SDÜ FEN EDEBİYAT FAKÜLTESİ
FEN DERGİSİ (E-DERGİ). 2008, 3(1), 98-106

IDDM YARDIMIYLA TERS MATRİS HESAPLAMA

Ali Osman ÇIBIKDİKEN [*], **Kemal AYDIN** [**]

[*] Selçuk Üniversitesi, Kadınhanı MYO, Bilgisayar Teknolojileri ve Proglama, Kadınhanı, KONYA, e-posta: aocdiken@selcuk.edu.tr
[**] Selçuk Üniversitesi, Alaaddin Keykubat Kampusü, Fen-Edebiyat Fakültesi, Matematik Bölümü, KONYA, e-posta: kaydin@selcuk.edu.tr
Alınış: 01 Ocak 2008, Kabul: 24 Nisan 2008

Özet: Bu çalışmada, lineer cebirsel denklem sistemlerinin çözümü için (KESKİN & AYDIN 2007) de verilen iteratif boyut indirgeme metodunu (IDDM) kullanarak verilen regüler matrisin tersini bulan bir metod geliştirdik. Bu metod üzerine kurulan bir algoritma ve Maple programı verdik. Ayrıca bazı nümerik örnekler de verdik.

Anahtar sözcükler: lineer sistem, iteratif metot, ters matris.

CALCULATION INVERSE OF MATRIX WITH IDDM

Abstract: In this study, we have developed a method to calculate inverse of given regular matrix with Iterative Decreasing Dimension Method (IDDM) that its given in (KESKİN & AYDIN 2007) to solve the linear algebraic equations systems. We have given a algorithm which is based on this method and Maple program. So we also have given some numerical examples.

Key words: linear system, iterative method, inverse matrix.

Mathematics Subject Classifications (2000): 65F10, 15A09.

1. GİRİŞ

$A = (a_{ij})$ matrisi n boyutlu regüler matris olmak üzere,

$$A\,Y = I,\ I\text{-birim matris},\ Y = (\,Y_{ij}\,) \tag{1}$$

olacak şekilde Y matrisinin varlığı ve tekliği iyi bilinmektedir (MIRSKY 1955, GOLUB & VAN LOAN 1989). $l = 1(1)n$ ($l = 1, 2, \ldots, n$) için Y_l ler Y matrisinin ve I_l ler de I-birim matrisinin sütun vektörleri olmak üzere,

$$A\,Y = A\,[Y_1\ Y_2\ \ldots\ Y_n] = [AY_1\ AY_2\ \ldots\ AY_n] = [I_1\ I_2\ \ldots\ I_n] \tag{2}$$

şeklinde yazabiliriz. (2) eşitliğinden

$$A\,Y_l = I_l,\ l = 1(1)n\ ;\quad Y_l = (\,Y_{il}\,),\ I_l = (\,I_{il}\,),\ i = 1(1)n \tag{3}$$

lineer cebirsel denklem sistemlerini oluşturabiliriz. (3) sistemlerinin Y_l çözüm vektörleri, aranan Y matrisinin sütun vektörleridir. Y matrisi, regüler A matrisinin ters matrisidir (SHAMPINE vd. 1997).
(3) denklem sistemlerinin çözümü için (KESKİN & AYDIN 2007) de bir iteratif boyut indirgeme metodu (IDDM) ve bu metod üzerine kurulmuş bir algoritma (IDDA)

verilmiştir. Bu çalışmanın 2. bölümünde; IDDM geliştirilerek regüler A matrisinin tersini bulan bir metod, 3. bölümünde; verilen metodun üzerine kurulan bir algoritma ve bir Maple programı, 4. bölümde ise bazı örnekler verilmiştir.

2. METOD

Bu bölümde, (KESKİN & AYDIN 2007) da verilmiş olan IDDM metodunun bir özeti verilmiştir. Ayrıca bu çalışmadaki semboller, (KESKİN & AYDIN 2007) de kullanılan sembollere uygun olarak seçilmiştir.

2.1. SEMBOLLER

k	iterasyon sayısı (k = 1, 2, ..., n)	$k = 1(1)n$
m	indirgenen katsayı matrisinin boyutu	$m = n - k + 1$
$A^{(k)}$	indirgenen katsayı matrisi	$m \times m$ matris
Y_l	indirgenen sistemlerin çözüm vektörleri	m-vektör
$f^{(k)}$, $I_l^{(k)}$	indirgenen sistemlerin sağ taraf vektörleri	m-vektör
$A_1^{(k)}$	$A^{(k)}$ matrisinin 1. satır vektöründen oluşan matris	$1 \times m$ matris
$a_{1s}^{(k)}$	$A_1^{(k)}$ nin sıfırdan farklı ilk elemanı	skaler
$A_2^{(k)}$	$A^{(k)}$ matrisinin diğer satır vektörlerinden oluşan matris	$(m-1) \times m$ matris
$0_{p \times q}$	sıfır matrisi	$p \times q$ matris
$I_{p \times p}$	birim matrisi	$p \times p$ kare matris
$R^{(k)}$	$A_1^{(k)}\,\mathrm{Y}_l^{(k)} = 0$ sisteminin çözüm uzayının baz vektörlerinden oluşan matris	$m \times (m-1)$ matris
$u^{(k)}$, $u_l^{(k)}$	sırasıyla $f^{(k)}$, $I_l^{(k)}$ nın 1.elemanından oluşan vektörler	1-vektör
$v^{(k)}$, $v_l^{(k)}$	sırasıyla $f^{(k)}$, $I_l^{(k)}$ nın diğer elemanlarından oluşan vektörler	$(m-1)$-vektör
$Y_{lö}^{(k)}$	$A_1^{(k)}\, Y_l^{(k)} = u_l^{(k)}$ sisteminin *bir* özel çözümü	m-vektör
$Y_{lh}^{(k)}$	$A_1^{(k)}\, Y_l^{(k)} = 0$ sisteminin çözümü	m-vektör

2.2. İTERATİF BOYUT İNDİRGEME METODU (IDDM)

$AX = f$; A - $n \times n$ -regüler matris, X , f - n-vektör
lineer cebirsel denklem sistemini ele alalım. k = 1(1)n iterasyon sayısı, $m = n - k + 1$ indirgenen katsayı matrisinin boyutu, sırasıyla

$$A^{(k)} = \begin{cases} A & k = 1 \\ A_2^{(k\text{-}1)} R^{(k\text{-}1)} & k = 2(1)n \end{cases} \quad ; \quad f^{(k)} = \begin{cases} f & k = 1 \\ v^{(k-1)} - A_2^{(k-1)} X_0^{(k-1)} & k = 2(1)n \end{cases}$$

ile tanımlanan indirgenen katsayı matrisi ve indirgenen sistemin sağ taraf vektörü,

$$X_0^{(k)} = \begin{pmatrix} 0 & \dots & 0 & \dfrac{f_1^{(k)}}{a_{1\text{s}}^{(k)}} & 0 & \dots & 0 \end{pmatrix}^T$$

ile verilen bir özel çözüm,

$$R^{(k)} = \begin{cases} \begin{pmatrix} r_{1\times(m-1)}^{(k)} \\ I_{(m-1)\times(m-1)} \end{pmatrix}, & \text{s} = 1 \\ \begin{pmatrix} I_{(\text{s}-1)\times(\text{s}-1)} & 0_{(\text{s}-1)\times(m-\text{s})} \\ 0_{1\times(\text{s}-1)} & r_{1\times(m-\text{s})}^{(k)} \\ 0_{(m-\text{s})\times(\text{s}-1)} & I_{(m-\text{s})\times(m-\text{s})} \end{pmatrix}, & \text{s} = 2(1)m-1 \\ \begin{pmatrix} I_{(m-1)\times(m-1)} \\ 0_{1\times(m-1)} \end{pmatrix}, & \text{s} = m \end{cases}$$

olmak üzere verilen lineer denklem sisteminin çözümü,

$$X = X^{(1)} = \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} \mathrm{R}^{(j)} \right) X_0^{(i)} \quad ; \quad \prod_{j=1}^{i-1} R^{(j)} = \begin{cases} R^{(1)} R^{(2)} \dots R^{(\text{i}-1)} & ; \ i > 1 \\ I & ; \ i = 1 \end{cases}$$

ile verilmektedir (KESKİN & AYDIN 2007).

2.3. IDDM YARDIMIYLA TERS BULMA

$$A\, Y_l = I_l \ ; \ l = 1(1)n \tag{4}$$

lineer cebirsel denklem sistemlerini ele alalım. Burada

$$I_l = (I_{il}), \ i, l = 1(1)n \ ; \ I_{il} = \begin{cases} 1 & i = l \\ 0 & i \neq l \end{cases}$$

şeklinde tanımlıdır. Şimdi ***l* = 1 için** (4) sistemine IDDM yi uygulayalım:

***l* = 1 için**; (4) sistemini

$$A\, Y_1 = A^{(1)}\, Y_1^{(1)} = I_1^{(1)} = I_1 \tag{5}$$

şeklinde yeniden yazalım. (5) sistemini,

$$A_1^{(1)}\, Y_1^{(1)} = u_1^{(1)}, \quad A_1^{(1)} = \left(a_{1j}^{(1)}\right)^{j=1(1)n}, \ u_1^{(1)} = \left(I_{11}^{(1)}\right) \tag{6}$$

$$A_2^{(1)}\, Y_1^{(1)} = v_1^{(1)}, \quad A_2^{(1)} = \left(a_{ij}^{(1)}\right)_{i=2(1)n}^{j=1(1)n}, \quad v_1^{(1)} = \left(I_{il}^{(1)}\right)_{i=2(1)n} \tag{7}$$

olacak şekilde ikiye bölelim. (6) sisteminin bir özel çözümü

$$Y_{1ö}^{(1)} = \begin{pmatrix} 0 & \dots & 0 & \dfrac{I_{11}^{(1)}}{a_{1s}^{(1)}} & 0 & \dots & 0 \end{pmatrix}^{T} \tag{8}$$

olarak seçilebilir. Burada $a_{1s}^{(1)}$ $(1 \le s \le n)$, $A_1^{(1)}$ matrisinin sıfırdan farklı ilk elemanıdır. (7) sisteminin homojen çözümü

$$Y_{1h}^{(1)} = R^{(1)} Y_1^{(2)} \tag{9}$$

olarak bulunur. Burada $Y_1^{(2)}$, Y_{i1} $(i = 1(1)n,\ i \neq s)$ parametrik değişkenlerden oluşan $(n-1)$ boyutlu vektör ve $R^{(1)}$ çözüm uzayının baz vektörlerinden oluşan

$$R^{(1)} = \begin{cases} \begin{pmatrix} r_{1\times(n-1)}^{(1)} \\ I_{(n-1)\times(n-1)} \end{pmatrix} & s = 1 \\ \begin{pmatrix} I_{(s-1)\times(s-1)} & 0_{(s-1)\times(n-s)} \\ 0_{1\times(s-1)} & r_{1\times(n-s)}^{(1)} \\ 0_{(n-s)\times(s-1)} & I_{(n-s)\times(n-s)} \end{pmatrix} & s = 2(1)n-1 \\ \begin{pmatrix} I_{(n-1)\times(n-1)} \\ 0_{1\times(n-1)} \end{pmatrix} & s = n \end{cases}$$

şeklinde $n \times (n-1)$ matristir. Burada $r_{1\times(n-s)}^{(1)} = r^{(1)} = \begin{pmatrix} r_{1(s+1)}^{(1)} & r_{1(s+2)}^{(1)} & \dots & r_{1n}^{(1)} \end{pmatrix}$; $r_{1j}^{(1)} = -\dfrac{a_{1j}^{(1)}}{a_{1s}^{(1)}}$, $j = s+1(1)n$ dir. (5) sisteminin genel çözümü ise

$$Y_1^{(1)} = Y_{1ö}^{(1)} + Y_{1h}^{(1)} = Y_{1ö}^{(1)} + R^{(1)} Y_1^{(2)} \tag{10}$$

oluşan matristir. $Y_1^{(1)}$ çözümü (7) de yerine yazar ve düzenlersek $(n\text{-}1)$ boyutlu

$$A^{(2)}\ Y_1^{(2)} = I_1^{(2)}; \quad A^{(2)} = A_2^{(1)} R^{(1)}, \quad I_1^{(2)} = v_1^{(1)} - A_2^{(1)}\ Y_{1ö}^{(1)} \tag{11}$$

yeni denklem sistemine ulaşılır. (6)-(10) arası adımlar (11) sistemine uygulanırsa biri diğerini takip eden sistemler $A^{(k)} = A_2^{(k-1)} R^{(k\text{-}1)}$, $I_1^{(k)} = v_1^{(k-1)} - A_2^{(k-1)}\ Y_{1ö}^{(k-1)}$, $k = 2(1)n$ olmak üzere

$$A^{(k)}\ Y_1^{(k)} = I_1^{(k)} \tag{12}$$

şeklinde elde edilir. $k = n$ olması durumunda ortaya çıkan lineer denklem sisteminin boyutunun 1 olacağı açıktır. Böylece $k = n(-1)2$ için $Y_1^{(k)}$ vektörlerini, $Y_1^{(k-1)}$ vektörlerinde yerine koyarsak (5) sisteminin çözümü

$$Y_1 = Y_1^{(1)} = \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} R^{(j)} \right) Y_{1ö}^{(i)} \tag{13}$$

olarak bulunur. Buraya kadar olan kısım IDDM nin (5) sistemine adım adım uygulanmasıdır.

***l* = 2(1)*n* için**; $l = 1$ durumunda bulunan $R^{(k)}$ ve $A^{(k)}$ matrisleri sağ taraf vektörlerine bağlı olmadığından

$$A\ Y_l = I_l\,, \quad l = 2(1)n \tag{14}$$

sistemlerinde (6)-(13) arasındaki hesaplamalardan sadece

$$Y_{\text{lö}}^{(k)} = \begin{pmatrix} 0 & \dots & 0 & \dfrac{I_{11}^{(k)}}{a_{1s}^{(k)}} & 0 & \dots & 0 \end{pmatrix}^T, \; I_l^{(k)} = v_l^{(k\text{-}1)} - A_2^{(k-1)} Y_{l\text{ö}}^{(k-1)}, \; k = 2(1)n$$

vektörlerini hesaplamak yeterli olacaktır. Bu vektörler hesaplandıktan sonra (14) sistemlerinin çözüm vektörleri

$$Y_l = Y_l^{(1)} = \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} R^{(j)} \right) Y_{l\text{ö}}^{(i)}, \quad l = 2(1)n$$

şeklinde elde edilir.

Şimdi (KESKİN & AYDIN 2007) de verilen Teorem 1 in, $A\ Y = I$ sistemine genişletilmiş hali olan ve A matrisinin tersini veren teoremi verelim.

Teorem 1. A regüler matris, I–birim matris olmak üzere $A\ Y = I$ sisteminin çözümü,

$$Y = A^{-1} = [Y_1\ Y_2\ \dots\ Y_n]\ ; \quad Y_l = \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} R^{(j)} \right) Y_{l\text{ö}}^{(i)}, \quad l = 1(1)n$$

dir. Burada $\prod_{j=1}^{i-1} R^{(j)} = \begin{cases} R^{(1)} R^{(2)} \dots R^{(i-1)} & ; \; i > 1 \\ I & ; \; i = 1 \end{cases}$ şeklinde tanımlanmaktadır.

İspat. İspat, KESKİN & AYDIN (2007)'deki Teorem 1 in ispatından ve metodun açıklamasından açıktır.

3. ALGORİTMA

Veri: A - regüler matris.
Adım 1. n = boyut(A).
Adım 2. $k = 1(1)n-1$, $m = n - k + 1$; $A^{(1)} = A$

2.1. $A^{(k)} = \begin{pmatrix} A_1^{(k)} \\ A_2^{(k)} \end{pmatrix}$; $A_1^{(k)} = \left(a_{1j}^{(k)} \right)^{j=1(1)m}$, $A_2^{(k)} = \left(a_{ij}^{(k)} \right)_{i=2(1)m}^{j=1(1)m}$

2.2. $a_{1s}^{(k)}$ bul. $a_{1s}^{(k)}$, $a_{1j}^{(k)} \neq 0$, $j = 1(1)m$ ilk eleman. s_k = s al.

2.3. Eğer $1 \leq s < m$ ise,

$$r^{(k)} = \begin{pmatrix} r_{1\times(s+1)}^{(k)} & r_{1\times(s+2)}^{(k)} & \dots & r_{1\times m}^{(k)} \end{pmatrix}; \; r_{1j}^{(k)} = -\frac{a_{1j}^{(k)}}{a_{1s}^{(k)}} \; ; j = s+1(1)m$$

2.4. $R^{(k)}$ hesapla.

$$R^{(k)} = \begin{cases} \begin{pmatrix} r^{(k)}_{1\times(m-1)} \\ I_{(m-1)\times(m-1)} \end{pmatrix} & s = 1 \\ \begin{pmatrix} I_{(s-1)\times(s-1)} & 0_{(s-1)\times(m-s)} \\ 0_{1\times(s-1)} & r^{(k)}_{1\times(m-s)} \\ 0_{(m-s)\times(s-1)} & I_{(m-s)\times(m-s)} \end{pmatrix} & s = 2(1)m-1 \\ \begin{pmatrix} I_{(m-1)\times(m-1)} \\ 0_{1\times(m-1)} \end{pmatrix} & s = m \end{cases}$$

2.5. $A^{(k+1)} = A_2^{(k)} R^{(k)}$ hesapla.

Adım 3. $l = 1(1)n$,

3.1. $I_l^{(1)} = I_l$, $I_l = (I_{il})$, $i = 1(1)n$; $I_{il} = \begin{cases} 1 & i = l \\ 0 & i \neq l \end{cases}$

3.2. $k = 1(1)n-1$, $m = n - k + 1$, s = s$_k$

3.2.1. $I_l^{(k)} = \begin{pmatrix} u_1^{(k)} \\ v_1^{(k)} \end{pmatrix}$; $u_1^{(k)} = \left(I_{1l}^{(k)}\right)$, $v_1^{(k)} = \left(I_{il}^{(k)}\right)_{i=2(1)m}$

3.2.2. $Y_{l\ddot{o}}^{(k)} = \begin{pmatrix} 0 & \ldots & 0 & \dfrac{I_{1l}^{(k)}}{a_{1s}^{(k)}} & 0 & \ldots & 0 \end{pmatrix}^{\mathrm{T}}$

3.2.3. $I_l^{(k+1)} = v_l^{(k)} - A_2^{(k)} Y_{l\ddot{o}}^{(k)}$

3.3. $k = n$; $A^{(n)} = \left(a_{11}^{(n)}\right)$, $I_l^{(n)} = \left(I_{1l}^{(n)}\right)$

3.3.1. $Y_{l\ddot{o}}^{(n)} = \left(\dfrac{I_{1l}^{(n)}}{a_{11}^{(n)}} \right)$

3.4. $Y_l = \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} R^{(j)} \right) Y_{l\ddot{o}}^{(i)}$

Çıktı:

$Y = A^{-1} = [\ Y_1\ Y_2\ \ldots\ Y_n\ ]$

Örnek 1. $A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 4 \\ -1 & 0 & 6 \end{bmatrix}$ matrisinin tersini bulalım.

Veri: $A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 4 \\ -1 & 0 & 6 \end{bmatrix}$

Adım 1. n = boyut(A) = 3

Adım 2. $\boldsymbol{k = 1}$, $\boldsymbol{m = 3}$, $A^{(1)} = A$

s = s$_1$ = 1 $\Rightarrow$ $r^{(1)} = \begin{bmatrix} -2 & 0 \end{bmatrix}$, $R^{(1)} = \begin{bmatrix} -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$, $A^{(2)} = \begin{bmatrix} 1 & 4 \\ 2 & 6 \end{bmatrix}$

$\boldsymbol{k = 2}$, $\boldsymbol{m = 2}$

$s = s_2 = 1 \Rightarrow r^{(2)} = [-4],\ R^{(2)} = [-4 \quad 1]^T,\ A^{(3)} = [-2]$

Adım 3. $\boldsymbol{l = 1} \Rightarrow I_1^{(1)} = [1 \quad 0 \quad 0]^T$

$\boldsymbol{k = 1}, \boldsymbol{m = 3}, s = s_1 = 1 \Rightarrow Y_{1ö}^{(1)} = [1 \quad 0 \quad 0]^T,\ I_1^{(2)} = [0 \quad 1]^T,$

$\boldsymbol{k = 2}, \boldsymbol{m = 2}, s = s_2 = 1 \Rightarrow Y_{1ö}^{(2)} = [0 \quad 0]^T,\ I_1^{(3)} = [1],$

$\boldsymbol{k = 3}, \boldsymbol{m = 1} \Rightarrow Y_{1ö}^{(3)} = [-0.5]$

$$Y_1 = \sum_{i=1}^{3}\left(\prod_{j=1}^{i-1} R^{(j)}\right) Y_{1ö}^{(i)} = [-3 \quad 2 \quad -0.5]^T$$

$\boldsymbol{l = 2} \Rightarrow I_2^{(1)} = [0 \quad 1 \quad 0]^T$

$\boldsymbol{k = 1}, \boldsymbol{m = 3}, s = s_1 = 1 \Rightarrow Y_{2ö}^{(1)} = [0 \quad 0 \quad 0]^T,\ I_2^{(2)} = [1 \quad 0]^T,$

$\boldsymbol{k = 2}, \boldsymbol{m = 2}, s = s_2 = 1 \Rightarrow Y_{2ö}^{(2)} = [1 \quad 0]^T,\ I_2^{(3)} = [-2],$

$\boldsymbol{k = 3}, \boldsymbol{m = 3} \Rightarrow Y_{2ö}^{(3)} = [1]$

$$Y_2 = \sum_{i=1}^{3}\left(\prod_{j=1}^{i-1} R^{(j)}\right) Y_{2ö}^{(i)} = [6 \quad -3 \quad 1]^T$$

$\boldsymbol{l = 3} \Rightarrow I_3^{(1)} = [0 \quad 0 \quad 1]^T$

$\boldsymbol{k = 1}, \boldsymbol{m = 3}, s = s_1 = 1 \Rightarrow Y_{3ö}^{(1)} = [0 \quad 0 \quad 0]^T,\ I_3^{(2)} = [0 \quad 1]^T,$

$\boldsymbol{k = 2}, \boldsymbol{m = 2}, s = s_2 = 1 \Rightarrow Y_{3ö}^{(2)} = [0 \quad 0 \quad 0]^T,\ I_3^{(3)} = [1],$

$\boldsymbol{k = 3} \Rightarrow Y_{3ö}^{(3)} = [-0.5]$

$$Y_3 = \sum_{i=1}^{3}\left(\prod_{j=1}^{i-1} R^{(j)}\right) Y_{3ö}^{(i)} = [-4 \quad 2 \quad -0.5]^T$$

Çıktı: $Y = A^{-1} = \begin{bmatrix} -3 & 6 & -4 \\ 2 & -3 & 2 \\ -0.5 & 1 & -0.5 \end{bmatrix}$

4. MAPLE PROGRAMI

```
>   restart:
>   with(LinearAlgebra, SubMatrix, SubVector, MatrixMatrixMultiply,
MatrixVectorMultiply, RowDimension);
>   with(linalg, blockmatrix);
> tersbul:= proc(A::Matrix)

#IDDM Yardımıyla Ters Matris Hesaplama için Maple Procedure

global B, A1, A2, Y0, r, B1, B2, B3, B4, B5, B6, H1, H2, R, RR, S, Y, C, e, d,
c, s, u, v;
local n, j, m, k, z, i, l;
n := RowDimension(A); B[1] := A;
printf("A[1]="); print(B[1]);
for k from 1 to n-1 do m := n-k+1;
A1[k]:= SubMatrix(B[k], 1..1, 1..m); A2[k]:= SubMatrix(B[k], 2..m, 1..m);

#A1 Matrisinde 0 dan Farklı İlk Elemanın Yeri Bulunuyor
```

```
  for z from 1 to m do
    if A1[k][1, z] <> 0 then c[k]:= z; break; end if:
  end do:

#r Vektörü Oluşturuluyor
  s:= c[k]; r[k]:= Matrix(1, 1 .. m-s);
    for z from 1 to m-s do r[k][1, z]:= -B[k][1, s+z]/B[k][1, s]; end do:

#R[k] Matrisi Oluşturuluyor
  H1:= Matrix(m-1, m-1, shape = identity); H2:= Matrix(1, m-1, 0);
  B1:= Matrix(s-1, s-1, shape = identity); B2:= Matrix(s-1, m-s, 0);
  B3:= Matrix(1, s-1, 0); B4:= r[k]; B5:= Matrix(m-s, s-1, 0);
  B6:= Matrix(m-s, m-s, shape = identity);
    if s = 1 then R[k]:= convert(blockmatrix(2, 1, [r[k], H1]), Matrix); end
if:
    if s = m then R[k]:= convert(blockmatrix(2, 1, [H1, H2]), Matrix); end if:
    if s >= 2 and s <= m-1 then
      R[k]:= convert(blockmatrix(3, 2, [B1, B2, B3, B4, B5, B6]), Matrix); end
if:
      B[k+1]:= MatrixMatrixMultiply(A2[k], R[k]);
  printf("R[%d]=", k);
  print(R[k]);
  printf("A[%d]=", k+1);
  print(B[k+1]);
end do:
for l from 1 to n do
  e[l]:= Vector(1 .. n);
    for i from 1 to n do
      if i = l then e[l][i]:= 1; else e[l][i]:= 0; end if:
    end do:
  d[l][1]:= e[l];
  for k from 1 to n-1 do m:= n-k+1; s:= c[k];
    u[l][k]:= SubVector(d[l][k], 1); v[l][k]:= SubVector(d[l][k], 2 .. m);
    Y0[l][k]:= Vector(1 .. m);
    for z from 1 to m do Y0[l][k][z]:= 0; end do: Y0[l][k][s]:=
u[l][k][1]/A1[k][1, s];
      d[l][k+1]:= v[l][k]-MatrixVectorMultiply(A2[k], Y0[l][k]);
  end do:
Y0[l][n]:= d[l][n]/B[n][1, 1];

#Y Çözümleri Hesaplanıyor
  RR[0]:= Matrix(n, n, shape = identity); Y[l] := Y0[l][1];
  for i to n-1 do
    RR[i]:= R[i];
    RR[i]:= MatrixMatrixMultiply(RR[i-1], RR[i]);
    S[i]:= MatrixVectorMultiply(RR[i], Y0[l][i+1]); Y[l]:= Y[l]+S[i];
  printf("Y[%d]=", l);
  print(Y[l]);
  end do:
end do:

#Ters Matris Oluşturuluyor
C:= Y[1];
for j from 2 to n do C := convert(blockmatrix(1, 2, [C, Y[j]], Matrix)); end
do:
end proc:
```

```
> A:= Matrix([[1,2,0],[0,1,4],[-1,0,6]]);
```

$$A := \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 4 \\ 1 & 0 & 6 \end{bmatrix}$$

```
> Y:= tersbul(A);
```

$$A[1]=\begin{bmatrix}1&2&0\\0&1&4\\1&0&6\end{bmatrix},\ R[1]=\begin{bmatrix}2&0\\1&0\\0&1\end{bmatrix};\ A[2]=\begin{bmatrix}1&4\\2&6\end{bmatrix},\ R[2]=\begin{bmatrix}4\\1\end{bmatrix};\ A[3]=[2]$$

$$Y[1]=\begin{bmatrix}3\\2\\\frac{1}{2}\end{bmatrix},\ Y[2]=\begin{bmatrix}6\\3\\1\end{bmatrix},\ Y[3]=\begin{bmatrix}4\\2\\\frac{1}{2}\end{bmatrix},\ Y:=\begin{bmatrix}3&6&4\\2&3&2\\\frac{1}{2}&1&\frac{1}{2}\end{bmatrix}$$

```
> A:= Matrix([[1,2,0,3],[-2,1,1,0],[-3,6,0,1],[0,2,-3,4]]);
```

$$A:=\begin{bmatrix}1&2&0&3\\2&1&1&0\\3&6&0&1\\0&2&3&4\end{bmatrix}$$

```
> Y:= tersbul(A);
```

$$A[1]=\begin{bmatrix}1&2&0&3\\2&1&1&0\\3&6&0&1\\0&2&3&4\end{bmatrix},\ R[1]=\begin{bmatrix}2&0&3\\1&0&0\\0&1&0\\0&0&1\end{bmatrix};\ A[2]=\begin{bmatrix}5&1&6\\12&0&10\\2&3&4\end{bmatrix},\ R[2]=\begin{bmatrix}\frac{1}{5}&\frac{6}{5}\\1&0\\0&1\end{bmatrix}$$

$$A[3]=\begin{bmatrix}\frac{12}{5}&\frac{22}{5}\\\frac{17}{5}&\frac{8}{5}\end{bmatrix},\ R[3]=\begin{bmatrix}\frac{11}{6}\\1\end{bmatrix};\ A[4]=\begin{bmatrix}\frac{47}{6}\end{bmatrix}$$

$$Y[1]=\begin{bmatrix}\frac{19}{94}\\\frac{3}{47}\\\frac{16}{47}\\\frac{21}{94}\end{bmatrix},\ Y[2]=\begin{bmatrix}\frac{24}{47}\\\frac{15}{47}\\\frac{14}{47}\\\frac{18}{47}\end{bmatrix},\ Y[3]=\begin{bmatrix}\frac{7}{94}\\\frac{11}{47}\\\frac{4}{47}\\\frac{17}{94}\end{bmatrix},\ Y[4]=\begin{bmatrix}\frac{8}{47}\\\frac{5}{47}\\\frac{11}{47}\\\frac{6}{47}\end{bmatrix};\ Y:=\begin{bmatrix}\frac{19}{94}&\frac{24}{47}&\frac{7}{94}&\frac{8}{47}\\\frac{3}{47}&\frac{15}{47}&\frac{11}{47}&\frac{5}{47}\\\frac{16}{47}&\frac{14}{47}&\frac{11}{47}&\frac{5}{47}\\\frac{21}{94}&\frac{18}{47}&\frac{17}{94}&\frac{6}{47}\end{bmatrix}$$

KAYNAKLAR

GOLUB H, VAN LOAN CF, 1989. *Matrix Computation*, Baltimore, The John Hopkins University Press, 2nd Edn.

KESKİN T, AYDIN K, 2007. Iterative Decreasing Dimension Algorithm, *Computers & Mathematics with Applications*, 53(7) 1153-1158.

MIRSKY L, 1955. *An Introduction to Linear Algebra*, London, Oxford University Press, Amen House.

SHAMPINE LF, ALLEN Jr.RC., PRUESS S, 1997. *Fundamentals of Numerical Computing*, John Wiley & Sons Inc.