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

$$\rightarrow \underline{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ . & . & \cdots & . \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Matris

Bant matris

$$\underline{A}_{10x10} = \begin{bmatrix} 26 & 3 & -5 & 7.7 & & & & & & \\ -3 & 16 & 12 & 8 & 3 & & & & & \\ 6 & 6.9 & 11.8 & 4 & 4.1 & 2 & & & & \\ & -6 & -6 & 11 & 0 & 5 & 4 & & & \\ & & 7 & 6 & 99 & -4 & -3 & 2 & & \\ & & & 9.1 & 9 & 44.7 & 1 & 5 & 2 & \\ & & & & 1 & 5 & 22 & 9 & 9 & 1 \\ & & & & & 3 & 3.3 & 11 & 8 & 4 \\ & & & & & & 8 & 3 & 11 & 2 \\ & & & & & & & 9 & 9 & 10 \end{bmatrix}$$

1

GİRİŞ

Matrisler, tanımlar

1. GİRİŞ: Matrisler, tanımlar

Nümerik analiz nedir?

Nümerik analiz, ya da diğer adıyla sayısal analiz, klasik matematiğin bir dalıdır. Matematiğin analitik çözüm (formül) üretemediği veya ürettiği çözümün uygulama açısından çok karmaşık olduğu durumlarda veya deneye-ölçüme dayalı problemlerde nümerik analiz yöntemlerine başvurulur. Temel ilkeler

- Karmaşık yerine basit
- Doğrusal olmayan yerine doğrusal
- Sonsuz bilinmeyen yerine sonlu bilinmeyen
- Kesin çözüm yerine yaklaşık çözüm

ile yetinmekten ibarettir.

Birkaç örnek:

1. $ax^2+bx+c=0$ ikinci derece polinomunun kökleri (denklemi sağlayan x değerleri) iki tane ve

$$x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}\,,\quad x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}$$

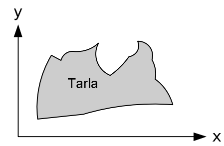
dir. Üçüncü ve dördüncü derece polinomların çözümü için de formüller vardır, fakat kullanışsızdır. n. dereceden olan $a_{0+}a_1\,x+a_2x^2+\ldots+\,a_n\,x^n\,=\,0$ denkleminin n tane kökü vardır. Fakat $n\geq 5$ durumunda x_1, x_2, …, x_n köklerinin formülü yoktur ve olamayacağı Norveçli matematikçi N. H. **Abel**[1] tarafından 1824 yılında ispatlanmıştır. Çözüm sadece nümerik yöntemlerle bulunabilir.

2. Klasik matematik integral hesabı için çok sayıda yöntem sunar, fakat

$$\int_a^b e^{-x^2}dx\,,\quad \int_a^b \frac{\mathrm{Sin}\,x}{x}dx\,,\quad \int_a^b \sqrt[3]{\mathrm{Sin}\,x}\,dx\,,\quad \int_a^b \mathrm{Sin}\,x^2dx,\quad \int_a^b \frac{1}{\ln x}dx,\quad \int_a^b \sqrt{1+x^4}\,dx,\quad \int_a^b \frac{dx}{\sqrt{1+x^4}}$$

gibi çok basit görünen bazı fonksiyonların integrali bilinen klasik yöntemlerle alınamaz. Tek çare nümerik çözümdür.

3. Sağdaki şekilde görülen tarlanın alanını integral alarak bulmak için tarlayı sınırlayan çevre eğrisinin fonksiyonunun bilinmesi gerekir. Fakat böyle bir fonksiyon yoktur. Çevre üzerinde alınan noktaların koordinatları ölçülür ve alan nümerik olarak hesaplanır. Ne kadar çok nokta ölçülürse alan o denli gerçeğe yakın olur.



4. $2x_1+6x_2=-32$
$-7x_1+4x_2=37$

gibi iki bilinmeyen ve iki doğrusal denklemden oluşan sistemi formüllerle çözerek $x_1=-7$ ve $x_2=-3$ bulabiliriz. Bilinmeyen ve denklem sayısı çok fazla ise, diyelim 1000000 denklem ve 1000000 bilinmeyen, formül kullanılamayacağı gibi elle çözüm için insan ömrü de yetmez.

5. İnşaat, makina, elektrik, elektronik, kimya gibi mühendislik dallarında, tıpta ve meteorolojide x, y, z koordinat eksenlerini ve t zaman değişkenini içeren kısmi türevli diferansiyel denklemlerin çözülmesi (zamana bağlı olarak cismin yer değiştirmesinin, şekil değiştirmesinin, iç kuvvet dağılımının belirlenmesi) matematikte bilinen diferansiyel denklem çözüm yöntemleri ile yapılamaz. Çözüm için nümerik yöntemler kullanılır. Modern yöntemler, problemin matematik modelinin kurulmasından çözümüne kadar yoğun olarak matrisler ile çalışır. Problem, bilinmeyen sayısı çok büyük olan bir doğrusal denklem sistemine dönüştürülür. Bu amaca yönelik nümerik yöntemlerden biri, günümüzde hemen her alanda yaygın olarak kullanılan, "Sonlu Elemanlar Metodu" dur.

[1] Niels Henrik **Abel** (1802 –1829) , Norveçli matematikçi

.Nümerik metotların dezavantajları

1. Nümerik analiz sayılar ile bir nevi oyun oynayarak çözüm üretmektir, denilebilir. Oyun genelde kazanılır, fakat kaybedilebilir de! Söylenmek istenen şudur: Çözüm genelde bulunur, ancak bulunamayabilir de.

2. Nümerik metotların çoğu belli bir hesap kuralının, belki yüzlerce binlerce kez, tekrarlanması ile adım adım sonuca varırlar(iterasyon). Bir tek sayının hesaplanabilmesi için binlerce hatta milyonlarca dört işlem (toplama, çıkarma, çarpma, bölme) gerekir. Bu nedenle el hesaplarına uygun değildir. Bilgisayar, uygun yöntem ve program kullanımı zorunludur. Direkt çözümler bile milyarlarca dört işlem gerektirir. Örneğin, n=10000 bilinmeyenli bir denklem sisteminin Gauss indirgeme yöntemi ile çözümünde yaklaşık $2n^3/3 \approx 7x10^{11}$ (700 milyar!) dört işlem vardır. Sayıların bu denli çok dört işleme tabi tutulması yuvarlama hatalarının giderek büyümesine neden olur. Peki, bu kadar büyük denklem sistemi uygulamada karşımıza çıkar mı? Evet, hatta çok daha büyükleri! Hemen söyleyelim n=374000000 (üç yüz yetmiş dört milyon) bilinmeyenli denklem sistemi günümüzde çözülebilmiştir. Hem de Türkiye'de: http://www.cem.bilkent.edu.tr/world_record

3. Nümerik çözüm yaklaşıktır, bir miktar kabul edilebilir hata içerir.

4. Aynı problemin çözümünde nümerik metotların bazıları sonuç verebilir, bazıları veremeyebilir.

5. Nümerik metotları öğrenmek ve çözüm üretmek için temel matematik ve mekanik bilgisi yanında bilgisayar, programlama ve paket yazılım kullanma becerisi de gerekir. Hem nümerik analiz hem de bilgisayar ve programlama dilleri yoğun olarak matrisler ile çalıştığından matris matematiği öğrenilmek zorundadır. Bir yapının nümerik metotlar ile hesabı için mukavemet, yapı statiği ve dinamiği gibi bilgiler de gerekir. Kısacası nümerik metotlar klasik bilgiler üzerine inşa edilmiş pratik bir çözüm yoludur, temel bilgiler olmadan bir işe yaramaz.

Tarihçe

Bugünkü bulgulara göre, nümerik analiz metotlarının tarihi yaklaşık 3650 yıl önce başlar. M.Ö. 1650 yıllarına ait bir papirüs (Rhind (veya Ahmes) papirüsü) basit bir denklemin kökünün nümerik çözümünü açıklamaktadır. Milat öncesinin en büyük matematikçisi kabul edilen Arşimet (Yaklaşık M.Ö 287- M.Ö. 212) eğrisel yüzeyli cisimlerin alanını, hacmini ve π sayısını hesaplamış, nümerik en küçük kavramını kullanmıştır.

M.Ö. 200 civarına ait Chiu Chang Suan Shu (aritmetik hakkında dokuz bölüm) adlı Çince kitapta aşağıdaki soru yer almaktadır: http://meyer.math.ncsu.edu/Meyer/PS_Files/GaussianEliminationHistory.pdf

Three sheafs of a good crop, two sheafs of a mediocre crop, and one sheaf of a bad crop are sold for 39 dou. Two sheafs of good, three mediocre, and one bad are sold for 34 dou; and one good, two mediocre, and three bad are sold for 26 dou. What is the price received for each sheaf of a good crop, each sheaf of a mediocre crop, and each sheaf of a bad crop?

Üç demet iyi kalite ekin mahsulü, iki demet orta kalite ekin mahsulü ve bir demet kötü kalite ekin mahsulü 39 dou'ya satılıyor. İki demet iyi, üç demet orta ve bir demet kötü 34 dou'ya; ve bir iyi, iki orta ve üç kötü 26 dou'ya satılıyor. Her bir iyi kalite, orta kalite ve kötü kalite ekin mahsulü için alınan fiyat nedir?

x iyi, y orta ve z kötü kalitedeki mahsulün birim fiyatı(dou) olmak üzere, cevabın üç bilinmeyenli denklem sisteminin çözümü olduğu anlaşılıyor:

$$\left.\begin{array}{l} 3x+2y+z=39 \\ 2x+3y+z=34 \\ X+2y+3z=26 \end{array}\right\} \text{Çözüm: } x=9.25,\ y=4.25,\ z=2.75 \text{ dou}$$

O çağda +, -, x, ÷ gibi dört işlem operatörleri, eşitlik işareti, değişken kavramı ve 0 rakamı henüz yoktu[2]. Çinliler bu doğrusal denklem sistemini, sayıları bambu çöpleri ile temsil ederek, çözmüşlerdi. Günümüzde Gauss indirgeme yöntemi veya benzeri birçok yöntem çözüm için kullanılmaktadır.

[2] Sıfır rakamını ilk kez 628 yılında Hintli matematikçi **Brahmagupta** (598-668) kullanmıştır. + ve − işareti 1489, = işareti 1557, x işareti 1628, ÷ işareti 1659 yıllarında kullanılmaya başlamıştır.

Nümerik analizin temellerini atanlar

John **Napier** (1550-1617), Isaac **Newton** (1643-1727), Gottfried **Leibniz** (1646–1716), Leonhard **Euler** (1707-1783), Joseph-Louis **Lagrange** (1736-1813), Pierre-Simon-Marquis de **Laplace**(1749-1827), Karl Friedrich **Gauss** (1777-1855), James Joseph **Sylvester**(1814 -1897).

Nümerik analiz yöntemlerinde adı sıkça geçenler

Niels Henrik **Abel** (1802-1829), Alexander Craig **Aitken** (1895 - 1967), Bernard **Bolzano**(1781-1848) George **Boole**(1815-1864), Isaac **Barrow**(1630-1677), Leonard **Bairstow**(1880-1963), Augustin Louis **Cauchy**(1789-1857),
Pafnuty Lvovich **Chebyshev**(1821-1894) , Prescott Duran **Crout** (1907-1984), Mayric Hascall **Doolittle**(1830-1913), André-Louis **Cholesky**(1875-1918), Gabriel **Cramer**(1704-1752), Roger **Cotes**(1682-1716), Arthur **Cayley**(1821-1895), Ferdinand Georg **Frobenius**(1849-1917), Leonardo **Fibonacci**(1170-1250), Joseph **Fourier**(1768-1830), Jørgen Pedersen **Gram**(1850-1916), Hermann **Grassmann**(1809-1877), Charles **Hermite**(1822-1901), David **Hilbert**(1862-1943), Alston Scott **Householder**(1904–1993), C. A. R. **Hoare**(1934 -), Carl Gustav Jacob **Jacobi**(1804-1851), Marie Ennemond Camille **Jordan**(1838-1922), Wilhelm **Jordan**(1842-1899), Martin Wilhelm **Kutta**(1867-1944) , Joseph-Louis **Lagrange** (1736 –1813), Adrien Marie **Legendre**(1752-1833), Cornelius **Lanczos**(1893-1974), Richard Edler von **Mises**(1883-1953), David E. **Müller**(1924 -), Lewis Fry **Richardson**(1881-1953), Carl David Tolmé **Runge**(1856-1927), Werner **Romberg**(1909-2003), Thomas **Simpson**(1710-1761), Ludwig von **Seidel**(1821-1896), Erhard **Schmidt**(1876-1959), Johan Frederik **Steffensen** (1873–1961), James Hardy **Wilkinson(1919-1986)**.

1930-1945 li yıllarda ilk bilgisayarların ortaya çıkmasıyla nümerik analiz yöntemleri önem kazanmaya başladı. Modern nümerik analizin 1947 yılında John **Neumann** ve Herman **Goldstine** tarafından yayınlanan "Numerical Inverting of Matrices of High Order , Bulletin of the AMS, Nov. 1947" ile başladığı kabul edilir. Bu yayında yuvarlama hataları ilk kez araştırılmıştır. Günümüzde kullanılan nümerik analiz yöntemlerinin çoğu 1950-1970 yıllarında geliştirilmiştir.

Bilgisayarların ve programlama dillerinin giderek gelişmesi, devrim yaratan ve günümüz vazgeçilmez nümerik metodu olan, "Sonlu Elemanlar Metodu-SEM" nun 1960 lı yıllarda doğmasına neden oldu. Bu konudaki ilk yayın: "M. J. **Turner**, R. W. **Clough**, H. C. **Martin** and L. J. **Topp**, "Stiffness and Deflection Analysis of Complex Structures," J. of Aero. Sci., 23 (9), Sept. 1956". Finite Element Method(Sonlu Elemanlar Metodu) adının ilk kullanıldığı yayın: "**Clough**, Ray W.: The finite element method in plane stress analysis. Proceedings, 2nd Conference on Electronic Computation, A.S.C.E. Structural Division, Pittsburgh, Pennsylvania, Sept. 1960". İlk Sonlu elemanlar kitabı: O. C. **Zienkiewicz,** The Finite Element Method, 1967.

Sonlu Elemanlar Metodunun ilkleri

Lord **Rayleigh**(1842-1919), Boris Grigorievich **Galerkin(**1871-1945), Walther **Ritz**(1878-1909), Richard **Courant(**1888-1972), Alexander **Hrennikoff**(1896-1984), John **Argyris(**1913-2004), James Hardy **Wilkinson(1919-1986),** Olgierd Cecil **Zienkiewicz(**1921-2009), Ray W. **Clough**, 1920-), William **Prager**(1903-1980).

İnşat mühendisliği yapı statiği dersinin bilgisayar öncesi vazgeçilmez nümerik metotları:

Cross metodu: Hardy **Cross**(1885-1959), Kani Metodu: Gaspar **Kani(**1910-1968),

Ders notunun kapsamı ve amacı

İnşaat mühendisliği lisans ve lisansüstü programlarında verilen derslerin birleştirilmesi ve gözden geçirilmesiyle oluşan bu ders notları öncelikle inşaat mühendisliği dalında okuyan öğrencilere yöneliktir. Bu bir kısıtlama ya da diğer dalların işine yaramaz anlamında değildir. Yoğun teori yerine öz ve pratik bilgiler içerir: Matris cebiri, ters matris, denklem sistemi çözümü, en küçük kareler metodu, standart ve genel özdeğer problemi. Ayrıca, doğrusal olmayan fonksiyonların kökleri, Min-Max belirleme, eğri uydurma, belirli integral, sıralama-arama teknikleri konularında yazarın hazırladığı veya başka kaynaklardan edinilmiş çok sayıda, BASIC programlama dilinde kodlanmış programa ve test örneklerine yer verilmiştir.

Açık kodlu bu programların bazılarını lisansüstü öğrencilerinin istedikleri bir güncel görsel programlama diline çevirmeleri, gerekli teorik bilgileri araştırmaları istenmektedir. Bunun amacı; öğrenciyi araştırmaya, yoğun bilgisayar kullanmaya, görsel bir programlama dilini öğrenmeye ve nihayet her zaman kullanabileceği kendi eseri nümerik analiz kitapçığını oluşturmaya zorlamaktır. Öğrencinin ayrıca kendi sonuçlarını, sadece bu ders notlarındaki sonuçlar ile değil, **Mathematica**, **MATLAB**, **Mathcad**, **Maple**, **REDUCE**, **Sage** gibi profesyonel yazılımlardan biri ile de doğrulaması istenmektedir. Bundan amaç, öğrencinin konuya yönelik profesyonel yazılımları tanıması ve kullanım becerisi edinmesini sağlamaktır.

Ders notlarında zaman zaman iyileştirme yapılmaktadır. Güncel olanı aşağıdaki adresten indirilebilir: http://mmf2.ogu.edu.tr/atopcu/index_dosyalar/BilgisayarDestekliNumerikAnaliz.htm

MATRİSLER

Güncel yaşamımızda tablolar veya çizelgeler kullanırız. Örnek: Satranç tahtası, ders çizelgesi, yıllık takvim, bir ayın günleri, spor toto ve sayısal loto kuponu gibi. Tablonun satırları ve kolonları vardır. Bir satır ve kolonun kesiştiği noktada bir nesne (eleman) vardır.

Satranç tahtası 8 satırlı ve 8 kolonlu bir tablodur, Her satır ve kolonun kesiştiği noktada bir eleman (fil, kale, at, piyon, boş, …) vardır. Bu ve benzeri tablolara matematikte matris adı verilir. Bu ders kapsamında matrislerin elemanları genelde sabit sayılardan ve değişkenlerden oluşacaktır. O halde; matris birçok sayının veya değişkenin bir araya getirildiği bir tablodur, bir şemadır, matrisin sayısal bir değeri yoktur.

Bilgisayarların 1945 li yıllardan itibaren gelişimine paralel olarak, gerçekte asırlar önce temelleri atılmış olan sayısal yöntemler, özellikle, 1960 lı yıllarda yeniden ele alınarak geliştirilmişlerdir. Bu yöntemlerin gerektirdiği milyonlarca verinin derli-toplu ifadesinin, teori ve bilgisayarda işlenmesinin en modern ve kolay yolu matris[3] notasyonunu kullanmaktır.

Matris notasyonu

m bilinmeyen ve n denklemli doğrusal bir denklem sistemi klasik matematikte

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m &= b_2 \\ &\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m &= b_n \end{aligned}$$

$$n = m = 3 \text{ için Örnek:} \quad \begin{aligned} 16x_1 - 2x_2 + 1.5x_3 &= -9.7 \\ 6x_1 + 22.5x_2 + 5.6x_3 &= 18.3 \\ 2.2x_1 - 2x_2 + 15.4x_3 &= 23.8 \end{aligned} \qquad (1.1)$$

şeklinde yazılır. Burada x_1, x_2, ..., x_m bilinmeyenleri, b_1, b_2, ..., b_n denklemlerin karşı tarafındaki terimleri, a_{ij} terimi i. denklemdeki x_j bilinmeyeninin katsayısını göstermektedir. Aynı türden olan büyüklükleri tablo(matris) olarak bir araya getirirsek

a_{ij} katsayılarının tablosu(matrisi): $\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ . & . & \ldots & . \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix}$ Örnek: $\begin{bmatrix} 16 & -2 & 1.5 \\ 6 & 22.5 & 5.6 \\ 2.2 & -2 & 15.4 \end{bmatrix}$

Bilinmeyenlerin tablosu(matrisi): $\begin{bmatrix} x_1 \\ x_2 \\ . \\ x_m \end{bmatrix}$ Örnek: $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Karşı taraftaki terimlerin tablosu(matrisi): $\begin{bmatrix} b_1 \\ b_2 \\ . \\ b_n \end{bmatrix}$ Örnek: $\begin{bmatrix} -9.7 \\ 18.3 \\ 23.8 \end{bmatrix}$

1.1 bağıntısı

Katsayılar matrisi *Bilinmeyenler matrisi(vektör)* *Karşı taraf matrisi(vektör)*

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ . & . & \ldots & . \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_m \end{bmatrix}$$

Örnek:

$$\begin{bmatrix} 16 & -2 & 1.5 \\ 6 & 22.5 & 5.6 \\ 2.2 & -2 & 15.4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -9.7 \\ 18.3 \\ 23.8 \end{bmatrix} \qquad (1.2)$$

[3] Matris kelimesi ilk kez 1850 yılında İngiliz James Joseph **Sylvester** (1814-1897) tarafından kullanıldı.

Şeklinde yazılabilir. 1.1 ile 1.2 in aynı anlamda olabilmesi için bir çarpım kuralı koymak gerekir: **Katsayılar matrisinin bir satırındaki terimler bilinmeyenler matrisindeki terimler ile sırasıyla çarpılıp toplanacaktır. Bu toplam karşı taraf matrisinin aynı satırındaki değere eşitlenecektir.**

1.2 bağıntısı 1.1 denklem sisteminin matris notasyonunda yazılmış açık şeklidir. Daha az yer tutması ve daha kısa yazılması için matrislere ad verilir:

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_m \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_n \end{bmatrix} \qquad (1.3)$$

Bu tanımlamaya bağlı olarak 1.2 bağıntısı kısaca

$$\underline{A}\ \underline{x}=\underline{b} \qquad (1.4)$$

Şeklinde yazılır. Bu bağıntı 1.1 bağıntısındaki doğrusal cebirsel denklem sisteminin matris notasyonunda yazılmış en kısa şeklidir. $\underline{A}$ denklem sisteminin katsayılar matrisi, $\underline{x}$ bilinmeyenler matrisi(vektörü), $\underline{b}$ karşı taraf matrisi(vektörü) adını alır. Dikkat edilirse; $\underline{A}$, $\underline{x}$ ve $\underline{b}$ büyüklüklerinin altı çizilidir. Bunun nedeni, matrisleri sayısal büyüklüklerden ayırabilmektir. $\underline{A}$ bir matris, A ise bir sayıdır.

Matrisleri sayısal büyüklüklerden ayırabilmek için kaynaklarda farklı farklı gösterim kullanılır. Bazı yazarlar matrislerin altını çizer, bazıları koyu harf kullanır, bazıları köşeli parantez, bazıları yuvarlak parantez kullanır. Aşağıda sıkça kullanılan notasyonlar verilmiştir:

$$\underline{A} = \underline{A}_{nxm} = \mathbf{A} = \mathbf{A}_{nxm} = [A] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} = (A) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

nxm indisine matrisin boyutu denir. Matrisin n satırlı, m kolonlu ve nxm elemanlı olduğunu belirtir. a_{ij}, matrisin i. satır ve j. kolonundaki elemandır. Bu ders notunda, yazım ve çizim kolaylığı nedeniyle, matrisler altı çizilerek ve köşeli parantez kullanılarak vurgulanacaktır. Gerekli olduğunda matrisin boyutu indis olarak yazılacaktır.

Matris tipleri

Herhangi bir matrisin elemanları gerçek(reel) ve karmaşık(kompleks) sayılardan oluşabilir. Bu ders notu kapsamında, aksi söylenmedikçe, sadece gerçek elemanlı matrisler ele alınacaktır. Uygulamada sıkça karşılaşılan matris tipleri ve özellikleri aşağıda özetlenmiştir.

Dikdörtgen matris: Satır ve kolon sayısı farklı, nxm elemanı olan matristir.

$$\underline{A}_{nxm} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \qquad \textit{Örnek:} \quad \underline{A} = \begin{bmatrix} 7 & -5 & -6.9 \\ 6.1 & 0 & 13 \end{bmatrix}$$

Kare matris: n satır, n kolon ve nxn elemanlı matristir.

$$\underline{A}_{nxn} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & \dots & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \qquad \textit{Örnek: } \underline{A} = \begin{bmatrix} 7 & -5 & 6.9 \\ 6.1 & 0 & 13 \\ 22.6 & 5.5 & 10 \end{bmatrix}$$

Satır matrisi veya satır vektörü: Tek bir satırı, n kolonu ve n elemanı olan matristir. Satır vektörü de denir.

$$\underline{A}_{1xn} = \begin{bmatrix} a_{11} & a_{12} & . & a_{1n} \end{bmatrix} \qquad \textit{Örnek: } \underline{A} = \begin{bmatrix} -9.73 & 0.964 & -5.55 & 8 \end{bmatrix}$$

Kolon matris veya kolon vektörü: Tek bir kolonu, n satırı ve n elemanı olan matristir. Kolon vektörü de denir. Yer kazanmak amacıyla { ve } parantezleri kullanılarak satır şeklinde de yazılır. Burada dikkat edilmesi gereken nokta { ve } parantezleri kullanılarak yazılmış matrisin bir satır matrisi değil kolon matrisi olduğudur.

$$\underline{A}_{nx1} = \begin{bmatrix} a_{11} \\ a_{21} \\ . \\ a_{n1} \end{bmatrix} = \{a_{11} \quad a_{21} \quad . \quad a_{n1}\} \qquad \textit{Örnek: } \underline{A} = \begin{bmatrix} 46.32 \\ 9 \\ -109.4 \\ 86 \end{bmatrix} = \{46.32 \quad 9 \quad -109.4 \quad 86\}$$

Diyagonal (köşegen) matris: Bir kare matrisin satır ve kolon numaraları eşit olan a_{11}, a_{22}, …, a_{nn} elemanlarına diyagonal veya köşegen eleman ve bu elemanların yer aldığı hatta ana diyagonal veya ana köşegen denir. Köşegen elemanları dışındaki tüm elemanları sıfır olan matrise diyagonal veya köşegen matris denir. Hepsi olmamak kaydıyla, Köşegen elemanlardan bazıları da sıfır olabilir. Yer kazanmak ve köşegen dışındaki sıfırları yazmamak için diag[…] veya ⌈…⌋ şeklinde de yazılır. Kaynaklarda genellikle D harfi ile gösterilir.

$$\underline{D}_{nxn} = \begin{bmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \dots & 0 \\ . & & . & . \\ 0 & 0 & \dots & d_{nn} \end{bmatrix} = diag\begin{bmatrix} d_{11} & d_{22} & . & d_{nn} \end{bmatrix} = \lceil d_{11} \ d_{22} \dots \ d_{nn} \rfloor$$

Ana köşegen

$$\textit{Örnek: } \underline{D} = \begin{bmatrix} -33.3 & 0 & 0 & 0 \\ 0 & 1.9 & 0 & 0 \\ 0 & 0 & 72 & 0 \\ 0 & 0 & 0 & 0.87 \end{bmatrix} = diag\begin{bmatrix} -33.3 & 1.9 & 72 & 0.87 \end{bmatrix} = \lceil -33.3 \ \ 1.9 \ \ 72 \ \ 0.87 \rfloor$$

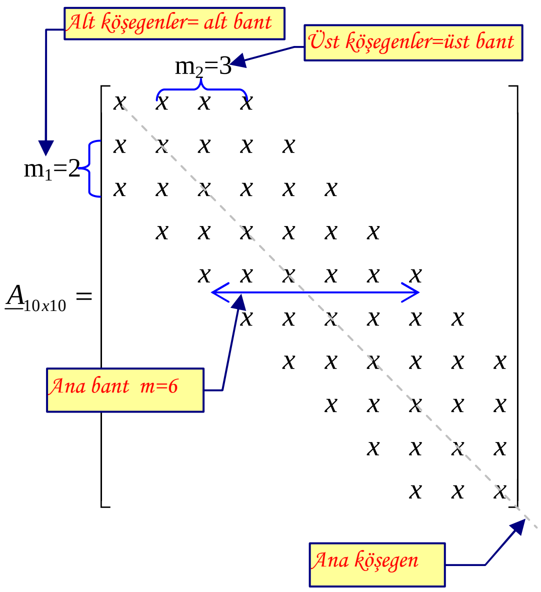
Alt üçgen matris: Diyagonalin üstündeki tüm elemanları sıfır olan kare matristir. Tümü olmamak kaydıyla, diyagonal ve altındaki elemanlardan bazıları da sıfır olabilir. Kaynaklarda çoğunlukla L (Lower) harfi ile gösterilir.

$$\underline{L}_{nxn} = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ . & & \dots & . \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \qquad \text{Örnek : } \underline{L} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 5.5 & 13 & 0 & 0 \\ 16 & 7 & -7 & 0 \\ -4 & 22 & 0 & 13.2 \end{bmatrix}$$

Üst üçgen matris: Diyagonalin altındaki tüm elemanları sıfır olan kare matristir. Tümü olmamak kaydıyla, diyagonal ve üstündeki elemanlardan bazıları da sıfır olabilir. Kaynaklarda çoğunlukla U (Upper) harfi ile gösterilir.

$$\underline{U}_{nxn} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ . & . & \dots & . \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \qquad Örnek: \ \underline{U} = \begin{bmatrix} 3 & 5.5 & 16 & -4 \\ 0 & 13 & 7 & 22 \\ 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 13.2 \end{bmatrix}$$

Bant matris: Matrisin sıfırdan farklı elemanları ana köşegen civarında toplanmış matristir. Ana köşegenine paralel köşegenlerine yan köşegen denir. Ana köşegenin altındakilere alt köşegen, üstündekilere üst köşegen de denir. m_1 alt ve m_2 üst köşegeni olan bir matrisin Ana bant genişliği $m=m_1+m_2+1$ olur. m_1 e alt bant m_2 ye üst bant genişliği de denir. Ana bant dışındaki tüm elemanları sıfır olan matrislere bant matris denir. Aşağıdaki şekilde 10x10 boyutlu, $m_1=2$ alt ve $m_2=3$ üst bant genişlikli bir matris ve örneği görülmektedir. Bant içindeki her bir eleman x ile gösterilmiştir. Ana bant genişliği $m=m_1+m_2+1=2+3+1=6$ ve $i>j+m_1$ veya $j>i+m_2$ için $a_{ij}=0$ dır. Bant dışındaki sıfır elemanlar yazılmamıştır.



Örnek:

$$\underline{A}_{10x10} = \begin{bmatrix} 26 & 3 & -5 & 7.7 & & & & & & \\ -3 & 16 & 12 & 8 & 3 & & & & & \\ 6 & 6.9 & 11.8 & 4 & 4.1 & 2 & & & & \\ & -6 & -6 & 11 & 0 & 5 & 4 & & & \\ & & 7 & 6 & 99 & -4 & -3 & 2 & & \\ & & & 9.1 & 9 & 44.7 & 1 & 5 & 2 & \\ & & & & 1 & 5 & 22 & 9 & 9 & 1 \\ & & & & & 3 & 3.3 & 11 & 8 & 4 \\ & & & & & & 8 & 3 & 11 & 2 \\ & & & & & & & 9 & 9 & 10 \end{bmatrix}$$

Bant matris

Üçlü köşegen matris: Bir alt ve bir üst köşegeni olan bant matristir. Uygulamada doğrudan karşılaşılabildiği gibi tam dolu matrisler bir takım işlemler sonucu bazen üçlü köşegen matrise dönüştürülür. Aşağıdaki örnekte gösterilmeyen elemanlar sıfırdır.

$$\underline{A}_{8x8} = \begin{bmatrix} x & x & & & & & & \\ x & x & x & & & & & \\ & x & x & x & & & & \\ & & x & x & x & & & \\ & & & x & x & x & & \\ & & & & x & x & x & \\ & & & & & x & x & x \\ & & & & & & x & x \end{bmatrix} \qquad Örnek: \ \underline{A}_{8x8} = \begin{bmatrix} 12 & 8 & & & & & & \\ -8 & 20 & 6 & & & & & \\ & 3 & 18 & -3 & & & & \\ & & 0 & 33 & 7 & & & \\ & & & -4 & 40 & 0 & & \\ & & & & 9 & 22 & 2.7 & \\ & & & & & 8 & 20 & 9.1 \\ & & & & & & 5 & 10 \end{bmatrix}$$

Seyrek matris: Uygulamada karşılaşılan matrisler çok büyük boyutlu, 1000000x1000000, 100000000 x100000000 gibi, fakat çoğu kez çok seyrek elemanlı olurlar. Elemanların %95-99 u sıfırdır. Bu matrislere seyrek (İngilizce: Sparse) matris denir. Bu tür matrisler ile çalışılırken, çok özel yöntemler kullanılarak, sıfırların depolanmasından ve sıfır sayısı ile gereksiz dört işlem yapmaktan kaçınılır. Aşağıdaki örnekte sıfır elemanlar yazılmamıştır.

$$\underline{A}_{10x10} = \begin{bmatrix} & & 2 & & & & & & & \\ 6 & & & & & & & & 6 & \\ & & 1 & 2 & & & & & & \\ & & & & & & 7 & & & \\ & 2 & & & 4 & & & & & \\ & & & & & 8 & & & & \\ & & & & & & & -9 & & 2 \\ -6 & & & 3 & 3 & & & & & \\ & & & & & & & 3 & & \\ -4 & & & & & & & & & 1 \end{bmatrix}$$

Seyrek matris

Sıfır matris ve birim matris: Tüm elemanları 0 (sıfır) olan matrise sıfır matris, diyagonal elemanları 1 (bir) olan *kare* matrise de birim matris denir. Genellikle, sıfır matris O ile, birim matris I ile gösterilir. Aşağıdaki birim matrisin sıfırları yazılmamıştır.

Örnek: $\underline{O} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ *Sıfır matris* $\qquad \underline{I} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$ *Birim matris*

Bölünmüş matris ve alt matris: Bir $\underline{A}_{nxm}$ matrisi yatay ve düşey çizgilerle daha küçük matrislere bölünebilir. Bu durumda A matrisi elemanları da matris olan bir matris olur. Bu elemanlara alt matris, matrisin kendisine de bölünmüş veya bloklanmış matris denir.

Örnek: $\underline{A} = \left[\begin{array}{c|c|c} -1 & 0 \quad 0 & 0 \\ \hline 0 & 1 \quad 2 & 3 \\ 0 & 1 \quad -2 & 3 \end{array}\right] = \begin{bmatrix} \underline{a}_{11} & \underline{a}_{12} & \underline{a}_{13} \\ \underline{a}_{21} & \underline{a}_{22} & \underline{a}_{23} \end{bmatrix}$ *Bölünmüş matris* *Alt matris*

Alt matrisler:

$\underline{a}_{11} = [-1]$, $\underline{a}_{12} = [0 \quad 0]$, $\underline{a}_{13} = [0]$

$\underline{a}_{21} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\underline{a}_{22} = \begin{bmatrix} 1 & 2 \\ 1 & -2 \end{bmatrix}$, $\underline{a}_{23} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$

Matris neden bölünür?

- Çok sayıda sıfırları alt matrislere toplamak ve sıfır alt matrisler ile işlemleri önlemek için.
- Sıfır alt matrisleri depolamamak için.
- Matris işlemlerini eleman- eleman değil, blok-blok yapabilmek için.
- Çok büyük matrislerin işlemlerini taksit-taksit yapabilmek için.

Çok büyük matrisleri depolamak için hiçbir bilgisayarın ana belleği yetmez. Büyük matris uygun alt matrislere bölünür, alt matrisler çevre birimlerinde (hard disk) depolanır. Ana belleğe sığacak kadar alt matris hard diskten okunur, matris işlemleri taksit-taksit yapılır.

Yılın günlerini gösteren takvim bölünmüş matrise ve alt matrislere bir örnektir. Her ay bir alt matristir.

Bölünmüş matris

Alt matris

Büyük matris nedir?

Basit bir tanımla, büyük matris kullanılan bilgisayarın, işletim sisteminin ve programlama dilinin üstesinden gelemediği matristir(bilgisayarın çok belleği olması tamamının kullanılabileceği anlamına gelmez. İşletim sistemi ve programa dili kullanılabilir bellek miktarını sınırlar). 1950 li yıllarda 20x20 boyutlu matris büyük sayılırdı. 1960 lı yıllarda 100x100 boyutlu, 1970 li yıllarda 1000x1000 boyutlu, 1980 li yıllarda 10000x10000 boyutlu, 1990 lı yıllarda 100000x100000 boyutlu matris büyük idi. Günümüzde 1000000x1000000 boyutlu matris büyük, 100000000x100000000 boyutlu matris çok büyüktür denilebilir.

Matris büyüdükçe bellek ve işlemci gücü yetmez olur, yuvarlama hataları büyür, sonuç güvensiz olur. Çok büyük denklem sistemlerini çözmek için çok özel çözüm yöntemlerine, çok özel(yüzlerce-binlerce işlemcili, birkaç terabyte ana bellekli) ve pahalı(birkaç milyon dolar) bilgisayara gereksinim olur.

Çok büyük denklem sistemi ile çok özel alanlarda karşılaşılır: uzay, fizik, tıp, biyoloji, meteoroloji, askeri, benzetim araştırmaları.

374 milyon bilinmeyenli denklem sistemi(matrisin boyutu: 374000000x374000000) Bilkent üniversitesinde 2009 yılında çözülerek dünya rekoru kırılmıştır: http://www.cem.bilkent.edu.tr/world_record

Kullanılabilir ana belleği 4 Gigabyte varsayımıyla, 1000-2000 $ fiyatındaki bir dizüstü bilgisarla kaç denklem çözülebilir? Yaklaşık bir hesap yapalım. Bir Gigabyte belleğin donanım, işletim sistemi ve yazılımlar tarafından kullanıldığını varsayarsak, matrisi depolamak için 3 Gigabyte boş ana bellek kalır. 1 Gigabyte=bir milyar byte demektir, boş bellek=3 milyar byte=$3x10^9$ byte. Ondalık bir sayıyı bilgisayarda depolamak için en az 8 byte kullanılır. Depolanabilecek ondalık sayı sayısı=$3x10^9/8=375x10^6$ adet olur. Depolanabilecek matrisin boyutu nxn=$375x10^6$ → n≈19000 civarında olacaktır.

19000x19000 boyutlu matrisin tamamı ana bellekte depolanabilir. Matris simetrik ise, yarısının depolanması, bant ve simetrik ise sadece yarı bantın depolanması yeterli olduğundan matris çok daha büyük olabilir. Bölünmüş ve alt matris tekniği ve hard disk kullanılması durumunda çözülebilecek denklem sayısı daha da artar. Ancak, hatırlatmakta yarar var: Matris büyüdükçe ondalık sayı için sadece 8 byte kullanmak yuvarlama hatalarını artırır, azaltmak için 10-16 byte kullanmak gerekecektir.

Her biri 100 m^2 4 daireli ve 10 katlı bir apartmanın Sonlu Elemanlar Metodu ile statik hesapları yaklaşık 5000-20000 bilinmeyenli bir denklem sisteminin çözümünü gerektirir, bir dizüstü ile rahatlıkla çözülebilir. Bundan şu sunuca varabiliriz. Günümüzde kullanılan 1000-2000 dolarlık dizüstü bilgisayar 1970-1980 lı yılların bir kaç milyon Amerikan dolarlık süper bilgisayarlarından çok daha güçlüdürler.

Simetrik matris ve ters simetrik matris: Bir kare matrisin tüm elemanları için $a_{ij}=a_{ji}$ ise bu matrise simetrik matris denir. Bir kare matriste i≠j için $a_{ij}=-a_{ji}$ ise ve $a_{ii}=0$ ise bu matrise ters simetrik matris denir.

Örnek: $$\underline{A}=\begin{bmatrix}1&2&3&-3\\2&1&0&0\\3&0&5&-1\\-3&0&-1&7\end{bmatrix}=\begin{bmatrix}1&2&3&-3\\&1&0&0\\\text{Sim}&&5&-1\\&&&7\end{bmatrix},\quad \underline{A}=\begin{bmatrix}0&6&3\\-6&0&5\\-3&-5&0\end{bmatrix}$$

Simetrik matris

Ters simetrik matris

Simetrik bant matris: Uygulamada sıkça karşılaşılan bir matris tipidir. Sonlu elemanlar metodunun denklem sisteminin katsayılar matrisi bu türdendir. Bu nedenle çok önemlidir.

Alt bant genişliği $m_1=2$ *Üst bant genişliği* $m_2=2$ *Yarı bant genişliği* $b=3$ *Ana bant genişliği=5*

$$\underline{A}=\begin{bmatrix}23&5&-1&&&&&\\5&20&6&-4&&&&\\-1&6&18&3&-3&&&\\&-4&3&42&11&3&&\\&&-3&11&33&4&2&\\&&&3&4&19&0&4\\&&&&2&0&10&-4\\&&&&&4&-4&16\end{bmatrix}=\begin{bmatrix}23&5&-1&&&&&\\&20&6&-4&&&&\\&&18&3&-3&&&\\&&&42&11&3&&\\&&&&33&4&2&\\\mathit{Sim}&&&&&19&0&4\\&&&&&&10&-4\\&&&&&&&16\end{bmatrix}$$

Simetrik bant matrisin üst ve alt köşegen sayısı aynıdır, $m_1=m_2$. Ana bant genişliği $m=2m_1+1$ dir. $b=m_1+1$ değerine özel olarak yarı bant genişliği denir. Yukarıdaki örnekte $m_1=m_2=2$, bant genişliği m=5 ve yarı bant genişliği b=3 tür. Simetrik bant matrislerin sadece yarı bant genişliği içinde kalan elemanları bilgisayarda depolanarak bellekten, işlem sayısından ve hesap zamanından tasarruf edilir.

Transpoz matris: $\underline{A}_{nxm}$ matrisinin satırları yeni bir matrisin kolonları olarak yazılırsa bu yeni matrise $\underline{A}$ nın transpoz matrisi denir ve yeni matris $\underline{A}^T$ ile veya $\underline{A}'$ gösterilir. Transpoz matrisin boyutu mxn olur. Bazı Türkçe kaynaklar Transpoz matrisi devrik matris olarak isimlendirmektedirler.

$$\text{Örnek}: \ \underline{A}_{3x4} = \begin{bmatrix} -3 & 7.1 & 0 & 5.5 \\ 6 & 3.3 & 11 & -8 \\ 9.4 & 2.9 & 16 & 1 \end{bmatrix} \Rightarrow \underline{A}^T_{4x3} = \begin{bmatrix} -3 & 6 & 9.4 \\ 7.1 & 3.3 & 2.9 \\ 0 & 11 & 16 \\ 5.5 & -8 & 1 \end{bmatrix}$$

Bazı önemli özellikler:

- Bir kolon matrisin(kolon vektörün) transpozu satır matrisi, bir satır matrisin(satır vektörün) transpozu kolon matrisidir.

$$\text{Örnek:} \ \underline{A}_{3x1} = \begin{bmatrix} 4.83 \\ -16.72 \\ 19.07 \end{bmatrix} \rightarrow \underline{A}^T = \begin{bmatrix} 4.83 & -16.72 & 19.07 \end{bmatrix}, \quad \underline{A}_{1x3} = \begin{bmatrix} -7 & 9.6 & 2 \end{bmatrix} \rightarrow \underline{A}^T = \begin{bmatrix} -7 \\ 9.6 \\ 2 \end{bmatrix}$$

- $\underline{A}_{nxm}$ matrisin transpozunun transpozu yine aynı $\underline{A}_{nxm}$ matrisidir $(\underline{A}^T)^T = \underline{A}$

$$\text{Örnek:} \ \underline{A}_{2x3} = \begin{bmatrix} -5 & -3 & 7.7 \\ 5.7 & 3 & 2 \end{bmatrix} \rightarrow \underline{A}^T = \begin{bmatrix} -5 & 5.7 \\ -3 & 3 \\ 7.7 & 2 \end{bmatrix} \rightarrow (\underline{A}^T)^T = \underline{A}_{2x3} = \begin{bmatrix} -5 & -3 & 7.7 \\ 5.7 & 3 & 2 \end{bmatrix}$$

- Simetrik $\underline{A}_{nxn}$ matrisinin tarnspozu gene $\underline{A}_{nxn}$ matrisidir: $\underline{A}^T = \underline{A}$

$$\text{Örnek:} \ \underline{A}_{3x3} = \begin{bmatrix} 5 & 1 & 3 \\ 1 & 7 & -1 \\ 3 & -1 & 4 \end{bmatrix} \rightarrow \underline{A}^T_{3x3} = \underline{A}_{3x3} = \begin{bmatrix} 5 & 1 & 3 \\ 1 & 7 & -1 \\ 3 & -1 & 4 \end{bmatrix}$$

Bu nedenle $\underline{A}^T = \underline{A}$ simetriklik koşulu olarak kullanılır.

Bölünmüş matrisin transpozu: Elemanları alt matris olan bölünmüş bir matrisin transpozu alt matrislerin de transpozu alınarak oluşturulur.

Bölünmüş matris

$$\underline{A} = \left[\begin{array}{c|ccc} -1 & 0 & 0 & 0 \\ \hline 0 & 1 & 2 & 3 \\ 0 & 1 & -2 & 3 \end{array}\right] = \begin{bmatrix} \underline{a}_{11} & \underline{a}_{12} & \underline{a}_{13} \\ \underline{a}_{21} & \underline{a}_{22} & \underline{a}_{23} \end{bmatrix}, \quad \underline{A}^T = \begin{bmatrix} \underline{a}^T_{11} & \underline{a}^T_{21} \\ \underline{a}^T_{12} & \underline{a}^T_{22} \\ \underline{a}^T_{13} & \underline{a}^T_{23} \end{bmatrix} = \left[\begin{array}{c|cc} -1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 2 & -2 \\ \hline 0 & 3 & 3 \end{array}\right] \qquad (1.5)$$

$\underline{a}_{21}$ $\underline{a}_{22}$ Alt matris

İki matrisin Eşitliği: Bir $\underline{A}$ matrisi ile bir $\underline{B}$ matrisinin eşit olabilmesi için her iki matrisin boyutların aynı olması ve karşılıklı elemanlarının eşit olması gerekir. $\underline{A}_{nxm}=\underline{B}_{nxm}$ ise $a_{ij}=b_{ij}$ dir.

$$\underline{A} = \begin{bmatrix} 2 & 0 & 1 \\ 9 & 4 & 0.25 \end{bmatrix}, \quad \underline{B} = \begin{bmatrix} \sqrt{x} & 0 & y \\ z^2 & x & -2t \end{bmatrix}, \qquad \underline{A} = \underline{B}$$

olduğu biliniyorsa x=4, y=1, z=±3, t=-0.125 olmak zorundadır.

Simetrik pozitif tanımlı matris: Simetrik bir A matrisi ($\underline{A}=\underline{A}^T$) ile elemanlarının en az biri sıfırdan farklı olan, bunun dışında tamamen keyfi bir $\underline{x} \neq \underline{0}$ kolon vektörü verilmiş olsun.

$$P = \underline{x}^T \underline{A}\, \underline{x}$$

çarpımı sabit bir sayı olur. Eğer

P>0 ise A pozitif tanımlıdır (positive definit)
P<0 ise A negatif tanımlıdır (negative definit)
P≥0 ise A yarı pozitif tanımlıdır (positive semidefinit)
P≤0 ise A yarı negatif tanımlıdır (negative semidefinit)

denir. Simetrik matrisler uygulamada sıkça görülür. Matrisin hem simetrik hem de pozitif tanımlı olması denklem sistemlerinin çözümünde önemli rol oynar, kolaylık sağlar.

Herhangi bir matrisin pozitif tanımlı olup olmadığını anlamak genelde basit değildir. Çoğu kez matrisin fiziksel anlamı yorumlanarak karar verilir. Fiziksel anlamı nedir? İlerleyen konularda tekrar ele alınarak açıklanacaktır. Şimdilik, iki basit örnek ile açıklayalım.

Örnekler:

$$\underline{A}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \underline{x}=\begin{bmatrix}x_1\\ x_2\end{bmatrix}, p=\underline{x}^T\underline{A}\underline{x}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=x_1^2+x_2^2$$

olduğundan, $x_1 \neq 0$ veya $x_2 \neq 0$ olduğu sürece p>0 dır, dolayısıyla A pozitif tanımlıdır.

$$\underline{B}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \underline{x}=\begin{bmatrix}x_1\\ x_2\end{bmatrix}, p=\underline{x}^T\underline{B}\underline{x}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=x_1x_2+x_1x_2$$

dır. $x_1 \leq 0$ ve $x_2 \geq 0$ için p≤0 olacağından B pozitif tanımlı değildir.

Diyagonal ağırlıklı(diagonal dominant) matris: Bir kare matrisin i. satırındaki diyagonal elemanının mutlak değeri aynı satırdaki diğer elemanların mutlak değerlerinin toplamına eşit veya büyükse

$$|a_{ii}| \geq \sum_{j=1}^{n} |a_{ij}| \ , \quad (\text{bütün } i=1,2,...,n \ \text{ ve } i \neq j \ \text{ için})$$

matris diyagonal ağırlıklıdır(İngilizce diagonal dominant: diyagonal ağırlıklı, egemen, baskın) denir. Eğer

$$|a_{ii}| > \sum_{j=1}^{n} |a_{ij}| \ , \quad (\text{bütün } i=1,2,...,n \ \text{ ve } i \neq j \ \text{ için})$$

sağlanıyorsa matris kesin diyagonal ağırlıklı (İng. strictly diagonal dominant: kesin diyagonal ağırlıklı, egemen, baskın) denir. Kesin diyagonal ağırlıklı matrisin determinantı[4] sıfırdan farklıdır.

Örnekler:

$$\underline{A}=\begin{bmatrix}-6 & 3 & -3\\ 4 & 8 & -3\\ 1 & 3 & 5\end{bmatrix} \rightarrow \left.\begin{matrix}|-6|=|3|+|-3|\\ |8|>|4|+|-3|\\ |5|>|1|+|3|\end{matrix}\right\}$$ olduğundan A diyagonal ağırlıklıdır, det A≠0 veya det A=0 olabilir.

$$\underline{B}=\begin{bmatrix}-7 & 3 & -3\\ -2 & 8 & -3\\ 1 & 3 & 5\end{bmatrix} \rightarrow \left.\begin{matrix}|-7|>|3|+|-3|\\ |8|>|-2|+|-3\\ |5|>|1|+|3|\end{matrix}\right\}$$ olduğundan B kesin diyagonal ağırlıklıdır, det B≠0 dır.

[4] Determinant kavramı bir sonraki bölümde verilecektir.

Matrisin kesin diyagonal dominant olması, özellikle iterasyon ile denklem sistemi çözümünde, önemli rol oynar.

Matrislerin bilgisayar belleğinde depolanma biçimleri

Bilgisayar belleği tek boyutlu bir dizidir, byte-byte i izler. Matrisler bu diziye, programlama tekniğine bağlı olarak, tek veya iki boyutlu olarak depolanabilir. Farklı tipte matrislerin depolama biçimleri aşağıda özetlenmiştir.

Vektör:

teori

Bilgisayarda tek boyutlu depolanır

$$\underline{A}=\begin{bmatrix} a_1 \\ a_2 \\ . \\ a_n \end{bmatrix} \text{ veya } \underline{A}=[a_1 \quad a_2 \quad . \quad a_n] \rightarrow [a_1\, a_2 \ldots a_n]$$

Dikdörtgen veya kare matris:

teori

Bilgisayarda iki boyutlu depolanır

Tek boyutlu ve satır-satır depolama

$$\underline{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ . & . & \cdots & . \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ . & . & \cdots & . \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

→veya→

$$[a_{11}\, a_{12} \ldots a_{1m} \mid a_{21}\, a_{22} \ldots a_{2m} \mid \ldots \mid a_{n1}\, a_{n2} \ldots a_{nm}]$$

$$[a_{11}\, a_{21} \ldots a_{n1} \mid a_{12}\, a_{22} \ldots a_{n2} \mid \ldots \mid a_{1m}\, a_{2m} \ldots a_{nm}]$$

Tek boyutlu ve kolon-kolon depolama

Diyagonal matris:

teori

Bilgisayarda sadece diyagonal elemanlar tek boyutlu depolanır

$$\underline{A}=\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ . & . & . & . \\ & & & a_{nn} \end{bmatrix} \rightarrow [a_{11}\, a_{22} \ldots a_{1n}]$$

Alt üçgen matris:

teori

Bilgisayarda iki boyutlu depolanır (üçgen depolama mümkün değil!)

Tek boyutlu ve satır-satır depolama

$$\underline{L}=\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ . & . & . & . \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ . & . & . & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

→veya→

$$[a_{11}|\, a_{21}\, a_{22}| \ldots |a_{n1}\, a_{n2} \ldots a_{nn}]$$

$$[a_{11}\, a_{21} \ldots a_{n1}|\, a_{22} \ldots a_{n2}| \ldots |\, a_{nn}]$$

Tek boyutlu ve kolon-kolon depolama

Üst üçgen matris:

Tek boyutlu ve satır-satır

$$\underline{U}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ . & . & . & . \\ & & & a_{nn} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ . & . & . & . \\ & & & a_{nn} \end{bmatrix}$$

→veya→

$$[a_{11}\, a_{12} \ldots a_{1n}|\, a_{22} \ldots a_{2n}| \ldots |\, a_{nn}]$$

$$[a_{11}\, |a_{12}\, a_{22}| \ldots |a_{1n}\, a_{2n} \ldots a_{nn}]$$

Tek boyutlu ve kolon-kolon depolama

Simetrik olmayan bant matris:

teori

bant

$$\underline{A} = \begin{bmatrix} x & x & x & x & & & & & & \\ x & x & x & x & x & & & & & \\ x & x & x & x & x & x & & & & \\ & x & x & x & x & x & x & & & \\ & & x & x & x & x & x & x & & \\ & & & x & x & x & x & x & x & \\ & & & & x & x & x & x & x & x \\ & & & & & x & x & x & x & x \\ & & & & & & x & x & x & x \\ & & & & & & & x & x & x \end{bmatrix} \rightarrow \text{Model} \rightarrow \begin{array}{c} \text{bant} \\ \begin{bmatrix} 0 & 0 & x & x & x & x \\ 0 & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Sadece bant içindeki elemanlar depolanır

Model: Bant genişliğini eşitlemek için gösterilen sıfırlar varmış gibi depolanır.

bant

Bilgisayarda iki boyutlu depolama

$$\begin{bmatrix} 0 & 0 & x & x & x & x \\ 0 & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & 0 & 0 & 0 \end{bmatrix} \rightarrow \text{veya} \rightarrow$$

Tek boyutlu ve satır-satır depolama

[0 0 x x x x | 0 x x x x x | x x x x x x | ... | x x x x x x | x x x x x 0 | x x x x 0 0 | x x x 0 0 0]

[0 0 x x x x x x x x | 0 x x x x x x x x x | ... | x x x x x x x x x 0 | x x x x x x x x 0 0 | x x x x x x x 0 0 0]

Tek boyutlu kolon-kolon depolama

Simetrik bant matrisin üst bantı:

Yarı bant

teori

$$\underline{A} = \begin{bmatrix} x & x & x & x & & & & & \\ & x & x & x & x & & & & \\ & & x & x & x & x & & & \\ & & & x & x & x & x & & \\ & & & & x & x & x & x & \\ & \textit{Simetrik} & & & & x & x & x & x \\ & & & & & & x & x & x \\ & & & & & & & x & x \\ & & & & & & & & x \end{bmatrix} \rightarrow \begin{array}{c} \text{Yarı bant} \\ \begin{bmatrix} x & x & x & x & & & & & & & & \\ & x & x & x & x & & & & & & & \\ & & x & x & x & x & & & & & & \\ & & & x & x & x & x & & & & & \\ & & & & x & x & x & x & & & & \\ & \textit{Simetrik} & & & & x & x & x & x & & & \\ & & & & & & x & x & x & 0 & & \\ & & & & & & & x & x & 0 & 0 & \\ & & & & & & & & x & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Sadece üst yarı bant içindeki elemanlar depolanır

Model: yarı bant genişliğini eşitlemek için gösterilen sıfırlar varmış gibi depolanır.

Yarı bant

Bilgisayarda iki boyutlu depolama

$$\rightarrow \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & 0 \\ x & x & 0 & 0 \\ x & 0 & 0 & 0 \end{bmatrix} \rightarrow \text{veya} \rightarrow$$

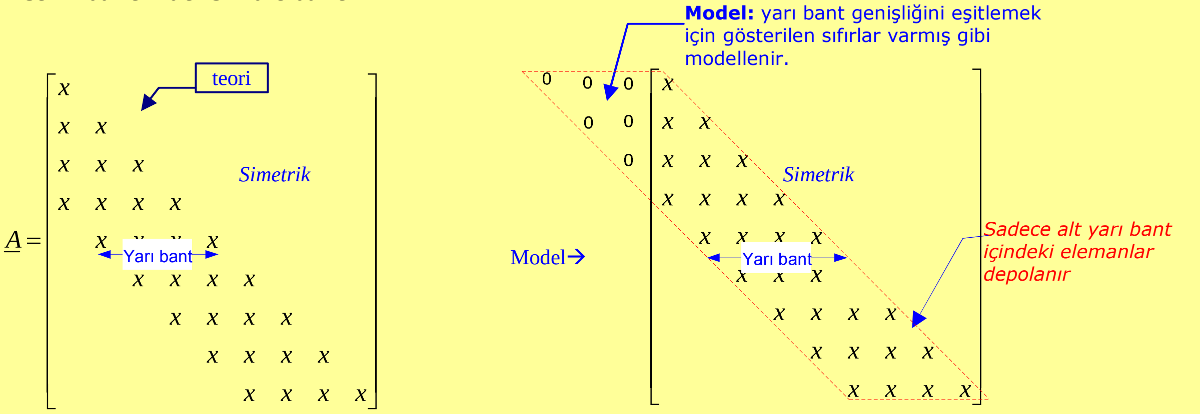
tek boyutlu satır-satır depolama

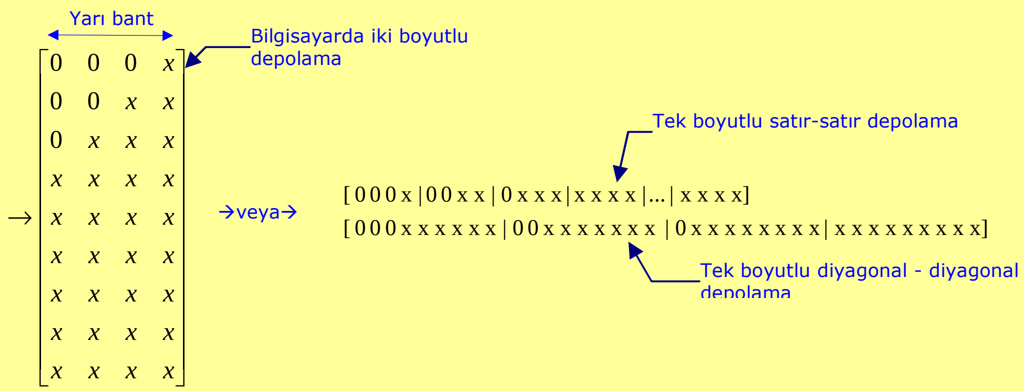
[x x x x | ... | x x x x | x x x 0 | x x 0 0 | x 0 0 0]

[x x x x x x x x x | x x x x x x x x 0 | x x x x x x x 0 0 | x x x x x x 0 0 0]

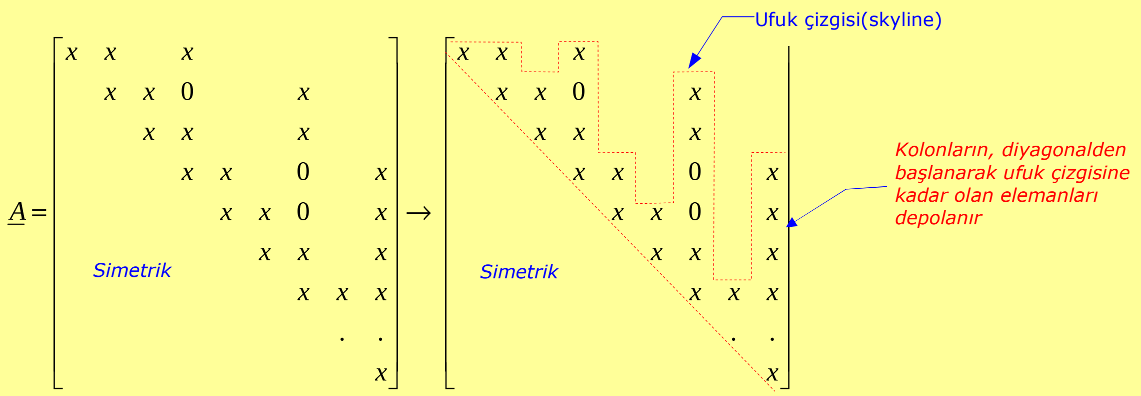
Tek boyutlu diyagonal - diyagonal depolama

Simetrik bant matrisin alt bantı:





Ufuk çizgisi (Skyline depolama tekniği): Sadece simetrik pozitif tanımlı matrisler için kullanılabilir.



Seyrek(sparse) matris(iterasyon yöntemleri için):

$$\underline{A}=\begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} & 0 \\ 0 & a_{22} & 0 & 0 & a_{25} \\ a_{31} & 0 & a_{33} & a_{34} & 0 \\ 0 & a_{42} & 0 & a_{44} & 0 \\ 0 & a_{52} & 0 & 0 & a_{55} \end{bmatrix}$$

Seyrek matris: Sıfırdan farklı elemanı çok az(%1 civarında olan matris)

Sadece sıfırdan farklı elemanlar depolanır

$$\underline{A} \rightarrow [a_{11}\ a_{13}\ a_{14}\ a_{22}\ a_{25}\ a_{31}\ a_{33}\ a_{34}\ a_{42}\ a_{44}\ a_{52}\ a_{55}]$$

Matrisin sadece sıfırdan farklı olan elemanlar tek boyutlu A vektöründe depolanır

$$ia \rightarrow [1\ 1\ 1\ 2\ 2\ 3\ 3\ 3\ 4\ 4\ 5\ 5]$$

Sıfırdan farklı elemanların satır numaraları *ia* vektöründe depolanır

$$\text{ja} \rightarrow [1\ 3\ 4\ 2\ 5\ 1\ 3\ 4\ 2\ 4\ 2\ 5]$$

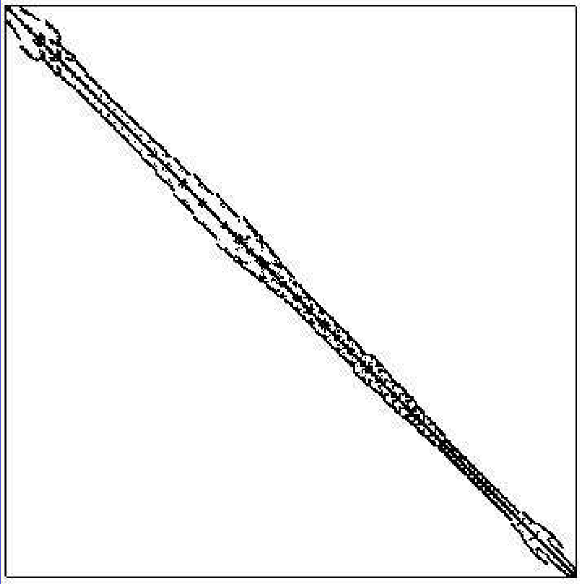
Sıfırdan farklı elemanların kolon numaraları *ja* vektöründe depolanır

Ek bilgi: Seyrek ve çok büyük matrisler uygulamada, özellikle sonlu elemanlar metodunun denge denklemlerinde, karşımıza çıkar. Matris simetrik pozitif tanımlı ve bant yapısına sahiptir. Denklem sistemi Cholesky(bant), Cholesky(Skyline) direkt veya CG(Conjugate Gradient) iterasyon metodu ile çözülür(bu metodlar ilerleyen konularda ele alınacak).

Cholesky(bant veya ufuk çizgisi) direkt metodunda denklem sistemi çözülürken bant içindeki veya ufuk çizgisi altındaki, başlangıçta sıfır olan, elemanlar sıfırdan farklı olurlar. Bu nedenle çok büyük sistemlerinde çok fazla bellek gerekir, ayrıca yuvarlama hataları giderek büyür.

CG metodunda matrisin sadece sıfırdan farklı elemanları, tercihen simetri dikkate alınmadan, depolanır. Çünkü programlama açısından daha uygundur. Denklem sistemi çözülürken matrisin elemanlarının değeri hiç değişmez. İterasyon sayısı yaklaşık denklem sayısı kadardır. Yuvarlama hataları da Cholesky metoduna nazaran daha azdır.

Uygulamadan alınan aşağıdaki A=Bcsstk17 seyrek matrisi için bu metotların ana bellek gereksinimi karşılaştırılacaktır:



A=Bcsstk17 matrisi

Matris bilgileri:
Simetrik: evet
Pozitif tanımlı: evet
Diyagonal ağırlıklı: Hayır
Boyutu: 10974x10974
Yarı bant genişliği: 522
Sıfırdan farklı eleman sayısı: 219812
Doluluk oranı: $219812/10974^2=0.0018\approx$ % 0.2

Matrisin depolanabilmesi için ana bellek gereksinimi:

Cholesky(Skyline) direkt metodu: 28483995x8=22787160 byte

Cholesky(bant) direkt metodu: 10974x522x8=45827424 Byte

CG itersayon metodu:
A için: 219812x8=17296 byte
ia ve *ja* için: 2x10974x2=43896 byte
ara değer vektörleri için: 4x10974x8=351168 byte
toplam: 17296+43896+351168=412360 byte

Oran:
Skyline/CG=22787160/412360≈55
Cholesky/CG=45827424/412360≈111

Görüldüğü gibi, CG iterasyon yöntemi direkt çözümlere nazaran 50-100 kat daha az ana bellek gerektirmektedir

Not:

1) Yukarıda ele alınan matrisin kaynaklardaki adı Bcsstk17 dir, araştırmalarda kullanılmaktadır. Bu matrisin tüm bilgilerine (*A , ia, ja* dahil) http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/bcsstruc2/bcsstk17.html adresinden erişilebilir.

2) Yukarıda verilen dışında başka seyrek matris depolama biçimleri de vardır. Seyrek(sparse) matrisler için detaylı bilgi: http://www-users.cs.umn.edu/~saad/books.html