17428

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

DOĞRUSAL DENKLEM SİSTEMLERİNİN ÇÖZÜMÜ İÇİN ETKİN BİR ALGORİTMA

Sezgin AKBULUT

Yönetici: Doç. Dr. Şeref MİRASYEDİOĞLU

Yüksek Lisans Tezi

ÖZET

Bu çalışmanın amacı, $x_1, x_2, \ldots, x_n$ bilinmeyenler olmak üzere n bilinmeyenli m tane doğrusal denklemden oluşan,

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

doğrusal denklem sisteminin çözümü için etkin bir algoritma geliştirmek ve C dilinde programını yazmaktır.

SUMMARY

The aim of the research is to construct an efficient algorithm for solving the system of m simultaneous linear equations in the n unknowns $x_1, x_2, \ldots, x_n$:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

and to give its program in C.

TEŞEKKÜR

Bu çalışma süresince bana bilimsel destek sağlayan ve yardımlarını esirgemeyen, tez yöneticim sayın hocam Doç.Dr. Şeref MİRASYEDİOĞLU'na ve tezin yazımı için gerekli olan VENTURA PUBLUSHING paket programının kullanımında yardımlarını esirgemeyen Arş. Gör. Bünyamin YILDIZ'a teşekkürlerimi sunarım.

İÇİNDEKİLER

1. GİRİŞ

Bilgisayar teknolojisinin gelişmesi ve buna paralel olarak kullanım alanlarının yoğun olarak artması matematiğin önem ve katkısını arttırmaktadır. Bu alan, doğal olarak matematiğin kendi içindeki gelişmeleri etkilemekte ve yeni araştırma alanlarının ortaya çıkmasına neden olmaktadır. 1950'li yılardan sonra bilgisayar teknolojisindeki hızlı gelişmeye paralel olarak matematikte de sayısal hesaplama teknikleri üzerinde çeşitli yöntemler geliştirilir.Özellikle Nümerik Analiz konusunda sayısal çözümler için algoritmalar önerilir (Carnahan, et al., 1969).

1980'li yılların başından itibaren sembolik işlemler yapabilen C, PASCAL, PROLOG, LİSP gibi fonksiyonel anlamda derleyicilerin geliştirilmesi, bilgisayar desteğinde hesaplama tekniklerine yeni bir boyut kazandırır. Bu doğrultuda, matematikte de fonksiyonel anlamda hesaplama teknikleri için yeni çalışmalar başlar.

Bilgisayar desteği ile çözümlemenin temel amacı, matematiksel modellerle ifade edilmiş problemlerin çözümünde belli sayıda ve sırası belirlenmiş işlemleri bilgisayar desteği ile yaparak, belirli bir hassaslığa sahip sonuçları elde edebilmek için kullanılabilecek yöntemlerin bulunması, geliştirilmesi, var olanların irdelenmesi ve en etkin olanlarının saptanması olduğu söylenebilir.

Doğrusal denklem sistemlerinin bilgisayar desteği ile çözümleri üzerine (Abramchuk, 1989; Aktaş, 1981; Stevvart, 1973; Valitskij, 1989; Yunusoğlu, 1974; Ikramov, 1989) gibi çalışmalar yapılmıştır. Bu çalışmalarda, doğrusal denklem sistemlerinin bilgisayar desteğinde çözümleri için, Gauss-Yoketme, Gauss-Jordan, Gauss-Seidel, Cholesky yöntemlerinin algoritma ve programları ayrıntılı biçimde verilmiştir.

Bu araştırmanın amacı, n bilinmeyenli m tane doğrusal denklemden oluşan doğrusal denklem sisteminin

i. $m > n$
ii. $m = n$
iii. $m < n$

koşulları altında, fonksiyonel anlamda çözümünü yapacak etkin bir algoritma geliştirmek ve C dilinde programını yazmaktır.

i. Doğrusal denklem sistemlerinin matematiksel çözümü araştırıldı.
ii. Bu konuda yapılan çalışmalar ve algoritmalar irdelendi.
iii. Mevcut algoritmalardan yararlanılarak etkin bir algoritama geliştirildi.

Geliştirilen program, verilen algoritmanın C programlama dilinde yazılması ile gerçekleştirilmiştir. Adı geçen program Atatürk Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümündeki MS-DOS işletim dizgesi altında çalışan mikrobilgisayarında geliştirilmiştir. Kurulan algoritmanın çevirimi C derleyicisi aracılığında gerçekleştirildiğinden, programın C derleyicisi olan herhangi bir bilgisayar dizgesinde kullanılması mümkündür.

Bu çalışma beş bölümden oluşmaktadır. Birinci bölüm giriştir. İkinci bölüm doğrusal denklem sistemlerinin matematiksel çözümünü kapsar. Üçüncü bölümde doğrusal denklem sistemlerinin çözümünde kullanılan algoritmalar, dördüncü bölümde geliştirilen algoritma, akış diyagramı ve C programı yazılmıştır. Beşinci bölümde ise sonuç verilmiştir.

2.ÖN BİLGİLER

2.1. Doğrusal Denklem

2.1.1. Tanım : $a_1,a_2,...,a_n,b$ R de herhangi sabit elamanlar ve $x_1,x_2,...,x_n$ de bilinmeyenler olmak üzere

$$\sum_{i=1}^{n} a_i x_i = b \qquad (1)$$

ifadesine, R üzerinde bir doğrusal denklem denir. Burada a_1, a_2, ..., a_n sayılarına bu doğrusal denklemin katsayıları, b ye de sabit terim denir. Eğer,

$$x_1 = c_1, x_2 = c_2, ..., x_n = c_n$$

değerleri (1) denklemini sağlıyor ise yani,

$$\sum_{i=1}^{n} a_i c_i = b$$

ise $c_1, c_2, ..., c_n$ ye (1) denkleminin bir çözümü denir.

2.2. Doğrusal Denklem Sistemleri

2.1.1. Tanım : a_{ij} , b $\in$ R olmak üzere, n tane bilinmeyen ve m tane doğrusal denklemden oluşan ,

$$\begin{array}{c} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \end{array} \qquad (2)$$

sistemine doğrusal denklem sistemi denir. Özel olarak, $b_1 = b_2 = ... = b_m = 0$ ise (2) doğrusal denklem sistemine homojen doğrusal denklem sistemi denir.

(2) sistemi kısaca,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad i = 1,2,...,m$$

şeklinde yazılır.

2.3. Doğrusal Denklem Sistemlerinin Çözümü

2.3.1. Tanım : $c_j \in R$ olmak üzere,

$$x_1 = c_1,\ x_2 = c_2,\ \dots,\ x_n = c_n$$

değerleri (2) sistemindeki denklemlerin hepsini sağlıyor ise yani,

$$\sum_{j=1}^{n} a_{ij} c_j = b_i \qquad i = 1,2,\ \dots,\ m$$

ise $(c_1, c_2, \dots, c_n)$ n-lisine (2) sisteminin bir çözümü denir.Bu çözümlerin tümünü içeren kümeye çözüm kümesi denir.

2.3.2. İndirgeme İşlemi

(2) doğrusal denklem sistemi eşdeğer, yani bu sistemle aynı çözüme sahip olan daha basit bir sisteme indirgenebilir. Bunun için aşağıdaki işlemlerin yapılması gerekir.

i. Verilen denklemler, birinci denklemdeki x_1 bilinmeyeninin katsayısı $a_{11} \neq 0$ olacak şekilde sıralanır.

ii. L_i, i'nci denklem olmak üzere her $i > 1$ için L_i denklemi yerine $a_{11}L_i - a_{i1}L_1$ denklemi alınır.

iii. (2) denklem sistemi kendisine eşedeğer olan

$$\begin{aligned} a'_{11}x_1 + a'_{12}x_2 + \dots + a'_{1n}x_n &= b'_1 \\ a'_{2j_2}x_{j_2} + \dots + a'_{2n}x_n &= b'_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ a'_{mj_2}x_{j_2} + \dots + a'_{mn}x_n &= b'_m \end{aligned} \qquad (3)$$

sistemine indirgenir. Burada $a'_{11} \neq 0$, $x_{j_2} \neq x_1$, $1 < j_2 < \dots < j_r$ ve x_{j_2}, ilk denklemden başka denklemler arasında sıfırdan farklı katsayıya sahip olan en küçük indisli bilinmeyendir.

iv. $a_{11} \neq 0, a_{22} \neq 0, \dots, a_{rj_r} \neq 0$ için indirgeme işlemi ardarda yapılarak (2) sistemi ya anlamsız ya da eşdeğeri olan

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{2j_2}x_{j_2} + \dots + a_{2n}x_n &= b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ a_{rj_r}x_{j_r} + \dots + a_{rn}x_n &= b_r \end{aligned} \qquad (4)$$

sistemine indirgenir.

v. İndirgeme işlemleri sırasında

$$0x_1 + 0x_2 + \dots + 0x_n = b \qquad b \neq 0$$

gibi bir denklem ortaya çıkarsa, bu denklem anlamsızdır ve sistemin çözümü yoktur.

vi. İndirgeme işlemleri sırasında

$$0x_1 + 0x_2 + ... + 0x_n = 0$$

şeklinde bir denklem ortaya çıkarsa, bu denklem sistemden çıkarılabilir. Çözüme etkisi olmaz. Yani, elde edilen sonuç

$$0x_1 + 0x_2 + ... + 0x_n = 0$$

denklemini daima sağlar.

2.3.3. Teorem : Basamak formundaki (4) sisteminin çözümünde aşağıdaki gibi iki hal vardır (Stoll, 1952).

i. $r = n$ ise sistemin tek çözümü vardır.

ii. $r < n$ ise sistemin çözümü, n-r tane bağımsız bilinmeyene keyfi değerler verilerek bulunur.

2.4. Homojen Doğrusal Denklem Sistemlerinin Çözümü

n bilinmeyenli m denklemden oluşan homojen doğrusal denklem sistemi genel olarak

$$\sum_{j=1}^{n} a_{ij}x_j = 0 \qquad i = 1,2,...,m$$

şeklinde yazılabilir. Homojen doğrusal denklem sisteminin bir çözümü her zaman vardır. Çünkü, (0,0,...,0) çözümü böyle bir denklemi daima sağlar ve

$$x_1 = x_2 = ... = x_n = 0$$

çözümüne bu sistemin aşikar çözümü denir. Homojen doğrusal denklem sisteminin aşikar çözümünden başka çözümlerinin bulunup bulunmadığı herhangi bir doğrusal denklem sistemi için izlenen yoketme yöntemi ile bulunabilir. Basamak formuna indirgenmiş homojen doğrusal denklem sistemleri için aşağıdaki teoremi verebiliriz.

2.4.1. Teorem : Basamak formuna indirgenmiş homojen doğrusal denklem sisteminin çözümünde aşağıdaki iki hal vardır (Stoll, 1952).

i. $r = n$ ise sistemin tek çözümü aşikar çözümdür.

ii. $r < n$ ise daima aşikar olmayan bir çözüm vardır.

2.5. Doğrusal Denklem Sistemleri ve Matrisler

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad i = 1,2,..., m \tag{5}$$

doğrusal denklem sistemi,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ b_m \end{bmatrix} \tag{6}$$

matris denklemine eşdeğerdir. Bu denklem kısaca $AX=B$ şeklinde gösterilir. (5) sisteminin her çözümü (6) matris denkleminin de bir çözümüdür. Homojen doğrusal denklem sistemi matris formunda, $AX=0$ şeklinde yazılır.

2.5.1. Tanım :

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

matrisine, (5) doğrusal denklem sisteminin katsayılar matrisi denir.

2.5.2. Tanım :

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

matrisine, (5) doğrusal denklem sisteminin genişletilmiş katsayılar matrisi denir.

2.5.3. Tanım : V, K cismi üzerinde bir vektör uzayı olsun.

$$a_1v_1 + a_2v_2 + \dots + a_nv_n = 0 \tag{7}$$

olacak şekilde hepsi birden sıfır olmayan $a_1, a_2, \dots, a_n \in K$ skalerleri bulanabiliyor ise $v_1, v_2, \dots, v_n \in V$ vektörlerine K cismi üzerinde doğrusal bağımlıdır denir. Eğer (7) ifadesinin sağlanması $a_1, a_2, \dots, a_n \in K$ skalerlerinin sıfır olmasını gerektiriyorsa $v_1, v_2, \dots, v_n \in V$ vektörlerine doğrusal bağımsızdır denir (Stoll, 1952).

2.5.4. Tanım : Verilen bir A matrisinin doğrusal bağımsız satır (ya da sütun) vektörlerinin maksimum sayısına A matrisinin rankı denir ve $r(A)$ ile gösterilir.

2.5.5. Teorem : n bilinmeyenli m tane doğrusal denklemden oluşan doğrusal denklem sisteminin katsayılar matrisi A ve genişletilmiş katsayılar matrisi G olsun. Bu durumda

i. $r(A) = r(G)$ ise denklem sisteminin tek çözümü vardır.
ii. $r(A) \neq r(G)$ ise denklem sisteminin çözümü yoktur.
iii. $r(A) = r(G)$ iken $r(A) < n$ ve $r(G) < n$ ise denklem sisteminin parametreye bağlı çözümü vardır (Stoll, 1952).

2.5.6. Örnek

$$x+y-z=1$$
$$x+2y-2z=0$$
$$-2x+y+z=4$$

denklem sisteminin çözümü aşağıdaki gibi verilebilir
Çözüm: Verilen denklem sisteminin katsayılar matrisi,

$$A=\begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{bmatrix}$$

ve genişletilmiş katsayılar matrisi

$$G=\begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 2 & -2 & 0 \\ -2 & 1 & 1 & 4 \end{bmatrix}$$

olur. G matrisi elementer işlemlerle,

$$G \sim \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

formuna indirgenir. Buradan $r(A)=3$ ve $r(G)=3$ olduğu görülür. $r(A)=r(G)$ olduğundan denklem sisteminin tek çözümü vardır. Bu çözüm,

$$x=2,\ y=2,\ z=3$$

olur.

2.5.7. Örnek

$$x+2y+3z=4$$
$$x-y+2z=1$$

denklem sisteminin çözümü aşağıdaki gibi verilebilir.
Çözüm: Verilen deklem sisteminin katsayılar matrisi,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & 2 \end{bmatrix}$$

ve genişletilmiş katsayılar matrisi

$$G = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & -1 & 2 & 1 \end{bmatrix}$$

olur. G matrisi elementer satır işlemleri ile

$$G \sim \begin{bmatrix} 1 & 0 & 2.4 & 2 \\ 0 & 1 & 0.3 & 1 \end{bmatrix}$$

formuna indirgenir. Buradan $r(A) = r(G) = 2$ olduğu görülür. Fakat $r(A) < 3$ ve $r(G) < 3$ olduğundan denklem sisteminin sonsuz çözümü vardır. k herhangi bir parametre olmak üzere bu çözüm,

$$x = 2-2.4k$$
$$y = 1-0.3k$$

şeklindedir.

2.5.8. Örnek

$$x + 2y + 3z = 4$$
$$-x + 2y = 0$$
$$x + 2y + 3z = 1$$

denklem sisteminin çözümü aşağıdaki gibi verilebilir.

Çözüm : Verilen denklem sisteminin genişletilmiş katsayılar matrisi,

$$G = \begin{bmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & 0 & 0 \\ 1 & 2 & 3 & 1 \end{bmatrix}$$

şeklindedir. G matrisi elementer satır işlemleri ile

$$G \sim \begin{bmatrix} 1 & 0 & 1.4 & 2 \\ 0 & 1 & 0.8 & 1 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

formuna indirgenir. Buradan $r(A)=2$ ve $r(G)=3$ olduğu görülür $r(A)\neq r(G)$ olduğundan denklem sistemin çözümü yoktur.

3. DOĞRUSAL DENKLEM SİSTEMLERİNİN ÇÖZÜMÜNDE KULLANILAN ALGORİTAMALAR

Bu bölümde, doğrusal denklem sistemlerinin bilgisayar desteğinde çözümlerinin yapılabilmesi için bilinen,

i. Gauss-Yoketme
ii. Kofaktör
iii.Gauss-Jordan
iv. Doğrusal Yaklaşım
v. Gauss-Seidel
vi. Çarpan
vii.Cholesky

yöntemlerinin algoritmaları ve kullanılabilirliği sade bir biçimde verilecektir. Ancak, Doğrusal Yaklaşım yöntemi ile Gauss-Seidel yöntemi, Çarpan yöntemi ile Cholesky yöntemleri arasında çok az bir fark vardır. Bu nedenle, sadece Gauss-Seidel ve Cholesky yöntemlerinin algoritmaları verilecektir.

3.1. Algoritma (Gauss-Yoketme yöntemi)

B0: Başla;
BA: n: Bilinmeyen sayısı ve denklem sayısı,
G: Genişletilmiş katsayılar matrisi ve $a_{ij} \in G$,
B1: n ve G yi oku;
B2: G matrisinin esas köşegeni üzerindeki elemanı sıfır ise,
$i = k + 1 \qquad k = 1,2,\ldots,n;$
$a_{kj} = a_{kj} + a_{ij};$
işlemlerini yap;
B3: G matrisinin esas köşgeni altındaki elemanlarının sıfırlanması için,
$a_{kk} = 1;$
$a_{kj} = a_{kj}/a_{kk};$
$b_{kj} = b_{kj}/a_{kk};$
$a_{ij} = a_{ij}-a_{ik}.a_{kj} \qquad j = k,\ldots, n;$
$b_i = b_i-a_{ik}.a_{kj} \qquad i = k + 1,\ldots,n;$
işlemlerini yap;
B4: $k = k + 1;$
B5: $k \leq m$ ise B2'ye git;

B6: $x_n = b_n/a_{nn}$ işlemini yap;
B7: Yerine koyma işlemi ile $x_{n-1},..., x_2, x_1$ i hesapla;
B8: Sonucu yaz;
B9: Dur (Stevvart, 1973; Aktaş, 1981; Moore, 1972).

Gauss-Yoketme yönteminde, verilen doğrusal denklem sisteminin katsayılar matrisi indirgeme işlemi ile (kesim 2.3.2.) bir üçgen matrise dönüştürülür. Daha sonra yerine koyma yöntemi ile denklem sisteminin çözümü bulunur. Sonuçların doğruluğunu etkileyen en önemli faktörlerden biri işlem sayısıdır. İşlem sayısının artması, hata miktarının artmasına yol açar. Bu yöntem, programlama kolaylığı, bellekte tuttuğu yer, işlem sayısı ve işlem süresi açısından oldukça kullanışlıdır.

3.2 Algoritma (Kofaktör Yöntemi)

B0: Başla;
BA: n: Bilinmeyen ve denklem sayısı,
A: Katsayılar matrisi ve $a_{ij} \in A$,
A^{-1}: Katsayılar matrisinin tersi,
B: Kaynak vektörü,
B1: n, A ve B'yi oku;
B2: A matrisinin esas köşegeni üzerindeki eleman sıfır ise,
$i = k + 1$ $\quad k = 1,2, ..., n;$
$a_{kj} = a_{kj} + a_{ij};$
işlemlerini yap;
B3: Esas köşegen üzerindeki eleman için,
$a_{kk} = 1/a_{kk};$
işlemini yap;
B4: Kullanılan köşegen elemanının bulunduğu satır ve sütun da olmayan diğer bütün elemanlar için,

$$a_{ij} = a_{ij} - \frac{a_{ik} \cdot a_{kj}}{a_{kk}} \qquad i \text{ ve } j \neq k$$

işlemini yap;
B5: k'ncı satır ve sütunda ki elemanlar (kullanılan köşegen elemanı hariç) için,
$a_{kj} = -a_{kj}/a_{kk}$ $\quad j \neq k$ satır elemanları;
$aik = a_{ik}/a_{kk}$ $\quad i \neq k$ sütun elemanları;
işlemlerini yap;

B6: $k = k + 1$;

B7: $k \leq n$ ise B2'ye git;

B8: A^{-1} matrisini yaz;

B9: $A^{-1}.B$ işlemini yap;

B10: Sonucu yaz;

B11: Dur (Yunusoğlu, 1974; Al-Towaiq, 1989; Hearn, 1987).

Kofaktör yönteminde, verilen katsayılar matrisinin tersi bulunur. Daha sonra ters matris ile kaynak vektörü çarpılarak çözüm elde edilir. Bu yöntemde işlem sayısı ve zamanı Gauss-Yoketme yönteminden daha fazladır. İşlem sayısının artması programlama güçlüğüne sebep olmaktadır. Bu nedenle, derecesi büyük olan doğrusal denklem sistemlerinin çözümünde Gauss-Yoketme yöntemi daha kullanışlıdır.

3.3. Algoritma (Gauss-Jordan Yöntemi)

B0: Başla;

BA: n: Bilinmeyen ve denklem sayısı,

A: Katsayılar matrisi ve $a_{ij} \in A$,

I: Birim matris ve $h_{kj} \in I$,

B: Kaynak vektörü,

A^{-1}: A matrisinin tersi,

T: A matrisine birim matrisi ekleyerek oluşturulan genişletilmiş matris,

B1: A, I, B ve n'yi oku;

B2: T matrisinin esas köşegeni üzerindeki eleman sıfır ise,

$i = k + 1 \quad k = 1,2, ..., n$;

$a_{kj} = a_{kj} + a_{ij}$;

işlemlerini yap;

B3: T matrisi içerisinde, A matrisini I matrisine ve I matrisini de A^{-1} matrisine dönüştürmek için,

$a_{kk} = 1$;

$a_{kj} = a_{kj}/a_{kk} \quad j = k + 1, ..., n$;

$h_{kj} = h_{kj}/a_{kk} \quad j = 1, ..., k$;

$a_{ij} = a_{ij} - a_{ik}.a_{kj} \quad j = k, ..., n \quad i = 1,2, ..., n$;

$h_{ij} = h_{ij} - a_{ik}.h_{kj} \quad j = 1, ..., k \quad i \neq k$;

işlemlerini yap;

B4: $k = k + 1$;

B5: $k \leq n$ ise B2'ye git;

B6: $A^{-1}.B$ işlemini yap;

B7: Çözümü yaz;

B8: Dur (Stevvart, 1973; Abramchuk, 1989).

Gauss-Jordan yönteminde, doğrusal denklem sistemi katsayılar matrisinin tersi bulunarak çözülür. Ters matrisi bulabilmek için, katsayılar matrisine birim matris ilave edilerek genişletilir. Bu da bilgisayar belleğinde fazla yer kaplamasına sebep olur. İşlem sayısı ve işlem süresi bakımından Gauss-Yoketme yönteminden daha uzundur. Fakat diğer yöntemlerden daha kullanışlıdır.

3.4. Algoritma (Gauss-Seidel Yöntemi)

B0: Başla;

BA: n: Bilinmeyen ve denklem sayısı,

A: Katsayılar matrisi ve $a_{ij} \in A$,

B: Kaynak vektörü,

D: Dönüşüm sonucu oluşan katsayılar matrisi ve $d_{ij} \in D$,

C: Dönüşüm sonucu oluşan kaynak vektörü,

M: Enbüyük yaklaşım işlem sayısı,

B1: n, A ve B'yi oku;

B2: $a_{ii} = 0$ ise,

$s = i + 1 \qquad i = 1, \ldots, n;$

$a_{ij} = a_{ij} + a_{sj};$

işlemlerini yap;

B3: A ve B den hareket ederek C ve D'yi oluştur;

B4: X = C-DX' i yaz;

B5: D matrisinde, tek tek sütun elemanlarının mutlak değerlerinin toplamlarını hesapla, yöntemin uygulanıp uygulanamayacağını (yakınsaklık koşulunu) saptamak icin,

$$T_j = \sum_{i=1}^{n} |d_{ij}| < 1 \qquad j = 1,2,\ldots,n$$

işlemini yap;

B6: Eğer yöntem, verilen denklem sistemine uygulanabiliyor ise

$$x^{(0)} = |0.0|$$

yaklaşım işlemine gir;

B7: Yöntem uygulanamıyor ise B10' a git;

B8:

$$x_i^{(k)} = c_i - \left(\sum_{j=1}^{i-1} d_{ij} x_j^{(k)} + \sum_{j=1}^{n} d_{ij} x_j^{(k-1)}\right)$$

($i = 1,2,\ldots, n$ ve $k = 1,2,\ldots, M$) ile $x_i^{(k)}$'yı hesapla;

B9: Sonucu yaz;

B10: Dur (Stevvart, 1973; Koboyashi et al., 1988).

Gauss-Seidel yöntemi bir yaklaşım yöntemidir. Bu yöntem genelilikle, katsayılar matrisinde sıfırı çok olan çok büyük denklem sistemlerinin çözümü için kullanılır. Sıfırdan farklı eleman sayısı az olduğundan her yaklaşım işleminde yapılan aritmetik işlem sayısıda az olacaktır. Ayrıca yanlışların büyümesi de söz konusu değildir. Ancak yaklaşım yöntemi her zaman yakınsak olmayabilir. Bu sebeple her denklem sisteminin çözümü için bu yöntem uygulanamaz. Fakat katsayılar matrisinde sıfırı çok olan çok büyük denklem sistemlerinin çözümünde Gauss-Seidel yöntemi diğer yöntemlerden daha kullanışlıdır.

3.5. Algoritma (Cholesky Yöntemi)

B0: Başla;

BA: n: Bilinmeyen ve denklem sayısı,

A: Katsayılar matrisi ve $a_{ij} \in A$,

C: Kaynak vektörü,

Z: Alt üçgensel denklem sisteminde bilinmeyenler,

S: Alt üçgensel denklem sisteminin katsayılar matrisi,

B1: n, A ve C' yi oku;

B2: Alt üçgensel matrisi oluşturmak için,

$$s_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} s_{ik} \cdot s_{jk}\right) / s_{jj} \quad j = 1,2,\ldots, i-1;$$

$$s_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2\right)^{1/2} \qquad i = 2,\ldots, n;$$

$$s_{11} = (a_{11})^{1/2}$$

işlemlerini yap;

B3: $z_1 = c_1 / s_{11}$;

$$z_i = \frac{c_i - \sum_{j=1}^{i-1} s_{ij}.z_j}{s_{ii}} \quad ;$$

işlemlerini yap;

B4: Çözüm vektörünü bulmak için,

$x_n = z_n/s_{nn}$;

$$x_j = \frac{z_j - \sum_{i=j+1}^{n} s_{ij}.x_i}{s_{jj}} \qquad j = n-1,...,2,1;$$

işlemlerini yap;

B5: Sonucu yaz;

B6: Dur (Stevvart, 1973; Valitskij, 1989).

Cholesky yöntemi, katsayılar matrisinin simetrik olduğu durumlarda kullanışlıdır. Bu yöntem, programlama güçlüğü ve işlem süresi bakımından Gauss-Yoketme yönteminden daha uzundur. Fakat katsayılar matrisi simetrik olan denklem sistemlerinin çözümünde, diğer yöntemlerden daha kısa sürede sonuç verir. Ayrıca bellek yeri ve işlem zamanından kazanç sağlar. Cholesky yöntemi bu tip denklem sistemlerinin çözümünde diğer yöntemlerden daha kullanışlıdır.

4. DOĞRUSAL DENKLEM SİSTEMLERİNİN ÇÖZÜMÜ İÇİN GELİŞTİRİLEN ALGORİTMA VE PROGRAM

Bu bölümde, daha önce verilen Gauss-Yoketme, Kofaktör, Gauss-Jordan, Gauss-Seidel, Cholesky yöntemlerinden ve ikinci bölümdeki teoremlerden yararlanarak n bilinmeyenli m tane doğrusal denklemden oluşan doğrusal denklem sistemlerinin bilgisayar desteğinde çözümünü yapabilmek için geliştirilen algoritma, akış diyagramı ve C dilinde programı verilecektir.

4.1. Algoritma

B0: Başla;

BA: m: Denklem sayısı,

n: Bilinmeyen sayısı,

G: Genişletilmiş katsayılar matrisi ve $a_{ij} \in G$,

h: Devirli sayılardan oluşan hata miktarı,

B1: m, n, G'yi oku;

B2: $a_{kk} = 0$ ise

$i = k + 1$;

$a_{kj} = a_{kj} + a_{ik}$;

B3: $n \geq m$ ise B5'e git;

B4: $n < m$ ise B10'a git;

B5: G matrisinin esas köşegeni üzerinde olmayan elamanların sıfırlanması için (indirgeme işlemi 2.3.2.)

$a_{kj} = a_{kj}/a_{kk}$;

$a_{kj} = a_{kj} - a_{kj}.a_{ik}$ $\quad i = 1,2,\ldots, m$ ve $i \neq k$;

$| a_{kj} | \leq h$ ise $a_{kj} = 0$;

işlemlerini yap;

B6: $k = k + 1$;

B7: $k \leq m$ ise B2'ye git;

B8: Çözümü veya parametrik çözümü yaz;

B9: B14'e git;

B10: G matrisinin esas köşegeni üzerinde olmayan elamanların sıfırlanması için ,

$a_{kj} = a_{kj}/a_{kk}$;

$a_{kj} = a_{kj} - a_{kj}.a_{ik}$ $\quad i = 1,2,\ldots, m$ ve $i \neq k$;

$| a_{kj} | \leq h$ ise $a_{kj} = 0$;

işlemlerini yap;

B11: $k = k + 1$;

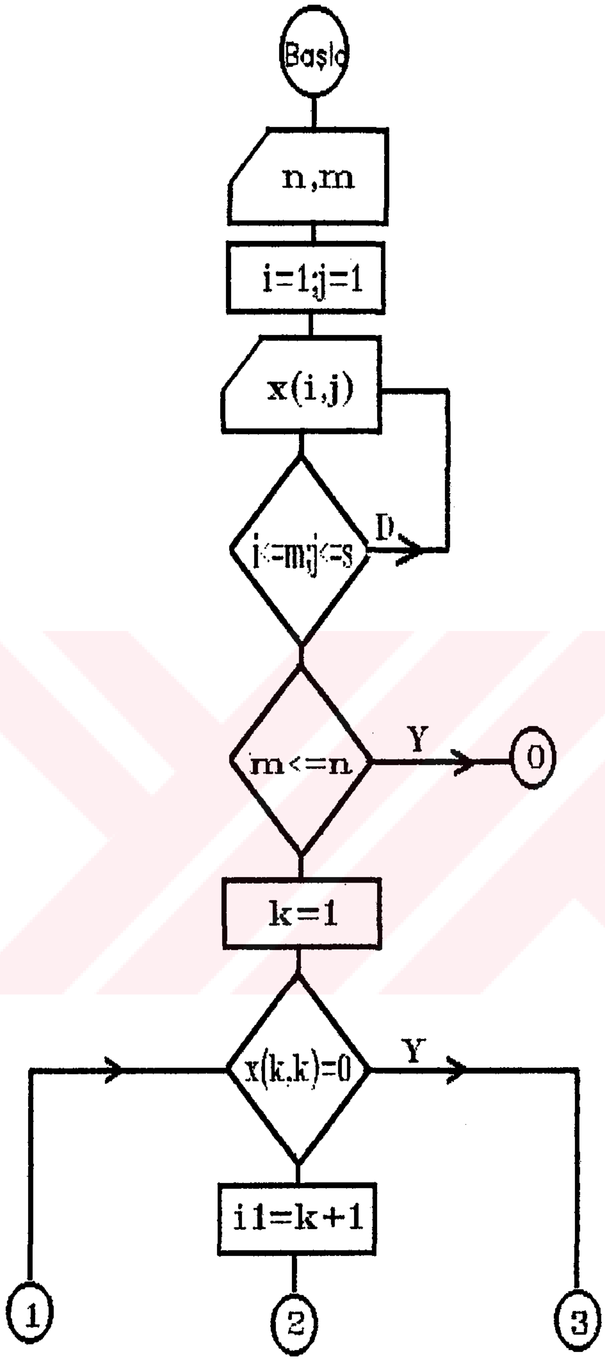
B12: $k < n$ ise B2'ye git;

B13: Çözümü yaz;

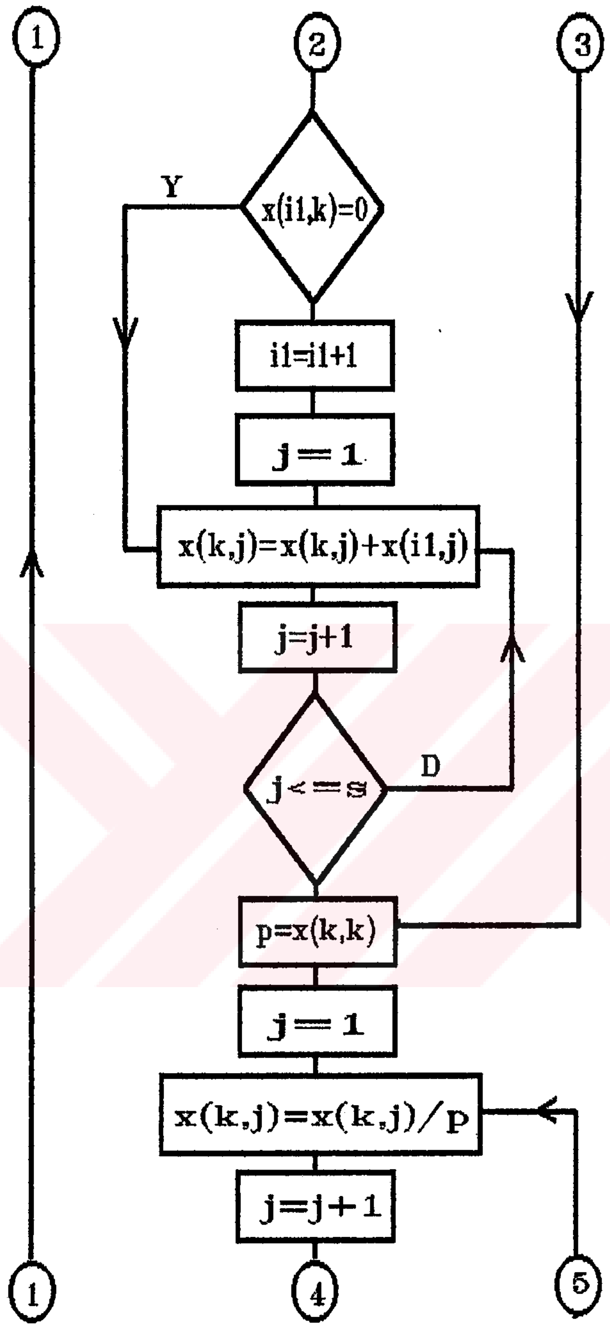
B14: Çözüm yoksa, çözüm yok yaz;

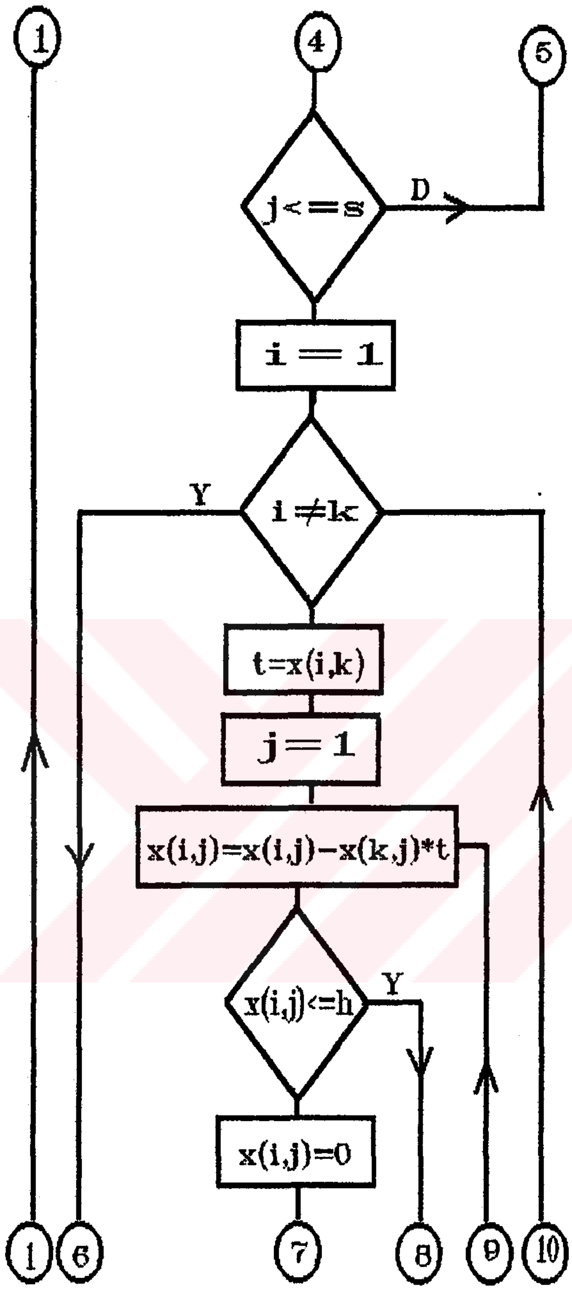
B15: Dur.

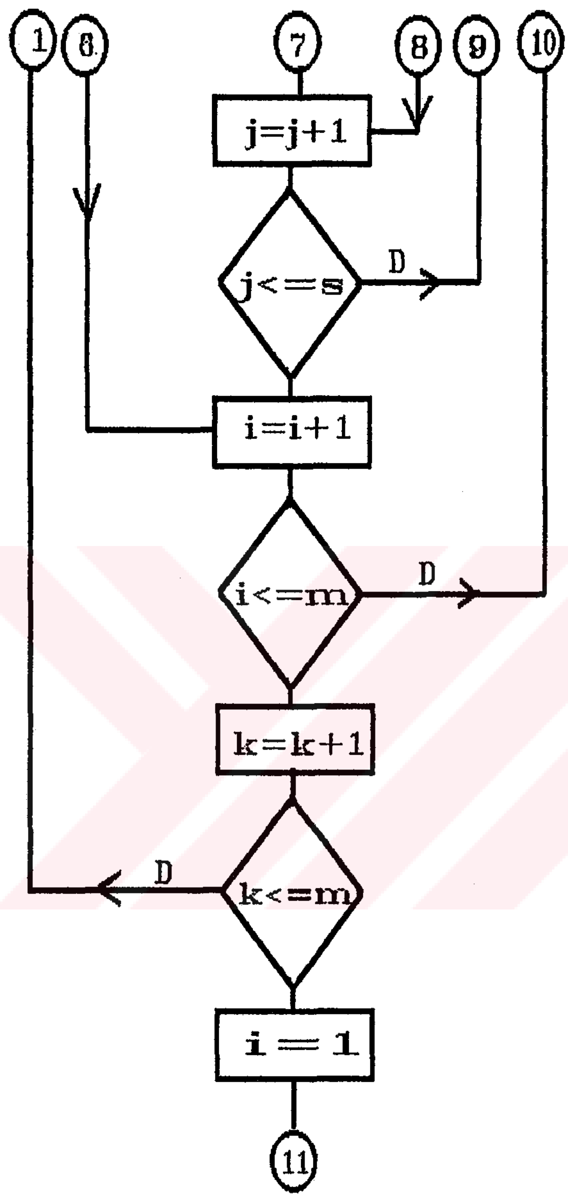
4.2. Akış Diyagramı

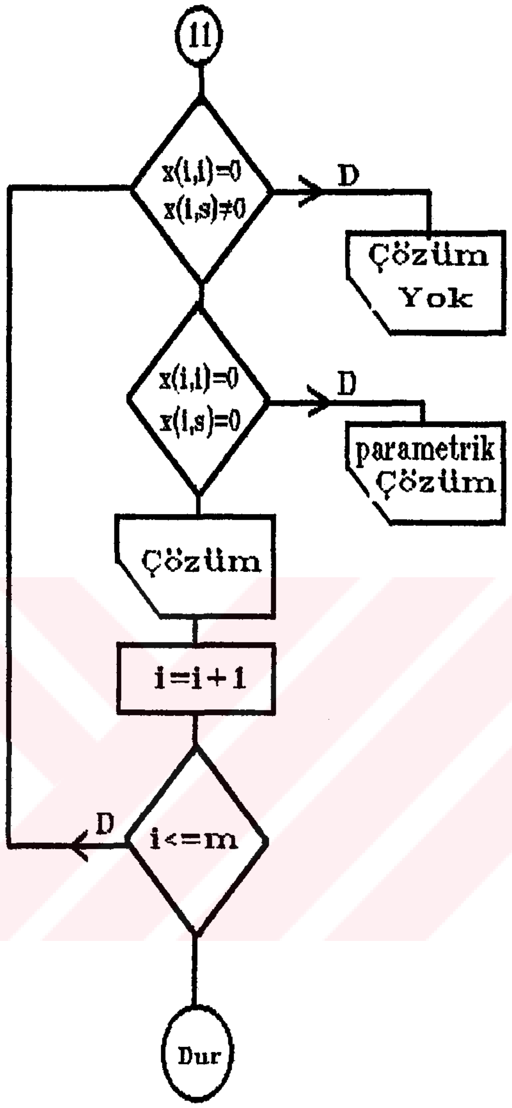


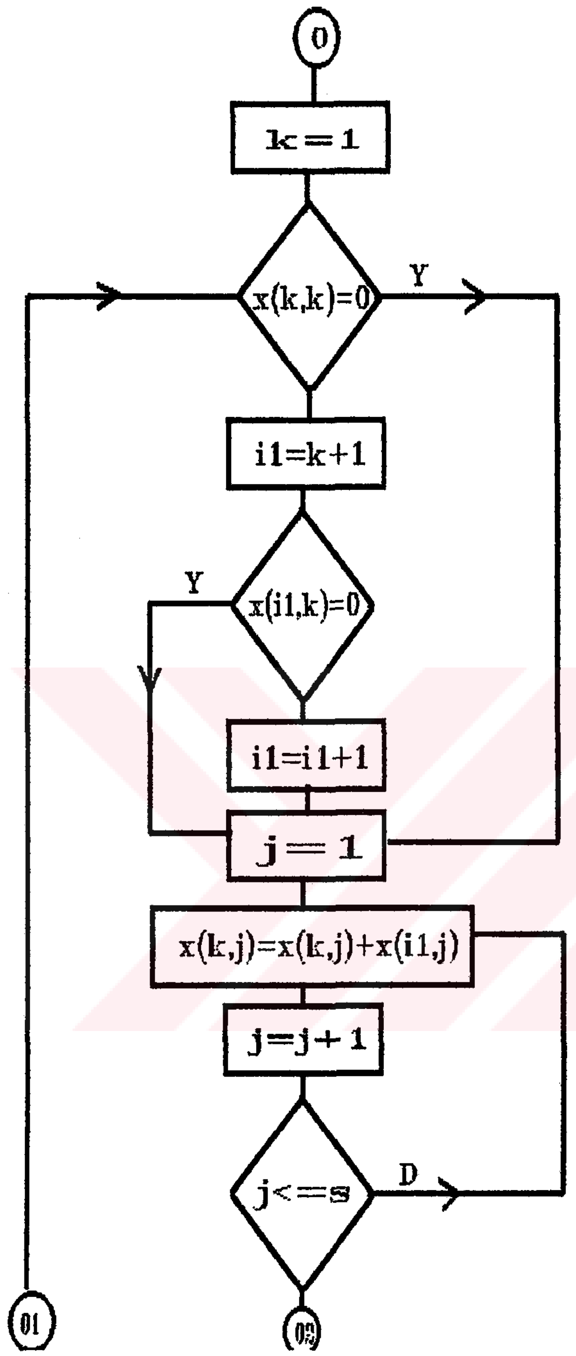
Başla
n,m
i=1;j=1
x(i,j)
i<=m;j<=s
D
m<=n
Y
0
k=1
x(k,k)=0
Y
i1=k+1
1
2
3

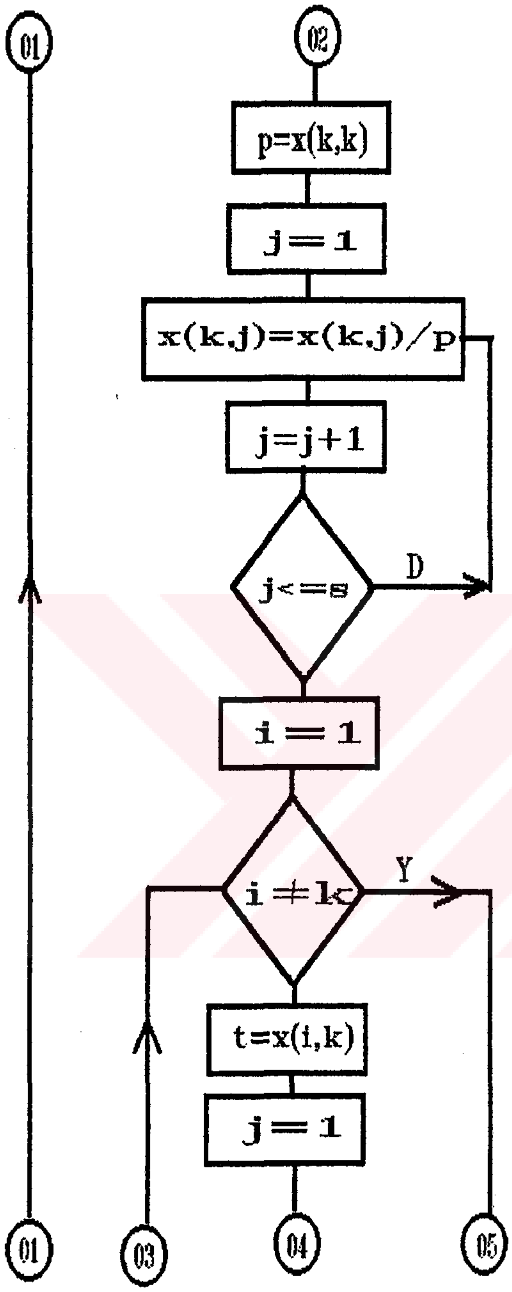

1
2
3
Y
x(i1,k)=0
i1=i1+1
j=1
x(k,j)=x(k,j)+x(i1,j)
j=j+1
D
j<=s
p=x(k,k)
j=1
x(k,j)=x(k,j)/p
j=j+1
1
4
5

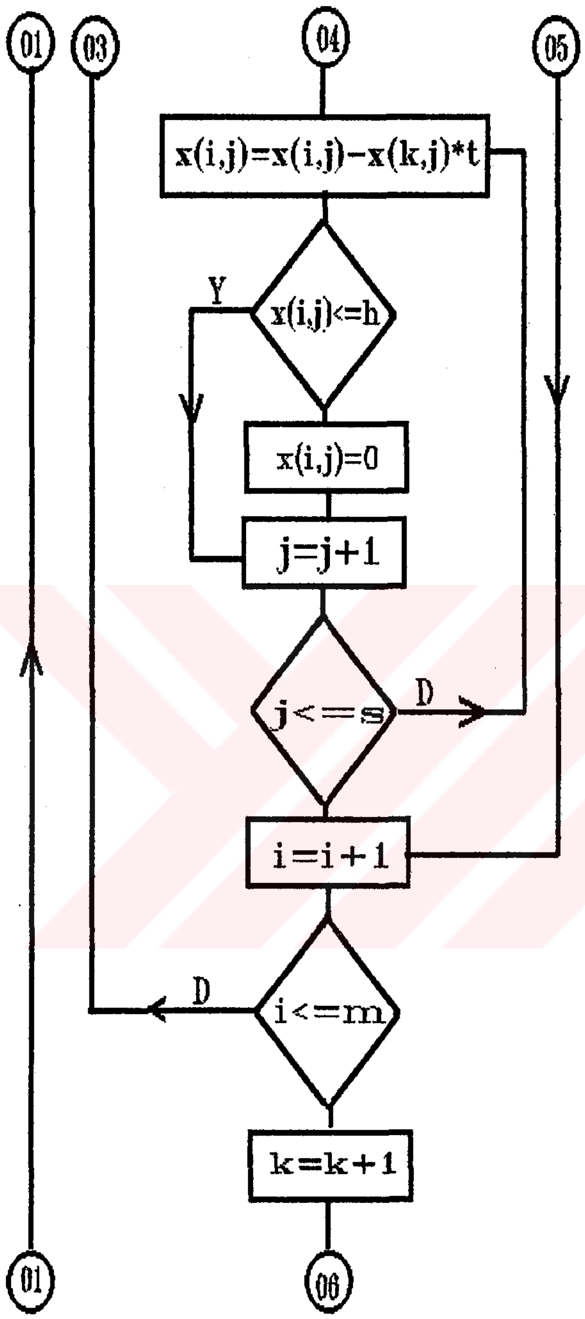

1
4
5
j<=s
D
i = 1
Y
i ≠ k
t=x(i,k)
j = 1
x(i,j)=x(i,j)-x(k,j)*t
x(i,j)<=h
Y
x(i,j)=0
1
6
7
8
9
10

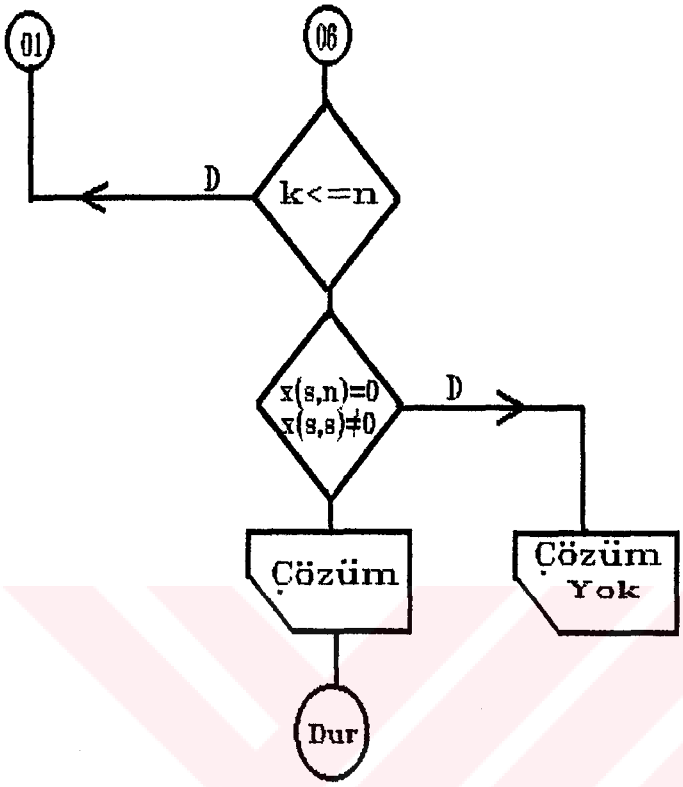

1
6
7
8
9
10
j=j+1
j<=s
D
i=i+1
i<=m
D
k=k+1
k<=m
D
i = 1
11


11
x(i,1)=0
x(i,s)≠0
D
Çözüm Yok
x(i,1)=0
x(i,s)=0
D
parametrik Çözüm
Çözüm
i=i+1
D
i<=m
Dur


0
k=1
x(k,k)=0
Y
i1=k+1
x(i1,k)=0
Y
i1=i1+1
j=1
x(k,j)=x(k,j)+x(i1,j)
j=j+1
j<=s
D
01


01
02
p=x(k,k)
j=1
x(k,j)=x(k,j)/p
j=j+1
j<=s
D
i=1
i≠k
Y
t=x(i,k)
j=1
01
03
04
05


01
03
04
05
x(i,j)=x(i,j)-x(k,j)*t
Y
x(i,j)<=h
x(i,j)=0
j=j+1
j<=s
D
i=i+1
D
i<=m
k=k+1
01
06


01
06
D
k<=n
x(s,n)=0
x(s,s)≠0
D
Çözüm
Çözüm
Yok
Dur

4.3. C Programı

/ *Bu program n bilinmeyenli m tane doğrusal denklemden oluşan doğrusal denklem sistemlerinin çözümünü yapar. */

/*x[i][j] ve c[i][j] : Genişletilmiş katsayılar matrisinin elemanlarını taşıyan değişkenler.
sat,sut : Çıkış için format düzenleyiciler.
K,K1,K2,K3,K4,K5: Parametreler.
qw[i][j]:c[i][j]'nin işaretini belirleyen değişken.
p: Köşegen üzerindeki elemanları taşıyan değişken.
t: p'nin bulunduğu sütundaki elemanları taşıyan değişken.*/

```
#include < stdio.h >
#include < math.h >
#include < string.h >
#include < graph.h >
char tk[1];
float x[20][20],c[20][20],qw[20][20];
main()
        {
int n,m,fark,s,i,i1,j,k,sat,sut;
float t,p;
float h=0.000001;
-clearscreen(GCLEARSCREEN);
printf(Değişken sayısını giriniz\n");
        scanf("%d",&n);
        printf("Denklem sayısını giriniz\n");
        scanf("%d",&m);
        fark=n-m; s=n+1; sat=6;
        printf("Genişletilmiş katsayılar matrisini giriniz\n");
for(i=1;i< =m;i+ +){
        sat=sat+1;sut=2;
for(j=1;j< =s;j+ +){
        sut=sut+6
        -settextposition(sat,sut);
        scanf("%f",&x[i][j]);
        }
}
printf("Çözüm\n");
        if(m< =n){
/*m< =n için genişletilmiş katsayılar matrisinin birim matrise dönüştürülmesi.*/
for(k=1;k< =m;k+ +){
        if(x[k][k]= =0){
        i1=k+1;
        if(x[i1][k]= =0)
        i1=i1+1;
for(j=1;j< =s;j+ +)
        x[k][j]=x[k][j]+x[i1][j];
}
        p=x[k][k];
```

```
for(j=1;j<=s;j++)
            x[k][j]=x[k][j]/p;
for(i=1;i<=m;i++){
            if(i!=k){
            t=x[i][k];
for(j=1;j<=s;j++){
            x[i][j]=x[i][j]-x[k][j]*t;
            if(fabs(x[i][j]<=h)
            x[i][j]=0;
                                    }

      }
}
for(i=1;i<=m;i++){
for(j=1;j<=s;j++)
            c[i][j]=x[i][j];
}
/*m<=n için birim matrise bağlı çözümlerin hesabı.*/
            if(c[m][m]==0&& c[m-1][m-1]==1&& c[m][s]==0){
for(i=1;i<=m-1;i++){
            if(c[i][n]<0){
            qw[i][n]=(-1)*c[i][n];strcpy(tk,"+");
            }
            if(c[i][n]>0){
            qw[i][n]=c[i][n];strcpy(tk,"-");
      }

            printf("X%d=%2.3f%s%2.3f*K\n",i,c[i][s],tk,qw[i][n]);
}
            printf("X%d=%s\n",m,"K");
            exit(0);
            }
if(c[m-1][m-1]==0&& c[m-1][s]==0&& c[m][m]==0&& c[m][s]==0){
for(i=1;i<=m-2;i++){
            if(c[i][n]<0){
            qw[i][n]=(-1)*c[i][n];strcpy(tk,"+");
                                    }

            if(c[i][n]>=0){
            qw[i][n]=c[i][n];strcpy(tk,"-");
            }
            if(c[i][n-1]<0){
            qw[i][n-1]=(-1)*c[i][n-1]);strcpy(tk,"+");
      }

            if(c[i][n-1])>0){
            qw[i][n-1]=c[i][n-1];strcpy(tk,"-");
}
            printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1\n",i,c[i][s],tk,qw[i][n],
            tk,qw[i][n-1]);
                                    }

            printf("X%d=%s\n",m,"K");
            printf("X%d=%s\n",m-1,"K1");
```

```
                exit(0);
                                            }
                if(c[m][m]==1&& n==m){
for(i=1;i<=m;i++)
                printf("X%d=%f\n",i,c[i][s]);
                }
                if(c[m][s]!=0&& c[m][m]==0){
                printf("Çözüm Yok\n");
                exit(0);
        }
                if(fark==1&& c[m][m]==0&& c[m][s]==0){
for(i=1;i<=m-1;i++){
                if(c[i][n]<0){
                qw[i][n]=(-1)*c[i][n];strcpy(tk,"+");
}
                if(c[i][n]>=0){
                qw[i][n]=c[i][n];strcpy(tk,"-");
                                            }
                if(c[i][n-1]<0){
                qw[i][n-1]=(-1)*c[i][n];strcpy(tk,"+");
                }
                if(c[i][n-1]>=0){
                qw[i][n-1]=c[i][n];strcpy(tk,"-");
                }
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1\n",i,c[i][s],tk,qw[i][n],tk,qw[i][n-1]);
        }
printf("Z=K1\nW=K\n");
                exit(0);
                                            }

                }
        }
for(i=1;i<=m;i++){
                if(c[i][n]<0){
                qw[i][n]=(-1)*c[i][n];strcpy(tk,"+");
}
                if(c[i][n]>=0){
                qw[i][n]=c[i][n];strcpy(tk,"-");
                                            }
                if(fark==1)
                printf("X%d=%2.3f%s%2.3f*K\n",i,c[i][s],tk,qw[i][n]);
                if(c[i][n-1]<0){
                qw[i][n-1]=(-1)*c[i][n-1];strcpy(tk,"+");
                }
                if(c[i][n-1]>=0){
                qw[i][n-1]=c[i][n-1];strcpy(tk,"-");
}
                if(fark==2)
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1\n",i,c[i][s]tk,qw[i][n],tk,
                qw[i][n-1]);
                if(c[i][n-2]<0){
```

```
                qw[i][n-2]=(-1)*c[i][n-2];strcpy(tk,"+");
                                             }
                if(c[i][n-2]>=0){
                qw[i][n-2]=c[i][n];strcpy("-");
                }
                if(fark==3)
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1%s%2.3f*K2\n",i,c[i][s],tk,
                qw[i][n],tk,qw[i][n-1],tk,qw[i][n-2]);
                if(c[i][n-3]<0){
                qw[i][n-3]=(-1)*c[i][n-3];strcpy(tk,"+");
}
                if(c[i][n-3]>=0){
                qw[i][n-3]=c[i][n];strcpy(tk,"-");
                if(fark==4)
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1%s%2.3f*K2%s%2.3f*K3\n",i,
                c[i][s],tk,qw[i][n],tk,qw[i][n-1],tk,qw[i][n-2],tk,qw[i][n-3]);
                if(c[i][n-4]<0){
                qw[i][n-4]=(-1)*c[i][n-4];strcpy(tk,"+");
                                             }
                if(c[i][n-4]>=0){
                qw[i][n-4]=c[i][n-4];strcpy(tk,"-');
                }
                if(fark==5)
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1%s%2.3f*K2%s%2.3f*K3%s
%2.3f*K4\n",i,c[i][s],tk,qw[i][n],tk,qw[i][n-1],tk,qw[i][n-2],tk,
                qw[i][n-3],tk,qw[i][n-4]);
                if(c[i][n-5]<0){
                qw[i][n-5]=(-1)*c[i][n-5];strcpy(tk,"+");
        }
                if(c[i][n-5]>=0){
                qw[i][n-5]=c[i][n-5];strcpy(tk,"-");
if(fark==6)
printf("X%d=%2.3f%s%2.3f*K%s%2.3f*K1%s%2.3f*K2%s%2.3f*K3%s
%2.3f*K4%s%2.3f*K5\n",i,c[i][s],tk,qw[i][n],tk,qw[i][n-1],tk,qw[i][n-2],
                tk,qw[i][n-3],tk,qw[i][n-4],tk,qw[i][n-5]);
                if(fark>6){
printf("Parametreye bağlı çözüm var\n");
                exit(0);
}
                if(fark==1)
printf("Z=K\n");
                if(fark==2)
printf("Z=K1\n"W=K\n");
                if(fark==3)
printf("W1=K\nW=K1\nZ=K2\n")
                if(fark==4)
printf("W2=K\nW1=K1\nW=K2\nZ=K3\n");
```

```
                    if(fark = =5)
printf("W3 = K\nW2 = K1\nW1 = K2\nW = K3\nZ = K4\n");
                    if(fark = =6)
printf("W4 = K\nW3 = K1\nW2 = K2\nW1 = K3\nW = K4\nZ = K5\n");
                    if(m > n){
/* m > n için genişletilmiş katsayılar matrisinin birim matrise dönüştürülmesi */
for(k = 1;k < =n;k + +){
                    if(x[k][k] = =0){
                    i1 = k + 1;
                    if(x[i1][k] = =0)
                    i1 = i1 + 1;
for(j = 1;j < =s;j + +)
                    x[k][j] = x[k][j] + x[i1][j];
                                                  }
                    p = x[k][k];
for(j = 1;j < =s;j + +)
                    x[k][j] = x[k][j]/p;
for(i = 1;i < =m;i + +){
                    if(i! = k){
                    t = x[i][k];
for(j = 1;j < =s;j + +){
                    x[i][j] = x[i][j]-x[k][j]*t;
                    if(fabs(x[i][j] < =h)
                    x[i][j] = 0;
                                                  }
                    }
          }
for(i = 1;i < =m;i + +){
for(j = 1;j < =s;j + +)
                    c[i][j] = x[i][j];
}
/* m > n için birim matrise bağlı çözümlerin hesabı.*/
                    if(c[s][n] = =0 && c[m][s] ! =0){
printf("Çözüm Yok\n");
                    exit(0);
                                                  }
                    if(c[n][n] = =1){
for(i = 1;i < n;i + +)
                    printf("X%d = %2.3f\n",i,c[i][s]);
                                                  }
                    }
          }
}
```

4.4. Program Uygulaması

Bu alt kesimde, geliştirilen program yardımıyla doğrusal denklem sistemlerinin farklı şartları için çözümleri bulunan birer örnek verilecektir.

4.4.1. Örnek

Denklem sistemi

$$x + 2y + 3z = 4$$
$$-y + 2z = 1$$
$$2x + 4y + 6z = 8$$

olsun.

Programın ürettiği çözüm:

X = 6.0-7.0*K
Y = -1.0 + 2.0*K
Z = K

4.4.2 Örnek

Denklem sistemi

$$2x - y + 9z = 6$$
$$4x + y + 6z = 7$$
$$2x - 4y + 21z = 10$$

olsun.

Programın ürettiği sonuç

Çözüm yok.

4.4.3. Örnek

Denklem sistemi

$$2x + z = 3$$
$$2x + 4y - z = 1$$
$$x - 8y - 3z = -2$$

olsun.

Programın ürettiği sonuç

X = 1
Y = 0
Z = 1

5. SONUÇ

Kuralan algoritmalar irdelenerek, n bilinmeyenli m tane doğrusal denklemden oluşan doğrusal denklem sistemlerinin,

i. $m > n$
ii. $m = n$
iii. $m < n$

koşulları altında çözümlerini bulan etkin bir algoritma geliştirildi ve C dilinde programı yazıldı. Geliştirilen program Reduce adlı bir program ile karşılaştırıldı(Hearn, 1987). Sonuçta önerilen programın daha hızlı (işlemlerin aldığı zaman bakımından) olduğu gözlendi.

KAYNAKLAR

Abramchuk, V.S., 1989, Investigation of nümerical stability of algorithms for solution of systems of linear equations. Dokl. Akad. Nauk Ukr. SSR, Ser. A, no.8,3-5.

Aktaş,Z, 1981, Sayısal Çözümleme. ODTÜ-Bilgisayar Mühendisliği Bölümü, Ankara, S. 91-144.

Al-Towaiq, M., 1989, A direct method for the solution of large systems of linear equations using incomplete LU-factorization. Math. Jap. 34, no. 5, 675-682.

Carnahan, B., Luther, H.A. and Wilkes, J.O., 1969, Applied Nümerical Methods. New York, P.269-340.

Hearn, C.A., 1987, Reduce User's Manuel. RAND Puplication CP78, p. 32-43.

Ikramov, Kh.D., 1989, Nümerical methods of linear algebra (Solution of linear equtions). Math. Kibern. no.4, 48-49.

Kobayashi, H., Fujise, T., Furukawa, A., 1988, Solving systems of algebraic equations by a general elimination method. J.Symbolic comput. 5, no. 3, 303-320.

Moore, J.T., 1972, Elemantary Linear and Matrix Algebra. Tokyo, p. 47-82.

Stevvart, G.W., 1973, Computer Science and Applied Mathematics. New York, p. 106-148.

Stoll, R.R., 1952, Linear Algebra and Matrix Theory. New York, p. 1-77.

Valitskij, Yu.N., 1989, A method for the solution of a system of linear equations with a Vandermonde determinant. Sci.-Methodical Collect., 5, Kiev, 39-47.

Yunusoğlu, A., 1974, Uygulamalı Sayısal Çözümleme. Türkiye Bilişim Derneği, Ankara., S. 103-161.