ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
Mühendislik Mimarlık Fakültesi
İnşaat Mühendisliği Bölümü
E-Posta: ogu.ahmet.topcu@gmail.com
Web: http://mmf2.ogu.edu.tr/atopcu

Bilgisayar Destekli

Nümerik Analiz

Ders notları 2014

Ahmet TOPÇU

m=n

Kare matrisin tersi *Birim matris*

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ . & . & \dots & . \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & \dots & \\ & 1 & \dots & \\ . & . & \dots & . \\ & & \dots & 1 \end{bmatrix} \rightarrow \underline{\mathbf{A}}\ \underline{\mathbf{A}}^{-1} = \underline{\mathbf{I}}$$

Kare matris

8

TERS MATRİS HESABI

GAUSS-JORDAN tekniği

Wilhelm **Jordan (**1842-1899)

8. TERS MATRİS HESABI

Bölüm 2 sayfa 27 de ters matrisin tanımı, özellikleri ve adjoint matris yöntemiyle sayısal bir hesap örneği verilmişti. $\underline{A}_{nxn}$ matrisi ve tersi $\underline{A}_{nxn}^{-1}$ tanım gereği

$$\underline{A}_{nxn}\,\underline{A}_{nxn}^{-1}=\underline{I}_{nxn} \tag{8.1}$$

bağıntısını sağlar. Ters matrisi $\underline{x}_{nxn}=\underline{A}_{nxn}^{-1}$ ile gösterirsek

$$\underline{A}_{nxn}\,\underline{x}_{nxn}=\underline{I}_{nxn} \tag{8.2}$$

$$\overset{\underline{A}}{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}} \overset{\underline{A}^{-1}}{\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ . & . & \dots & . \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}} = \overset{\underline{I}}{\begin{bmatrix} 1 & & \dots & \\ & 1 & \dots & \\ . & . & \dots & . \\ & & \dots & 1 \end{bmatrix}} \tag{8.3}$$

ifadesinden hesaplanacak $\underline{x}$ matrisi $\underline{A}$ nın tersi olur. Bu ifade n karşı taraflı ve n bilinmeyen vektörlü bir denklem sistemidir. $\underline{A}$ matrisi **GAUSS, DOOLİTTLE, CROUT** veya **CHOLESKY** metodlarından biri ile üçgen çarpanlarına ayrılarak $\underline{x}$ in her bir vektörü hesaplanabilir. Ancak bu tarz hesap yolu iyi değildir. Çünkü hem $\underline{A}$ nın hem de tersi olan $\underline{x}$ in aynı anda bilgisayar belleğinde tutulması gerekir. Gerekli bellek $2n^2$ dir.

Ters matrisin hesabı için **GAUSS, DOOLİTTLE, CROUT** veya **CHOLESKY** metotları yerine **GAUSS-JORDAN**[1] metodu kullanılır. **GAUSS-JORDAN** metodunda tüm işlemler $\underline{A}$ nın üzerinde yapılır, ters matris $\underline{A}$ nın üstüne depolanır. Dolayısıyla, diğer metotların yarısı kadar, sadece n^2 bellek gerekir. Bu nedenle, diğer metotlara nazaran daha çok işlem gerektirmesine rağmen, **GAUSS-JORDAN** metodu tercih edilir. Ters matris hesabı için iterasyon metotları uygun değildir, çünkü $\underline{A}$ seyrek veya bant matris olsa bile tersi daima tam dolu bir matristir.

GAUSS-JORDAN tekniği ile ters matris hesabı

GAUSS-JORDAN tekniğine göre $\underline{A}_{nxn}$ düzenli(tekil olmayan) matrisinin tersi birbirini izleyen transformasyonlar ile hesaplanır. $\underline{A}_{nxn}$ matrisi, her adımda bir tane olmak üzere, n adımda belirlenebilen $\underline{T}_1$, T_2, …, $\underline{T}_n$ transformasyon matrisleri ile soldan ardışık çarpılarak birim matrise dönüştürülür. nxn boyutlu transformasyon matrisleri öyle seçilirler ki

$$\underline{T}_n \dots \underline{T}_2\,\underline{T}_1\,\underline{A}=\underline{I}$$

olur. Buradan, Ters matrisin tanımı gereği

$$\underline{A}^{-1}=\underline{T}_n \dots \underline{T}_2\,\underline{T}_1$$

olduğu anlaşılır.

Transformasyon matrislerinin belirlenmesi:

$$\underline{A}_{nxn}=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

verilmiş olsun. $\underline{T}_1$, T_2, …, $\underline{T}_n$ transformasyon matrislerinin nasıl hesaplanacağı adım adım açıklanacaktır.

[1] Wilhelm **Jordan** (1842 – 1899), Alman: 1887 de yayınlandı.

1. adım: A matrisinin 1. kolonunun elemanları yardımıyla oluşturulan nxn boyutlu T_1 matrisi A_{nxn} matrisi ile soldan çarpılarak B matrisine dönüştürülür:

$$\underline{T}_1 = \left[\begin{array}{c|cccc} \frac{1}{a_{11}} & 0 & 0 & \dots & 0 \\ \hline -\frac{a_{21}}{a_{11}} & 1 & 0 & \dots & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 & \dots & 0 \\ . & . & . & \dots & . \\ -\frac{a_{n1}}{a_{11}} & 0 & 0 & \dots & 1 \end{array}\right] , \quad \underline{B} = \underline{T}_1\underline{A} = \left[\begin{array}{c|cccc} 1 & b_{12} & b_{13} & \dots & b_{1n} \\ \hline 0 & b_{22} & b_{23} & \dots & b_{2n} \\ 0 & b_{32} & b_{33} & \dots & b_{3n} \\ . & . & . & \dots & . \\ 0 & b_{n2} & b_{n3} & \dots & b_{nn} \end{array}\right] , \quad (a_{11} \neq 0)$$

Görüldüğü gibi, T_1 öyle seçilmiştir ki çarpım sonrası $B=T_1A$ matrisinin birinci kolonu birim matrisin birinci kolonuna dönüşmüştür. T_1 in özelliği nedeniyle B nin elemanları

$$b_{1j} = \frac{a_{1j}}{a_{11}} \qquad ve \quad b_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \ , \qquad (i \neq 1)$$

ile hesaplanmaktadır.

Açıklama: Bu formüllerden anlaşılacağı gibi, B nin 1. kolonunun hesaplanması gerekmez. b_{ij} elemanını hesaplamak için T_1 matrisinin i. satırındaki tüm elemanların A nın j. kolonundaki elemanlar ile çarpılıp toplanması da gerekmez. T_1 in i. satırındaki ilk eleman $-a_{i1}/a_{11}$, A nın j. kolonundaki ilk elemanla çarpılır ve bulunan değer a_{ij} ile toplanırsa b_{ij} bulunur. Gerçekte nxn boyutlu olan T_1 in, yapısı incelendiğinde, tamamının oluşturulmasına gerek olmadığı, sadece 1. kolonunun bir vektör olarak hesaplanmasının yeterli olduğu anlaşılır. Ayrıca, B nin 1. kolonu sonraki hesaplarda kullanılmayacağı için, $B=T_1A$ hesaplandıktan sonra T_1 in 1. kolonu B nin 1. kolonuna yazılarak saklanabilir. Bu açıklamanın amacı, gereksiz dört işlemin önlenmesine yöneliktir.

2. adım: B matrisinin elemanları yardımıyla T_2 hesaplanır, B ile soldan çarpılır C matrisi oluşur:

$$\underline{T}_2 = \left[\begin{array}{c|c|ccc} 1 & -\frac{b_{12}}{b_{22}} & 0 & \dots & 0 \\ \hline 0 & \frac{1}{b_{22}} & 0 & \dots & 0 \\ \hline 0 & -\frac{b_{32}}{b_{22}} & 1 & \dots & 0 \\ . & . & . & \dots & . \\ 0 & -\frac{b_{n2}}{b_{22}} & 0 & 0 & 1 \end{array}\right] , \quad \underline{B} = \underline{T}_2\underline{T}_1\underline{A} = \left[\begin{array}{c|c|ccc} 1 & 0 & c_{13} & \dots & c_{1n} \\ \hline 0 & 1 & c_{23} & \dots & c_{2n} \\ \hline 0 & 0 & c_{33} & \dots & c_{3n} \\ . & . & . & \dots & . \\ 0 & 0 & c_{n3} & \dots & c_{nn} \end{array}\right] , \quad (b_{22} \neq 0)$$

$$c_{2j} = \frac{b_{2j}}{b_{22}} \qquad ve \quad c_{ij} = b_{ij} - \frac{b_{i2}}{b_{22}} a_{1j} \ , \qquad (i \neq 2)$$

n. adım: Bu adımdan bir önceki adımda hesaplandığı varsayılan $N=T_{n-1} \dots T_2 T_1A$ matrisinin elemanlarından T_n belirlenir ve N ile soldan çarpılırsa birim matris oluşur.

$$\underline{T}_n = \left[\begin{array}{cccc|c} 1 & 0 & \dots & 0 & -\frac{n_{1n}}{n_{nn}} \\ 0 & 1 & \dots & 0 & -\frac{n_{2n}}{n_{nn}} \\ . & . & \dots & . & . \\ 0 & 0 & \dots & 1 & -\frac{n_{n-1}}{n_{nn}} \\ \hline 0 & 0 & \dots & 0 & \frac{1}{n_{nn}} \end{array}\right] , \quad \underline{I} = \underline{T}_n \dots \underline{T}_2\underline{T}_1\underline{A} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & .. & 0 \\ . & . & . & \dots & . \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \quad (n_{nn} \neq 0)$$

Böylece A yı birim matrise dönüştüren transformasyon matrisleri bulunmuş olur. Tanım gereği, A nın tersi

$A^{-1}=T_n \dots T_2\ T_1$

çarpımından hesaplanır.

a_{11}, b_{22}, … , n_{nn} sayılarına "**pivot eleman**", bu elemanların bulunduğu satır ve kolona "**pivot satırı**" ve "**pivot kolonu**" denir. Pivot kelimesi kilit, rol oynayan, yönlendiren olarak tercüme edilebilir. Pivot eleman hesabın kaderini belirler. Çünkü, transformasyon matrislerinin hesabında geçen pivot elemanların $a_{11}\neq 0$, $b_{22}\neq 0$, … , $n_{nn}\neq 0$ olması gerekir, aksi durumda hesaba devam edilemez. Pivot elemanın sıfır olması halinde, örneğin $b_{22}=0$ ise, $\underline{B}$ matrisinin 2. satırında pivot eleman aranır ve 2. kolon sonraki kolonlardan biri ile değiştirilir, pivot elemanın sıfırdan farklı olması sağlanır ve hesaba devam edilir. Hangi kolonun hangi kolon ile değiştirildiği bir yere kaydedilir. Çünkü iki kolonun yerinin değiştirilmesi sonuçta hesaplanan ters matrisin aynı nolu satırlarının değiştirilmesini gerektirir.

Sıfırdan farklı pivot elemanın bulunamaması durumunda; örneğin $\underline{T}_2$ nin hesaplanması sırasında $b_{22}=b_{23}=\ldots=b_{2n}=0$ ise; kolon değiştirmek bir işe yaramaz. Bunun anlamı; $\underline{A}$ matrisinin tekil, yani det $\underline{A}=0$ olduğu, $\underline{A}$ nın tersinin hesaplanamayacağı, $\underline{A}^{-1}$ in tanımsız olduğudur.

Pivot elemanın sıfırdan farklı, fakat mutlak değerce çok küçük olması da nümerik sorun yaratır. Transformasyon matrislerinin elemanları pivot elemana bölünerek hesaplandığından, pivot elemanın çok küçük olması bölümün çok büyük bir sayı olmasına, bununla çarpılan/toplanan sayıların daha da büyümesine ve yuvarlama hatalarına neden olur. Büyük matrislerde bu işlemler milyonlarca defa yapıldığından, yuvarlama hataları birike birike sonucun yanlış hesaplanmasına neden olur.

Yuvarlama hatalarını azaltmak için her adımda pivot aranır. Mutlak değerce en büyük sayı satırda aranır, bu sayı pivot olacak şekilde kolonların yeri değiştirilir.

Örnek:

$$\underline{A}=\begin{bmatrix}2 & 1 & 4\\ 2 & 2 & 2\\ 1 & 2 & 0\end{bmatrix},\quad \underline{A}^{-1}=?$$

1. adım: $\underline{T}_1$ i oluşturmadan önce 1. satırdaki mutlak değerce en büyük sayı aranır. Üçüncü kolondaki 4 sayısı pivot adayıdır. 1. kolon ile 3. kolona yer değiştirilir, $a_{11}=4$ pivot eleman olur. Sonra $\underline{T}_1$ oluşturularak $\underline{A}$ ile soldan çarpılır $\underline{B}$ matrisi hesaplanır. $\underline{B}$ nin 1. kolonu birim matrisin birinci kolonuna dönüşür.

1. ve 3. kolonun yerleri değiştirildi!

$$\underline{A}=\begin{bmatrix}4 & 1 & 2\\ 2 & 2 & 2\\ 0 & 2 & 1\end{bmatrix},\qquad \underline{T}_1=\begin{bmatrix}\frac{1}{4} & 0 & 0\\ -\frac{2}{4} & 1 & 0\\ -\frac{0}{4} & 0 & 1\end{bmatrix}=\begin{bmatrix}0.25 & 0 & 0\\ -0.5 & 1 & 0\\ 0 & 0 & 1\end{bmatrix},\quad \underline{B}=\underline{T}_1\underline{A}=\begin{bmatrix}1 & 0.25 & 0.5\\ 0 & 1.5 & 1\\ 0 & 2 & 1\end{bmatrix}$$

2.adım: $\underline{B}$ nin 2. satırındaki mutlak değerce en büyük sayı aranır. $b_{22}=1.5$ pivot elemandır. Kolon değiştirilmesi gerekmez. $\underline{T}_2$ oluşturulur $\underline{B}$ ile soldan çarpılır, $\underline{C}$ matrisi hesaplanır. $\underline{C}$ nin 2. kolonu birim matrisin ikinci kolonuna dönüşür.

$$\underline{T}_2=\begin{bmatrix}1 & -\frac{0.25}{1.5} & 0\\ 0 & \frac{1}{1.5} & 0\\ 0 & -\frac{2}{1.5} & 1\end{bmatrix}=\begin{bmatrix}1 & -0.166667 & 0\\ 0 & 0.666667 & 0\\ 0 & -1.333333 & 1\end{bmatrix},\quad \underline{C}=\underline{T}_2\underline{B}=\begin{bmatrix}1 & 0 & 0.333333\\ 0 & 1 & 0.666667\\ 0 & 0 & -0.333333\end{bmatrix}$$

3. adım: b_{33}=-0.333333 pivot elemandır. $\underline{T}_3$ oluşturulur, $\underline{C}$ ile soldan çarpılırsa $\underline{I}$ birim matrisi oluşur:

$$\underline{T}_3 = \begin{bmatrix} 1 & 0 & -\dfrac{0.333333}{-0.333333} \\ 0 & 0 & -\dfrac{0.666667}{-0.333333} \\ 0 & 0 & \dfrac{1}{-0.333333} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{bmatrix}, \qquad \underline{I} = \underline{T}_3 \underline{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ters matrisin hesabı: $\underline{A}^{-1}=\underline{T}_3\,\underline{T}_2\,\underline{T}_1$ den hesaplanır:

$$\underline{A}^{-1} = \underline{T}_3\underline{T}_2\underline{T}_1 = \begin{bmatrix} 1 & 0 & -\dfrac{0.333333}{-0.333333} \\ 0 & 0 & -\dfrac{0.666667}{-0.333333} \\ 0 & 0 & \dfrac{1}{-0.333333} \end{bmatrix} \begin{bmatrix} 1 & -\dfrac{0.25}{1.5} & 0 \\ 0 & \dfrac{1}{1.5} & 0 \\ 0 & -\dfrac{2}{1.5} & 1 \end{bmatrix} \begin{bmatrix} \dfrac{1}{4} & 0 & 0 \\ -\dfrac{2}{4} & 1 & 0 \\ -\dfrac{0}{4} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1.5 & 1 \\ 1 & -2 & 2 \\ -2 & 4 & -3 \end{bmatrix}$$

1. ve 3. satırların yerleri değişmeli!

1. ve 3. kolonun yerleri değiştirildiğinden 1. ve 3. satırın yerleri değiştirilmelidir:

$$\underline{A}^{-1} = \begin{bmatrix} -2 & 4 & -3 \\ 1 & -2 & 2 \\ 1 & -1.5 & 1 \end{bmatrix}$$

Kontrol: $\underline{A}\ \underline{A}^{-1}=\underline{I}$ olmalı:

$$\underline{A}\underline{A}^{-1} = \begin{bmatrix} 2 & 1 & 4 \\ 2 & 2 & 2 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} -2 & 4 & -3 \\ 1 & -2 & 2 \\ 1 & -1.5 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Açıklama: Teoriyi açıklamak amacıyla $\underline{T}_1$, $\underline{T}_2$, …, $\underline{T}_n$, $\underline{A}$, $\underline{B}$, $\underline{C}$, …, $\underline{N}$ gibi çok sayıda matris gerekli olmakla birlikte, programlama için bu matrislerin hemen hiçbirine gerek yoktur. Tüm transformasyon matrisleri için sadece n elemanlı bir vektör, kolon değişimlerini saklamak için n elemanlı bir vektör ve nxn boyutlu $\underline{A}$ matrisi yeterlidir. i. transformasyon sonrası $\underline{A}$ nın i. vektörünün birim matrisin i. vektörüne dönüşeceği bilindiğinden ve i. kolon sonraki hesaplarda gerekli olmadığından, $\underline{T}_i$ nin i. kolonu $\underline{A}$ nın i. kolonuna depolanabilir. Tüm işlemler $\underline{A}$ matrisi üzerinde yapılabilir, bellek harcanmamış olur.

Ters matris hesabı fazla dört işlem(yaklaşık $6n^3$) gerektirir, zorunlu olmadıkça hesabından kaçınılır. Zorunlu durumda GAUSS-JORDAN tekniği tercih edilir. Çünkü tüm diğer yöntemler iki kat daha fazla bellek gerektirir. Ayrıca, varsa, $\underline{A}$ nın bant ve simetrisinden yararlanılamaz; çünkü ters matris daima tam doludur.

Yukarıda verilen yöntemin i. adımında pivot eleman i. satırda aranmakta, gerekirse, kolonlara yer değiştirilmektedir. Buna satırda pivot arama denir. Satırda pivot arama yerine, i. kolonda da pivot aranabilir ve gerekirse satırlara yer değiştirilebilir. Buna kolonda pivot arama denir. Hatta, i. satır ve kolon ile n. satır ve kolonun sınırladığı matris içinde pivot eleman aranarak, gerekirse hem satırlara ve hem de kolonlara yer değiştirilir. Buna da tam pivot arama denir.

Tam pivot arama kullanılırsa yuvarlama hataları en aza indirgenir, ancak hesap süresi çok uzar, genelde tercih edilmez.

Ters matris ile denklem sistemi çözümü

$\underline{A}\underline{x}=\underline{b}$ denklem sisteminin çözümü ters matris hesaplandıktan sonra, teorik olarak, $\underline{x}=\underline{A}^{-1}\underline{b}$ çarpımından da bulunabilir. Ancak, çok fazla işlem gerektirdiğinden ve $\underline{A}^{-1}$ tam dolu olduğundan, bu çözüm yolu hiç bir zaman tercih edilmez.

Nümerik hesaplarda, zorunlu olmadıkça, ters matris hesabından kaçınılır!