Polinom Denklemlerin Mutlak Değerce En Büyük Reel Kökünü Veren Bir Metot

A Method to Find the Dominant Real Root of a Polynomial Equation

ANTALYA
İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

Arif Kerem DAYI[1*] **Özlem MOĞOL**[1]

[1]Karşıyaka Aydoğan Yağcı Bilim ve Sanat Merkezi, İzmir / Türkiye
[1]Karşıyaka Aydoğan Yağcı Science and Art Center, İzmir / Turkey
[*]arifkeremdayi@gmail.com
ORCID: 0000-0002-1743-3879 **ORCID:** 0000-0001-6609-9806

MAKALE BİLGİSİ / ARTICLE INFORMATION

Geliş Tarihi / Date Received
06.11.2019

Kabul Tarihi / Date Accepted
09.06.2020

Yayın Tarihi / Date Published
Temmuz / July 2020

Yayın Sezonu / Pub Date Season
Haziran - Aralık / June - December

ATIF / CITE as

Dayı, A. K. ve Moğol, Ö. (2020). "Polinom Denklemlerin Mutlak Değerce En Büyük Reel Kökünü Veren Bir Metot" / "A Method to Find the Dominant Real Root of a Polynomial Equation". bilar: Bilim Armonisi Dergisi, 3 (1): 37-51. doi: 10.37215/bilar.643505.

https://dergipark.org.tr/tr/pub/bilar

Polinom Denklemlerin
Mutlak Değerce En Büyük Reel Kökünü Veren Bir Metot

A Method to Find the Dominant Real Root of a Polynomial Equation

ANTALYA
İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

ÖZET

Fibonacci Tavşan Problemi önemli bir sayı dizisi olan Fibonacci dizisini vermektedir. Bu dizi incelendiğinde karşılaşılan indirgeme bağıntısının karakteristik denklemi 2. dereceden bir polinom denklem olduğu görülür. Bu dizinin ardışık terimlerinin oranının limiti ise bu 2. derece polinom denklemin köklerinden biri olan Fi sayısına yakınsar. Peki neden Fi sayısına yakınsar? Sadece pozitif kök olduğu için mi?

Bu çalışmada tavşan problemindeki bazı veriler değiştirilmiştir. Değiştirilen veriler "Canlının yetişkinliğe ulaşma süresi" ve "Canlının yetişkinliğe ulaştıktan sonra hangi aralıklarla doğum yaptığı" dır. Bu değişkenlerin aldıkları değerlere göre ortaya, farklı indirgeme bağıntıları çıkmıştır. Bu indirgeme bağıntılarının karakteristik denklemlerinin 2. dereceden daha yüksek mertebelerde polinomlar oldukları görülmüştür. Bu dizilerin karakteristik denklemi olan Polinom denklemlerin köklerinden birinin Fibonacci dizisinde uygulanan algoritmada olduğu gibi "Dizinin ardışık terimlerin oranlarının limiti" ile bulunabileceği gösterilmiştir. "Bu durumda oluşan dizinin ardışık terimlerin oranının limit değeri her zaman bir köke yakınsar mı? Hangi köke yakınsar? Bu metot hangi koşullar altında sağlanır?" sorularına cevap verilmiştir.

Bu çalışmada yüksek mertebeden polinom denklemlerin köklerinden mutlak değerce en büyük kök reel ise bu reel kökün bulunması için bir metot ortaya koyulmuştur. Bu metot aynı zamanda polinom denklemlerin reel kökleri için üst veya alt sınır bulunmasına da katkı sağlamaktadır.

Anahtar Sözcükler: İndirgemeli diziler, Polinom denklemler, Fibonacci tavşan problemi.

ABSTRACT

The solution to the Rabbit Problem is the famous Fibonacci sequence. When this sequence is analyzed, it is found that the characteristic equation of this sequence is a quadratic equation. Furthermore, the ratio of the consecutive terms of the sequence converges to one of the roots of the characteristic equation. So why does it converge to the number of phi? Is it just because it is a positive root?

In this study, some data in the rabbit problem have been changed. The changed data are "The time that the creature reaches adulthood" and "At what intervals the creature gives birth after reaching adulthood". According to the values of these variables, different recurrence relations emerged. It has been observed that the characteristic equations of these recurrence relations are polynomials of higher order than the 2nd degree. It has been shown that one of the roots of the polynomial equations, which are the characteristic equation of these sequences, can be found with the "Limit of the rates of consecutive terms" as in the algorithm applied in the Fibonacci sequence. "In this case, does the limit value of the rate of consecutive terms of the resulting array always converge to a root? Which root does it converge? Under what conditions is this method provided?" questions were answered.

This paper suggests a new method to find the dominant root –the root with the largest modulus- of a polynomial equation. The method also helps to find lower and upper bounds for the roots.

Keywords: Recursive sequence, Polynomial equations, The Fibonacci sequence.

*Bu çalışma, TÜBİTAK 50. Lise Öğrencileri Araştırma Projeleri Yarışması'nda Türkiye Finalinde sergilenmiştir (2019).

1. GİRİŞ

Matematik olimpiyat sınavlarında karşılaştığımız yüksek dereceli polinom denklemlerin köklerini bulmayı gerektiren sorular için çoğu zaman uzun ve tahmin edilmesi zor çarpanlara ayırma işlemleri gerekebilir. Bu sorular farklı soru tipleri için farklı yöntemler denemeyi gerektirebilir. Bu nedenle köklerin varlığı, köklerin reel, sanal, rasyonel ya da tamsayı olup olmadığı, kök katsayı ilişkisi, katsayıların işaretleri gibi kök hakkında edinilecek her türlü bilgi çok değerlidir.

Fibonacci tavşan problemini çözerken, Fibonacci dizisinin ardışık terimlerinin $\lim_{n\to\infty}\left(\frac{F_{n+1}}{F_n}\right)$ değeri ikinci dereceden $x^2 - x - 1 = 0$ denkleminin çözüm kümesinin elemanlarından birine yakınsadığını görürüz. Fibonacci dizisinin karakteristik denkleminin köklerinden birinin Altın oranı vermesi şu soruyu akla getirmektedir; Her polinom denklemin bir kökünün yaklaşık değerine benzer şekilde ulaşabilir miyiz?

Bölüm 2 de: Çalışmamızın çıkış noktası olan Fibonacci tavşan problemi incelenecektir, problemin değişkenleriyle oynandığında farklı diziler ve denklemlere ulaşıldığı ve metodumuzun burada da çalıştığı görülecektir. Daha sonra daha genel yüksek mertebeden denklemler ele alınarak metodun hangi koşullarda çalıştığı görülecektir. Bu metodun reel kökler için üst sınırı veren mevcut yöntemlerle karşılaştırması yapılacaktır. Ayrıca metot Matematik Olimpiyatlarında çıkmış bazı sorular üzerinde uygulanacaktır.

Bölüm 3 de: Metodun ispatı yapılacak ve metodun algoritmasını veren Python programlama dili kodları verilecektir.

2. MATERYAL VE METOT

2.1. Fibonacci Tavşan Problemi ile Polinom Denklemlerin Çözüm Kümelerinin İlişkilendirilmesi

Bu bölümde Fibonacci tavşan probleminde verilen üreme döngüsü değiştirilerek yeni problemler kurgulanacaktır. Amacımız Fibonacci tavşan probleminde Fi sayısını veren algoritmayı kurguladığımız bu problemlere uygulayarak çözümlerinde zorluk yaşadığımız bazı polinom denklemlerin köklerinin yaklaşık değerlerini bulmakta kullanmaktır. Bu bölümde "Problem değiştirildiğinde her zaman metot çalışır mı? Ya da hangi durumlarda çalışır? Bulduğumuz kök, denklemin hangi köküdür?" sorularına cevap verilmiştir.

2.1.1. Fibonacci Tavşan Problemi

Fibonacci tavşan problemi kısaca şöyle tanımlanabilir. "Bir çift tavşan, iki aylıktan önce yeni bir çift tavşan dünyaya getiremez. Fakat bir tavşan çifti yetişkinliğe ulaştığında, her ay bir çift yeni tavşan dünyaya getirir. Bir çift yeni doğan tavşanla başladığımızda bundan sonra her ayın başında kaç tavşan çiftine sahip olacağımız bilgisini veren ve her ay sonunda karşılaşılan toplam tavşan sayısının oluşturduğu bu dizi Fibonacci dizisidir (Barbeau 1989).

Bu bölümde problemlerimizde kullandığımız değişkenleri tanımlayalım.

Yetişkin olmayan; 1 aylık canlıları K_1 ile 2 aylık olan canlıları K_2 ile i aylık canlıları K_i ile gösterelim. Yetişkinlik düzeyine ulaşan canlıları K_y ile gösterelim.

α = "Canlının yetişkinliğe ulaşma süresi'',

β = "Canlının yetişkinliğe ulaştıktan sonra hangi aralıklarla doğum yaptığı" olsun.

Bu durumda Fibonacci dizisindeki değişkenleri $\alpha = 2$, $\beta = 1$ olarak aldığımızda karşılaştığımız tavşan popülasyonuna ait veriler Çizelge 1 deki gibi olur.

Çizelge 1. Aylara göre tavşan popülasyonu

Ay	K_1	K_2	K_y	Toplam
1	1	0	0	1
2	0	1	0	1
3	1	0	1	2
4	1	1	1	3
5	2	1	2	5
6	3	2	3	8
7	5	3	5	13

$n \geq 1$ ve $n \in Z^+$ olmak üzere Fibonacci indirgeme bağlantısı, $F_{n+2} = F_{n+1} + F_n$ olur. Her terim kendisinden önceki iki terimin toplamıdır. Fibonacci dizisi $(F_n) = (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots)$ve ardışık terimlerinin oranının

$$\left(\frac{F_{n+1}}{F_n}\right) = (1, 2, 1.5, 1.666, 1.625, 1.615, 1.619, 1.617, 1.618, \ldots)$$

$\lim_{n\to\infty}\left(\frac{F_{n+1}}{F_n}\right) \cong \mathbf{1.618}$ olduğu görülmektedir.

Bu limit değeri ise (F_n) dizisinin karakteristik denklemi olan ikinci dereceden $x^2 - x - 1 = 0$ denkleminin, çözüm kümesinin Ç.K. $= \{\mathbf{1.6180}, -0.6180\}$ elemanlarından birine yakınsadığı bilinmektedir.

$\lim_{n\to\infty}\left(\frac{F_{n+1}}{F_n}\right)$ limit, değeri ikinci dereceden $x^2 - x - 1 = 0$ denkleminin çözüm kümesinin elemanlarından birine yakınsar. Yakınsadığı bu kök ise, bizim altın oran sabiti olarak bildiğimiz Fi sayısını vermektedir. Peki, Fi sayısına yakınsamasının sebebi sadece pozitif kök olması mıdır?

Bu çalışmada $\lim_{n\to\infty}\left(\frac{F_{n+1}}{F_n}\right)$ değerinin neden pozitif köke yakınsadığı, neden diğer köke yakınsamadığı, bu algoritmanın her zaman çalışıp çalışmadığı, bu algoritma ile yüksek mertebeden polinom denklemlerin köklerinin bulunup bulunamayacağı sorularına cevap aranmıştır.

2.2. Uyarlama Problemler

2.2.1. Problem

Bir x canlısı, üç aylıktan önce yeni bir çift tavşan dünyaya getiremez. Fakat bir tavşan çifti yetişkinliğe ulaştığında, her ay bir çift yeni tavşan dünyaya getirir. Bir çift yeni doğan tavşanla başladığımızda bundan sonra her ayın başında kaç tavşan çiftine sahip olacağımız bilgisini veren ve her ay sonunda karşılaşılan toplam x canlı sayısının oluşturduğu çizelgeyi inceleyelim. Bu problem için değişkenlerimiz $\alpha = 3$, $\beta = 1$ olarak alınmış ve tavşan popülasyonundaki değişimler Çizelge 2 yi oluşturmuştur.

Çizelge 2. Aylara göre tavşan popülasyonu

Ay	K_1	K_2	K_3	K_y	**Toplam**
1	1	0	0	0	1
2	0	1	0	0	1
3	0	0	1	0	1
4	1	0	0	1	2
5	1	1	0	1	3
6	1	1	1	1	4
7	2	1	1	2	6
8	3	2	1	3	9
9	4	3	2	4	13
10	6	4	3	6	19

Bu durumda karşılaşılan indirgemeli dizi

$$a_n = a_{n-1} + a_{n-3}$$

$$(a_n) = (1, 1, 1, 2, 3, 4, 6, 9, 13, 19, 28, 41, \ldots)$$

$$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 2, 1.5, 1.333, 1.5, 1.5, 1.44, 1.4615, 1.4736, 1.4642, \ldots)$$

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong \mathbf{1.46}$$

ve bu dizinin karakteristik denklemi olan 3. dereceden $x^3 = x^2 + 1$ polinom denklemi elde edilir. Bu denklemin (Wolframalpha 1987) internet sitesinden elde edilen kökleri aşağıdaki gibi bulunur.

Ç.K. = {**1.4656**, $-0.23279 \pm 0.79255i$} elemanlarından birine yakınsadığı bilinmektedir.

Bu dizinin terimlerine bakarak $\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right)$ limit değeri hesaplandığında sonucun mutlak değerce büyük reel köke yakınsadığı görülecektir.

2.2.2. Problem

Bir x canlısı, üç aylıktan önce yeni bir çift tavşan dünyaya getiremez. Fakat bir tavşan çifti yetişkinliğe ulaştığında, iki ayda bir çift yeni tavşan dünyaya getirir. Bir çift yeni doğan tavşanla başladığımızda bundan sonra her ayın başında kaç tavşan çiftine sahip olacağımız bilgisini veren ve her ay sonunda karşılaşılan toplam x canlı sayısının oluşturduğu çizelgeyi inceleyelim. $\alpha = 3,\ \beta = 2$

Çizelge 3. Aylara göre tavşan popülasyonu

Ay	K_1	K_2	K_3	K_{y_1}	K_{y_2}	Toplam
1	1	0	0	0	0	1
2	0	1	0	0	0	1
3	0	0	1	0	0	1
4	1	0	0	1	0	2
5	0	1	0	0	1	2
6	1	0	1	1	0	3
7	1	1	0	1	1	4
8	1	1	1	1	1	5
9	2	1	1	2	1	7
10	2	2	1	2	2	9
11	3	2	2	3	2	12
12	4	3	2	4	3	16
13	5	4	3	5	4	21
14	7	5	4	7	5	28

Bu durumda karşılaşılan indirgemeli dizi

$a_n = a_{n-2} + a_{n-3}$

$(a_n) = (1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, \ldots)$

$$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 2, 1, 1.5, 1.3333, 1.25, 1.4, 1.285, 1.3333, 1.333, 1.3125, 1.3333, 1.3234, \ldots)$$

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong \mathbf{1.32}$$

ve bu dizinin karakteristik denklemi olan $x^3 = x + 1$ üçüncü derece denklemin Ç.K. = {**1.3247**, $-0.66236 \pm 0.56228i$} olduğu bilinmektedir.

$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right)$ limit değeri hesaplandığında sonucun mutlak değerce en büyük reel köke yakınsadığı görülmüştür.

Peki mutlak değerce eşit zıt işaretli iki reel kök olması durumunda ne olur? Sorusuna cevap aramak için Problem 2.2.3 ü inceleyelim.

2.2.3. Problem

Bir x canlısı, dört aylıktan önce yeni bir çift tavşan dünyaya getiremez. Fakat bir tavşan çifti yetişkinliğe ulaştığında, her iki ayda bir çift yeni tavşan dünyaya getirir. Bir çift yeni doğan tavşanla başladığımızda bundan sonra her ayın başında kaç tavşan çiftine sahip olacağımız bilgisini veren ve her ay sonunda karşılaşılan toplam x canlı sayısının oluşturduğu çizelgeyi inceleyelim. K_{y_1} canlının yetişkin olarak geçirdiği 1. Ay ve K_{y_2} canlının yetişkin olarak geçirdiği 2. Ayı ifade etsin. Bu problemde fibonacci probleminde değişkenlerin iki katları verildiğinde aşağıdaki gibi Fibonacci dizisinin terimlerinin tekrarlarından oluşan bir çizelge ile karşılaşılmıştır. Bu durumda değişkenlerimiz $\alpha = 4, \ \beta = 2$ olacaktır.

Çizelge 4. Aylara göre tavşan popülasyonu

Ay	K_1	K_2	K_3	K_4	K_{y_1}	K_{y_2}	Toplam
1	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1
3	0	0	1	0	0	0	1
4	0	0	0	1	0	0	1
5	1	0	0	0	1	0	2
6	0	1	0	0	0	1	2
7	1	0	1	0	1	0	3
8	0	1	0	1	0	1	3
9	2	0	1	0	2	0	5
10	0	2	0	1	0	2	5
11	3	0	2	0	3	0	8
12	0	3	0	2	0	3	8
13	5	0	3	0	5	0	13

Bu durumda karşılaşılan indirgemeli dizi

$$a_n = a_{n-2} + a_{n-4}$$

$$(a_n) = (1, 1, 1, 1, 2, 2, 3, 3, 5, 5, 8, 8, 13, 13, \ldots)$$

$$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 1, 2, 1, 1.5, 1, 1.66667, 1, 1.6, 1, 1.625, 1, 1.615, 1, 1.619, 1, 1.617, 1, 1.618, \ldots)$$

İlk bakışta bu problemde uyguladığımız algoritma çalışmadı gibi görünmektedir. Dizinin parçalı yakınsak olması, mutlak değerleri eşit ve zıt işaretli iki kök durumuyla karşılaştığımızı göstermektedir. Mutlak değerce eşit zıt işaretli iki kök olduğu için çarpımlarının karekökü bizi pozitif köke ulaştırmaktadır.

$$\lim_{n\to\infty} \frac{a_{n+1}}{a_n} \cdot \frac{a_{n+2}}{a_{n+1}} = \lim_{n\to\infty} \frac{a_{n+2}}{a_n} = 1.618\ldots$$

$$\sqrt{\lim_{n\to\infty} \frac{a_{n+2}}{a_n}} \cong \sqrt{1.618\ldots} \cong \mathbf{1.2720\ldots}$$

(a_n) İndirgemeli dizisine karşılık gelen karakteristik denklem $x^4 = x^2 + 1$ ve çözüm kümesi Ç.K. $= \{-1.2770, \ \mathbf{1.2770}, \ -0.78615i, \ 0.78615i\}$ olduğu bilinmektedir. Yaptığımız çalışmalarda gördük ki eğer $\frac{a_{n+1}}{a_n}$ dizisi parçalı yakınsak ise yani iki farklı değere birden yakınsıyorsa mutlak değerce en büyük kökten iki tane vardır ve bunlar zıt işaretledir. Bununla ilgili çıkmış olimpiyat sorusu ilerleyen bölümlerde verilecektir.

2.3. Metodun Uygulamaları

Çalışmamızda asıl amacımız "Verilen herhangi bir polinom denklemin mutlak değerce en büyük kökü eğer bir reel sayı ile algoritmamızı kullanarak bu köke ulaşabilir miyiz?" sorusuna cevap aramaktır. Bu nedenle bu bölümde farklı kök durumları içeren denklemlere örnekler verilecektir. Uygulamalarda kolaylık sağlaması amacıyla metodun bilgisayar kodları yazılmıştır. Bilgisayar kodları Bölüm 3 de yer almaktadır.

2.3.1. Örnek

İkinci dereceden iki farklı reel kök içeren $x^2 = 2x + 4$ denklemini ele alalım. Bu denklemi aşağıda verilen bağıntının indirgeme dizisinin karakteristik denklemi olarak düşünelim.

$a_n = 2a_{n-1} + 4a_{n-2}$

(a_n) dizisinin başlangıç koşullarını $a_1 = 1$ ve $a_2 = 2$ seçersek dizinin terimleri

$(a_n) = (1, 2, 8, 24, 80, 256, 832, 2688, 8704, 28160, \ldots)$ oluşturulduğunda ardışık terimler oranı

$\left(\frac{a_{n+1}}{a_n}\right) = (1, 2, 4, 3, 3.33, 3.2, 3.25, 3.23077, 3.2381, 3.23529, 3.23636, 3.23596, \ldots)$ olmak üzere

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong \mathbf{3.23}$$

olur ve 3.23 değerine yakınsadığı görülmektedir. Bu denklemin gerçek çözümünü incelediğimizde Ç.K. $= \{1 - \sqrt{5}\,,\ \mathbf{\sqrt{5} + 1}\}$ yani Ç.K. $= \{-1.2361,\ \mathbf{3.2361}\}$. Limit değerinin, çözüm kümesinin mutlak değerce en büyük olan köküne yakınsadığı görülmektedir.

2.3.2. Örnek

Üçüncü dereceden üç farklı reel köke sahip $2x^3 - 17x^2 + 48x - 45 = 0$ denklemini ele alalım. Bu karakteristik denkleme karşılık gelen indirgeme bağıntısı

$2a_{n+3} = 17a_{n+2} - 48a_{n+1} + 45a_n$

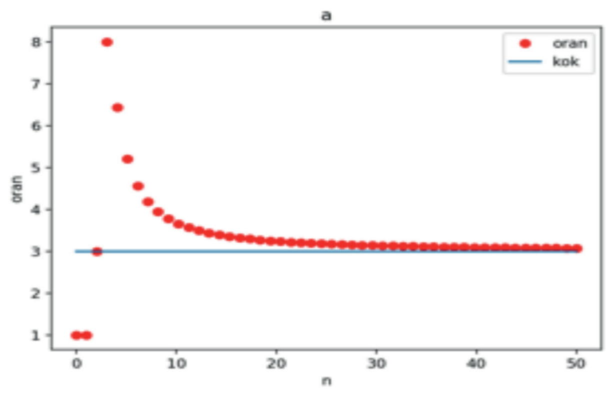
olur ve (a_n) dizisinin başlangıç koşullarını $a_1 = a_2 = 1$ ve $a_3 = 2$ seçelim. Dizinin terimleri

$(a_n) = (1, 1, 2, 15.5, 106.25, 576.125, 2695.8, \ldots)$ olarak bulunur. Ardışık terimlerin oranı ise

$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1.0, 3.0, 8.0, 6.4375, 5.20874, 4.56337, 4.18733, 3.94592, 3.77951, 3.65866, 3.56736, 3.49625,$
$3.43948, 3.39326, 3.355, 3.32289, 3.29562, 3.27222, 3.25195, 3.23426, 3.2187, 3.20494, 3.19269,$
$3.18173, 3.17188, 3.16298, 3.15491, 3.14757, 3.14085, 3.1347, 3.12904, 3.12383, 3.119, 3.11453,$
$3.11037, 3.1065, 3.10288, 3.0995, 3.09633, 3.09335, 3.09055, 3.08791, 3.08541, 3.08306, 3.08083,$
$3.07871, 3.0767, 3.07479, 3.07298)$ olacaktır. Bu dizinin limitinin

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong \mathbf{3}$$

olduğu Grafik 1. den de görülebilir. Bu örnekte olduğu gibi dizinin köke yakınsama hızı yavaş olabilmektedir.



Grafik 1. N. adımda ardışık terimler oranı

Şimdi bu denklemin gerçek çözümünü inceleyelim. Karakteristik denklemin gerçek çözümünü çarpanlarına ayırarak bulalım. $(x - 3)^2 (2x - 5) = 0$ ve çözüm kümesi Ç.K. $= \{3, 2.5\}$ olduğu görülmektedir.

Mutlak değerce en büyük olan kökün çakışık iki kök olması durumunda da algoritmanın çalıştığı görülmektedir.

2.3.3. Örnek

Üçüncü dereceden reel köke sahip olan $x^3 - 2x^2 + x - 2 = 0$ denklemini inceleyelim. Bu karakteristik denkleme karşılık gelen indirgemeli dizi

$$a_{n+3} = 2a_{n+2} - a_{n+1} + 2a_n$$

olur. Başlangıç koşullarını $a_1 = a_2 = a_3 = 1$ alalım. Dizinin terimleri ve ardışık terimler oranı

$$(a_n) = (1, 1, 1, 3, 7, 13, 25, 51, 103, 205, 408, \ldots)$$

$$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 3, 2.33, 1.85714, 1.92307, 2.04, 2.0196, 1.99029, 1.99024, \ldots)$$

olduğu görülür. Bu dizinin limiti ise

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong \mathbf{2}$$

olduğu görülmektedir. Verilen 3. Derece denklemin çarpanlarına ayrılmış halinin $(x - 2)(x^2 + 1) = 0$ ve çözüm kümesinin de Ç.K. $= \{2, \pm i\}$ olduğu bilinmektedir. Dizinin limit değeri de kümenin en büyük değerine 2'ye yakınsamaktadır.

2.3.4. Örnek

Mutlak değerce en büyük kökü negatif reel sayı olan 4. dereceden bir polinom denklemin köklerini inceleyelim.

$$x^4 + 3x^3 - 3x^2 + 3x - 4 = 0$$

$$a_n = -3a_{n-1} + 3a_{n-2} - 3a_{n-3} + 4a_{n-4}$$

$$(a_n) = (1, 1, 1, 2, -2, 13, -47, 194, -770, 3089, \ldots)$$

$$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 2, -1, -6.5, -3.61, -4.12, -3.96, -4.00, -3.99, \ldots)$$

$$\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right) \cong -\mathbf{4}$$

Denklemin çözüm kümesi Ç.K. $= \{-4, 1, \pm i\}$ olduğu bilinmektedir. Bu bilgi Teorem 2.4 ile birlikte kullanıldığında reel köklerin hangi aralıkta bulunduğuna dair bilgi sahibi olmamıza yardımcı olur.

2.4. Polinom Denklemlerin Reel Köklerinin Üst Sınırını Veren Teorem

(Barbeau 1989) kitabında reel katsayılı bir polinomun, reel köklerinin üst sınırını veren bir formül sunmuştur.

2.4. Teorem

$\mathbf{a_n > 0}$ olmak üzere $f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ polinomun en az bir negatif katsayısı bulunsun. En yüksek dereceli terimden başlamak üzere sağa doğru gidildikçe karşılaştığımız ilk negatif katsayıdan önceki "sıfır" veya "pozitif" katsayılı terim sayısı **k** olsun.

$G = \{\max|a_i| : a_i < 0\}$ olmak üzere polinomun pozitif reel köklerinin üst sınırını veren bağıntı, $f(x)$'in herhangi pozitif bir kökü **r** olmak üzere $r < 1 + \sqrt[k]{\frac{G}{a_n}}$ olarak ifade edilebilir.

2.4.1. Örnek

$5x^3 - 16x^2 + 8x - 1 = 0$ polinomu verilsin. Teorem 2.1 e göre $k = 1$, $G = 16$ olduğu için polinomun en büyük reel kökü için $1 + \frac{16}{5} = \frac{21}{5} \sim 4$.

Üst sınırı yaklaşık 4 olarak hesaplamaktadır. Bizim metodumuz ile mutlak değerce en büyük kökü aradığımızda

$5a_n = 16a_{n-1} - 8a_{n-2} + a_{n-3}$

$(a_n) = (1, 1, 2, 5, 13, 34, 89, 233, 610, \ldots)$

$\left(\frac{a_{n+1}}{a_n}\right) = (1, 2, 2.5, 2.6, 2.61538, 2.61765, 2.61798, \ldots)$

limit değerinin 2.61 e yaklaştığı görülür. Dolayısı ile polinomun köklerinin [-2.61, 2.61] aralığında olduğunu söyleyebiliriz. Teorem 2.4 de bulduğumuz pozitif değer 2.61 den küçük olsaydı kökler için bulduğumuz sınır aralığını daraltmamıza imkan sağlayabilirdi.

Problemde verilen denklemin çözüm kümesi Ç.K. $= \{\frac{1}{5}, \frac{3 \pm \sqrt{5}}{2}\}$ ve bu kümede yer alan kök değerlerinin $x = 0.2$, $x \sim 0.38197$, $x \sim \mathbf{2.6180}$ olduğu görülmektedir.

Sonuç olarak mutlak değerce en büyük kökü veren Teorem 3.1 ile pozitif üst sınırı veren Teorem 2.4 birlikte kullanıldığında reel köklerin alt ve üst sınırları hakkında bilgi sahibi olabiliriz.

Bu duruma bir örnek daha vermek gerekirse, Örnek 2.3.4 de bizim metodumuzla bulduğumuz alt sınır ve Teorem 2.4 den gelen üst sınır birlikte kullanılarak reel köklerin hangi aralıkta olduğu söylenebilir. Örnek 2.3.4 de;

$x^4 + 3x^3 - 3x^2 + 3x - 4 = 0$ polinomum reel köklerin üst sınırı Teorem 2.4 e göre $1 + \sqrt{4} = 3$ olduğu için, polinomun reel köklerinin [-4,3] aralığında olduğunu söylemek mümkündür.

2.5. Metodun Olimpiyat Sorularına Uygulaması

Polinom denklemlerin köklerinin bulunması, kök katsayı ilişkisi, polinomlarda bölme ve çarpanlarına ayrılması Matematik Olimpiyatlarında önemli bir yer tutmaktadır. Tüm bu soru tipleri için bir tek kökün varlığı hakkında bilgi sahibi olmak bile oldukça önemlidir.

2.5.1. Örnek

$x^3 - x^2 - x - \frac{1}{3} = 0$ polinomunun en büyük gerçel kökü nedir? (Tübitak UMO 2005).

A) $\frac{\sqrt[2]{3} - \sqrt[2]{2}}{2}$ B) $\frac{\sqrt[3]{3} - \sqrt[3]{2}}{2}$ C) $\frac{1}{\sqrt[3]{3} - 1}$ D) $\frac{1}{\sqrt[3]{4} - 1}$ E) Hiçbiri

Çözüm

$3a_n = 3a_{n-1} + 3a_{n-2} + a_{n-3}$

$(a_n) = (1, 1, 2, 3.33, 5.66, 9.66, 16.44, \ldots)$

$\left(\frac{a_{n+1}}{a_n}\right) = (1, 2, 1.66, 1.7, 1.705, 1.7011, 1.7027, 1.70238, \ldots)$

Limit değeri 1.70 e yakınsadığı görülmektedir. Şıklara bakıldığında A ve B şıkları 1 den küçük olduğu için C şıkkı 2 den büyük olduğu için elenir. D şıkkı ise 1.7024 olduğu için doğru seçenek D seçeneğidir.

2.5.2. Örnek

$x^5 + x^4 - x^3 - x^2 - 2x - 2 = 0$ polinomunun kaç gerçel kökü vardır? (Tübitak UMO 2002).

Çözüm

$a_n = -a_{n-1} + a_{n-2} + a_{n-3} + 2a_{n-4} + 2a_{n-5}$

$(a_n) = (1, 1, 1, 1, 2, 4, 3, 7, 6, 16, 11, 31, 22, 64, 43, 127, 86, 256, ...)$

$\left(\frac{a_{n+1}}{a_n}\right) = (1, 1, 1, 2, 2, 0.75, 2.33, 0.85, 2.666, 0.68, 2.81, 0.70, 2.90, 0.67, 2.95, 0.67, ...)$

$\frac{a_{n+1}}{a_n}$ dizisinin parçalı yakınsak olduğu görülmektedir bu da bize mutlak değerce en büyük kökten iki tane ve zıt işaretli olduğunu hatırlatmaktadır. Bkz problem 2.2.3. Bu çözüm ile ilgili açıklama 3. Bölümde verilecektir.

$$\lim_{n\to\infty} \frac{a_{n+1}}{a_n} . \frac{a_{n+2}}{a_{n+1}} = \lim_{n\to\infty} \frac{a_{n+2}}{a_n} = 2.9 \text{ x } 0.7$$

$$\sqrt{\lim_{n\to\infty}\left(\frac{a_{n+1}}{a_n}\right)} \cong \sqrt{2.03...} \cong 1.4247$$

Buradan limitin $\sqrt{2}$ ye yaklaştığı ve burada zıt işaretli iki reel kökün varlığından söz edilebilir. Yani ilk denememiz gereken $\pm\sqrt{2}$ köklerinin denklemi sağlayıp sağlamadığıdır bunun için polinom $x^2 - 2$ e bölünür. Sonuç olarak $(x^2 - 2)(x + 1)(x^2 + 1)$ şeklinde çarpanlarına ayrılır ve 3 reel köke sahip olduğu söylenebilir.

2.6. Metodun Nümerik Analiz ile İlişkilendirilmesi

Polinomların yaklaşık değerlerini bulmak için kullanılan nümerik metodların hızlı bir şekilde köke yakınsaması için kökün içinde bulunduğu bir aralık tanımlanmalıdır. Örneğin $x = 2$ köküne ulaşmak ve iterasyonun hızlı yakınsaması için iterasyonun $[1.8, 2.3]$ aralığında uygulanması istenir. Kökler hakkında hiçbir bilgi verilmediği durumda metot çalışmayabilir. Çalışmamızın bize sağladığı kök bilgisinden yola çıkarak diğer mevcut reel köklerin bulundukları yerler de kabaca söyleyebiliriz. Mutlak değerce en büyük reel kök bulunduktan sonra polinomun türevi alınarak elde edilen denkleme aynı algoritma uygulandığında mutlak değerce en büyük apsisli tepe noktası bulunacaktır. Her ne kadar kökler 2. Dereceden büyük polinomlarda her zaman tepe noktasına göre simetri göstermese de kökün bulunduğu yer konusunda (iterasyona başlayacağım uygun bir aralık) bilgi sahibi olmamıza fayda sağlayabilir. Bu metot geliştirilerek tüm reel köklerin nümerik yolla bulunmasına kolaylık getirilebilir.

Olimpiyat sınavlarında da sıklıkla karşılaştığımız yüksek mertebeden polinomun köklerini bulmayı gerektiren sorularda verilen polinomları çarpanlara ayırmak çoğu zaman kolay değildir. Reel köklerin sayısı ve işaretleri konusunda bilgiye ulaşmak mümkündür bunun için çeşitli yöntemler verilmektedir. Olimpiyat sınavında rasyonel bir kök hakkında ulaşılacak basit ve küçücük bir bilgi bile işimizi kolaylaştırmaya yardımcı olacaktır. Bu sebeple bu metodun kullanışlı olacağını düşünmekteyiz.

2.6.1. Örnek

Ç.K. $= \{3, 1, -2\}$ olan 3. derece $x^3 - 2x^2 - 5x + 6 = 0$ denklemini ele alalım. Üçüncü derece bir denklemde mutlak değerce en büyük kökün bulunması diğer köklerin bulunması için yeterlidir. Mutlak değerce en büyük kökü kendi metodumuzla bulduktan sonra denklem polinom bölmesiyle 2. dereceye düşürülür ve 2. derece denklemin kökleri bulunur.

Ancak diğer kökleri de kendi metodumuzla bulmak için 3. derece denklemin katsayılarını Python da yazdığımız programa girerek çalıştırdığımızda mutlak değerce en büyük kökün 3 e yaklaştığı görülür. Diğer kökleri bulmak için polinomun türevi alınıp algoritma tekrar çalıştırıldığında ise 2 ye yakınsayacaktır. Dolayısıyla diğer kökün 3 apsisli noktanın 2 ye göre simetriği olan 1 civarında olacağı söylenebilir. Polinom bölmesi yerine bu algoritma kullanılabilir.

3. BULGULAR

Bu bölümde 2. bölümde verdiğimiz problem ve örneklerde uyguladığımız metot, Teorem 3.1 olarak verilmiş ve ispatı yapılmıştır. Ardından metodun algoritmasını veren bilgisayar kodları verilmiştir.

3.1. Teorem

$A_0 \neq 0$, $A_0, A_1, A_2, \ldots, A_k$ ve $n, k \in Z^+$ olmak üzere n. dereceden polinom denkleminin

$$A_0x^n + A_1x^{n-1} + A_2x^{n-2} + \ldots + A_k = 0 \qquad (1)$$

ve indirgeme bağıntısına sahip (u_n) dizisinin karakteristik denklemi

$$A_0u_{n+k} + A_1u_{n+k-1} + A_2u_{n+k-2} + \ + \ldots + A_ku_n = 0 \qquad (2)$$

olsun. n. dereceden (1) denkleminin kökleri $r_1, r_2, r_3, \ldots, r_n \in \mathbb{C}$ ve $i = 1, 2, 3, \ldots, n$ için r_i sayıları birbirinden farklı olması gerekmeyen, karmaşık veya reel sayılar olsun. Bu köklerden mutlak değerce en büyük kökümüz *reel kök* olsun. Bu durumda

a) $1 < i \leq k$ olmak üzere $-1 < \frac{r_i}{r_1} \leq 1$ için $|r_1| \geq |r_2| \geq |r_3| \geq \ldots |r_k|$ ve $r_1 \in \mathrm{R}$ ise $\lim\limits_{n\to\infty}\frac{u_{n+1}}{u_n} = r_1$

b) $1 < i \leq k$ olmak üzere $-1 = \frac{r_i}{r_1}$ olacak şekilde, r_1 köküne mutlak değerce eşit zıt işaretli en az bir kök bulunsun. Bu köke r_2 diyelim. $|r_1| = |r_2| \geq |r_3| \geq \ldots |r_k|$ ve $r_1, r_2 \in \mathrm{R}$ ve $r_1 > 0 > r_2$ ise $\lim\limits_{n\to\infty}\sqrt{\frac{u_{n+2}}{u_n}} = r_1$

c) $|r_1| \geq |r_2| \geq |r_3| \geq \ldots |r_k|$ ve mutlak değerce en büyük kökün reel olma zorunluluğu olmasın fakat $r_1^t \in \mathrm{R}$ olsun. Bu durumda ise $\lim\limits_{n\to\infty}\frac{u_{n+t}}{u_n} = r_1^t$ olur.

İspat

k. dereceden polinom denkleminin mutlak değerce en büyük kökü yada kökleri reel olsun.

a) $1 < i \leq k$ olmak üzere $-1 < \frac{r_i}{r_1} \leq 1$ için $|r_1| \geq |r_2| \geq |r_3| \geq \ldots |r_k|$ ve $r_1 \in \mathrm{R}$ alalım. u_n dizisinin genel terimi, $u_n = P_1(n)r_1^n + P_2(n)r_2^n + \ldots + P_k(n)r_k^n$ olacaktır. Ardışık terimlerin oranı

$$\frac{u_{n+1}}{u_n} = \frac{P_1(n+1)r_1^{n+1} + P_2(n+1)r_2^{n+1} + \ldots + P_k(n+1)r_k^{n+1}}{P_1(n)r_1^n + P_2(n)r_2^n + \ldots + P_k(n)r_k^n}$$

ile gösterilebilir. Kesrin pay ve paydasını $P_1(n)r_1^n$ terimine bölerek limitini aldığımızda

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n} = \frac{\frac{P_1(n+1)}{P_1(n)}r_1 + \frac{P_2(n+1)}{P_1(n)}r_2\left(\frac{r_2}{r_1}\right)^n + \ldots + \frac{P_k(n+1)}{P_1(n)}r_k\left(\frac{r_k}{r_1}\right)^n}{1 + \frac{P_2(n)}{P_1(n)}\left(\frac{r_2}{r_1}\right)^n + \ldots + \frac{P_k(n)}{P_1(n)}r_k\left(\frac{r_k}{r_1}\right)^n}$$

$1 < i \leq k$ için $-1 < \frac{r_i}{r_1} \leq 1$ olur. Bu durumda $\lim\limits_{n\to\infty}\frac{P_1(n+1)}{P_1(n)}r_1 = r_1$ olarak bulunur. Limit değerin mutlak değerce en büyük reel köke yakınsadığı görülmüş olur.

b) u_n dizisinin parçalı yakınsak olduğu bu durumda. $1 < i \leq k$ olmak üzere $-1 = \frac{r_i}{r_1}$ olacak şekilde, r_1 köküne mutlak değerce eşit zıt işaretli en az bir kök bulunsun. Bu köke r_2 diyelim.

$|r_1| = |r_2| \geq |r_3| \geq \ldots |r_k|$ ve $r_1, r_2 \in \mathrm{R}$ ve $r_1 > 0 > r_2$ ise

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}.\lim_{n\to\infty}\frac{u_{n+2}}{u_{n+1}}=\frac{\frac{P_1(n+1)}{P_1(n)}r_1+\frac{P_2(n+1)}{P_1(n)}r_2\left(-1\right)^n+...+\frac{P_k(n+1)}{P_1(n)}r_k\left(\frac{r_k}{r_1}\right)^n}{1+\frac{P_2(n)}{P_1(n)}\left(-1\right)^n+...+\frac{P_k(n)}{P_1(n)}r_k\left(\frac{r_k}{r_1}\right)^n}x\frac{\frac{P_1(n+2)}{P_1(n+1)}r_1+\frac{P_2(n+2)}{P_1(n+1)}r_2\left(-1\right)^{n+1}+...+\frac{P_k(n+2)}{P_1(n+1)}r_k\left(\frac{r_k}{r_1}\right)^{n+1}}{1+\frac{P_2(n+1)}{P_1(n+1)}\left(\frac{r_2}{r_1}\right)^n+...+\frac{P_k(n+1)}{P_1(n+1)}r_k\left(\frac{r_k}{r_1}\right)^n}$$

$$=\frac{r_1+\frac{P_2(n+1)}{P_1(n)}r_2\left(-1\right)^n}{1+\frac{P_2(n)}{P_1(n)}\left(-1\right)^n}x\frac{\frac{P_1(n+2)}{P_1(n+1)}r_1+\frac{P_2(n+2)}{P_1(n+1)}r_2\left(-1\right)^{n+1}}{1+\frac{P_2(n+1)}{P_1(n+1)}\left(-1\right)^{n+1}}$$

$$=\frac{r_1\left[1-\frac{P_2(n+1)}{P_1(n)}\left(-1\right)^n\right]}{1+\frac{P_2(n)}{P_1(n)}\left(-1\right)^n}x\frac{r_1\left[1-\frac{P_2(n+2)}{P_1(n+1)}\left(-1\right)^{n+1}\right]}{1+\frac{P_2(n+1)}{P_1(n+1)}\left(-1\right)^{n+1}}$$

$$=r_1^2\frac{\left[1+\frac{P_2(n+1)}{P_1(n)}\left(-1\right)^{n+1}\right]}{1+\frac{P_2(n)}{P_1(n)}\left(-1\right)^n}x\frac{\left[1+\frac{P_2(n+2)}{P_1(n+1)}\left(-1\right)^n\right]}{1+\frac{P_2(n+1)}{P_1(n+1)}\left(-1\right)^{n+1}}$$

$$=r_1^2$$

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}.\lim_{n\to\infty}\frac{u_{n+2}}{u_{n+1}}=\lim_{n\to\infty}\frac{u_{n+2}}{u_n}=r_1^2$$

$\sqrt{\lim_{n\to\infty}\frac{u_{n+2}}{u_n}}=r_1$ ile hesaplanır.

Sonuç olarak Polinom denklemlerde mutlak değerce en büyük kökün kompleks sayı olması dışındaki tüm durumlarda bu metot ile en az bir kök bulmak mümkündür.

c) Polinom denkleminin mutlak değerce en büyük kökü yada kökleri reel olma şartı gerekmeksizin.

$1 < i \le k$ olmak üzere $-1 < \frac{r_i}{r_1} \le 1$ için $|r_1| \ge |r_2| \ge |r_3| \ge \ldots |r_k|$ ve $r_1{}^t \in R$ olsun. u_n dizisinin genel terimi, $u_n = P_1(n)r_1^n + P_2(n)r_2^n + ... + P_k(n)r_k^n$ olacaktır. Ardışık terimlerin oranı

$$\frac{u_{n+t}}{u_n}=\frac{P_1(n+t)r_1^{n+t}+P_2(n+t)r_2^{n+t}+...+P_k(n+t)r_k^{n+t}}{P_1(n)r_1^n+P_2(n)r_2^n+...+P_k(n)r_k^n}$$

ile gösterilebilir. Kesrin pay ve paydasını $P_1(n+t)r_1^n$ terimine bölerek limitini aldığımızda

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}=\frac{\frac{P_1(n+t)}{P_1(n+t)}r_1+\frac{P_2(n+t)}{P_1(n+t)}r_2^t\left(\frac{r_2}{r_1}\right)^n+...+\frac{P_k(n+t)}{P_1(n+t)}r_k^t\left(\frac{r_k}{r_1}\right)^n}{\frac{P_1(n)}{P_1(n+t)}\frac{r_1^t}{r_1^t}+\frac{P_2(n)}{P_1(n+t)}\left(\frac{r_2}{r_1}\right)^n+...+\frac{P_k(n)}{P_1(n+t)}\left(\frac{r_k}{r_1}\right)^n}$$

$1 < i \le k$ için $-1 < \frac{r_i}{r_1} \le 1$ olur. Bu durumda $\lim_{n\to\infty}\frac{u_{n+t}}{u_n}=r_1^t$ olarak bulunur.

Limit değerinin mutlak değerce en büyük reel olması gerekmeyen kökün t. kuvvetine yakınsadığı görülmüş olur. Yani kompleks kök mutlak değerce en büyük kök ise ve t. kuvveti reel ise dizinin terimlerinde kökü bulmak mümkün olabilir.

Hata Analizi

İkinci dereceden bir polinomun kökünü bulmak için uygulanan metodun n. adımında yapılan hata, e_n

ile gösterilsin. Yani hata, Hata Analizi

$e_n = \left| \dfrac{Ar_1^{n+1} + Br_2^{n+1}}{Ar_1^n + Br_2^n} - r_1 \right|$ ile ifade edilebilir. Şimdi hata için bir üst sınır bulalım.

$$e_n = \left| \frac{Ar_1^{n+1} + Br_2^{n+1} - Ar_1^{n+1} - Br_2^n.r_1}{Ar_1^n + Br_2^n} \right| \leq \frac{2\,|\,B\,|}{A} \left(\frac{r_2}{r_1} \right)^{n-1} .r_2$$

Hatanın üstel olarak azalması metodun iyi çalıştığını gösterir.

Metodun Bilgisayar Kodları

Bu bölümde uyguladığımız algoritmanın daha yüksek mertebeden polinom denklemlerin çözüm kümelerinde çalışıp çalışmadığını inceleyebilmemizde bize kolaylık sağlaması amacıyla yazdığımız programın kodu yer almaktadır. PYTHON programında yazılan algoritma, katsayılarını girdiğimiz polinom denklemi karakteristik denklem kabul eden indirgeme dizisinin terimlerini ve ardışık terimlerinin oranını çıktı olarak vermektedir. Programda katsayıları girilen n. dereceden indirgeme dizisine başlangıç koşulu olarak, ilk (n-1) terime 1 değeri, n. terime ise 2 değeri atanmaktadır.

```
class Polynomial: # Herhangi bir polinom oluşturmak için kullanacağımız sınıf

    def __init__(self):

        self.coeffs = []

        self.elements = []

        self.degree = 0


    def init_coeffs(self, arr): # katsayıları belirlediğimiz fonksiyon

        self.coeffs = arr

        self.degree = len(arr) - 1


    def init_series(self, num_elems):

    # 0 dan num_elems'e kadar olan elemanları bulup bir array(dizi)'nin içine kaydeder


        self.elements = []

        self.elements = (self.degree -1) * [1] + [2]


        for j in range(num_elems-self.degree):


            acc = 0

            for i in range(1, self.degree+1):

                acc -= self.elements[-i] * self.coeffs[i]/self.coeffs[0]


            self.elements.append(acc)


    def get_ratio(self):

    # 0 dan n e kadar dizinin elemanlarının olduğu array'in

    # sonuncu elemanı(n'inci) ile sondan 1 önceki elemanı(n-1'inci)'nın
```

```
        # oranını verir
                return self.elements[-1]/self.elements[-2]

        def get_value(self, x):
         # P(x) i hesaplar
                accumulator = 0


                for i in range(0, self.degree + 1): # sum over i: a_i * x^i
                        accumulator += self.coeffs[i] * math.pow(x, self.degree - i)
                return accumulator

        def print_info(self):

                ratios = []
                for i in range(len(self.elements)):

                        if i > 0:
                                ratios.append(round(self.elements[i]/self.elements[i-1], 5))
                        else:
                                ratios.append(1)

                print("an= " + str(self.elements))
                print("an+1/an= " + str(ratios))

p1 = Polynomial() # p1 isimli bir polinom oluşturduk


coeffs = [1, -9, 35, -79, 104,-60]
"""
        polinomun katsayılarını belirliyoruz
        yani polinomun derecesi d ise x^dli terimin katsayısı ilki x^(d-1) in ikinci yani
        [x^d, x^(d-1), ..., x^1, x^0] şeklinde gider
"""
coeffs = input("Polinomun katsayılarını aralarında boşluk bırakarak giriniz. ").split(' ')

for i in range(len(coeffs)):
        coeffs[i] = int(coeffs[i])

p1.init_coeffs(coeffs) # coeffs adlı arrayi kullanarak polinomun katsayılarını belirledik
p1.init_series(20) # dizinin ilk n elemanını buldurur. (20 yazarsa 20, 30 yazarsa 30 vs.)
```

```
"""
        p1.get_ratio() son eleman ile sondan önceki elemanın orаınını verir

        yani p1.init_series(20) çağırırsak sonra da p1.get_ratio() çağırdığımızda

        a_20/a_19 değerini verir
"""
p1.print_info()
```

4. BULGULAR

Bu çalışmada aşağıdaki sonuçlara varılmıştır.

Polinom denklemlerin mutlak değerce en büyük kök(ler)ü reel ise, bu denklemi karakteristik denklemi kabul eden u_n indirgeme dizisinin ardışık terimleri oranı bu köke yakınsar.

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}=r_1$$

Polinom denklemlerin mutlak değerce en büyük kökleri ters işaretli ve reel ise, bu denklemi karakteristik denklemi kabul eden parçalı yakınsak u_n indirgeme dizisinin pozitif olan kökü

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}.\lim_{n\to\infty}\frac{u_{n+2}}{u_{n+1}}=\lim_{n\to\infty}\frac{u_{n+2}}{u_n}=r_1^2$$

$\sqrt{\lim_{n\to\infty}\frac{u_{n+2}}{u_n}}=r_1$ ile hesaplanır.

Yöntemin dezavantajı; denklemin mutlak değerce en büyük kökü karmaşık (kompleks) ise metot çalışmaz. Ancak bu karmaşık kökün t. kuvveti reel ise ardışık terimlerin oranı t. kuvvete yakınsar. Ayrıca dizinin kaçıncı adımda köke yakınsadığını bileyemeyiz.

Mutlak değerce en büyük kökü veren Teorem 3.1 ile Pozitif üst sınırı veren Teorem 2.4 birlikte kullanıldığında reel köklerin alt ve üst sınırları hakkında bilgi sahibi olabiliriz.

Okuyuculara önerilerimiz; Bu çalışmada Fibonacci Tavşan probleminden yola çıkarak iki değişken üzerinden kurgulama yapılmıştır. α = Canlının yetişkinliğe ulaşma süresi, β = Canlının yetişkinliğe ulaştıktan sonra hangi aralıklarla doğum yaptığı. Bu probleme üçüncü bir değişken olarak canlının tek seferde dünyaya getirdiği yavru sayısı dahil edilerek daha üst seviyede zorluklar getirilebilir.

Karmaşık köklerin büyüklüğü (modülü, mutlak değeri) polinom grafiklerine nasıl yansımaktadır? Bu bilgiyle birleştirildiğinde metodumuzun farklı alanlarda soruların çözümüne de ışık tutacağını düşünüyoruz.

Ayrıca bu çalışmamızın hata analizi yapılarak genelleştirmelere ulaşılması ve dizinin kaçıncı adımda köke yakınsadığının belirlenmesi beklenmektedir.

KAYNAKLAR

Barbeau, E.J. (1989). Problem Books in Mathematics, Polynomials. Springer Verlag. New York-ABD.

TÜBİTAK UMO (2003). Erişim adresi: (www.tubitak.gov.tr/tr/olimpiyatlar/ulusal-bilim-olimpiyatlari/icerik-matematik) . Son ErişimTarihi:31.04.2020

TÜBİTAK UMO (2005). Erişim adresi: (www.tubitak.gov.tr/tr/olimpiyatlar/ulusal-bilim-olimpiyatlari/icerik-matematik). Son Erişim Tarihi:31.04.2020

Wolframalpha (1987). Erişim adresi: https://www.wolframalpha.com/input/?i=x%5E3%3Dx%5E2%2B1. Son Erişim Tarihi: 31.04.2020