

T.C.
TURGUT ÖZAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOĞRUSAL OLMAYAN CEBİRSEL DENKLEMLERİN
KÖKLERİNİ HESAPLAYAN UYGULAMA GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Hazırlayan
Gürhan USTALI

Tez Danışmanı
Doç. Dr. Ünal GÖKTAŞ

Ankara-2014

Bilimsel Etik Bildirim Sayfası

Turgut Özal Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

-tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,

-görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,

-başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,

-atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,

-kullanılan verilerde herhangi bir tahrifat yapmadığımı

-ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı beyan ederim.

22/01/2014

Gürhan USTALİ

ONAY

Gürhan USTALİ tarafından hazırlanan “Doğrusal Olmayan Cebirsel Denklemlerin Köklerini Hesaplayan Uygulama Geliştirilmesi” başlıklı bu çalışma, 22/01/2014 tarihinde yapılan savunma sınavı sonucunda oybirliği ile başarılı bulunarak jürimiz tarafından Elektrik ve Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans tezi olarak kabul edilmiştir.

.....

Doç. Dr. Ünal GÖKTAŞ (Başkan)

.....

Doç. Dr. İlyas ÇANKAYA

.....

Yrd. Doç. Dr. Tahir MALAS

ÖNSÖZ

Bu tez çalışmasında amaç, doğrusal olmayan cebirsel denklemlerin köklerini, pertürbasyon ve iterasyon yöntemlerini kullanarak hesaplayan bir *Mathematica* uygulaması geliştirmektir. Bunun için hazırlanan yazılım paketi, dokuz adet fonksiyondan oluşmaktadır. Geliştirilen uygulamada, verilen denklemin köklerini bulmak için, denklem polinom ise başlangıç değeri tahmini yapılarak, denklem polinom değil ise verilen başlangıç değeri kullanılarak, üretilen pertürbasyon ve iterasyon formülleri ile gerekli hesaplamalar yapılmaktadır.

Bu tezin hazırlanmasında maddi, manevi her türlü desteğiyle yanımda olan danışman hocam Doç. Dr. Ünal GÖKTAŞ'a, mesai arkadaşlarım Edip ŞENYÜREK ve Orhan DEMİRSÖZ'e ve son olarak çok kıymetli aileme sonsuz teşekkürlerimi sunarım.

ÖZET

USTALİ, Gürhan, Doğrusal olmayan cebirsel denklemlerin köklerini hesaplayan uygulama geliştirilmesi, Yüksek Lisans Tezi, Ankara, 2014.

Gerçek hayattaki problemlerin çözümüne yönelik öncelikle belirli matematiksel modellerin hazırlanması büyük önem arz etmektedir. Problemin matematik modeli bazen doğrusal, çoğunlukla da doğrusal olmayan denklem ya da denklem grupları şeklinde ortaya çıkmaktadır. Bu matematik modellerin, diğer bir ifadeyle problemin indirgenmiş olduğu denklemlerin yaklaşık çözümlerine ulaşabilmek için bu güne kadar birçok yöntem ve algoritma geliştirilmiş ve uygulanmıştır. En çok bilinen yöntemler, Newton-Raphson metodu ve Householder iterasyonu gibi, denklemlerin sayısal çözümlerini hesaplamak için kullanılan yöntemlerdir.

Bu çalışmada geliştirilen *Mathematica* uygulamasında, polinom denklemler için köklerin büyüklüklerinin tahmini ve pertürbasyon-iterasyon metodları kullanılarak doğrusal olmayan cebirsel denklemlerin köklerinin hesaplanmasını sağlayan fonksiyonlar geliştirilmiştir. Uygulama çalıştırıldığında, önce kök bulma algoritması üretilmekte ve daha sonra denklemin polinom olup olmadığının tespiti yapılmaktadır. Eğer denklem polinom ise kök bulmaya başlamadan önce başlangıç değeri tahmini yapılmaktadır. Üretilmiş olan algoritma ve verilmiş olan ya da hesaplanan tahmini başlangıç değeri kullanılarak verilen denklemin kökleri hesaplanmaktadır.

Anahtar Kelimeler

1. Pertürbasyon ve iterasyon metodları
2. Doğrusal olmayan cebirsel denklemler
3. Polinomlar
4. Kök değerleri

ABSTRACT

USTALI, Gürhan, Developing an application for the computation of the roots of nonlinear algebraic equations, Master's Thesis, Ankara, 2014.

The mathematical modelling of the real life problems is very essential in order to find the solution to the problem. The mathematical model of the problem sometimes leads to a linear, but mostly to a nonlinear equation or a system of equations. A lot of algorithms have been developed and applied for finding the solutions of these type of equations. Well-known algorithms that are used to find the solutions of these equations are Newton-Raphson and Householder Iteration.

In the developed *Mathematica* application, by using perturbation-iteration methods, functions for the estimation of the magnitude of the roots of polynomial equations and for the computation of the roots of nonlinear algebraic equations are developed. When the application is called, first, the algorithm to find the roots is determined and then the check, for whether the equation is polynomial or not, takes place. If the equation is polynomial, an estimation for the initial value is computed. With the generated algorithm and with the given or estimated initial value, the roots of the given equation are then computed.

Key Words

1. Perturbation-Iteration algorithms
2. Non-Linear algebraic equations
3. Polynomials
4. Values of root

İÇİNDEKİLER

ÖNSÖZ.....	i
ÖZET.....	ii
ABSTRACT	iii
İÇİNDEKİLER	iv
SİMGE VE KISALTMALAR.....	vii
TABLolar	viii
ŞEKİLLER	x
GİRİŞ	1

BİRİNCİ BÖLÜM

PERTÜRBASYON-İTERASYON ALGORİTMALARI

1.1. Tek Düzeltme Terimli Pertürbasyon Algoritmaları.....	5
1.2. Çift Düzeltme Terimli Pertürbasyon Algoritmaları.....	8
1.3. Üç Düzeltme Terimli Pertürbasyon Algoritmaları	10

İKİNCİ BÖLÜM

POLİNOMLAR İÇİN BAŞLANGIÇ KÖK TAHMİNİ

2.1. Tüm Katsayıların Benzer Büyüklükte Olduğu Polinomlar	12
2.2. Katsayılarından Biri Diğerlerine Göre Büyük Olan Polinomlar	13
2.3. Katsayılarından Biri Diğerlerine Göre Küçük Olan Polinomlar	13
2.4. Katsayılarından İki Diğeri Diğerlerine Göre Büyük Olan Polinomlar	14

ÜÇÜNCÜ BÖLÜM

PIAFindRoots PAKETİ

3.1. PIAFindRoots Paketi ve Kullanımı	17
3.1.1. Uygulamanın Akış Şeması.....	19
3.1.2. Kurulum ve Yükleme.....	23
3.1.3. PIAFindRoots ve GeneratePIA Fonksiyonları	24

DÖRDÜNCÜ BÖLÜM**ÖRNEKLER**

4.1. Örnekler	26
SONUÇ	30
KAYNAKÇA	31
EKLER	34

SİMGE VE KISALTMALAR

PIA: Pertürbasyon İterasyon Algoritması

ϵ : Pertürbasyon parametresi

x_0 : Aranana kökü elde etmek için gereken başlangıç değeri

x_i : i 'inci iterasyon sonucu hesaplanan kök değeri

TABLOLAR

Tablo 1: $e^{-x} - x = 0$ denkleminin kökleri [26]	28
Tablo 2: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26].....	28
Tablo 3: $x^3 + x^2 - x = 2$ denkleminin kökleri [2].....	28
Tablo 4: $x^3 + 0.5x^2 - x + 1200 = 0$ denkleminin kökleri [2].....	29
Tablo 5: $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$ denkleminin kökleri [29]	29
Tablo 6: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26].....	34
Tablo 7: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26].....	34
Tablo 8: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26].....	34
Tablo 9: $x^3 + 2x^2 - x = 1$ denkleminin kökleri [2]	35
Tablo 10: $x^3 + 0.5x^2 - x = -2$ denkleminin kökleri [2].....	35
Tablo 11: $x^3 + 0.5x^2 - 1.1x = -0.7$ denkleminin kökleri [2]	35
Tablo 12: $x^3 + 0.8x^2 + 1.4x = -0.6$ denkleminin kökleri [2].....	35
Tablo 13: $x^3 + x^2 + x = -1$ denkleminin kökleri [2]	36
Tablo 14: $x^4 - 1.2x^3 + 0.9x^2 + 0.8x + 1.3 = 0$ denkleminin kökleri [2]	36
Tablo 15: $x^4 + 0.6x^3 - 1.1x^2 + 0.8x - 0.6 = 0$ denkleminin kökleri [2]	36
Tablo 16: $x^4 + 1.4x^3 - 1.5x^2 + 0.8x + 0.6 = 0$ denkleminin kökleri [2]	37
Tablo 17: $x^4 - 1.4x^3 - 1.5x^2 - 0.8x + 1.3 = 0$ denkleminin kökleri [2]	37
Tablo 18: $x^4 - 0.9x^3 - 1.3x^2 - 0.6x + 0.7 = 0$ denkleminin kökleri [2]	37
Tablo 19: $x^4 + x^3 + 0.3x^2 - 0.6x - 1.5 = 0$ denkleminin kökleri [2]	38
Tablo 20: $x^4 + 0.8x^3 + 1.3x^2 + 1.4x - 1.1 = 0$ denkleminin kökleri [2]	38
Tablo 21: $x^4 + 1.6x^3 + 0.5x^2 - 1.4x - 0.4 = 0$ denkleminin kökleri [2]	38
Tablo 22: $x^3 + 12x^2 + x - 1.5 = 0$ denkleminin kökleri [2].....	38
Tablo 23: $x^3 + 12x^2 + x - 1.5 = 0$ denkleminin kökleri [2].....	38
Tablo 24: $x^4 + 1100x^3 - x^2 + 1.2x - 0.7 = 0$ denkleminin kökleri [2]	39
Tablo 25: $x^5 + 1.1x^4 + 0.8x^3 - 0.6x^2 - 116x - 2 = 0$ denkleminin kökleri [2]	39
Tablo 26: $x^5 + 1.1x^4 + 0.8x^3 - 0.6x^2 - 116x - 2 = 0$ denkleminin kökleri [2]	39
Tablo 27: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29].....	39
Tablo 28: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29].....	39

Tablo 29: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29].....	39
Tablo 30: $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$ denkleminin kökleri [29]	40
Tablo 31: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]	40
Tablo 32: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]	40
Tablo 33: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]	40
Tablo 34: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29].....	40
Tablo 35: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29].....	41
Tablo 36: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29].....	41
Tablo 37: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]	41
Tablo 38: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]	41
Tablo 39: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]	42
Tablo 40: $x^7 - 1.3x^5 + 101x^4 + 0.8x^2 + 140000 = 0$ denkleminin kökleri [29].....	42
Tablo 41: $1.3x^5 + 61x^4 + 0.8x^2 - 900x + 1.4 = 0$ denkleminin kökleri [29].....	43
Tablo 42: $1.3x^5 + 61x^4 + 0.8x^2 - 900x + 1.4 = 0$ denkleminin kökleri [29].....	43
Tablo 43: $x^7 - 56x^6 - 1.3x^5 + 1967x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]	43

ŞEKİLLER

Şekil 1: Fonksiyon listesi.....	18
Şekil 2: PIAFindRoots fonksiyonunun kullanımı	18
Şekil 3: PIAFindRoots fonksiyonu varsayılan değerler	19
Şekil 4: EstimateRootsOfPolynomials fonksiyonunun kullanımı	20
Şekil 5: PIAFindRoots fonksiyonunun örnek kullanımı ve çıktıları	21
Şekil 6: PIAFindRoots akış şeması	22
Şekil 7: PIAFindRoots paketi kurulumu.....	23
Şekil 8: Get komutunun kullanımı.....	23
Şekil 9: PIAFindRoots fonksiyonunun örnek kullanımı ve çıktıları	26

GİRİŞ

Mühendislikten ekonomiye kadar tüm bilim dallarında, hayata dair olayların ve oluşumların çeşitli bilimsel modellerle ifade edildiği görülür. Bu modeller, insanoğlunun kullanımına sunulacak her türlü makine, cihaz, yapı ve sistemin tasarımında ve işletilmesinde kullanılır. Gerçek hayat problemlerinin çözümü adına bilimsel yöntemlerle inceleme yapılırken, değeri değıştikçe (bir titreşim sistemine uygulanan kuvvet gibi), olay ve oluşumların seyrini değıştiren büyüklükler bulunur. Sürekli değışen bu büyüklükler değışken olarak adlandırılır. Değışkenlerin birden fazla olması, değışkenler arası ilişkiler, olay ve oluşumlar hakkında daha özel yorumlamalara imkân verir. Değışkenler arası ilişkiler genellikle tablolar, grafikler, cebirsel veya diferansiyel denklemler ya da denklem sistemleri ile ifade edilir [1].

Gerçek hayat problemlerinin matematiksel modelleri çıkarılırken, birçok problemin doğrusal olmayan cebirsel denklem şeklinde modellenebileceği görülmüştür [2-3]. Örneğin titreşim üreten bir sistemin doğal frekanslarını bulmak, köklerinin çözümlenmesini gerektiren bir polinom denkleme indirgenebilir [2].

Problemler, denklemler şeklinde ifade edilip tanımlandıktan sonra çözümünün bulunması, başka bir ifadeyle denklemin çözüm kümesini oluşturan köklerinin bulunması gerekir. Doğrusal olan denklemler için bu kökler daha kolay bulunabilirken, doğrusal olmayan denklemler için kökleri bulmak oldukça zordur.

Denklemlerin köklerinin bulunmasına ilişkin birçok yöntem geliştirilmiştir. Özellikle bilgisayarların yaygın kullanımıyla birlikte sayısal yöntemler, denklemlerin çözümleri için en çok yararlanılan yöntemlerden olmuştur [2, 4]. Bunlara, lineer interpolasyon, sabit nokta iterasyonu ve Newton-Raphson yöntemi gibi metotlar örnek olarak verilebilir [4-5].

Pertürbasyon teorisi, cebirsel denklemlerin, diferansiyel denklemlerin ve fark denklemlerinin yaklaşık çözümlerini hesaplamak için yaygın olarak kullanılmaktadır [3, 5-10,29-30]. Pertürbasyon teorisi, tam olarak çözülemeyen bir problemin yaklaşık çözümünü bulmak için kullanılan matematiksel metotlar içerir [6, 11]. Eğer problem, bir “küçük” terim eklenerek tam olarak çözülebilen probleme indirgenebiliyorsa pertürbasyon teorisi uygulanabilir demektir. Pertürbasyon teorisinin amacı, bir problemin çözümünü, tam olarak çözülebilen başka bir problemden sapmayı ölçerek, bir “küçük” parametrenin kuvvet serisi cinsinden bulmaktır. Kuvvet serisindeki ilk terim, tam olarak çözülebilen problemin çözümü iken, diğer terimler, çözümde başlangıç problemine göre sapmayı tanımlar. Pertürbasyon yöntemlerinde önce kök, bir pertürbasyon serisine açılır ve daha sonra her bir düzeltme terimi yaklaşık kökü bulabilmek için bir kere hesaplanır [12-13]. Ancak düzeltmeler üzerinde hiç bir iterasyon uygulanmamaktadır.

Pertürbasyon metotlarının uygulanmasındaki zorluklar, denklemde küçük bir parametreye ihtiyaç duyulması veya denklemde küçük bir parametrenin yapay olarak tanımlanması zorunluluğudur. Bu nedenle pertürbasyon metotlarının oluşturduğu dezavantajları gidermek üzere iterasyon-pertürbasyon metodu [14], iterasyon tekniği [15-17], harmonik denge metodu [18-20], parametre açılımı metodu [21], Lindstedt-Poincaré metodu [22-24] ve homotopi pertürbasyon metodu [25] gibi metotlar geliştirilmiştir [11].

Lineer olmayan sistemlerin çözümleri için geçerli alternatif bir yöntem olarak pertürbasyon-iterasyon (veya iterasyon-pertürbasyon) metotlarının kullanıldığı görülür [11]. Genellikle denklemler iterasyon işlemlerini uygulamadan önce alternatif bir forma sokulur. Bazı algoritmalar ise sadece özel problemler üzerinde çalışmak için geliştirilmiştir. Ön dönüşüm ve başlangıç kabulleri gerektirmeyen ve bütün denklem tipleri için geçerli olan bir yaklaşım olarak pertürbasyon-iterasyon metodu geliştirilmiştir [26].

Pertürbasyon-iterasyon yöntemi, pertürbasyon seri açılımındaki terim sayısına (n), Taylor seri açılımındaki terim sayısına (m) ve sonuçta elde edilen denklemlerin nasıl ayrıştırıldığına bağlı olarak, diğer iterasyon formülleri ile birlikte iyi bilinen

Newton-Raphson ve Householder iterasyon formüllerini de içermektedir [4, 7-8, 26-28].

İkinci bölümde de belirtildiği gibi iterasyon algoritmalarında bir köke yaklaşmak için başlangıç değerinin belirlenmesi ya da tahmin edilmesi oldukça önemlidir. Polinomlar ile çalışılırken, pertürbasyon teorisinin büyüklük sırası kavramı kullanılarak iyi bir başlangıç değeri tahmini elde etmeye yönelik, Pakdemirli ve Yurtsever [2] polinomları, terimlerinin katsayı büyüklüklerine göre sınıflandırmak üzere iki teorem geliştirmişlerdir. Daha sonra Pakdemirli ve Elmas [29] iki farklı sınıfı daha içeren iki teorem daha geliştirmiştir.

Denklemlerin köklerinin bulunmasına ilişkin yöntemlerin temel noktası, yaklaşık çözümler üretmesidir [4]. Bu durum, beraberinde iki problemin ortaya çıkmasına sebep olur. Bunlardan ilki yaklaşık çözümler bulunurken hatanın en aza indirilmesi, ikincisi ise çözüme yaklaşmak için gereken iterasyon sayısının en aza indirilmesidir [5-6].

Bu çalışmada pertürbasyon ve iterasyon metotlarının birlikte ele alındığı pertürbasyon-iterasyon metodu kullanılmıştır [26]. Aslında Newton-Raphson ve ikinci düzeltmesi, Householder iterasyonu gibi iyi bilinen formüller pertürbasyon-iterasyon metotlarının da bir sonucudur. Pertürbasyon açılımında ele alınan terimlerin sayısına (n), Taylor seri açılımdaki terimlerin sayısına (m) ve elde edilen denklemlerin nasıl ayrıştırıldığına bağlı olarak farklı iterasyon formülleri oluşturulabilir [5, 7-8, 12-13, 26-27]. Polinom denklemlerinde kök bulunurken gereken başlangıç değeri tahmini için Pakdemirli, Yurtsever ve Elmas tarafından geliştirilen teoremlerden yararlanılmıştır [2, 29].

Bu tez çalışmasında, polinomları ele alırken kullanılan başlangıç kök değeri tahmininin hesaplanması da dâhil olmak üzere, $1 \leq n \leq m$ şartını sağlayan genel n ve m tam sayı değerleri için pertürbasyon-iterasyon algoritmalarının tüm basamaklarını otomatik olarak yürüten, PIAFindRoots adını verdiğimiz *Mathematica* yazılım paketi geliştirilmiştir.

Bu tez çalışmasının içeriđi řu řekilde dzenlenmiřtir. Doğrusal olmayan cebirsel denklemlerin yaklaşık çözümlerini bulmak için kullanılan pertürbasyon-iterasyon algoritmaları birinci bölümde, polinomlar için başlangıç kök tahmininin detayları ikinci bölümde incelenmektedir. Üçüncü bölümde *Mathematica* ile geliştirilen PIAFindRoots isimli yazılım paketi ve kullanımı hakkında bilgi verilmiştir. Dördüncü bölümde, yapılan testlerde elde edilen sonuçlara yer verilmiştir. Son olarak elde edilen sonuçlar ve gelecekte yapılması muhtemel çalışmalarla ilgili önerilerde bulunulmuştur.

BİRİNCİ BÖLÜM

PERTÜRBASYON-İTERASYON ALGORİTMALARI

Cebirsel denklemlerin yaklaşık çözümlerinin bulunmasında pertürbasyon metotlarından yararlanılması yeni değildir. Örneğin Hinch'in kitabında [10] pertürbasyon metotları ve iterasyon algoritmaları ayrı ayrı ele alınmıştır. Pakdemirli ve Boyacı [26] pertürbasyon ve iterasyon algoritmalarını beraberce ele almışlardır.

Bu bölümde pertürbasyon teorisi kullanılarak çeşitli iterasyon formülleri elde edilmektedir. Pertürbasyon açılımındaki düzeltme terimlerinin sayısına ve Taylor açılımındaki terimlerin sayısına bağlı olarak, farklı kök bulma formülleri oluşturulabilir. Bu formüller algoritma olarak adlandırılmış ve düzeltme terimlerinin sayısına göre tek düzeltme terimli, çift ve üç düzeltme terimli pertürbasyon algoritmaları olmak üzere sınıflandırılmışlardır [12-13, 28]. Pertürbasyon açılımından sonra ortaya çıkan denklemleri ayrıştırma biçimi de kök bulma formüllerini değiştirir [26]. Ardışık iterasyonlar sonucunda hesaplanan köklerin arasındaki fark, belirlenen tolerans değerinin altına düşünceye kadar iterasyon hesaplamalarına devam edilir.

1.1. Tek Düzeltme Terimli Pertürbasyon Algoritmaları

Pertürbasyon açılımındaki terimlerin sayısına, Taylor serisi açılımındaki terimlerin sayısına ve sonuçlanan denklemlerin nasıl gruplandığına bağlı olarak farklı iterasyon formülleri ortaya çıkmıştır [26]. n değeri pertürbasyon serisindeki düzeltme

terim sayısı olmak üzere, tek düzeltme terimli pertürbasyon algoritmaları $n = 1$ değerinin kullanıldığı durumlardır.

Doğrusal olmayan bir denklem olan

$$f(x) = 0 \quad (1)$$

denklemini ele alalım. Kökleri bulabilmek için başlangıçta varsayılan köke tek düzeltme terimi eklenebilir. Bu durumda,

$$x = x_0 + \epsilon x_1 \quad (2)$$

olacaktır. Bu denklemde, x_1 aranan kök iken bu kökün elde edilmesi için gereken başlangıç değeri x_0 ve ϵ ise pozitif ve 1'den çok çok küçük pertürbasyon parametresidir. Yapılan hesaplama sonucunda bulunan x değeri artık x_1 yerine, x_1 değeri de artık x_0 yerine kullanılmaktadır. Yeni değerlerle yeni bir x değeri hesaplanmaktadır. Bu işlem bir önceki iterasyonda hesaplanan x değeri ile hesaplanan yeni x değeri arasındaki fark başlangıçta belirlenen tolerans değeri aralığında oluncaya kadar tekrarlanır. (2) eşitliğindeki ifade (1) de yerine konulup, m değeri Taylor seri açılımındaki ϵ 'un derecesi olmak üzere, $m = 1$ alınarak Taylor serisi açılımı yapıldığında:

$$f(x_0) + f'(x_0) \epsilon x_1 \cong 0 \quad (3)$$

eşitliği elde edilir. Bu denklem, ϵx_1 için çözümlendiğinde ve (2) denkleminde yerine konulduğunda

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (4)$$

eşitliği elde edilir. Bu eşitliğin tolerans değeri elde edilene kadar ardışık tekrarlandığı var sayılırsa, genel bir ifade olarak

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (5)$$

eşitliğine ulaşılır.

Elde edilen bu denklem, iterasyon denklemi olup Newton-Raphson olarak adlandırılır.

Householder'a göre klasik pertürbasyonlarda (4) nolu denklemdeki düzeltme terimi hesaplandıktan sonra tolerans değerine kadar yineleme yapılacak olursa (5) nolu denklem elde edilecektir [26]. Pertürbasyon teorisi bağlamında açılımın genişlemesi

$$\left| \frac{f(x_n)}{f'(x_n)} \right| \ll |x_n| \quad \text{yada} \quad \left| \frac{f(x_n)}{x_n f'(x_n)} \right| \ll 1 \quad (6)$$

durumunda geçerli olacaktır. Bu karşılaştırmada \ll çok çok küçük ifadesi için kullanılmaktadır. Pakdemirli ve Boyacı [26], 0.2'ye kadar olan değerlerin 1'den çok çok küçük kabul edilebileceğini ve iterasyonun bu koşullar altında aranılan köke yakınsayabileceğini ifade etmişlerdir.

Alternatif olarak $m = 2$ alınarak üç terimli Taylor serisi açılımı yapıldığında

$$f(x_0) + f'(x_0)\epsilon x_1 + \frac{f''(x_0)}{2}\epsilon^2 x_1^2 \cong 0 \quad (7)$$

denklemini elde edilir. ϵx_1 çözümlenip, (2) nolu denklemde yerine konulduğunda farklı bir iterasyon denklemi olarak

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \mp \frac{1}{f''(x_n)} \sqrt{(f'(x_n))^2 - 2f(x_n)f''(x_n)} \quad (8)$$

elde edilir. Burada son iki terimin toplamının birinci düzeltme terimi olduğu dikkati çekmektedir. Bu ifade Newton-Raphson metodunun düzeltilmesi değildir. Her ne kadar türetilen bu denklem pertürbasyon yönüyle incelendiyse de bu metot bir geometrik ortalama metodudur [26].

Newton-Raphson metodunda köke yakınsama tanjant çizgileri yoluyla yapılmaktadır ve tanjant parabolü yoluyla köke yakınsama vardır. Ancak $f'(x_n) \cong 0$ olduğu zaman Newton-Raphson algoritmasında yakınsama problemi oluşmaktadır [26]. Aynı şekilde (8) nolu denklemde $f''(x_n) \cong 0$ olduğu zaman yakınsama problemi ortaya çıkmaktadır [26].

1.2. Çift Düzeltme Terimli Pertürbasyon Algoritmaları

Çift düzeltme terimi yaklaşımında tek düzeltme terimli yaklaşıma bir düzeltme terimi daha eklenmektedir:

$$x = x_0 + \epsilon x_1 + \epsilon^2 x_2 \quad (9)$$

(9) denklemini (1) içine yerleştirilerek Taylor seri açılımı yapıldığında:

$$f(x_0) + f'(x_0)(\epsilon x_1 + \epsilon^2 x_2) + \frac{f''(x_0)}{2}(\epsilon x_1 + \epsilon^2 x_2)^2 \cong 0 \quad (10)$$

elde edilir. Yüksek mertebeli terimler göz ardı edilerek son terimin karesi değerlendirildiğinde

$$f(x_0) + \epsilon x_1 f'(x_0) + \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) \cong 0 \quad (11)$$

elde edilir. (11) nolu denklem,

$$f(x_0) + \epsilon x_1 f'(x_0) = 0, \quad \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) = 0 \quad (12)$$

şeklinde bölünebilir. Bu denklemler sırasıyla ϵx_1 ve $\epsilon^2 x_2$ için çözümlendiğinde

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(x_n)(f(x_n))^2}{2(f'(x_n))^3} \quad (13)$$

algoritması elde edilmektedir.

Elde edilen bu algoritma Householder iterasyon algoritması olarak bilinmektedir [26]. Bu algoritma aynı zamanda Adomian ayrıştırma metodu ile de türetilebilir [26-27]. Pertürbasyon kuramı yönüyle bakıldığında, Taylor serisi açılımı terimleri ile pertürbasyon açılımı terimleri daha tutarlı olduğundan, (8) nolu denkleme

göre (13) nolu Householder İterasyon denklemi daha iyidir. Bununla birlikte, (13) nolu iterasyonun (8) nolu formüle kıyasla ilave bir düzeltme terimi bulunmaktadır. Ayrıca, (8) nolu denklemin kolayca atılamayan geometrik kökleri vardır.

Çift düzeltme terimli bir diğer örnek ise (9) nolu denklemde kullanıldığı gibi Taylor açılımındaki bir fazla terim ile

$$f(x_0) + f'(x_0)(\epsilon x_1 + \epsilon^2 x_2) + \frac{f''(x_0)}{2}(\epsilon x_1 + \epsilon^2 x_2)^2 + \frac{f'''(x_0)}{6}(\epsilon x_1 + \epsilon^2 x_2)^3 \cong 0 \quad (14)$$

denklemi elde edilir. Sıralarına göre terimler gruplanınca

$$f(x_0) + \epsilon x_1 f'(x_0) + \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) + \epsilon^3 \left(x_1 x_2 f''(x_0) + x_1^3 \frac{f'''(x_0)}{6} \right) \cong 0 \quad (15)$$

elde edilir. Farklı bir gruplandırma yapılacak olursa,

$$f(x_0) + \epsilon x_1 f'(x_0) = 0 \text{ ve} \\ \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) + \epsilon^3 \left(x_1 x_2 f''(x_0) + x_1^3 \frac{f'''(x_0)}{6} \right) = 0 \quad (16)$$

elde edilecektir. Her iki denklem ϵx_1 ve $\epsilon^2 x_2$ için çözüldüğünde:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2}{6(f'(x_n))^2} \frac{3f'(x_n)f''(x_n) - f(x_n)f'''(x_n)}{(f'(x_n))^2 - f(x_n)f''(x_n)} \quad (17)$$

denklemi elde edilecektir. Elde edilen bu son iterasyonda Newton-Raphson algoritmasına ek bir düzeltme terimi vardır [26].

1.3. Üç Düzeltme Terimli Pertürbasyon Algoritmaları

Üç düzeltme terimi yaklaşımında ise çift düzeltme terimli yaklaşıma bir düzeltme terimi daha eklenmektedir:

$$x = x_0 + \epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3. \quad (18)$$

(18) denklemini (1) içine yerleştirilerek Taylor serisi açılımı yapıldığında

$$\begin{aligned} f(x_0) + f'(x_0)(\epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3) + \frac{f''(x_0)}{2}(\epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3)^2 \\ + \frac{f'''(x_0)}{6}(\epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3)^3 \cong 0 \end{aligned} \quad (19)$$

elde edilir. Sıralarına göre terimler gruplandığında

$$\begin{aligned} f(x_0) + \epsilon x_1 f'(x_0) + \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) \\ + \epsilon^3 \left(x_3 f'(x_0) + x_1 x_2 f''(x_0) + x_1^3 \frac{f'''(x_0)}{6} \right) \cong 0 \end{aligned} \quad (20)$$

elde edilir. Bu denklem de ayrıştırıldığında

$$\begin{aligned} f(x_0) + \epsilon x_1 f'(x_0) &= 0, \\ \epsilon^2 \left(x_2 f'(x_0) + x_1^2 \frac{f''(x_0)}{2} \right) &= 0, \\ \epsilon^3 \left(x_3 f'(x_0) + x_1 x_2 f''(x_0) + x_1^3 \frac{f'''(x_0)}{6} \right) &= 0 \end{aligned} \quad (21)$$

elde edilecektir. (21) nolu denklemler, $\epsilon x_1, \epsilon^2 x_2, \epsilon^3 x_3$ için (18) nolu denklemde yerlerine konulup çözümlendiğinde

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} + (f(x_n))^3 \frac{f'(x_n) f'''(x_n) - 3(f''(x_n))^2}{6(f'(x_n))^5} \quad (22)$$

elde edilecektir. Bu denklemde son terimin Householder İterasyon algoritmasının düzeltme terimi olduğu görülmektedir [27].

İKİNCİ BÖLÜM

POLİNOMLAR İÇİN BAŞLANGIÇ KÖK TAHMİNİ

Pakdemirli ve arkadaşları [2, 29] tarafından pertürbasyon yöntemleri kullanılarak polinom denklemlerinin köklerini bulmaya yönelik iyi bir başlangıç değeri tahmini için bazı teoremler geliştirilmiştir.

İterasyon algoritmalarında bir köke yaklaşmak için başlangıç tahmini oldukça önemlidir. Polinomlardaki terimlerin katsayılarının büyüklüğü bu katsayıların logaritmasına göre değerlendirilmektedir. Polinomlar ile çalışırken Pakdemirli ve Yurtsever [2], polinomları terimlerin katsayılarının büyüklüğüne göre sınıflandırmışlardır. Bunun için bütün terimlerin katsayılarının benzer olduğu ve terimlerden bir tanesinin diğerlerine göre çok büyük olduğu iki teorem geliştirmişlerdir. Daha sonra Pakdemirli ve Elmas [29], ilave iki teorem daha geliştirmişlerdir. Bu teoremler, pertürbasyon teorisinin büyüklük sırası kavramına dayanmakta ve iyi bir başlangıç tahmini elde edebilmek için kullanılmaktadır.

2.1. Tüm Katsayıların Benzer Büyüklükte Olduğu Polinomlar

Teorem: Tüm katsayıların benzer büyüklükte olduğu $a_i (i = 0, 1, \dots, n)$ katsayıları benzer büyüklükte olmak üzere

$$a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0 = 0, \quad (23)$$

şeklindeki polinom denklemlerinde kök $O(1)$ büyüklüğündedir [2].

Örneğin; $x^3 + x^2 - x - 2 = 0$, denkleminde polinomun her bir teriminin katsayısının logaritması alındığında, katsayıların benzer bir büyüklükte olduğu görülebilir.

2.2.Katsayılarından Biri Diğerlerine Göre Büyük Olan Polinomlar

Teorem: Katsayılardan biri diğerlerine göre büyük ve diğer tüm katsayıların benzer büyüklük mertebesinde olması durumunda, yani

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_m x^m + \dots + a_1 x + a_0 = 0 \quad (28)$$

denkleminde eğer $a_m \sim O\left(\frac{1}{\epsilon^k}\right)$ ($k > 0$) olup, diğer katsayılar $O(1)$ ise, bu tip polinomlarda $m \neq 0$ ise kök $O\left(\epsilon^{k/m}\right)$ büyüklüğünde $m \neq n$ ise kök $O\left(\frac{1}{\epsilon^{k/(n-m)}}\right)$ büyüklüğünde olacaktır [2].

Örneğin; $x^3 + 0.5x^2 - x + 1200 = 0$, denkleminde, polinomun her bir teriminin katsayısının logaritması alındığında sabit terimin katsayısından (1200) elde edilen büyüklüğün diğerlerine göre çok çok büyük olduğu görülebilir.

2.3. Katsayılarından Biri Diğerlerine Göre Küçük Olan Polinomlar

Teorem: Katsayılardan biri diğerlerine göre küçük ve diğer tüm katsayıların benzer büyüklük mertebesinde olduğu [29],

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_m x^m + \dots + a_1 x + a_0 = 0,$$

$$a_m \sim O(\epsilon^k), \epsilon \ll 1, k > 0, a_i \sim O(1), i \neq m, \quad (32)$$

şeklindeki polinomlarda köklerin büyüklüklerinin tespiti sırasında

- a) $m \neq 0$ ve $m \neq n$ ise; $x \sim O(1)$
- b) $m = n$ ise $n r - k = (n - 1)r$ yada $r = k, r > 0$ olmak üzere $x \sim O(1/\epsilon^r)$
- c) $m = 0$ ise $r = k$ olmak üzere $x \sim O(\epsilon^r)$

Örneğin; $x^6 - 1.1x^4 + 0.02x^3 - x + 1.1 = 0$, denkleminde, polinomun her bir teriminin katsayısının logaritması alındığında derecesi 3 olan terimden (0.02) elde edilen büyüklüğün diğerlerine göre çok çok küçük olduğu görülebilir.

2.4. Katsayılarından İki Diğeri Göre Büyük Olan Polinomlar

Teorem: Katsayılardan ikisi diğerlerine göre büyük ve diğer tüm katsayıların benzer büyüklük mertebesinde olması durumunda, bu tip polinomlarda köklerin büyüklüklerinin tespiti için [29]

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_m x^m + \dots + a_p x^p + \dots + a_1 x + a_0 = 0 \quad (38)$$

denkleminde $a_m \sim O(1/\epsilon^{k_1}), a_p \sim O(1/\epsilon^{k_2}), \epsilon \ll 1, m > p, k_1 > 0, k_2 > 0$ ve diğer tüm katsayıların $O(1)$ olması halinde 3 ana durum olmak üzere alt durumlarıyla toplamda 7 durum ortaya çıkmaktadır.

1. $x \sim O(1), k_1 = k_2$ durumunda kökler $O(1)$ olarak beklenir.
2. $x \sim O(\epsilon^r), r > 0$ durumu
 - a. $rm - k_1 > rp - k_2$ ise $r > (k_1 - k_2)/(m - p)$ dir. Bu durum $rp - k_2 = 0$ ya da $r = k_2/p$ olmasını gerektirir. Bu koşullarla hesaplanan r değerine göre

$k_2/p > (k_1 - k_2)/(m - p)$ olacaktır. $\frac{k_2}{p} \geq \frac{k_1 - k_2}{m - p}$, $a_0 \neq 0, p \neq 0$ olmak üzere

kök büyüklüğü $x \sim O\left(\epsilon^{k_2/p}\right)$ olarak beklenir.

b. $rm - k_1 < rp - k_2$ ise $r < (k_1 - k_2)/(m - p)$ dir. Bu durum $rm - k_1 = 0$ ya da $r = k_1/m$ olmasını gerektirir. Bu koşullarla hesaplanan r değerine göre

$k_1/m < (k_1 - k_2)/(m - p)$ olacaktır. $\frac{k_1}{m} \leq \frac{k_1 - k_2}{m - p}$, $a_0 \neq 0, m \neq 0$ olmak üzere

kök büyüklüğü $x \sim O\left(\epsilon^{k_1/m}\right)$ olarak beklenir.

c. $rm - k_1 = rp - k_2$ ise

i) $r = (k_1 - k_2)/(m - p) > 0$ yada $k_1 > k_2(m > p)$ ve $rm - k_1 > 0$ için kök yoktur.

ii) $rm - k_1 < 0$ için ise $\frac{k_1}{m} > \frac{k_1 - k_2}{m - p}$, $\frac{k_2}{p} > \frac{k_1 - k_2}{m - p}$, $k_1 > k_2, a_0 \neq 0$ koşuluyla

kök büyüklüğü $x \sim O\left(\epsilon^{(k_1 - k_2)/(m - p)}\right)$ olarak beklenir.

iii) $rm - k_1 = rp - k_2 = 0$ ise $r = k_1/m = k_2/p$ olacaktır. $k_1/m = k_2/p = (k_1 - k_2)/(m - p)$ olması koşulu ile kökler $O\left(\epsilon^{k_1/m}\right)$ yada $O\left(\epsilon^{k_2/p}\right)$

mertebeğinde beklenir.

3. $x \sim O(1/\epsilon^r)$, $r > 0$ durumu

a. $rm + k_1 > rp + k_2$ ise $r > (k_2 - k_1)/(m - p)$ dir. Bu durum $rn = rm + k_1$ ya da $r = k_1/(n - m)$ olmasını gerektirir. Bu koşullarla hesaplanan r değerine göre $k_1/(n - m) > (k_2 - k_1)/(m - p)$ olacaktır.

$\frac{k_1}{n - m} > \frac{k_2 - k_1}{m - p}$, $m \neq n$ olmak üzere kök büyüklüğü $O\left(\frac{1}{\epsilon^{k_1/(n - m)}}\right)$

mertebeğinde beklenir.

b. $rm + k_1 < rp + k_2$ ise $r < (k_2 - k_1)/(m - p)$ dir. Bu $rn = rp + k_2$ yâda $r = k_2/(n - p)$ olmasını gerektirir. Bu koşullarla hesaplanan r değerine göre $k_2/(n - p) < (k_2 - k_1)/(m - p)$ olacaktır. $k_2/(n - p) < (k_2 - k_1)/(m - p)$, $p \neq n$ olmak üzere kök büyüklüğü

$O\left(\frac{1}{\epsilon^{k_2/(n - p)}}\right)$ mertebesinde beklenir.

- c. $rm + k_1 = rp + k_2$ ise $r = (k_2 - k_1)/(m - p) > 0$ yada $k_1 > k_2$ dir.
 $\frac{k_1}{n-m} > \frac{k_2-k_1}{m-p}, \frac{k_2}{n-p} > \frac{k_2-k_1}{m-p}, k_2 > k_1$ olmak üzere kök büyüklüğü
 $O\left(\frac{1}{\epsilon^{(k_2-k_1)/(m-p)}}\right)$ mertebesinde beklenir. $rm + k_1 = rn, r = k_1/(n -$
 $m) = k_2/(n - p)$ ise $k_1/(n - m) = k_2/(n - p) = (k_2 - k_1)/(m -$
 $p), n \neq m, n \neq p$ koşulları ile kök değerleri $O\left(\frac{1}{\epsilon^{k_1/(n-m)}}\right) \equiv$
 $O\left(\frac{1}{\epsilon^{k_2/(n-p)}}\right)$ mertebesinde olur.

Örneğin; $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$, denkleminde, polinomun her bir teriminin katsayısının logaritması alındığında derecesi 1 olan (565) ve derecesi 4 olan (533) terimlerden elde edilen büyüklüklerin diğerlerine göre çok çok büyük olduğu görülebilir.

ÜÇÜNCÜ BÖLÜM

PIAFindRoots PAKETİ

Bu tez çalışması kapsamında geliştirilen uygulamaya **PIAFindRoots** adı verilmiştir. **PIAFindRoots** uygulaması *Mathematica* [31] tümleşik ortamında tasarlanmıştır. *Mathematica*, Wolfram Research, Inc. firması tarafından geliştirilen bir hesaplama, görselleştirme ve yazılım geliştirme sistemidir. Her türlü mühendislik alanında, matematik, fizik, kimya, ekonomi gibi hemen hemen bütün bilim dallarında analizler yapılabilen, hem sembolik hem de nümerik hesaplamaların yapılabildiği bir tümleşik ortam aracıdır [31-32].

3.1. PIAFindRoots Paketi ve Kullanımı

PIAFindRoots uygulamasının, biri ana fonksiyon olmak üzere toplam 9 adet fonksiyondan oluştuğu Şekil 1'de görülmektedir.

```

EstimateRootsOfPolynomials[{rhs_ == lhs_}, x_, opts___?OptionQ] /; PolynomialQ[rhs - lhs, x]
MemberOfLargeCoefficientQ[{rhs_ == lhs_}, x_, opts___?OptionQ] /; PolynomialQ[rhs - lhs, x]
MemberOfSmallCoefficientQ[{rhs_ == lhs_}, x_, opts___?OptionQ] /; PolynomialQ[rhs - lhs, x]
MemberOfTwoLargeCoefficientsQ[{rhs_ == lhs_}, x_, opts___?OptionQ] /; PolynomialQ[rhs - lhs, x]
OrderOfOneQ[{rhs_ == lhs_}, x_, opts___?OptionQ] /; PolynomialQ[rhs - lhs, x]
EstimateRootsOfPolynomials[___]
FindRoots[___]
FindRoots[{rhs_ == lhs_}, x_, n_Integer, m_Integer, fnc_, x0_, opts___?OptionQ]
GeneratePIA[___]
MemberOfLargeCoefficientQ[___]
MemberOfSmallCoefficientQ[___]
MemberOfTwoLargeCoefficientsQ[___]
Options[PIAFindRoots]
orderOfOneQ[___]
OrderOfOneQ[___]
PIAFindRoots[___]
GeneratePIA[x_, n_, m_, opts___] /; n <= m
orderOfOneQ[x_List] /; Length[x] >= 2
PIAFindRoots[{rhs_ == lhs_}, x_, n_Integer,
m_Integer, opts___?OptionQ] /; n <= m

```

Şekil 1: Fonksiyon listesi

Sembolik hesaplama yapabilen ve *Mathematica* kullanıcıları için tasarlanmış olan bu yazılım paketinde bulunan PIAFindRoots fonksiyonunun kullanımı Şekil 2’de görülmektedir.

```

PIAFindRoots[{E^-x - x == 0}, x, 1, 1]
Generated algorithm(s) is/are  $\left\{x - \frac{F[x]}{F'[x]}\right\}$ .
PIAFindRoots::notpol: The equation is not a polynomial.
Initial Value = 0.
□1 = 0.5
□2 = 0.566311
□3 = 0.567143
PIAFindRoots::tol: The ToleranceValue has been reached. >>
3 iteration(s) is/are needed.
Result is {0.567143}.
{0.567143}

```

Şekil 2: PIAFindRoots fonksiyonunun kullanımı

3.1.1. Uygulamanın Akış Şeması

Köklerinin hesaplanması istenen denklem, bağımsız değişken, pertürbasyon terim adedi, Taylor serisi terim adedi ana fonksiyona (PIAFindRoots) girdi olarak verilir. Eğer gerek duyuluyor ise kök bulmaya yönelik başlangıç değeri, verilen başlangıç değerinden köke yakınsama olmuyorsa bu değeri artırarak denemek için başlangıç değeri artış miktarı, tolerans değeri, maksimum döngü adedi, çok küçük sayı kriteri gibi seçenekler de bu fonksiyona parametre olarak verilebilir.

Uygulama öncelikle verilen parametrelerin atamalarını yapar. Bu parametreler Şekil 3'te görülmektedir. Fonksiyon çağrılırken verilmemiş parametreler için bu varsayılan değerler kullanılır.

```
Options[PIAFindRoots] =
{
  PerturbationParameter -> ε,
  MuchSmallerCriteria-> 0.1,
  ToleranceValue-> 10^-5,
  ChopTolerance-> 10^-7,
  MaxCountOfIteration-> 1500,
  NumberOfDigit->6,
  InitialValue-> 0.,
  InitialValueIncrement-> 0.5
};
```

Şekil 3: PIAFindRoots fonksiyonu varsayılan değerler

Sonrasında kökleri hesaplayacak algoritma üretilir. Verilen denklemin polinom olup olmadığı tespit edilerek, polinom değilse doğrudan kök, algoritmaya göre hesaplanır. Polinom ise başlangıç değeri verilip verilmediğine bakılır. Verilmişse yine doğrudan kök hesaplanır. Eğer başlangıç değeri verilmemişse ilgili polinomun köklerinin hangi aralıkta olduğu tahmin edilir.

```
EstimateRootsOfPolynomials[{x3 + x2 - x == 2}, x]
-----
Coefficients are {1, 1, -1, -2}.
Order of one      : True
Small coefficient  : False
Large coefficient  : False
Two large coefficients : False
The estimation type is order of one.
O(1) {1.}
-----
{1.}
```

Şekil 4: EstimateRootsOfPolynomials fonksiyonunun kullanımı

Diğer bir ifadeyle kökler için başlangıç değerleri tahmini yapılır ve kök hesaplama işlemi buna göre gerçekleştirilir. EstimateRootsOfPolynomials fonksiyonunun kullanımı Şekil 4'te gösterilmiştir. Şekil 5'te ise PIAFindRoots fonksiyonunun kullanım şekli ve çıktıları görülmektedir.


```

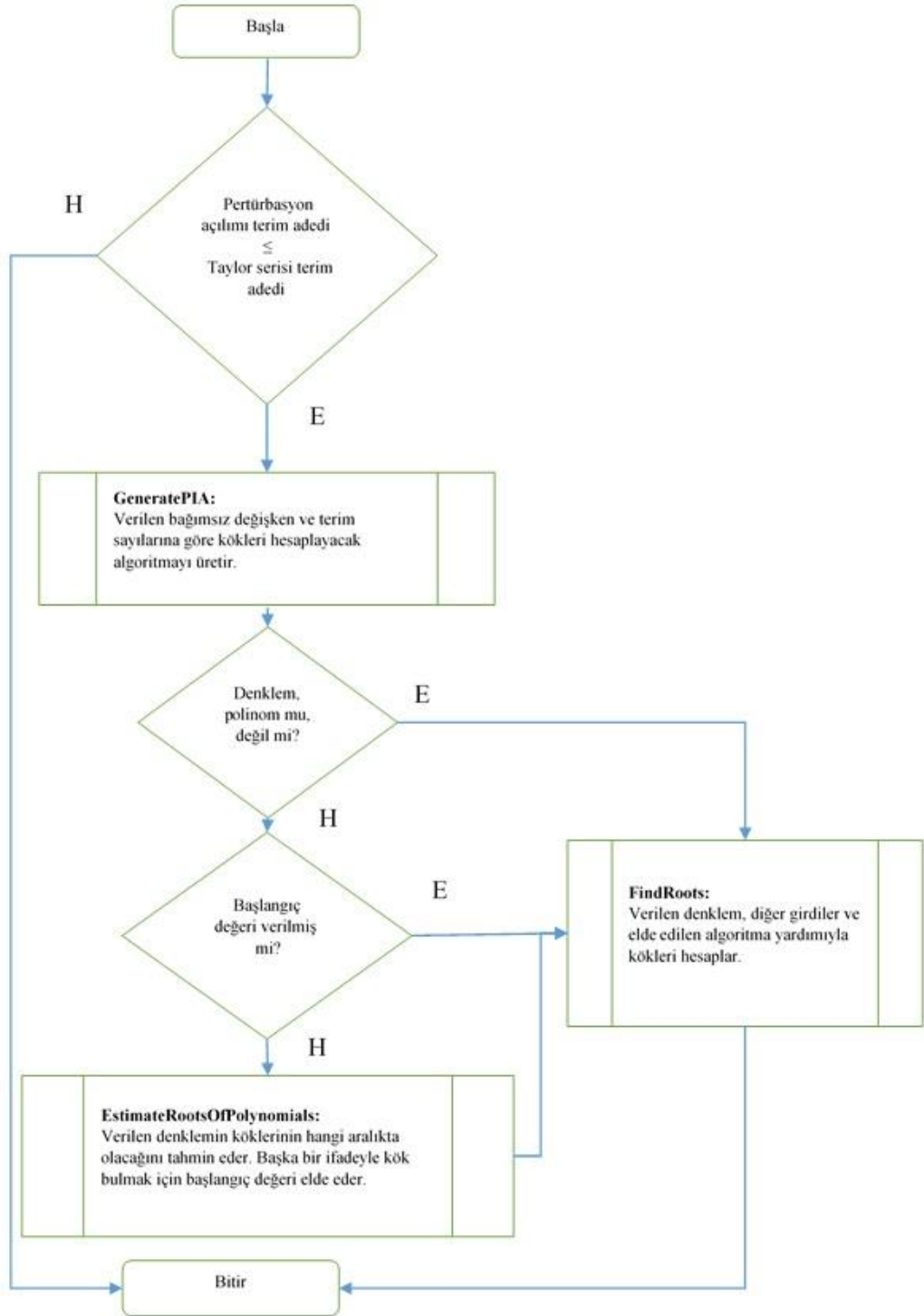
PIAFindRoots[{x^3 + 0.5 x^2 - x == -1200}, x, 1, 1]

Generated algorithm(s) is/are {x -  $\frac{F[x]}{F'[x]}$ }.

PIAFindRoots::pol: The equation is a polynomial.
-----
Coefficients are {1, 0.5, -1, 1200}.
Order of one          : False
Small coefficient     : False
Large coefficient     : True
Two large coefficients: False
The estimation type is one large coefficient.
(i) {10.7978}
-----
Estimation(s) is/are {10.7978}.
Initial Value = 10.7978
#1 = 3.82724
#2 = -23.1033
#3 = -16.2292
#4 = -12.4428
#5 = -11.0314
#6 = -10.8316
#7 = -10.8278
PIAFindRoots::tol: The ToleranceValue has been reached. >>
7 iteration(s) is/are needed.
Result is {-10.8278}.
{-10.8278}

```

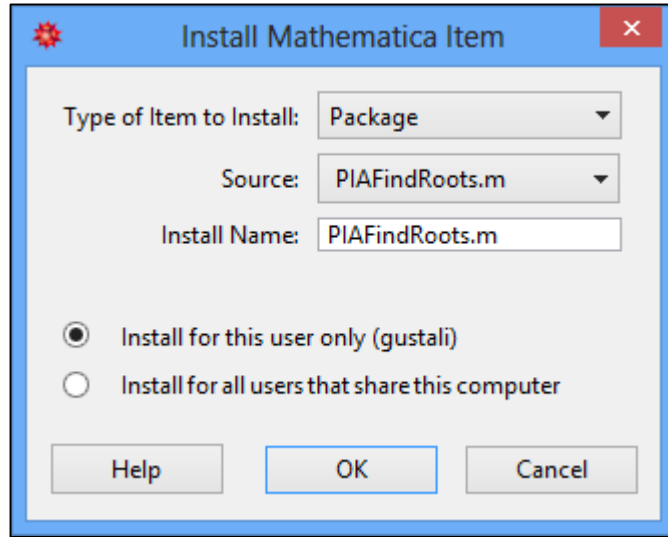
Şekil 5: PIAFindRoots fonksiyonunun örnek kullanımı ve çıktıları
Paketin akış şeması Şekil 6'da gösterilmiştir.



Şekil 6: PIAFindRoots akış şeması

3.1.2. Kurulum ve Yükleme

Mathematica uygulaması için geliştirmiş olduğumuz PIAFindRoots paketi [33], öncelikle *Mathematica*'ya yüklenmelidir. Bunun için fonksiyonları kullanmaya başlamadan öncelikle Şekil 7'de gösterildiği şekilde paket *Mathematica* ile entegre edilmelidir.



Şekil 7: PIAFindRoots paketi kurulumu

Daha sonra Şekil 8'de gösterildiği gibi Get komutuyla paketteki fonksiyonlar kullanılabilir hale getirilir.

```
In[1]:= Get["PIAFindRoots.m"]

Loading Package PIAFindRoots.m of December 31, 2013.
Package PIAFindRoots.m of December 31, 2013 is successfully loaded.
```

Şekil 8: Get komutunun kullanımı

3.1.3. PIAFindRoots ve GeneratePIA Fonksiyonları

Herhangi bir doğrusal olmayan denklemin köklerini hesaplamak için **PIAFindRoots** fonksiyonuna *denklem*, denklemde kullanılan bağımsız değişken *bd*, pertürbasyon seri açılımı terim sayısı *n* ve Taylor seri açılımı terim sayısı *m* verilerek $1 \leq n \leq m$ olmak üzere fonksiyon

PIAFindRoots[{*denklem*}, *bd*, *n*, *m*]

şeklinde çalıştırılmalıdır. Örneğin; $e^{-x} - x = 0$ denkleminin Newton-Raphson yöntemiyle köklerini bulabilmek için **PIAFindRoots**[{ $e^{-x} - x == 0$ }, *x*, 1, 1] komutu ile fonksiyonu çağırmak gerekir. Sonuç Tablo 1’de de ayrıca verildiği gibi {0.567143} olarak elde edilir.

Fonksiyonun çalışması esnasında, Şekil 5’deki gibi iki mesaj üretilmektedir. Bu mesajlardan birincisinde, verilen denklemin polinom olup olmadığı, ikincisinde ise belirlenmiş bir tolerans değerine erişildiği ifade edilmektedir. Arka planda PIAFindRoots fonksiyonu verilen *bd*, *n* ve *m* değerlerine göre, GeneratePIA[*bd*, *n*, *m*] fonksiyonunu çağırarak bir iterasyon algoritması üretmektedir. Örneğin; Newton-Raphson yöntemi için, GeneratePIA[*x*, 1, 1] komutu çağrılarak $\left\{x - \frac{F[x]}{F'[x]}\right\}$ algoritması elde edilir. Şekil 5’de örnek algoritma görülmektedir.

Daha sonra program, verilen denklemin polinom olup olmadığını tespit ederek ikinci aşamaya geçmektedir. Eğer denklem polinom değilse, doğrudan FindRoots fonksiyonu çağrılarak kökler hesaplanır. Denklem polinom ve başlangıç değeri verilmişse de FindRoots fonksiyonu çağrılarak kökler hesaplanır. Denklem polinom ve başlangıç değeri kullanıcı tarafından verilmemişse ikinci bölümde anlatılan sınıflandırma yöntemleriyle Şekil 4’te gösterildiği gibi EstimateRootsOfPolynomials fonksiyonu kullanılarak polinom denkleminin köklerinin hangi büyüklük

mertebesinde olduđu tahmin edilerek başlangıç deęerleri üretilir. Ardından FindRoots fonksiyonu bu başlangıç deęerlerine göre denklemin köklerini hesaplar.

DÖRDÜNCÜ BÖLÜM

ÖRNEKLER

4.1. Örnekler

Aşağıdaki tablolarda farklı pertürbasyon-iterasyon algoritmaları ile denklemlerin köklerinin bulunmasına ilişkin sonuçlar gösterilmiştir. Bu tablolarda;

- Doğrusal olmayan herhangi bir denklemin köklerinin Newton-Raphson, Householder gibi algoritmalarla kaç iterasyonda bulunabildiği karşılaştırılmış,
- Denklem polinom ise kök tahmin fonksiyonu ile kök bulmaya ilişkin başlangıç değeri ya da değerleri tahmini gösterilmiş,
- n ve m değerleri arttırıldıkça gereken iterasyon adedinin azalmış olduğu gösterilmiştir.

```
PIAFindRoots[{E^-x - x == 0}, x, 1, 1] 1
Generated algorithm(s) is/are {x - F[x]/F'[x]}. 2
PIAFindRoots::notpol: The equation is not a polynomial. 3
Initial Value = 0. 4
x1 = 0.5 5
x2 = 0.566311 6
x3 = 0.567143 7
PIAFindRoots::tol: The ToleranceValue has been reached. >> 8
3 iteration(s) is/are needed. 9
Result is {0.567143}. 10
{0.567143} 11
```

Şekil 9: PIAFindRoots fonksiyonunun örnek kullanımı ve çıktıları

Örneğin, Şekil 9'daki gibi $e^{-x} - x = 0$ denkleminin köklerini bulmak için uygulamanın çalışmasını ve sonuçlarını adım adım izah etmek gerekirse,

Satır 1: Öncelikle uygulamamıza denklem, denklemdeki bağımsız değişken, pertürbasyon açılımındaki terim adedi, Taylor serisi açılımındaki terim adedi ve gerekliyse kök bulmaya yönelik başlangıç değeri verilir. Şekil 9'da görüldüğü gibi; denklem $e^{-x} - x = 0$, bağımsız değişken x , pertürbasyon açılımındaki terim adedi ilk sıradaki 1 ve Taylor serisindeki terim adedi ikinci sıradaki 1'dir. Fonksiyonun buradaki çalıştırılmasında başlangıç değeri, tolerans değeri gibi parametreler verilmediği için bu gibi değerlerin varsayılan değerleri geçerlidir. Şekil 3'de de görüldüğü gibi;

- varsayılan başlangıç değeri 0 (sıfır),
- varsayılan tolerans değeri 10^{-5} ,
- varsayılan maksimum iterasyon adedi 1500 olarak belirlenmiştir.

Satır 2: Fonksiyon çağrıldığında önce kök bulmada kullanılacak algoritma verilen parametre değerlerine göre üretilir.

Satır 3: Denklemin polinom olup olmadığı belirlenir.

Satır 4-7: Bu satırlarda, başlangıç değeri ile birlikte her bir iterasyonda bulunan kökler gösterilmektedir.

Satır 8: Tolerans değerine erişildiği bu satırda ifade edilmiştir.

Satır 9: Kaç iterasyonda sonuca ulaşıldığı belirtilmiştir.

Satır 10: Bulunan kök değerlerinin gösterildiği satırdır.

Satır 11: Başka işlemlerde kullanılmak üzere fonksiyon bulduğu kökleri içeren bir liste döndürür.

Aynı işlemler Parabolik, Householder İterasyonu gibi diğer algoritmalarla tekrarlandığında Tablo 1 elde edilmektedir.

Tablo 1: $e^{-x} - x = 0$ denkleminin kökleri [26]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	Üçlü düzeltme (n=3, m=3)	n=3,m=6
Başlangıç Değeri	0	0	0	0	0
İlk İterasyon	0.5	0.585786	0.5625	0.567708	0.567139
İkinci İterasyon	0.566311	0.567143	0.567143	0.567143	
Üçüncü İterasyon	0.567143				

Şekil 9'daki $e^{-x} - x = 0$ denkleminin köklerini bulmak için uygulamanın çalışması sonucu elde edilen Tablo 1 gibi, $\tan(x) - \tanh(x) = 0$ denklemi için Tablo 2, $x^3 + x^2 - x = 2$ denklemi için Tablo 3, $x^3 + 0.5x^2 - x + 1200 = 0$ denklemi için Tablo 4 ve $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$ denklemi için Tablo 5 elde edilmiştir. Benzer örnekler tez çalışmasının Ek kısmında sunulmuştur.

Tablo 2: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26]

İterasyonlar	Newton-Raphson n=1,m=1	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=7
Başlangıç Değeri	3	3	3	3	3
İlk İterasyon	4.12583	9.06801	4.29596	3.84694	3.92179
İkinci İterasyon	3.971	10.4095+1.19756 i	3.99323	3.92659	3.9266
Üçüncü İterasyon	3.92863	11.1217+3.2845 i	3.92684		
Dördüncü İterasyon	3.92661	-4.15941-0.374611 i	3.9266		
Beşinci İterasyon	3.9266	-3.86281-1.12641 i			
..		..			
Dokuzuncu İterasyon		-2.55318*10669469+ 1.84865*10669469 i			

Tablo 3: $x^3 + x^2 - x = 2$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householde riterasyonu (n=2, m=2)	n=3,m=3	n=6, m=10
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	1.25	-0.207107	1.1875	1.21484	1.20557
İkinci İterasyon	1.20707	-1.25306	1.20556	1.20557	
Üçüncü İterasyon	1.20557	-1.03481+0.605879 i			
Dördüncü İterasyon		-1.10301+0.665527 i			
Beşinci İterasyon		-1.10278+0.665457 i			

Tablo 4: $x^3 + 0.5x^2 - x + 1200 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=3, m=9
Başlangıç Değeri	10	10	10	10	10
İlk İterasyon	2.75081	4.93443 + 6.91249 i	-2.43624	-8.62642	- 10.8278
İkinci İterasyon	-47.2191	-0.346032 - 2.32067 i	3169.43	- 11.036	
Üçüncü İterasyon	-31.7208	-9.90238 + 7.80886 i	1760.72	-10.8278	
..					
Altıncı İterasyon	-12.0409	-10.8278 - 2.3624*10 ⁻¹¹ i	301.767		
..					
Dokuzuncu İterasy.	-10.8278		51.5184		
..					
Ondokuzuncu İter.			-10.8278		

Tablo 5: $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=4
Başlangıç Değeri	7.9433	7.9433	7.9433	7.943 3	7.9433
İlk İterasyon	6.4851	6.30956 + 1.44759 i	5.83435	5.42463	5.12959
İkinci İterasyon	5.18945	5.02931 + 0.01134 i	4.08985	3.45923	3.04089
Üçüncü İterasyon	4.06209	3.73442 - 1.07711 i	2.7465	2.08796	1.68584
..
Altıncı İterasyon	1.73801	1.40002 + 0.01465 i	0.469366	-0.03370	- 0.0033628
Yedinci İterasyon	1.24512	0.895343 - 0.42457 i	-0.0286706	-0.00336	
Sekizinci İterasyon	0.822399	0.538223 + 0.02474 i	-0.00336418		
..					
Onbirinci İterasyon	- 0.0033185	-0.0033 + 2.56*10 ⁻⁶ i			
Onikinci İterasyon	- 0.0033628				

SONUÇ

Doğrusal olmayan cebirsel denklemlerin köklerini pertürbasyon-iterasyon yöntemlerini kullanarak hesaplayabilmek için el ile yapılması zor ve hassas hesaplamaları yapabilen bir *Mathematica* yazılım paketi geliştirilmiştir [33]. Bu paket kullanmak isteyen araştırmacıların hizmetine ücretsiz olarak sunulmaktadır [33].

Geliştirilen uygulama, $1 \leq n \leq m$ olmak üzere herhangi bir n ve m tam sayı değerleri için kullanılabilir kadar geneldir.

Pakdemirli ve Boyacı [26], çalışmalarında bir, iki ve üç düzeltme terimi kullanarak elde ettikleri algoritmalarda, düzeltme terimi sayısı arttıkça köke yaklaşmak için daha az iterasyona ihtiyaç duyulduğunu rapor etmişlerdir. Bu çalışmada daha yüksek düzeltme terimi sayısı ile de aynı gözlem tecrübe edilmiştir.

Kullanılan algoritmaların denklem sistemlerine uyarlanması ve uygulamanın da denklem sistemlerini kapsayacak şekilde geliştirilmesi sonraki çalışmalarımıza bırakılmıştır.

KAYNAKÇA

- [1] <http://members.comu.edu.tr/iturkyilmaz/Uygulamali.htm>, Erişim Tarihi: 10.04.2013.
- [2] **Pakdemirli, M. ve Yurtsever, H. A.** (2007). Estimating roots of polynomials using perturbation theory, *Applied Mathematics and Computation* 188, 2025–2028.
- [3] **Göktaş, Ü., Pakdemirli, M. ve Aksoy, Y.** (2011). Symbolic computation of Perturbation-Iteration solutions for differential equations, *2nd. International Symposium on Computing in Science & Engineering*, 100(14), 60-66.
- [4] <http://teknomak.org/sayisal-yontemler-ders-notlari/>, Erişim Tarihi: 23.04.2013.
- [5] **Ustali, G., Göktaş, Ü. ve Pakdemirli, M.** (2012). Symbolic computation of the roots of nonlinear algebraic equations using perturbation theory, *9th International Conference on Electronics Computer and Computation*, 171-173, Ankara, Türkiye.
- [6] **Holmes, M. H.** (2009). Introduction to the Foundations of Applied Mathematics Texts in Applied Mathematics, 56, 43-85, Springer.
- [7] **Pakdemirli, M., Boyacı, H. ve Yurtsever, H. A.** (2007). Perturbative derivation and comparisons of root-finding algorithms with fourth order derivatives, *Mathematical and Computational Applications*, 12, 117-124.
- [8] **Pakdemirli, M., Boyacı, H. ve Yurtsever, H. A.** (2008). A root-finding algorithm with fifth order derivatives, *Mathematical and Computational Applications*, 13, 123-128.
- [9] **Nayfeh, A. H.** (1981). Introduction to Perturbation Techniques, John Wiley and Sons, New York.
- [10] **Hinch, E. J.** (1991). Perturbation Methods, Cambridge University Press, New York.
- [11] **Şenol, M.** (2011). Diferansiyel Denklem Sistemlerinin Pertürbasyon-İterasyon Metodu İle Çözümü, Yüksek Lisans Tezi, Nevşehir, Nevşehir Üniversitesi.
- [12] **Aksoy, Y. ve Pakdemirli, M.** (2010). New perturbation-iteration solutions for Bratu-type equations, *Computers and Mathematics with Applications* 59, 2802-2808.

- [13] **Aksoy, Y.** (2013). Yeni bir Perturbasyon-İterasyon Metodu ve Mühendislik Problemlerine Uygulamaları, Doktora Tezi, Manisa, Celal Bayar Üniversitesi.
- [14] **He, J. H.** (2001). Iteration Perturbation Method for Strongly Nonlinear Oscillators, *J. Sound Vibr.* 7, 631-642.
- [15] **Mickens, R. E.** (1987). Iteration Procedure for Determining Approximate Solutions to Nonlinear Oscillator Equations, *J. Sound Vibr.* 116, 185–187.
- [16] **Mickens, R.E.** (2005). A Generalized Iteration Procedure for Calculating Approximations to Periodic Solutions of “Truly Nonlinear Oscillators”, *J. Sound Vibr.* 287, 1045–1052.
- [17] **Mickens, R.E.** (2006). Iteration Method Solutions for Conservative and Limit-cycle $x^{1/3}$ Force Oscillators, *J. Sound Vibr.* 292, 964–968.
- [18] **Mickens, R.E.** (2002). Generalized Harmonic Balance/numerical Method for Determining Analytical Approximation to the Periodic Solutions of the $x^{4/3}$ Potential, *Journal of Sound and Vibration*, 250, 951-954.
- [19] **Hu, H. ve Xiong, Z.G.** (2003). Oscillations in an $x^{(2m+1)/(2n+1)}$ Potential, *J. Sound Vibr.* 259(8), 977-980.
- [20] **He, J. H.** (2006). Non-Perturbative Methods for Strongly Nonlinear Problems, Dissertation. De-verlag Im Internet GmbH, Berlin.
- [21] **Wang, S.Q. ve He, J.H.** (2008). Nonlinear Oscillator with Discontinuity by Parameter-Expansion Method, *Chaos Solitons & Fractals* 35, 688-691.
- [22] **He, J. H.** (2002). Modified Linstedt–Poincare Methods for Some Non-linear Oscillations. Part I: Expansion of Constant, *J. Non-linear Mech.* 37, 309–314.
- [23] **Ramos, J.I.** (2007). On Linstedt–Poincare’ Techniques For the Quintic Duffing Equation, *Appl. Math. Comput.*, in Press, Doi:10.1016/j.amc.2007.03.050.
- [24] **Öziş, T. ve Yıldırım, A.** (2007). Determination of Periodic Solution of a $u^{1/3}$ Force by He’s Modified Linstedt–Poincare’ Method, *J. Sound Vibr.* 301, 415–419.
- [25] **Mickens, R.E.** (2001). Oscillations in an $4x^3$ potential, *Journal of Sound and Vibration* 246(2), 375-378.
- [26] **Pakdemirli, M. ve Boyacı, H.** (2007). Generation of root finding algorithms via perturbation theory and some formulas, *Applied Mathematics and Computation* 184, 783–788.
- [27] **Householder, A.S.** (1970). The Numerical Treatment of a Single Nonlinear Equation, McGraw-Hill, New York.
- [28] **Vatanserver, F.** (2006). İleri Programlama Uygulamaları, Seçkin Yayıncılık, Ankara, Türkiye.
- [29] **Pakdemirli, M. ve Elmas, N.** (2010). Perturbation theorems for estimating magnitudes of roots of polynomials, *Applied Mathematics and Computation* 216, 1645–1651.

- [30] **Nayfeh, A. H.** (2008). Perturbation Methods, John Wiley and Sons, New York.
- [31] **Wolfram, S.** (2003). The Mathematica Book, 5th ed., Wolfram Media, USA.
- [32] <http://lp.marmara.edu.tr/course.aspx?mod=2&kultur=tr-TR&program=249&did=24503&mid=16011>, Erişim Tarihi: 02.11.2013.
- [33] **Göktaş, Ü.** (2014) [Online] Available: <http://www.turgutozal.edu.tr/ugoktas/>.

EKLER

Tablo 6: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=9
Başlangıç Değeri	3.5	3.5	3.5	3.5	3.5
İlk İterasyon	4.04861	0.395126	3.93455	3.90387	3.9266 1
İkinci İterasyon	3.94263	0.201148 -0.114896 i	3.9266	3.9266	
Üçüncü İterasyon	3.92686	0.0671579 -0.115593 i			
Dördüncü İterasyon	3.9266	0.000221625 -0.077171 i			

Tablo 7: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=4
Başlangıç Değeri	4	4	4	4	4
İlk İterasyon	3.93225	3.21135	3.92692	3.92661	3.9266
İkinci İterasyon	3.92663	-10.681	3.9266		
Üçüncü İterasyon		-10.5184+0.195227 i			

Tablo 8: $\tan(x) - \tanh(x) = 0$ denkleminin kökleri [26]

İterasyonlar	Newton-Raphson (n=1, m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=3, m=5
Başlangıç Değeri	4.5	4.5	4.5	4.5	4.5
İlk İterasyon	4.33836	4.39218 -0.152415 i	4.2172	4.12778	3.97108
İkinci İterasyon	4.13168	4.27856 -0.466404 i	3.95617	3.92778	3.9266
Üçüncü İterasyon	3.97377	4.24693 -1.24828 i	3.92662	3.9266	
Dördüncü İterasyon	3.9289	5.2546 -3.22637 i			
Beşinci İterasyon	3.92661	-9.39715-6.00726 i			

Tablo 9: $x^3 + 2x^2 - x = 1$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=6, m=8
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.833333	0.0	0.810185	0.804527	0.801942
İkinci İterasyon	0.802935	-0.5	0.801939	0.801938	
Üçüncü İterasyon	0.801939	-0.554886			
Dördüncü İterasy.		-0.554958			

Tablo 10: $x^3 + 0.5x^2 - x = -2$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=3, m=9
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.166667	0.571429 -0.728431 i	-0.643519	-2.02598	-1.73775
İkinci İterasyon	2.6358	0.618629 -0.877125 i	-142.158	-1.74163	
Üçüncü İterasyon	1.69489	0.618877 -0.8763 i	-79.0533	-1.73775	
..		
Altıncı İterasyon	2.46172		-13.7165		
..		
Onikinci İterasyon	0.362684		-1.73775		
..	..				
Yirmüçüncü İter.	-1.73776				

Tablo 11: $x^3 + 0.5x^2 - 1.1x = -0.7$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=3, m=9
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.62069	0.585714 -0.377694 i	0.447046	0.447046	-1.52356
İkinci İterasyon	-0.0430534	0.512128 -0.444054 i	1918.04	1918.04	
Üçüncü İterasyon	0.614714	0.511779 -0.444447 i	1065.5	1065.5	
..	
Onuncu İterasyon	-0.20083		17.2576	20.5286	
..			..		
Yirmidördüncü İ.			-1.52356	-1.88173	
..			..		
Yirmialtıncı İter.				-1.52356	

Tablo 12: $x^3 + 0.8x^2 + 1.4x = -0.6$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=3, m=9
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.366667	0.210526 -0.613784 i	0.11263	-0.048825	-0.481305
İkinci İterasyon	-0.164792	-0.205299-1.02868 i	-0.521621	-0.512031	
Üçüncü İterasyon	-0.482201	-0.159074-1.10515 i	-0.481295	-0.481304	
Dördüncü İter.	-0.481305	-0.159347-1.10509 i			

Tablo 13: $x^3 + x^2 + x = -1$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=3, m=9
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.333333	0.25 -0.661438 i	0.037037	-0.176955	-1.0
İkinci İterasyon	-0.407407	-0.0071891-0.976625 i	-1.8821	-1.33175	
Üçüncü İterasyon	-1.41886	5.65947*10 ⁻⁷ -1.00001 i	-1.21905	-1.0113	
Dördüncü İter.	-1.11849		-1.0101	-1.0	
Beşinci İterasyon	-1.01248		-1.0		
..	..				
Yedinci İterasyon	-1.0				

Tablo 14: $x^4 - 1.2x^3 + 0.9x^2 + 0.8x + 1.3 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	Üçlü düzeltme (n=3, m=3)	n=4, m=4
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.484848	0.62069 -0.496919 i	0.135028	-0.232361	-0.683873
İkinci İterasyon	-0.872121	0.630149 -0.768675 i	0.27167	-1.28758	-0.751568
Üçüncü İter.	-0.750016	0.64353 -0.763238 i	-6.00908	898.712	
Dördüncü İter.	-0.75157		-4.10775	540.512	
..			..		
Dokuzuncu İter.			-1.73549	42.3111	
..				..	
Onduzuncu İter.				-0.75157	

Tablo 15: $x^4 + 0.6x^3 - 1.1x^2 + 0.8x - 0.6 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=9, m=11
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.840909	0.729448	0.802369	0.787906	0.775318
İkinci İterasyon	0.782595	0.775487	0.775414	0.775317	
Üçüncü İterasyon	0.775417	0.775317	0.775317		
Dördüncü İterasy.	0.775317				

Tablo 16: $x^4 + 1.4x^3 - 1.5x^2 + 0.8x + 0.6 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4	n=5, m=5
Başlangıç Değeri	1	1	1	1	1	1
İlk İterasyon	0.616667	0.655172 +0.381394 i	0.403597	0.21743 1	0.0255845	-0.189869
İkinci İterasyon	- 0.053845 2	0.311787 0.0881884 i	-5.9118	30.7733	4.62755	-0.405532
Üçüncü İterasyon	-0.621486	-0.297128+ 0.578764 i	-4.1008	18.3883	2.5373	-0.389974
..
Altıncı İterasyon	-0.389975	-0.389974- 1.46732*10 ⁻¹³ i	-2.20373	3.81992	-12.8816	
Yedinci İterasyon			-2.19333	2.23672	-7.47145	
..				
Onikinci İterasyon				-2.2413	-2.19333	
				..		
Ondördüncü İterasyon				2.19333		

Tablo 17: $x^4 - 1.4x^3 - 1.5x^2 - 0.8x + 1.3 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=10	n=5, m=5
Başlangıç Değeri	1	1	1	1	1	1
İlk İterasyon	0.65	0.658735	0.659187	0.63083 6	0.63241 6	0.630624
İkinci İterasyon	0.632569	0.632421	0.63243	0.63241 4		0.632414
Üçüncü İterasyon	0.632414		0.632414			

Tablo 18: $x^4 - 0.9x^3 - 1.3x^2 - 0.6x + 0.7 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=3, m=6
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.421053	0.594304	0.773874	0.027231	0.659206
İkinci İterasyon	0.517673	0.512335	0.526208	6.37397	0.511971
Üçüncü İterasyon	0.511987	0.51197	0.511973	3.99807	
Dördüncü İterasyon	0.51197			2.63199	
..				..	
Yedinci İterasyon				1.72074	

Tablo 19: $x^4 + x^3 + 0.3x^2 - 0.6x - 1.5 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=3, m=6
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.926829	0.91827	0.919974	0.918947	0.918717
İkinci İterasyon	0.918808	0.918718	0.918718	0.918718	
Üçüncü İterasyon	0.918718				

Tablo 20: $x^4 + 0.8x^3 + 1.3x^2 + 1.4x - 1.1 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=15, m=16
Başlangıç Değ.	1	1	1	1	1
İlk İterasyon	0.673077	0.463918 +0.251259 i	0.573392	0.528727	0.476387
İkinci İterasyon	0.511737	0.472438 +0.0100696 i	0.477571	0.476412	
Üçüncü İter.	0.477679	0.476396 +3.9849*10 ⁻⁷ i	0.476397	0.476397	
Dördüncü İter.	0.476399				

Tablo 21: $x^4 + 1.6x^3 + 0.5x^2 - 1.4x - 0.4 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=3, m=5
Başlangıç Değeri	1	1	1	1	1
İlk İterasyon	0.845238	0.780315	0.813018	0.802073	0.793553
İkinci İterasyon	0.797891	0.793556	0.793591	0.793553	
Üçüncü İterasyon	0.793587		0.793553		
Dördüncü İterasyon	0.793553				

Tablo 22: $x^3 + 12x^2 + x - 1.5 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=4
Başlangıç Değeri	10	10	10	10	10
İlk İterasyon	5.91774	3.55952 +3.33221 i	4.62399	3.9297	3.47924
İkinci İterasyon	3.37067	1.86182 +0.1027 i	1.97917	1.39283	1.07514
Üçüncü İterasyon	1.84883	0.311323 -0.431278 i	0.81217	0.504551	0.386757
..
Beşinci İterasyon	0.549543	0.310802 -7.33115*10 ⁻⁸ i	0.31215	0.310802	0.310802
Altıncı İterasyon	0.361436		0.310802		
..	..				
Dokuzuncu İter.	0.310802				

Tablo 23: $x^3 + 12x^2 + x - 1.5 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=4
Başlangıç Değeri	0.316228	0.316228	0.316228	0.316228	0.316228
İlk İterasyon	0.310845	0.310802	0.310803	0.310802	0.310802
İkinci İterasyon	0.310802				

Tablo 24: $x^4 + 1100x^3 - x^2 + 1.2x - 0.7 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=6, m=6
Başlangıç Değeri	0.1	0.1	0.1	0.1	0.1
İlk İterasyon	0.0849988	0.0817898	0.0828211	0.082298	0.0820849
İkinci İterasyon	0.0821712	0.0820763	0.0820764	0.082076	
Üçüncü İterasyon	0.0820764				

Tablo 25: $x^5 + 1.1x^4 + 0.8x^3 - 0.6x^2 - 116x - 2 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=6, m=6
Başlangıç Değeri	3.16228	3.16228	3.16228	3.16228	3.16228
İlk İterasyon	3.02111	3.00292	3.00684	3.00454	3.00399
İkinci İterasyon	3.00421	3.00398	3.00398	3.00398	
Üçüncü İterasyon	3.00398				

Tablo 26: $x^5 + 1.1x^4 + 0.8x^3 - 0.6x^2 - 116x - 2 = 0$ denkleminin kökleri [2]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=6, m=6
Başlangıç Değeri	0.01	0.01	0.01	0.01	0.01
İlk İterasyon	- 0.0172391	-0.0172428	-0.0172428	-0.017243	-0.017243

Tablo 27: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	0.004	0.004	0.004	0.004	0.004
İlk İterasyon	-0.00812	-0.00812	-0.00812	-0.00812	-0.00812

Tablo 28: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	17.7828	17.7828	17.7828	17.7828	17.7828
İlk İterasyon	17.1956	17.0821	17.1159	17.0988	17.0944
İkinci İterasyon	17.0953	17.0926	17.0926	17.0926	17.0926
Üçüncü İterasyon	17.0926				

Tablo 29: $-1.2x^6 + 0.5x^5 + 342x^4 - x^2 + 234x + 1.9 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	0.9261	0.9261	0.9261	0.9261	0.9261
İlk İterasyon	0.569698	Ulaşılamadı	0.400775	0.283732	0.19206
İkinci İterasyon	0.217789		-0.0318952	-0.01316	-0.007119
Üçüncü İterasyon	0.0014582		-0.00812142	-0.00811	-0.008119
Dördüncü İterasyon	-0.008119				

Tablo 30: $x^7 + 0.5x^5 + 533x^4 - x^2 + 565x + 1.9 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	0.0016	0.0016	0.0016	0.0016	0.0016
İlk İterasyon	- 0.0033628	-0.00336281	-0.0033628	-0.00336	- 0.0033628

Tablo 31: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	Üçlü düzeltme (n=3, m=3)	n=5, m=5
Başlangıç Değeri	0.0398	0.0398	0.0398	0.0398	0.0398
İlk İterasyon	0.0522756	0.0522797	0.0522797	0.052279	0.0522791

Tablo 32: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]

İterasyonlar	n=1,m=1	n=1, m=2	n=2, m=2	n=3,m=3	n=5, m=5
Başlangıç Değeri	0.8577	0.8577	0.8577	0.8577	0.8577
İlk İterasyon	0.555583	0.550323 +0.302071 i	0.407109	0.30705 1	0.230863
İkinci İterasyon	0.25811	0.293019 +0.000406686 i	0.0432028	0.04493 1	0.053169
Üçüncü İterasyon	0.0667941	0.0173485 -0.000308709 i	0.0522794	0.05227 9	0.0522791
Dördüncü İter.	0.0522788	0.0522873 +1.43897*10 ⁻⁷ i			

Tablo 33: $x^9 + 0.5x^8 + x^7 + 1.7x^6 + 42x^4 - x^2 + 23x - 1.2 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=5, m=5
Başlangıç Değ.	2.0893	2.0893	2.0893	2.0893	2.0893
İlk İterasyon	1.76662	1.75896 +0.32258 i	1.60903	1.50382	1.36096
İkinci İterasyon	1.44789	1.44534 +0.00211 i	1.13222	0.924728	0.655581
Üçüncü İter.	1.13072	1.12461-0.314986 i	0.672303	0.374212	0.0070632
Dördüncü İter.	0.820803	0.805538+0.00296 i	0.222405	0.031162	0.0522791
Beşinci İter.	0.519231	0.500952-0.29855 i	0.047314	0.052279	
Altıncı İter.	0.224709	0.253476-0.00566 i	0.0522791		
..			
Sekizinci İter.	0.0522787	0.0522-0.336*10 ⁻⁷ i			

Tablo 34: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=5, m=5
Başlangıç Değeri	0.002	0.002	0.002	0.002	0.002
İlk İterasyon	0.0018223	0.00180506	0.001808	0.001805	0.001805
İkinci İterasyon	0.0018052				

Tablo 35: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=5, m=5
Başlangıç Değeri	50.1187	50.1187	50.1187	50.1187	50.1187
İlk İterasyon	49.162	49.0558	49.0759	49.0637	49.0612
İkinci İterasyon	49.0621	49.0611	49.0611	49.0611	49.0611
Üçüncü İterasyon	49.0611				

Tablo 36: $-1.1x^6 + 54x^5 - 1.4x^4 + x^3 - 554x^2 + x = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=5, m=5
Başlangıç Değeri	2.1544	2.1544	2.1544	2.1544	2.1544
İlk İterasyon	2.21705	2.21222	2.21137	2.21222	2.2121
İkinci İterasyon	2.21213	2.21209	2.21209	2.21209	
Üçüncü İterasyon	2.21209				

Tablo 37: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	0.2399	0.2399	0.2399	0.2399	0.2399
İlk İterasyon	0.249094	0.248496	0.248409	0.248485	0.248475
İkinci İterasyon	0.248479	0.248476	0.248476		

Tablo 38: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	7.0795	7.0795	7.0795	7.0795	7.0795
İlk İterasyon	6.58728	6.27063	6.43752	6.36869	6.33143
İkinci İterasyon	6.33538	6.26343	6.27036	6.26611	6.26594
Üçüncü İterasyon	6.26998	6.26594	6.26594	6.26594	
Dördüncü İterasyon	6.26595				
Beşinci İterasyon	6.26594				

Tablo 39: $x^8 + 27x^7 - x^6 - 1300x^5 + 1.1x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3,m=3	n=4, m=4
Başlangıç Değeri	25.1189	25.1189	25.1189	25.1189	25.1189
İlk İterasyon	21.8074	21.3041		19.4759	18.8354
İkinci İterasyon	18.9356	18.4961		15.1261	14.1619
..
Beşinci İterasyon	12.468	11.605			6.84094
..
Yedinci İterasyon	9.57319	8.71021	6.74277	6.27278	6.26594
Sekizinci İterasyon	8.47373	7.76235	6.3211	6.26594	
..
Onuncu İterasyon	6.92722	6.41559	6.26594		
..
Onikinci İterasyon	6.30494	6.26594			
..
Ondördüncü İter.	6.26594				

Tablo 40: $x^7 - 1.3x^5 + 101x^4 + 0.8x^2 + 140000 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=7, m=7
Başlangıç Değeri	5.3527	5.3527	5.3527	5.3527	5.3527
İlk İterasyon	3.80204	4.32492 -1.45985 i	2.63226	1.38168	-8.87851
..
Dördüncü İterasyon	12399.8	4.5933 -2.69986 i	-115.463	-2.3549*10 ⁶	-6.00474
..	
Yirminci İterasyon	1052.61		-6.00474	-27980.2	
..	
Elliüçüncü İterasyon	6.07668			-6.00474	
..	..				
Altmışüçüncü İter.	-6.00475				

Tablo 41: $1.3x^5 + 61x^4 + 0.8x^2 - 900x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=5, m=5
Başlangıç Değeri	63.0957	63.0957	63.0957	63.0957	63.0957
İlk İterasyon	49.3	45.6996 +13.3176 i	43.8298	40.5648	36.5631
..					
Dördüncü İterasyon	22.9736	19.4808 -0.00137 i	13.975	10.042	6.45289
..
Sekizinci İterasyon	7.89208	5.48044 -0.00764 i	3.0758	2.41741	2.40981
Dokuzuncu İterasyon	6.02478	3.76416 +1.15798 i	2.51024	2.40981	
..
On birinci İterasyon	3.61522	2.3 -0.0412156 i	2.40981		
..
On üçüncü İterasyon	2.55721	2.40981 +8.45044*10 ⁻¹³ i			
..
Onaltıncı İterasyon	2.4098				

Tablo 42: $1.3x^5 + 61x^4 + 0.8x^2 - 900x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=5, m=5
Başlangıç Değeri	2.5119	2.5119	2.5119	2.5119	2.5119
İlk İterasyon	2.41771	2.40954	2.41079	2.40996	2.40982
İkinci İterasyon	2.40987	2.40981	2.40981	2.40981	
Üçüncü İterasyon	2.40981				

Tablo 43: $x^7 - 56x^6 - 1.3x^5 + 1967x^4 + 0.8x^2 - 0.9x + 1.4 = 0$ denkleminin kökleri [29]

İterasyonlar	Newton-Raphson (n=1,m=1)	Parabolik (n=1, m=2)	Householder iterasyonu (n=2, m=2)	n=3, m=3	n=4, m=4
Başlangıç Değeri	6.3096	6.3096	6.3096	6.3096	6.3096
İlk İterasyon	6.27729	6.27654	6.27657	6.27655	6.27655
İkinci İterasyon	6.27655		6.27655		