

MATEMATİK ÇARKI

Oğuzhan Türker, Ata Gülalan

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

oguzturker8@gmail.com, xavaneo@gmail.com

Özet

Programımız kullanıcının rastgele bir sayı girmesiyle başlayıp, girilen sayı değeri kadar programın rastgele sayılar üretmesiyle başlar. Ardından kullanıcıdan alınan sayı değeri kadar, rastgele sayı üretilir. Üretilen sayılara göre programın yapacağı dört matematiksel işlemden biri belirlenir. Kullanıcıdan istenen matris boyutu ve matris elemanlarına göre seçilmiş işlem yapılır. Son olarak ise sonuçlar ve ara işlemler ilgili metin belgesine yazdırılır ve program sonlandırılır.

Matematik çarkı, kullanıcının girdiği iki boyutlu tüm matrislerde işlem yapabilmek için doğru sonuçları üretirken, üç boyutlularda bir kısmını ve dört ve daha fazla boyutlu matrislerde ise hiç işlem yapamamaktadır.

1.Giriş

Matematik çarkı, üniversite lineer ve matematik derslerinin ilgisinde olduğu, kullanıcının girişlerine göre farklı sonuçlar ürettiği bir programdır. Matematik çarkı barındırdığı dört işlem için de matrise ihtiyaç duyar. Kullanıcının belirlediği kare matrislere göre işlemler yapıp, sonuçlar üretir.

Program insanların, matrislerin özdeğerlerinin ve öz vektörlerinin bulunmasında kolaylık sağlar. Ayrıca Nilpotent matris hakkında kullanıcıya bilgi verip, kullanıcıdan aldığı boyuta göre rastgele sayılar üretip Nilpotent matrisi verdiği örnek ile açıklar.

Çark, kullanıcının belirlediği işlemleri bir ve iki boyutta doğru yaparken üç boyutta işlemlerin %68 'ini yapabilmektedir.

Matematik çarkının yaptığı işlemlere ayrıca seçilen işlem adında, programın bulunduğu klasördeki metin belgesinden de ulaşabilirsiniz.

Matrisler üzerinde yapılan bu dört işlem hakkında daha çok bilgi almak isterseniz, diğer araştırmacıların çalışmalarına* bakabilirsiniz. [<http://kisi.deu.edu.tr/kemal.sehirli/%C3%96zde%C4%9Fferler.pdf>]

2.Temel Bilgiler

Program C programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak “CodeBlocks” kullanılmıştır.

3.Tasarım

Matematik çarkının programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

3.1 Algoritma

Program, öncelikle sizden çarkı kaç kere döndürmesi gerektiğini isteyecektir.

Verdiğiniz cevabın doğal sayı olması gerekmektedir. Bu cevap, çarkı istediğiniz kadar döndürecek, size her durduğunda bulduğu değerleri gösterecektir.

Son döndürdüğünüzde gelen değere göre çeşitli fonksiyonlar çağıracaktır.

Eğer gelen fonksiyon Nilpotent ise, sizden sadece matrisin boyutunu isteyecektir. Eğer gelen fonksiyon Nilpotent değil ise, sizden matrisin boyutu yanında matrisin elemanlarını da isteyecektir.

Bu boyut ve elemanların tam sayı olması gerekmektedir.

Verilen matrise göre işlemleri yapıp, üstte belirtilen dosyalara yazdırılacaklardır.

3.2 Denklemler

Özdeğer Denklemi;

$$(1) 0 = \det(L \cdot I_n - A) \quad (1)$$

$L = \text{Ozdeger}$,

$I_n = n$ boyutlu birim matris,

$A =$ kullanıcıdan alınan n boyutlu matris.

Schur Eşitsizliği;

$$(2) |L_1|^2 + \dots + |L_n|^2 \leq A_{(11)}^2 + \dots + A_{(nn)}^2$$

$i = 1 \rightarrow n, j = 1 \rightarrow n$.

Özvektör Denklemi;

$$(3) A \cdot (1)v = L \cdot (2)v; \quad (3)$$

$\cdot(1) =$ Matris çarpımı,

$\cdot(2) =$ Skaler çarpım.

$v =$ Özvektör.

Nilpotent Denklemi;

$$(4) A \cdot A = 0 \quad (\text{nilpotent})$$

$A \cdot A \neq 0$ (nilpotent değil),

$\cdot =$ Matris çarpımı.

3.3 Karşılaşılan problemler ve çözüm yaklaşımları

Özdeğer bulma fonksiyonunda 3x3 matris için değerlerin tam bulunmadığını fark ettik. Bunun üzerine matematiksel fonksiyonu açıklayan sitelerden yardım aldık. [Kaynakça 3.2]

Öz vektörde uygun kodu bulamadık, Java dilinden C'ye çevirerek çalıştırdık. [Kaynakça 2]

3.4 Kullanılan Fonksiyonlar

int det(int A[MAXNXXN][MAXNXXN], int n);

Verilen matrisin determinantını bulur.

int ozdegerbul(int n);

Verilen matrisin özdeğerini bulur.

int schur(int n);

Verilen matrisi ve bu matrisin özdeğerlerini kullanarak, schur teoremi ile sınırlarını bulur.

int ozvektorbul(int n);

Verilen matrisi ve bu matrisin özdeğerlerini kullanarak öz vektörlerini bulur.

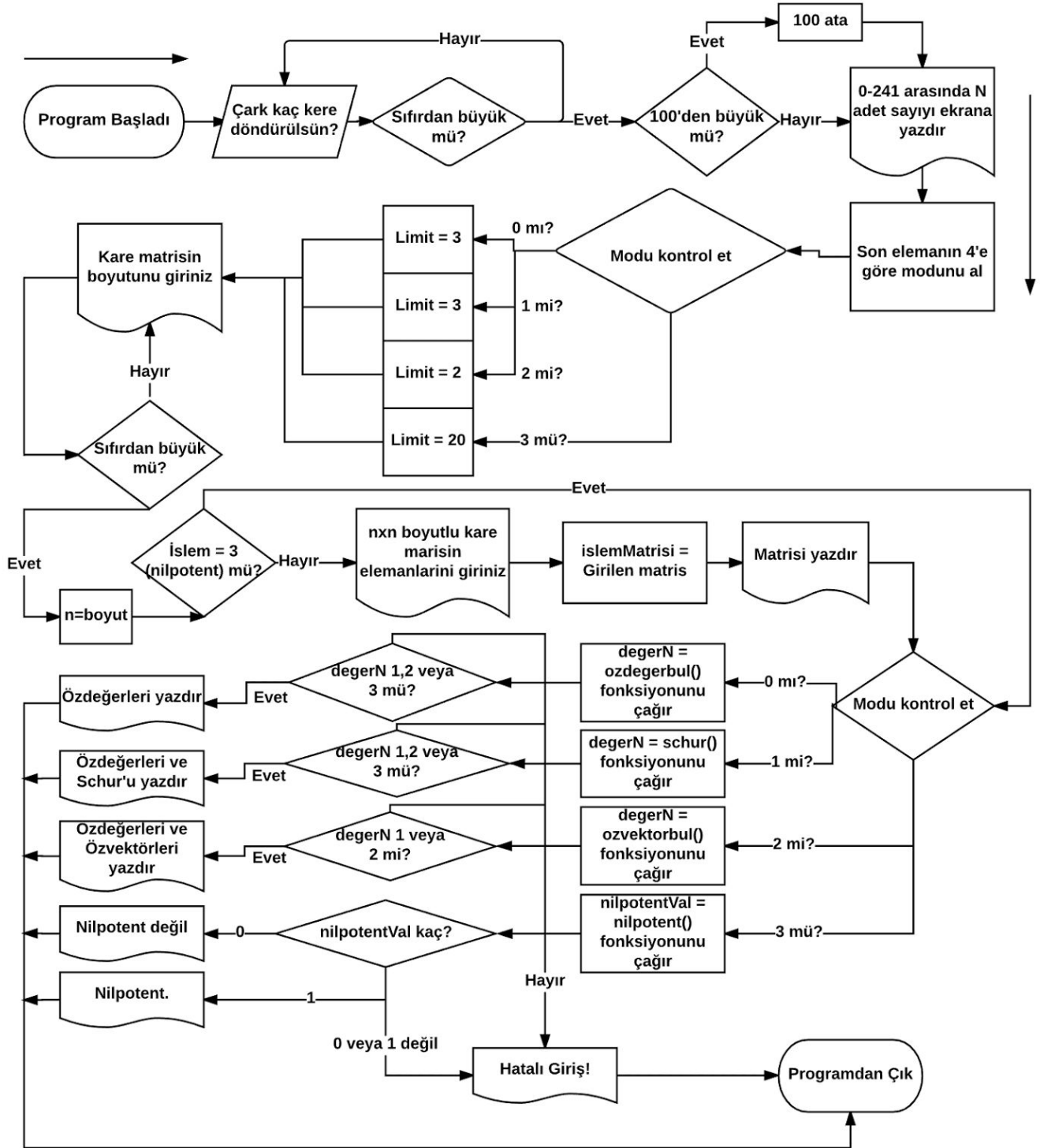
int nilpotent(int n);

Kullanıcının belirlediği matris boyutuna göre -15 ve 15 değerleri arasında rastgele sayılar üreterek matrisin elemanlarını oluşturur. Matrisin karesinin determinantı 0 ise matris nilpotenttir.

void dosyaSifirla();

Dosyalama işlemi sırasında, varolan dosyanın üzerine yazmaması için dosyanın içeriğini boşaltan fonksiyondur.

3.5 Akış Diyagramı



Şekil 1 (Akış Diyagramı)

4.Sonuçlar

Matematik çarkı dört ve daha fazla boyutlu kare matrisler için nilpotent bulup, özdeğer, özvektör ve schur eşitsizliği yapamamaktadır. Üç boyutlu kare matrislerde ise kompleks özdeğerler ve tüm özdeğerlere göre özvektörleri bulamamaktadır. Program ilk üç boyutlu matrisler için işlemlerin çoğunu yapabildiğinden dolayı proje hedefe yeteri kadar ulaşmıştır.

5. Bazı ekran çıktıları:

```
C:\Windows\System32\cmd.exe
Matematik çarkı kaç kere dönsün? 44
36, 180, 210, 220, 205, 109, 84, 54, 191, 39, 3, 187, 6, 73, 201, 168, 191, 212, 35, 225, 160,
147, 61, 48, 42, 30, 186, 87, 59, 159, 119, 42, 130, 77, 187, 217, 197, 65, 190, 202, 36, 5,
180, 71. (44 kere döndü)
Gelen işlem:3 (Nilpotent Bulma)
[n]x[n] Kare matrisin boyutunu giriniz: 6

----- Matris A -----
      5      -4      8      3      14      -12
      2     -15     -1     12      6     -14
     15     15     13    -14     13      -6
     13      4     14     -9    -10      -6
      5      6     14    -14      0      -3
     10      5      2      0    -12     15

----- Matris A*A -----
     126     196     362    -368     264    -292
      11     216     242    -352     -27    -108
     123     -98     248     -13     681    -513
      56     -28      4     112     550    -302
      35      29     14     17     464    -189
     180     -82    -37     230     16      59

Matris nilpotent değil.
```

Şekil 2 (Nilpotent Bulma)

```
C:\Windows\System32\cmd.exe
Matematik carki kac kere dunsun? 42
193, 107, 24, 112, 146, 200, 150, 191, 4, 4, 18, 138, 149, 79, 94, 102, 147, 238, 92, 67,
28, 51, 88, 101, 204, 132, 71, 11, 9, 65, 99, 240, 188, 118, 85, 103, 202, 163, 145, 192,
36, 192. (42 kere dondu)
Gelen islem:0 (Ozdeger Bulma)
[n]x[n] Kare matrisin boyutunu giriniz: 2
[2]x[2] boyutlu kare marisin elemanlarini giriniz:
[1]x[1] : 1
[1]x[2] : 3
[2]x[1] : 5
[2]x[2] : 8

----- Matris -----

      1      3
      5      8

Determinant = -7
Diskriminant = 109
Girilen matrisin ozdegerleri: 9.720153, -0.720153
```

Şekil 3 (Özdeğer Bulma)

```
C:\Windows\System32\cmd.exe
Matematik carki kac kere dunsun? 33
60, 173, 101, 192, 56, 160, 200, 201, 164, 172, 98, 31, 73, 100, 177, 26, 66, 114, 94, 96,
163, 37, 78, 132, 213, 158, 55, 186, 203, 30, 27, 157, 110. (33 kere dondu)
Gelen islem:2 (Ozvektor Bulma)
[n]x[n] Kare matrisin boyutunu giriniz: 2
[2]x[2] boyutlu kare marisin elemanlarini giriniz:
[1]x[1] : 1
[1]x[2] : 7
[2]x[1] : 3
[2]x[2] : 6

----- Matris -----

      1      7
      3      6

Determinant = -15
Diskriminant = 109
Girilen matrisin ozdegerleri: 8.720153, -1.720153
Girilen matrisin ozvektorleri: [0.671710, 0.740815]; [0.932098, -0.362207]
```

Şekil 4 (Özvektör Bulma)

6. Dosyalama örnekleri:

Örnek 1 (schur.txt):

Kullanıcıdan [2]x[2] boyutunda bir kare matris alındı.

----- Matris -----

1	2
3	4

Determinant: -2

Diskriminant: 33

Girilen matrisin ozdegerleri:

Ozdeger 1: 5.372281

Ozdeger 2: -0.372281

Schur Teoremi:

$|Ozdeger1|^2 + \dots + |Ozdeger n|^2 \leq$
 $A[i][j]^2 + \dots + A[n][n]^2 \quad i \rightarrow n, j \rightarrow n$
 $28.861 + 0.139 < 1 + 4 + 9 + 16$
 $29.000 \leq 30$

Örnek 2 (ozdeger.txt):

Kullanıcıdan [2]x[2] boyutunda bir kare matris alındı.

----- Matris -----

1	2
4	3

Determinant: -5

Diskriminant: 36

Girilen matrisin ozdegerleri:

Ozdeger 1: 5.000000

Ozdeger 2: -1.000000

Kaynakça

1. **[det()]**
C program to calculate the determinant of a NxN matrix - Stack Overflow
<https://stackoverflow.com/questions/41384020/c-program-to-calculate-the-determinant-of-a-nxn-matrix>
(Erişim Tarihi: 04.10.2017)
2. **[ozvektorbul()]**
How to solve for eigenvectors of a 2x2 Matrix using Java - Stack Overflow
<https://stackoverflow.com/questions/43727943/how-to-solve-for-eigenvectors-of-a-2x2-matrix-using-java>
(Erişim Tarihi: 08.10.2017)
3. **[ozdegerbul()]**
 1. Equation Solver in C - Code Project
<https://www.codeproject.com/Articles/42727/Equation-Solver-in-C>
(Erişim Tarihi: 06.10.2017)
 2. Solving Cubic Equations - 1728
<http://www.1728.org/cubic2.htm>
(Erişim Tarihi: 09.10.2017)
4. **[schur()]**
LİNEER ZAMAN GECİKMELİ SİSTEMLER İÇİN KONTROL PROBLEMLERİ (Bayram Barış KIZILSAÇ) - İTÜ
<https://polen.itu.edu.tr/bitstream/11527/9911/1/9282.pdf>
(Erişim Tarihi: 08.10.2017)