. İntegral alındıktan sonra, elde edilen ifade Chebyshev polinomları cinsinden yazılırsa;

$$= -\frac{1}{2}\left(\frac{-\cos(n+1)\theta}{n+1} + \frac{\cos(n-1)\theta}{n-1} \right) + c$$
$$= \frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1} \right) + c$$

olur. $[-1,1]$ aralığında $f(x)$ fonksiyonunun integralini alırsak,

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} \left[\frac{1}{2}a_0 T_0(x) + a_1 T_1(x) + a_2 T_2(x) + \ldots + a_N T_N(x) \right] dx = \int_{-1}^{1} \sum_{k=0}^{N'} a_k T_k(x)dx$$
$$= \left(\frac{a_0}{2} T_1(x) + \frac{a_1}{4} T_2(x) + \sum_{r=2}^{N} \frac{a_r}{2} \left(\frac{T_{r+1}(x)}{r+1} - \frac{T_{r-1}(x)}{r-1} \right) \right) \Bigg|_{-1}^{1}$$

bulunur. Burada sigmanın indisi tek ise $T_{r+1}(x)$ ve $T_{r-1}(x)$ indisleri çift, sigmanın indisi çift ise $T_{r+1}(x)$ ve $T_{r-1}(x)$ indisleri tek olacaktır. Dolayısıyla

$$\begin{aligned} T_{2N}(x)\big|_{-1}^{1} &= T_{2N}(1) - T_{2N}(-1) \\ &= \cos\left(2N \arccos 1\right) - \cos(2N \arccos(-1)) \\ &= \cos 0 - \cos(2N\pi) \\ &= 1 - 1 \\ &= 0 \end{aligned}$$

ve

$$\begin{aligned} T_{2N+1}(x)\big|_{-1}^{1} &= T_{2N+1}(1) - T_{2N+1}(-1) \\ &= \cos\left((2N+1) \arccos 1\right) - \cos((2N+1) \arccos(-1)) \\ &= \cos 0 - \cos((2N+1)\pi) \\ &= 1 - (-1) \\ &= 2 \end{aligned}$$

den

$$T_N(x)\big|_{-1}^{1} = \begin{cases} 2, & N \text{ tek} \\ 0, & N \text{ çift} \end{cases}$$

elde edilir. Buradan,

$r = 2$ ise

$$\frac{a_2}{2}\left(\frac{T_3(x)}{3} - \frac{T_1(x)}{1}\right)\Big|_{-1}^{1} = \frac{a_2}{2}\left(\frac{1}{3}T_3(x)\Big|_{-1}^{1} - T_1(x)\Big|_{-1}^{1}\right)$$
$$= \frac{a_2}{2}\left(\frac{2}{3} - 2\right) = a_2\left(-\frac{2}{3}\right) \quad ,$$

$r = 3$ ise

$$\frac{a_3}{2}\left(\frac{T_4(x)}{4} - \frac{T_2(x)}{2}\right)\Big|_{-1}^{1} = \frac{a_3}{2}\left(\frac{1}{4}T_4(x)\Big|_{-1}^{1} - \frac{1}{2}T_2(x)\Big|_{-1}^{1}\right)$$
$$= 0 \quad ,$$

$r = N \quad (N \text{ tek})$

$$\frac{a_N}{2}\left(\frac{T_{N+1}(x)}{N+1} - \frac{T_{N-1}(x)}{N-1}\right)\Big|_{-1}^{1} = \frac{a_N}{2}\left(\frac{1}{N+1}T_{N+1}(x)\Big|_{-1}^{1} - \frac{1}{N-1}T_{N-1}(x)\Big|_{-1}^{1}\right)$$
$$= 0 \quad ,$$

$r = N \quad (N \text{ çift})$

$$\frac{a_N}{2}\left(\frac{T_{N+1}(x)}{N+1} - \frac{T_{N-1}(x)}{N-1}\right)\Big|_{-1}^{1} = \frac{a_N}{2}\left(\frac{1}{N+1}T_{N+1}(x)\Big|_{-1}^{1} - \frac{1}{N-1}T_{N-1}(x)\Big|_{-1}^{1}\right)$$
$$= \frac{a_N}{2}\left(\frac{1}{N+1}.2 - \frac{1}{N-1}.2\right) = \frac{a_N}{2}.\frac{4}{1-N^2}$$
$$= a_N.\frac{2}{1-N^2} \quad ,$$

olup integral için

$$\int_{-1}^{1} f(x)dx = \begin{cases} a_o + a_2\left(\dfrac{2}{1-2^2}\right) + a_4\left(\dfrac{2}{1-4^2}\right) + \ldots + a_N\left(\dfrac{2}{1-N^2}\right), & (N \text{ çift}) \\ a_o + a_2\left(\dfrac{2}{1-2^2}\right) + a_4\left(\dfrac{2}{1-4^2}\right) + \ldots + a_{N-1}\left(\dfrac{2}{1-(N-1)^2}\right), & (N \text{ tek}) \end{cases}$$

ifadesini yazarız. Burada,

$$I_k = \begin{cases} 2, & k = 0 \\ 0, & k \text{ tek} \\ \dfrac{2}{1-k^2}, & k \text{ çift} \end{cases}$$

olarak alırsak,

$$\int_{-1}^{1} f(x)dx = \sum_{k=0}^{N''} I_k a_k$$

ifadesinden

$$\begin{aligned} \sum_{k=0}^{N''} I_k a_k &= \sum_{k=0}^{N''} I_k \frac{2}{N} \sum_{s=0}^{N''} f\left(\cos\frac{\pi s}{N}\right)\cos\frac{\pi k s}{N} \\ &= \sum_{s=0}^{N''} \frac{2}{N} \sum_{k=0}^{N''} I_k f\left(\cos\frac{\pi s}{N}\right)\cos\frac{\pi k s}{N} \end{aligned}$$

elde edilir. Yine,

$$t_s = \cos\left(\frac{s\pi}{N}\right), \qquad \delta_j = \begin{cases} 0{,}5, & k = 0, N \\ 1, & 0 < k < N \end{cases}$$

olarak alınırsa,

$$\int_{-1}^{1} f(x)dx = \sum_{s=0}^{N} \frac{2\delta_s}{N} \sum_{k=0}^{N''} I_k f(t_s)\cos\frac{k\pi s}{N}$$

bulunur.

$$b_s = \frac{2\delta_s}{N} \sum_{k=0}^{N''} I_k \cos\left(\frac{k\pi s}{N}\right)$$

olarak alınırsa da,

$$\int_{-1}^{1} f(x)dx = \sum_{s=0}^{N} b_s f(t_s) \tag{2.3.3}$$

elde edilmiş olur.

2.4. İntegral denklemleri

İntegral denklemler 19. yüzyılın başlarında ele alınmış ve üzerinde çalışmalara başlanmıştır. 1823 yılında, ABEL bir mekanik problemi çalışırken, integral denkleme rastlaması ve ilk olarak integral denklem deyimini 1888 yılında De Bais REYMOND bir çalışmasında kullanması, konu alanının başlangıcı hakkında bilgi verir. (Bocher M, 1913)

2.4.1. Volterra integral denklemleri

Matematikte Volterra integral denklemleri, integral denklemlerinin özel bir tipidir. Bunlar birinci tip ve ikinci tip olmak üzere iki gruba ayrılırlar.

Birinci tip lineer Volterra denklemi

$$f(t) = \int_a^t K(t,s)x(s)ds \tag{2.4.1}$$

ile tanımlanır.

İkinci tip lineer Volterra denklemi ise

$$x(t) = f(t) + \int_a^t K(t,s)x(s)ds \tag{2.4.2}$$

ile verilir.

Lineer Volterra integral denklemi eğer,

$$x(t) = f(t) + \int_a^t K(t-s)x(s)ds \tag{2.4.3}$$

ile verilirse konvolüsyon(girişim) denklemi olur.

İntegraldeki K fonksiyonuna genellikle kernel(çekirdek) fonksiyonu adı verilir. Böyle denklemler Laplace dönüşümleri yardımıyla analiz edilebilir ve çözülebilir. Volterra integral denklemleri; demografi uygulaması, elastik materyallerin çalışması gibi değişik alanlarda da önemli bir yere sahiptir.

2.4.2. Fredholm integral denklemleri

Matematikte Fredholm integral denklemi; çözümü Fredholm teorisinin,(yani Fredholm çekirdekleri ve Fredholm operatörleri çalışmasının), doğuşunu sağlayan integral denklemidir. İntegral denklemi Ivar Fredholm tarafından çalışılmıştır.

Birinci tip homojen Fredholm denklemi;

$$g(t) = \int_a^b K(t,s)\, f(s) ds \tag{2.4.4}$$

şeklindedir ve *f(s)* fonksiyonunu bulmak için sürekli çekirdek fonksiyonu *K(t,s)* ve *g(t)* fonksiyonu verilir. Eğer çekirdek fonksiyonu, sadece değişkenlerinden farklı olarak bir denklem olursa; yani $K(t,s) = K(t-s)$ ve integrallemenin limitleri $\pm\infty$ olursa, sonuç olarak denklemin sağ tarafı K ve f fonksiyonlarının konvolusyonu olarak tekrar yazılabilir ve bu yüzden çözüm

$$f(t) = F_w^{-1}\left[\frac{F_t[g(t)](w)}{F_t[K(t)](w)}\right] = \int_{-\infty}^{\infty} \frac{F_t[g(t)](w)}{F_t[K(t)](w)} e^{2\pi i w t} dw \tag{2.4.5}$$

olarak elde edilir. F_t ve F_w^{-1} sırasıyla doğrudan ve ters Fourier dönüşümüdür.

Homojen olmayan ikinci tip Fredholm denklemi,

$$f(t) = \phi(t) + \lambda \int_a^b K(t-s)\phi(s) ds$$

ile verilir. *K(t,s)* çekirdeği ve *f(t)* fonksiyonu verildiğinde, problem tipik olarak $\phi(t)$ fonksiyonunu bulmaya dönüşür. Bunu çözmek için standart yaklaşım seri olarak yazılan resolvent formal ismi kullanmaktır. Sonuç ise Liouville-Neumann serileri olarak bilinir.

2.5. Gauss – Lengendre İntegrasyonu

Nümerik matematikteki, önemli bir yere sahip olan integrasyon konusunu Oturanç ve ark., (2008) de detaylı olarak ele almıştır. Bu çalışmada eşit olmayan aralıklar için Gauss – Lengendre İntegrasyonu inceleyelim.

Gauss integrasyon formülleri genelde, mümkün olan en doğru yaklaşık sonucu elde edecek şekilde düzenlenmiştir. Gauss, çalışmalarında *f(x)*'in integralinin elde edilmesinde kullanılan x_i bağımsız değişken değerlerinin hatayı minimumda tutmak için eşit aralıklı olması gerektiği sonucuna varmıştı. Yaptığı çalışmalara göre;

$$I = \int_a^b f(x)dx$$

şeklinde verilen integral, $x = \frac{b+a}{2} + \frac{b-a}{2}t$ dönüşümü ile

$$I = \frac{b-a}{2}\int_{-1}^{1} F(t)dt$$

şeklinde yazılabilir. *F(t)*'nin integrali de,

$$\int_{-1}^{1} F(t)dt = \sum_{i=0}^{n} w_i F(t_i) + R_n$$

olarak verilebilir; öyle ki, t_i kökleri (yani interpolasyon noktalarının yerleri) R_n minimum yapacak şekilde belirlenir. t_i kökleri ve w_i ağırlık çarpanları [-1,1] aralığında t=0'a göre simetrik olacak şekilde düzenlenmiştir. Böylece I integrali aşağıdaki şekilde verilebilir;

$$I = \int_a^b f(x)dx = \frac{b-a}{2}\sum_{i=0}^{n} w_i F(t_i) + R_n$$

F(t) fonksiyonu yukarıda verilen *x=x(t)* dönüşümü ile *f(x)* fonksiyonundan elde edilir. t_i ve w_i değerlerinin irrasyonel sayılar olması hesaplamalarda zorluk çıkarırsa

da, az sayıda nokta ile yüksek doğruluklu sonuçlar sağlar. Özellikle bilgisayar uygulamalarında bu zorluk ortadan kalkmıştır ve integrandın hesaplamasının zor olduğu durumlarda Gauss integrasyonu avantaj sağlar. w_i ve t_i'lerin belirlenmesindeki yöntemlere göre değişen formüller çeşitli kaynaklarda verilmiştir.

Açıklanan yaklaşım ile t_i ve w_i'nin bulunmasında Legendre polinomlarından yararlanıldığı için bu isimle anılmaktadır. Burada integrali

$$\int_{-1}^{1} F(t)dt = \sum_{i=0}^{n} w_i F(t_i) + R_n$$

olarak hesaplanacak biçimde tanımlanıp, eğer *F(t)* fonksiyonu langrange polinomları yardımıyla yaklaştırılırsa;

$$F(t) = \sum_{i=0}^{n} L_i\, F(t_i) + \left[\prod_{i=0}^{n} (t - t_i) \cdot \frac{F^{(n+1)}(\xi)}{(n+1)!} \right] \;;\; L_i(t) = \prod_{\substack{j=0 \\ i \neq j}}^{n} \frac{(t - t_j)}{(t_i - t_j)}$$

dir.

Böylece

$$\int_{-1}^{1} F(t)\, dt \cong \sum_{i=0}^{n} F(t_i) \int_{-1}^{1} L_i(t)\, dt = \sum_{i=0}^{n} w_i F(t_i)$$

olduğundan ; $w_i = \int_{-1}^{1} L_i(t)\, dt$ yazılabilir. Diğer taraftan integrasyon sonucunda hata terimi

$$\int_{-1}^{1} \left[\prod_{i=0}^{n} (t - t_i) . \frac{F^{n+1}(\xi)}{(n+1)!} \right] dt \qquad (2.5.1)$$

olarak verilir. Bu noktada amaç (2.5.1) ile verilen hata terimi sıfır olacak şekilde t_i'lerin belirlenmesidir. Bunun için $\prod_{i=0}^{n} (t - t_i)$ ifadesi ortogonal legendre polinomlarına açılır. Elde edilen denklemlerden t_i kök değerleri w_i ağırlık çarpanı

değerleri bulunur. Bu değerler, çeşitli sayıda nokta kullanan formüller halinde Tablo 2.4' de verilmektedir.

Gauss – Legendre integrasyonu için hata;

$$e_n = \frac{2^{n+3}\left[(n+1)!\right]^4}{(2n+3)\left[(2n+2)!\right]^3} F^{(2n+2)}(\xi) \quad ; \quad \xi \in (-1,1)$$

olarak verilir. Uygulamalarda *x=x(t)* dönüşümü kullanarak *F(t)* fonksiyonu yerine, doğrudan;

$\int_a^b f(x)\,dx \cong \frac{b-a}{2} \sum_{i=0}^{n} w_i\, f(\frac{t_i(b-a)+b+a}{2})$ kullanılabilir.

2.6. Chebyshev Formülleri

Chebyshev formülleri $\int_{-1}^{1} F(t)dt \cong \frac{2}{n}\sum_{i=0}^{n} F(t_i)$ şeklinde verilen integrallerdir ve *n.*dereceden düşük polinomlar için integrali tam olarak verirler. Bütün $F(t_i)$ değerleri hep aynı ağırlık faktörü *2/n* ile çarpılırlar. *n=8* ve *n>10* için Chebyshev formülleri bulunmamaktadır. Burada da [*a,b*] aralığında bir integral *x=x(t)* dönüşümü ile [-1,1] aralığına indirgenebilir;

$x = \frac{b+a+(b-a)t}{2}$ ve $dx = \frac{b-a}{2}dt$ dönüşümü ile Chebyshev formülü için

$$I = \int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f(\frac{b+a+(b-a)t}{2})\,dt \cong \frac{b-a}{n}\sum_{i=1}^{n} F(t_i)$$

yazılabilir. Burada eğrinin derecesi ve nokta sayısı *n* ile gösterilmektedir. Chebyshev formülleri Tablo 2.5'te gösterilmiştir. (Oturanç ve ark., 2008)

Hata terimi $e_n = \frac{2\pi}{2^{2n+2}(2n+2)!} F^{(2n+2)}(\xi) \qquad \xi \in [-1,1]$

	t_i	w_i
İki nokta için n=1	±0.5773502691	1.0000000000
Üç nokta için n=2	0.0000000000 ±0.7745966692	0.8888888889 0.5555555556
Dört nokta için n=3	±0.3399810436 ±0.8611363116	0.6521451549 0.3478548451
Beş nokta için n=4	0.0000000000 ±0.5384693101 ±0.9061798459	0.5688888889 0.4786286705 0.2369268851

Tablo 2.4: Gauss – Legendre formülü için kök ve ağırlık çarpanı değerleri

n	3	5	7	9
T_i	0.00000000 ±0.70710678	0.00000000 ±0.37454141 ±0.83249749	0.00000000 ±0.32391181 ±0.52965678 ±0.88386170	0.00000000 ±0.16790618 ±0.52965678 ±0.60101866 ±0.91158931

Tablo 2.5: Chebyshev formülleri için bazı kök değerleri

3. DİFERANSİYEL, İNTEGRAL VE İNTEGRO-DİFERANSİYEL DENKLEMLERİNİN NÜMERİK ÇÖZÜMLERİ İÇİN LAGRANGE İNTERPOLASYON YÖNTEMİ

Bu bölümde; Rashed, (2004) de belirtilen Fredholm ve Volterra tipindeki doğrusal integral denklemlerin sayısal çözümlerini içeren iki yeni metot üzerinde çalışacağız. Aynı zamanda bu metotlar doğrusal integro-diferansiyel ve diferansiyel denklemlere genişletilecektir. Metotların uygulanabilirliği farklı örnekler üzerinde çalışılarak değerlendirilecektir.

Lagrange interpolasyonu için $y(x)$ fonksiyonu;

$$l_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \left(\frac{x - x_k}{x_j - x_k} \right)$$

olmak üzere

$$y(x) = \sum_{j=0}^{n} l_j(x) y(x_j), \ A \leq x \leq B \tag{3.1.1}$$

ile verilsin. (3.1.1) de A'dan x'e kadar integral alırsak;

$$b_j(x) = \int_A^x \prod_{\substack{k=0 \\ k \neq j}}^{n} \left(\frac{x - x_k}{x_j - x_k} \right) dt \tag{3.1.2}$$

olmak üzere

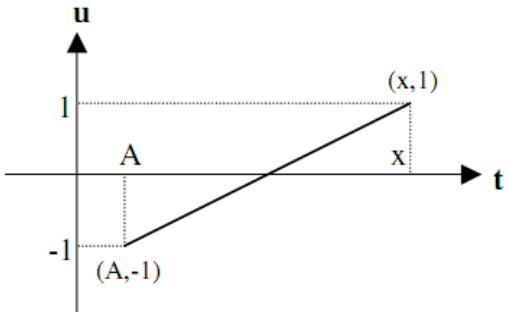
$$\int_A^x y(t)dt \approx \sum_{j=0}^{n} b_j(x)(x_j) \tag{3.1.3}$$

olur. (3.1.2) deki integralini (2.3.3)de verilen

$$\int_{-1}^{1} f(u)du = \sum_{j=0}^{N} b_s f(t_s)$$

eşitliğinden yararlanarak yaklaşık olarak hesap edilebilmesi için aralık sınırlarının [A,x] yerine [-1,1] olarak almamız gerekir.

Bunun için $u = \dfrac{2t - x - A}{x - A}$ ve $du = \dfrac{2}{x - A}$ dönüşümünü yaparak,



Şekil 3.1: Aralık sınırları için lineer dönüşüm

$$b_j(x) = \frac{x - A}{2} \int_{-1}^{1} \prod_{\substack{k=0 \\ k \neq j}}^{n} \left(\frac{\dfrac{x - A}{2} u + \dfrac{x + A}{2} - x_k}{x_j - x_k} \right) du \qquad (3.1.4a)$$

denklemini elde ederiz. (2.3.3) eşitliğinden yararlanarak,

$$d_s = \frac{2\delta_s}{N} \sum_{k}^{N} {}'' I_k \cos\left(\frac{ks\pi}{N} \right)$$

ve

$$I_k = \begin{cases} 2, & k = 0 \\ 0, & k \text{ tek} \\ \dfrac{2}{1 - k^2}, & k \text{ çift} \end{cases}$$

olmak üzere

$$b_j(x) = \frac{x - A}{2} \sum_{s=0}^{N} d_s \prod_{\substack{k=0 \\ k \neq j}}^{n} \left(\frac{\dfrac{(x - A)t_s + (x + A)}{2} - x_k}{x_j - x_k} \right) \qquad (3.1.4b)$$

elde edilir. Ayrıca Lagrange interpolasyon yöntemini kullanarak $\int_A^x \int_A^x y(t)dtdx$ belirsiz integrali

$$C_j(x) = \int_A^x (x-t) l_j(t)dt$$

$$= \frac{x-A}{2}\int_{-1}^{1}\left[x-\left(\frac{x-A}{2}\right)t-\left(\frac{x+A}{2}\right)\right]\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{\frac{x-A}{2}t+\frac{x+A}{2}-x_k}{x_j-x_k}\right)dt$$

$$= \left(\frac{x-A}{2}\right)\sum_{s=0}^{N} d_s\left[x-\left(\frac{x-A}{2}\right)t_s-\left(\frac{x+A}{2}\right)\right]\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{\frac{x-A}{2}t_s+\frac{x+A}{2}-x_k}{x_j-x_k}\right)dt$$

ve

$$t_s = \cos\left(\frac{s\pi}{N}\right), \qquad s = 0(1)N$$

olmak üzere

$$\int_A^x \int_A^x y(t)dtdx = \int_A^x (x-t)y(t)dt \approx \sum_{j=0}^{n} C_j(x)y_j \tag{3.1.5}$$

şeklinde hesaplanabilir.

3.1. Birinci Metot

İkinci tip lineer Volterra integral denklemi;

$$g(x) = y(x) + \lambda\int_A^x k(x,t)y(t)dt, \quad A \le x \le B \tag{3.2.1}$$

yaklaşık olarak;

$$g(x) = \sum_{j=0}^{n}\left\{l_j(x) + \lambda k(x,t_j)b_j(x)\right\}y_j, \quad A \le x \le B \tag{3.2.2}$$

ile verilebilir. Burada $b_j(x)$; (3.1.4b) de belirtilip, ve düğüm noktaları $x_j = \cos\left(j\pi/n\right),\ j = 0(1)n$ veya $x_j = A + ih, h = (B-A)/n,\ j = 0(1)n$ olur.

Lagrange interpolasyonunun, Chebyshev interpolasyonuna göre temel avantajı ilk olarak eşit aralıklarda sayısal çözümleri vermesidir.

$y(x)$ fonksiyonu için Chebyshev interpolasyonu;

$$l_j(x) = \frac{2\delta_s}{n} \sum_{j=0}^{n}{}'' T_k(x) \cos\left(\frac{kj\pi}{n}\right)$$

$$x_j = \cos\left(\frac{j\pi}{n}\right), \qquad j = 0(1)n$$

ve

$$\delta_j = \begin{cases} 0,5, & j = 0(1)n \\ 1, & 0 < j \ll n \end{cases}$$

olmak üzere

$$y(x) \approx \sum_{j=0}^{n} l_j(x) y(x_j), \quad -1 \le x \le 1$$

ile tanımlanır.

İkinci tip Fredholm integral denklemi;

$$g(x) = y(x) + \lambda \int_A^B k(x,t) y(t) dt, \quad A \le x \le B \tag{3.2.3}$$

$$g(x) = \sum_{j=0}^{n} \left\{ l_j(x) + \lambda k(x, t_j) b_j(1) \right\} y_j, \quad A \le x \le B \tag{3.2.4}$$

ile gösterilir.

3.2. İkinci Metot

(3.2.1) deki integral denklemi yaklaşık olarak;

$$g(x)=\sum_{j=0}^{n}\left\{l_j(x)+\lambda B(x,t_j)\right\}y_j \tag{3.3.1}$$

burada

$$B(x,t_j)=\int_A^x k(x,t)l_j(t)dt\approx\sum_{k=o}^{N}k(x,t_k)l_j(t_k)b_k(x) \tag{3.3.2a}$$

veya,

$$B(x,t_j)=\frac{x-A}{2}\int_{-1}^{1}k\left(x,\frac{x-A}{2}t+\frac{x+A}{2}\right)\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{\frac{x-A}{2}t+\frac{x+A}{2}-x_k}{x_j-x_k}\right)dt$$

$$\approx\left(\frac{x-A}{2}\right)\sum_{s=0}^{N}d_s k\left[x,\left(\frac{x-A}{2}\right)t_s-\left(\frac{x+A}{2}\right)\right]\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{\frac{x-A}{2}t_s+\frac{x+A}{2}-x_k}{x_j-x_k}\right) \tag{3.3.2b}$$

ile tanımlanır. Burada;

$$t_s=\cos\left(\frac{s\pi}{N}\right),\qquad s=0(1)N$$

dir.

Ayrıca son iki metot birinci tip lineer Volterra integral denklemini hesaplamak için kullanılır.

$$\int_A^x k(x,t)y(t)dt=g(x),\qquad A\le x\le B \tag{3.3.3}$$

birinci metot ile (3.3.3) denklemini aşağıda olduğu gibi,

$$\sum_{j=0}^{n} k(z_i,t_j)b_j(z_i)y_j = g(z_i), \quad i=0(1)n \tag{3.3.4}$$

hesaplayabiliriz. Burada;

$$z_i = \frac{B-A}{2}\cos\left(\frac{(i-0{,}45)\pi}{n}\right)+\frac{A+B}{2}, \quad i=0(1)n$$

ve

$$y_j = y(t_j), \quad t_j = A+ih, \quad h=\frac{B-A}{n}, \quad j=0(1)n$$

ile tanımlanır.

Diğer taraftan ikinci metodu kullanarak (3.3.3) integral denklemi yaklaşık olarak;

$$\sum_{j=0}^{n} B(z_i,t_j) = g(z_i), \quad i=0(1)n \tag{3.3.5}$$

hesaplanmıştır. Burada, $B(z_i,t_j), i, j = 0(1)n$ (3.3.2b)'de verilmiştir.

3.3. Lineer İntegro-Diferansiyel Denklemler

İkinci dereceden lineer integro-diferansiyel denklemler;

$$\begin{aligned} &w_1(x,y,y',y'')+\lambda\int_A^x k(x,t)w_2(t,y,y',y'')dt = g(x) \\ &y(A)=\alpha, \quad y'(A)=\beta \end{aligned} \tag{3.4.1}$$

olarak alırsak,

$$w_1(x,y(x),y'(x),y''(x)) = A_1(x)y''(x)+A_2(x)y'(x)+A_3(x)y(x)$$

$$w_2(t,y,y',y'') = A_4(t)y''(t)+A_5(t)y'(t)+A_6(t)y(t)$$

ile belirtilir.

$y''(x)$ fonksiyonunun Lagrange açılımı;

$$y''(x) \approx \sum_{j=0}^{n} y_j'' l_j(x), \quad A \le x \le B \tag{3.4.2}$$

$$y'(x) \approx y'(A) + \sum_{j=0}^{n} y_j'' b_j(x), \quad A \le x \le B \tag{3.4.3}$$

ve

$$y(x) \approx y(A) + (x-A)y'(A) + \sum_{j=0}^{n} y_j'' C_j(x), \quad A \le x \le B \tag{3.4.4}$$

olur. (3.4.1) de (3.4.2)- (3.4.4) eşitlikleri yerine konulursa;

$$\sum_{j=0}^{n} B_j(x) y_j'' = G(x) \tag{3.4.5}$$

olduğunda,

$$B_j(x) = A_1(x)l_j(x) + A_2(x)b_j(x) + A_3(x)C_j(x) + \lambda \int_A^x k(x,t)\left\{A_4(t)l_j(t) + A_5(t)b_j(t) + A_6(t)C_j(t)\right\} dt$$

ve

$$G(x) = g(x) - A_2(x)y'(A) - A_3(x)\left[y(A) + (x-A)y'(A)\right] - \lambda \int_A^x k(x,t)y'(A)dt - \lambda \int_A^x k(x,t) + \left[y(A) + (t-A)y'(A)\right]dt$$

yazılabilir. Böylece, (3.4.1) de integro-diferansiyel denklemin çözümünü, $x_j = A + ih, i = 0(1)n, h = (B-A)/n$ düğüm noktalarında bulmak için kullanılır. Ayrıca elde edilen son metotlar birinci tip integro-diferansiyel denklemleri için de genişletilebilir.

$$\int_A^x k(x,t)y''(t)dt = g(x)$$

yaklaşık olarak,

$$\sum_{j=0}^{n} k(z_i, t_j) y''_j b_j(z_i) = g(z_i), \quad i = 0(1)n$$

ile hesaplanabilir.

3.4. Nümerik Örnekler

(i) İkinci tip Volterra integral denklemi,

$$g(x) = y(x) + \lambda \int_A^x k(x,t) y(t) dt, \quad A \le x \le B$$

ile tanımlanmıştı.

1) $k(x,t) = xt, \quad g(x) = e^{-x^2} - \frac{\lambda x}{2}\left(e^{-x^2} - e^{-A^2}\right), \quad y(x) = e^{-x^2}$

2) $k(x,t) = e^{-x^2+t^2}, \quad g(x) = 2x + \lambda e^{-x^2}\left(e^{-x^2} - e^{-A^2}\right), \quad y(x) = 2x$

(ii) İkinci tip Fredholm integral denklemi;

$$g(x) = y(x) + \lambda \int_A^B k(x,t) y(t) dt$$

ile tanımlanmıştı.

3) $K(x,t) = e^{-x-t}, \quad g(x) = e^x + \lambda e^{-x}\left(B - A\right), \quad y(x) = e^x$

Örnek 1,2 ve 3 için, birinci metot ve ikinci metot kullanılarak, hesaplanan σ_n değerleri,

	Örnekler					
n	1		2		3	
	Metot 1	Metot 2	Metot 1	Metot 2	Metot 1	Metot 2
2	2.9E-03	3.2E-03	3.8E-02	5.3E-04	3.5E-16	6.4E-04
3	1.0E-03	4.1E-05	5.3E-03	1.8E-05	5.3E-16	9.2E-05
4	3.8E-05	4.2E-05	8.5E-04	5.1E-07	6.8E-16	1.4E-06

5	1.9E-05	2.3E-06	1.4E-04	1.2E-08	2.9E-16	1.6E-07
6	8.3E-07	6.0E-07	2.2E-05	2.4E-10	2.7E-16	2.3E-09
7	3.5E-07	5.8E-08	3.6E-06	4.3E-12	3.0E-16	2.1E-10
8	2.1E-08	8.1E-09	5.6E-07	6.8E-14	6.8E-16	2.7E-12
9	5.5E-09	1.1E-09	8.7E-08	7.9E-15	2.8E-16	1.9E-13
10	4.8E-10	9.4E-11	1.2E-08	1.5E-14	4.7E-16	1.9E-14

Tablo 3.1

olarak bulunmuştur.

4) $y(x)-\frac{1}{4}\int_0^{\pi/2} xty(t)dt=\sin(x)-\frac{\pi}{2},\quad y(x)=\sin(x)\quad 0\le x\le\frac{\pi}{2}$

(Delves, 1974)

5) $y(x)=x^2+\int_0^1 k(x,t)y(t)dt$

$$k(x,t)=\begin{cases}-t(1-x), & t\le x\\ -x(1-t), & t>x\end{cases}$$

$$y(x)=A_1e^x+B_1e^{-x}-2.0$$

$$A_1=\frac{3e-2}{e^2-1},\qquad B_1=\frac{e(2e-3)}{e^2-1}$$

Örnek 4 ve 5 için, birinci metot, ikinci metot ile trapezyum ve simpson metotları kullanılarak, hesaplanan σ_n değerleri,

n	Örnekler					
	4				5	
	Metot 1	Metot 2	Trapezium	Simpson	Metot 1	Metot 2
2	3.3E-03	3.3E-03	2.6E-02	3.3E-03	1.1E-02	4.6E-05
3	1.3E-03	3.7E-04	9.9E-03		1.7E-04	9.5E-06
4	1.7E-05	1.7E-05	5.2E-03	1.6E-04	3.5E-05	9.1E-08
5	9.4E-06	1.6E-06	3.2E-03		4.4E-07	1.4E-08

6	6.9E-08	6.9E-08	2.2E-03	2.9E-05	7.7E-08	1.1E-10
7	4.2E-08	4.9E-09	1.6E-03		7.7E-08	1.4E-11
8	1.9E-10	1.9E-10	1.2E-03	9.1E-06	1.1E-10	1.1E-13
9	1.3E-10	1.1E-11	9.2E-04		9.3E-13	9.8E-15
10	4.1E-13	4.1E-13	7.4E-04	3.7E-06	1.1E-13	1.7E-15

Tablo 3.2

olarak elde edilmiştir.

(iii) Birinci tip lineer Volterra integral denklemi,

$$\int_0^x k(x,t)y(t)dt = g(x)$$

için,

6) $k(x,t) = \cos(x-t), \quad g(x) = x\sin(x), \quad y(x) = 2\sin(x)$

7) $k(x,t) = \cos(x-t), \quad g(x) = \sin(x), \quad y(x) = 1.0$

(Baker, 1977)

örnekleri ve,

(iv) Birinci tip Fredholm integral denklemi;

8) $\int_0^1 k(x,t)y(t)dt = g(x)$

$$k(x,t) = \begin{cases} t(x-1), & t \le x \\ x(t-1), & t > x \end{cases}, \qquad g(x) = e^x + (1-e)x - 1.0, \qquad y(x) = e^x$$

(Baolian ve ark. 1979)

Örnek 6,7 ve 8 için, birinci metot ve ikinci metot kullanılarak, hesaplanan σ_n değerleri,

n	Örnekler					
	6		7		8	
	Metot 1	Metot 2	Metot 1	Metot 2	Metot 1	Metot 2
2	1.2E-01	5.2E-02	9.7E-04	6.3E-04	1.1E-01	2.2E-02
3	9.6E-03	3.4E-03	1.2E-03	2.1E-04	9.9E-03	1.5E-03
4	1.6E-03	2.8E-04	9.5E-06	5.8E-07	6.2E-04	7.7E-05
5	8.4E-05	6.7E-06	2.4E-06	7.1E-08	3.1E-05	3.2E-06
6	8.9E-06	5.5E-07	1.6E-08	1.2E-10	1.3E-06	1.2E-07
7	3.6E-07	9.3E-09	2.5E-09	2.9E-11	4.5E-08	3.6E-09
8	2.9E-08	5.6E-10	1.6E-12	4.9E-13	1.4E-09	9.9E-11
9	1.0E-09	7.4E-12	1.7E-12	4.6E-13	3.9E-11	8.4E-11
10	6.5E-11	5.4E-13	2.4E-14	8.1E-13	2.9E-12	1.3E-10

Tablo 3.3

olarak elde edilmiştir.

9) $\int_0^1 k(x,t)y(t)dt = g(x)$

$$k(x,t) = \begin{cases} t(x-1), & t \le x \\ x(t-1), & t > x \end{cases}, \qquad g(x) = \frac{x^3 - x}{6}, \qquad y(x) = x$$

(Baolian ve ark. 1979)

(v) İkinci tip integro-diferansiyel Volterra integral denklemi,

10) $y'' - y' + \frac{x^2 - 2}{2} y + \lambda \int_A^x \left[y''(t) + y'(t) - ty(t) \right] dt = g(x), \quad -1 \le x \le 1$

$$y(A) = A, \qquad y'(A) = 1.0$$

$$g(x) = \frac{x(x^2 - 2) - 2}{2} + \lambda(x - A) - \lambda \frac{(x^3 - A^3)}{3.0},$$

$$\lambda = 1.0, \qquad \lambda = 0.0, \qquad y(x) = x$$

11) $y'' - xe^{x}y' + y + \lambda\int_{A}^{x} e^{x+t}y(t)dt = g(x), \quad 0 \le x \le 1$

$y(A) = e^{-A}, \quad y'(A) = -e^{-A}$

$g(x) = 2e^{-x} - x + \lambda(x - A)e^{-x},$

$\lambda = 1.0, \qquad \lambda = 0.0, \qquad y(x) = e^{-x}$

Örnek 9,10 ve 11 için, birinci metot ve ikinci metot kullanılarak, hesaplanan σ_n değerleri

	Örnekler					
n	9		10		11	
	Metot 1	Metot 2	$\lambda = 1.0$	$\lambda = 0.0$	$\lambda = 1.0$	$\lambda = 0.0$
2	9.2E-16	2.8E-15	3.7E-18	0.0E+0.0	3.1E-03	4.4E-04
3	2.1E-15	7.7E-15	1.5E-15	7.1E-17	2.5E-04	1.0E-04
4	7.7E-15	5.9E-15	9.3E-18	0.0E+0.0	8.7E-06	1.0E-06
5	2.5E-14	7.8E-14	3.1E-16	1.1E-16	3.7E-07	3.1E-08
6	8.1E-15	8.8E-14	6.8E-17	2.5E-18	1.8E-08	1.6E-09
7	1.8E-14	1.2E-12	2.3E-16	2.0E-16	6.1E-10	4.8E-11
8	4.1E-14	6.8E-13	4.4E-17	0.0E+0.0	2.5E-11	1.8E-12
9	3.7E-14	2.4E-11	6.9E-16	2.3E-16	7.2E-13	5.2E-14
10	1.1E-13	6.8E-13	9.3E-18	5.8E-20	2.5E-14	1.6E-15

Tablo 3.4

olarak elde edilmiştir.

12) $y''(x) - (x^2 + x + 1)y'(x) + (x + 2)y(x) = g(x), \quad 0 \le x \le 1$

$y(0)=1, \qquad y'(0)=1.0$

$g(x)=(x^2+2x+4)e^x, \qquad y(x)=e^x$

ve

13) $\int_0^x \cos(x-t)y''(t)dt = 2\sin(x), \quad 0 \le x \le 1$

$y(0)=0, \qquad y'(0)=0.0$

$y(x)=x^2$

örnekleri için birinci metot ve ikinci metot kullanılarak, hesaplanan σ_n değerleri

	Örnekler		
n	12	13	
		Metot 1	Metot 2
2	5.2E-04	3.6E-04	8.4E-05
3	3.4E-04	1.6E-04	1.7E-05
4	1.4E-06	6.0E-07	2.4E-08
5	1.7E-07	2.2E-07	2.6E-09
6	2.4E-09	1.1E-09	2.9E-12
7	1.3 E-10	2.6E-10	6.7E-13
8	2.8E-12	1.3E-12	7.8E-16
9	9.8E-14	2.3E-13	4.4E-16
10	2.4E-15	9.9E-16	5.3E-16

Tablo 3.5

olarak elde edilmiştir.

4. DEĞERLENDİRMELER VE SONUÇLAR

(1) Geliştirilen metotlar $x_j = \cos\left(j\pi / n \right)$, $j = 0(1)n$ veya $x_j = A + ih$, $j = 0(10)n$, $h = (B-A)/n$ düğüm noktalarında çözümler verir.

(2) 6., 7. ve 8. örneklerde; geliştirilen metotların, birinci tip Volterra integral denklemlerini ve birinci tip integro-diferansiyel denklemlerini çözdüğünü gördük.

(3) 3-5, 8. ve 9. örneklerde, geliştirilen metotların, ikinci tip ve birinci tip Fredholm integral denklemlerine uygulanabildiğini gördük.

(4) FORTRAN, BASIC, C ve C++, MATLAB VE MAPLE da yazılmış olan formül, kişisel bilgisayarda kullanabilme olanağı sağladığından geliştirilen metotların ayrı bir avantajı olarak değerlendirilebilir.

(5) Çok hızlı yakınsamanın elde edilmesi gösterir ki; aslında geliştirilen metotlar başarılı bir şekilde Volterra integral denklemleri ile Fredholm integral denklemleri çözer.

(6) Örnek 5; singüler değerleriyle birlikte Green fonksiyonunun çekirdeğine sahiptir.

(7) Hesaplanan hatalar, σ_n;

$$\sigma_n = \left[\sum_{j=0}^{n} e_n^2(x_j)/n \right]^{1/2} \approx \left[\int_{-1}^{1} e_n^2(t)dt \right]^{1/2}$$

ile tanımlanır.

(8) Ayrıca bu metotlar, yan koşulları ile verilen integro-diferansiyel denklemlerin çözümlerini bulmak için de kullanılabilir.

KAYNAKLAR

Hildebrand F.B., *Introduction to Numerical Analysis*, (1956).

Davis P.J. , Rabinowitz P., *Numerical İntegration*, (1965).

Rashed M.T. , *Lagrange interpolation to compute the numerical solutions of differential, integral and integro-differential equations,* Applied Mathematics and Computation, 151 (2004) 869–878.

Oturanç.G, Keskin Y., Kurnaz A., Kiriş M., *Sayısal Analiz*, Konya, (2008)

Delves L.M., Walsh J., *Numerical Solution of Integral Equations*, Oxford University Press,London, New York, (1974)

Rashed M.T., *Numerical solution of a special type of integro-differential equations,* Applied Mathematics and Computation 143 (2003) 73–88

Baker C.T.H., The *Numerical Treatment of Integral Equations, Clarendon press*, Oxford, (1977)

Rashed M.T., *Numerical solutions of functional integral equations,* Applied Mathematics and Computation 156 (2004) 507–512

Baolian E., Delves L.M., *An augmented Galerkin matrix for the first integral equations*, J. Inst.Math. Appl. (1979) 157–174.

Delves L.M., Mohammed J.L., *Computational methods for integral equations*, Cambridge University Press, Great Britain, (1992)

Rashed M.T., *An expansion method to treat integral equations,* Applied Mathematics and Computation 135 (2003) 65–72

Davis, P.J., Rabinowitz,P., "Numerical Integration", (1965).

Rashed M.T., *Numerical solution of functional differential, integral and integro-differential equations,* Applied Mathematics and Computation 156 (2004) 485–492

Demirbaş M., *İntegral denklemlerinin yaklaşık çözümleri,* Isparta, (2002)

Sharma J.N., *Numerical methods for engineers and scientists 2nd Edition*, India, (2007)

Kartal Ş.Ş., *Bir fonksiyonun türevinin lagrange interpolasyonu ile hesaplanması,* İstanbul, (2008)

Press W.H., Tevkolsky S.A., Vetterling W.T., Flennery B.P., *N. Recipes in Fortran 2nd Edition,* Cambridge University Press, (1992)

Mathews J.H., Fink K.D., *Numerical Methods using Matlab 4th Edition*, (2004)

Porter D. and Stirling D.S.G., *Integral Equations*, Cambridge University Press, (1990)

Kopal, Z., *Numerical Analysis*, Wiley, New York, (1955)

Bocher M., *An Introduction to the Study of Integral Equations*, (1913)

http://numericalmethods.eng.usf.edu/mtl/gen/05inp/

http://en.wikipedia.org

http://www.gidb.itu.edu.tr/staff/yildiz/