



## Lineer Olmayan Denklemlerin Mathematica ile Nümerik Analizleri

Esin İBİŞ<sup>1\*</sup>

<sup>1</sup>Marmara Üniversitesi, Matematik, İstanbul, Türkiye

### Anahtar Kelimeler:

Lineer olmayan  
denklem, Nümerik  
analiz,  
Mathematica,  
Yöntem, Yazılım

### Özet

Nümerik Analiz, mühendislik ve doğa bilimlerinde önem arz eden bir bilimsel hesaplama yöntemidir. Bu hesaplama yöntemi, matematiksel analiz problemlerinin yaklaşık çözümlerinde kullanılan algoritmaları inceler. Mathematica ise etkileşimli ve kullanıcı dostu yapısıyla, uzun ve karmaşık kod yazma ihtiyacını azaltarak diğer yazılım dillerine kıyasla zaman kaybını büyük ölçüde ortadan kaldıran bir yazılım sistemidir. Bu makalede lineer olmayan denklemlerin Mathematica ile nümerik analizleri incelenmiştir. İlk olarak, lineer olmayan denklemlerin nümerik analizinde kullanılan Bisection Yöntemi, Sabit Nokta Yöntemi, Newton-Raphson Yöntemi, Secant Yöntemi ve Regula Falsi Yöntemi açıklanmıştır. Daha sonra ise verilen yöntemlerin ilgili örneği Mathematica kullanarak çözülmüş ve sonuçları elde edilmiştir.

\*e-posta: [esinibis97@gmail.com](mailto:esinibis97@gmail.com)

Bu makaleye atıf yapmak için:

Esin İBİŞ, "Lineer Olmayan Denklemlerin Mathematica ile Nümerik Analizleri", Bayburt Üniversitesi Fen Bilimleri Dergisi, C. 8, s 1, ss. 32-43

How to cite this article:

Esin İBİŞ, "Numerical Analysis of Non-Linear Equations with Mathematica", Bayburt University Journal of Science, vol. 8, no 1, pp. 32-43

## Numerical Analysis of Non-Linear Equations with Mathematica

### Keywords:

*Non-linear equation, Numerical analysis, Mathematica, Method, Software*

### Abstract

Numerical Analysis is a scientific calculation method that is important in engineering and natural sciences. This method of calculation examines the algorithms used in approximate solutions of mathematical analysis problems. Mathematica, on the other hand, is a software system that reduces the need to write long and complex code with its interactive and user-friendly structure, eliminating the need to waste time compared to other software languages. In this article, numerical analysis of nonlinear equations with Mathematica is examined. Firstly, the Bisection Method, Fixed Point Method, Newton-Raphson Method, Secant Method and Regula False Method used in the numerical analysis of nonlinear equations are explained. Then, the relevant example of the given methods was solved using Mathematica and the results were obtained.

## 1 GİRİŞ

Nümerik Analiz, sayısal analiz, nümerik çözümleme veya sayısal çözümleme mühendislik ve doğa bilimlerinde önem arz eden bir bilimsel hesaplama yöntemidir. Bu hesaplama yöntemi analitik olarak çözülemeyen matematiksel analiz problemlerinin yaklaşık çözümlerinde kullanılan algoritmaları inceler[1].

Doğa bilimleri ve mühendisliklerde problemlerin çözümü analitik çözüm ve nümerik çözüm olmak üzere ikiye ayrılır. Analitik çözümlerle, genellikle kabul edilmiş veriler üzerinden daha kesin sonuçlara ulaşılırken, nümerik çözümlerle belirli kabullerin etkisi ile kesin değerler elde edilememektedir.

Bazı problemlerin analitik çözümü çok karmaşık olabilir veya analitik çözüm bulunmayabilir. Analitik çözümlerin yapılamadığı durumlarda yaklaşık çözümler bulmak için nümerik analiz ortaya çıkar[2]. Nümerik analizde çözülmesi zor olan problemlere çözüm oluşturmaya algoritma oluşturma denir[3]. Algoritma, bir problemin çözümü için gerekli olan sonlu sayıdaki işlemleri adım adım belirli sıra ile tanımlayan düzendir[4]. Burada çözümlenmesi zor olan problemlerin çözümü için bilgisayarlar devreye girer. Algoritmaların bilgisayarın anlayacağı şekilde yani belirli kodlar ile makine diline çevrilmesi ile programlama yapılır ve çözülmesi zor olan problemler bilgisayar aracılığı ile kolayca çözümlenir.

Mathematica 1991 yılında Stephen Wolfram tarafından yayınlanan bir yazılım paket programıdır[5]. Bu program diğer yazılım dillerine kıyasla özellikle sembolik hesaplama, sayısal analiz, grafik ve görselleştirme, programlama kolaylığı ve geniş fonksiyon kütüphanesi gibi alanlarda sunduğu avantajlar nedeniyle tercih edilmektedir. MATLAB sayısal hesaplamalarda güçlü olsa da sembolik işlemler konusunda sınırlıdır, Maple sembolik hesaplama açısından güçlüdür ancak görselleştirme ve geniş kapsamlı analizlerde Mathematica kadar esnek değildir, Python ise açık kaynaklı olup geniş kütüphaneleri sayesinde esneklik sunar, ancak Mathematica'nın sunduğu entegre ve optimize edilmiş ortam kadar kullanıcı dostu değildir. Mathematica, Wolfram Engine sayesinde hem sembolik hem de sayısal hesaplamaları bir arada yapabilirken, karmaşık matematiksel işlemleri minimal kod ile gerçekleştirme imkânı sağlar. Bunun yanı sıra, birçok işlemi otomatikleştirilmiş algoritmalar ve optimizasyon teknikleri ile hızlandırarak kullanıcının en uygun yöntemi manuel olarak seçmesine gerek bırakmaz. MATLAB veya Python'da uzun algoritmalar gerektiren işlemler, Mathematica'da tek bir satırla yapılabilir. Ayrıca, yüksek kaliteli grafikler ve interaktif görselleştirme seçenekleri, otomatikleştirilmiş hesaplama yetenekleri ve Wolfram Alpha entegrasyonu gibi özellikler, bilimsel araştırmalar, akademik çalışmalar ve mühendislik projeleri için onu güçlü bir seçenek haline getirir. Wolfram Alpha ile doğrudan bağlantılı olması sayesinde birçok matematiksel ve bilimsel veriye anında erişim sağlar, böylece veri toplama ve analiz sürecinde ciddi bir zaman tasarrufu sunar. Etkileşimli ve kullanıcı dostu yapısıyla, uzun ve karmaşık kod yazma ihtiyacını azaltarak diğer dillere kıyasla zaman kaybını büyük ölçüde ortadan kaldırır. Bu yönleriyle Mathematica hem teorik hem de uygulamalı bilimlerde sıkça tercih edilmektedir.

Bu makalede lineer olmayan denklemlerin Mathematica ile nümerik analizleri incelenmiştir. Makalenin ilerleyen kısmında ilk olarak lineer olmayan denklemin ne olduğu ve nasıl çözümlenmesi gerektiği açıklanmıştır. Daha sonra lineer olmayan denklemlerin nümerik analizinde kullanılan Bisection Yöntemi, Sabit Nokta Yöntemi, Newton-Raphson Yöntemi, Secant Yöntemi ve Regula Falsi Yöntemi açıklanmıştır. Son olarak ise verilen yöntemlerin ilgili örneği Mathematica kullanarak çözülmüş ve sonuçları elde edilmiştir.

## 2 MATERYAL VE METOT

Matematik ve mühendisliğin çeşitli alanlarında karşımıza çıkan iki veya daha yüksek dereceden polinomlar veya trigonometrik, logaritmik, üstel gibi lineer olmayan terimler içeren denklemlere lineer olmayan veya nonlinear denklemler denir. Örneğin,  $x^3 + 2x^2 - 5\sin x = 0$  veya  $x - \tan x = e^{-x}$  denklemleri lineer olmayan denklemlerdir.

Lineer olmayan denklemlerin nümerik analizleri yani çözümleri  $f(x) = 0$  şeklinde verilen denklemin köklerinin bulunması problemidir. Bulunan bu köke  $f$  fonksiyonunun sıfırı da denir. Lineer olmayan bir denklem sistemi verildiğinde problemin çözümü için kullanılacak yöntem verilen denklemin birkaç kökünün veya bütün köklerinin bulunmasına köklerin gerçel veya kompleks olmasına, köklerin yaklaşık değerinin bilinip bilinmemesine bağlıdır[6-7-8]. Bir lineer olmayan denklemin bazı veya bütün köklerini bulmak için karşımıza çıkan başlıca yöntemler şunlardır:

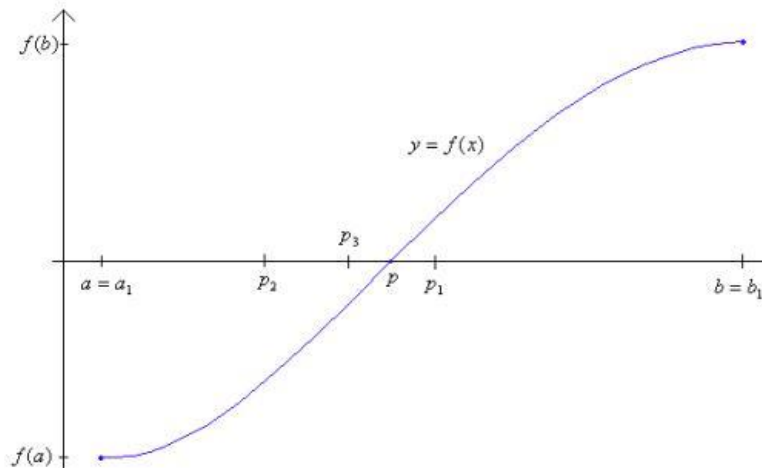
- Bisection Yöntemi (İkiye Bölme Yöntemi-Yarıya Bölme Yöntemi)
- Sabit Nokta Yöntemi
- Newton-Raphson Yöntemi
- Secant Yöntemi (Kiriş Yöntemi)
- Regula Falsi Yöntemi (Değişik İşaret Yöntemi)

### 2.1 Bisection yöntemi (İkiye bölme yöntemi-Yarıya bölme yöntemi)

Bisection Yöntemi Ara Değer Teoremine dayanır. Bu yöntemde,  $f$  fonksiyonu verilen  $[a, b]$  aralığında sürekli,  $f(a)$  ile  $f(b)$  zıt işaretli olmalıdır. Ara Değer Teoremi,  $f(p) = 0$  olmak üzere  $(a, b)$  aralığında bir  $p$  sayısının olduğunu söyler. Bu durum  $(a, b)$  aralığında birden fazla kök olduğu durumlarda da geçerlidir. Burada,  $(a, b)$  aralığında tek bir kökün olduğu varsayılarak işlemler yapılacaktır. Bu yöntemin amacı Şekil 1. de gösterildiği gibi  $p$  değerini içeren  $[a, b]$  aralığını yarıya bölerek ve ardışık işlemler uygulayarak denklemin köküne ulaşmaktır.

İlk olarak,  $a_1 = a, b_1 = b$  ve  $[a, b]$  aralığının orta noktası  $p_1$  alınarak,  $p_1 = \frac{a_1+b_1}{2}$  olmak üzere,

- 1) Eğer,  $f(p_1) = 0$  ise  $p = p_1$  alınır ve durulur.
  - 2) Eğer,  $f(p_1) \neq 0$  ise  $f(p_1)$  nin işareti ile  $f(a_1)$  ve  $f(b_1)$  nin işaretine bakılır.
    - a) Eğer,  $f(p_1)$  nin işareti ile  $f(a_1)$  nin işareti aynı ise  $a_2 = p_1$  ve  $b_2 = b_1$  alınarak aralık yeniden oluşturulur.  $p \in (p_1, b_1)$ .
    - b) Eğer,  $f(p_1)$  nin işareti  $f(b_1)$  nin işareti ile aynı ise  $a_2 = a_1$  ve  $b_2 = p_1$  alınarak aralık yeniden oluşturulur.  $p \in (a_1, p_1)$ .
- $|f(p)| < \epsilon$  durma koşulu sağlanana kadar yeni oluşturulan aralıklara da bu kural uygulanır.  $|f(p)| < \epsilon$  koşulunun sağlanması ile aranan kökün  $p$  olduğu sonucuna varılır[6-7-8].



Şekil 1. Bisection yöntemi grafiği [9]

## 2.2 Sabit nokta yöntemi

Verilen bir  $g(x)$  fonksiyonunda  $g(p) = p$  eşitliğini sağlayan  $p$  noktasına fonksiyonun sabit noktası denir[10].  $g(x)$  fonksiyonu,  $f(x) = 0$  fonksiyonunun  $x = g(x)$  formunda yazılmış halidir.

Kök bulma problemi ile sabit nokta problemi denk problemler olup, aralarındaki ilişki şöyle açıklanabilir:

Verilen bir  $f(p) = 0$  kök bulma problemi için  $g$  fonksiyonu sabit nokta  $p$  yi verecek şekilde birçok şekilde tanımlanabilir. Örneğin,  $g(x) = x - f(x)$  veya  $g(x) = x + f(x)$  olarak seçilebilir.  $g(x) = x - f(x)$  fonksiyonunda  $x$  yerine  $p$  yazılırsa  $g(p) = p + f(p)$  olur.  $f(p) = 0$  olduğu bilindiğine göre,  $g(p) = p$  elde edilir. Bulunan  $p$  noktası sabit noktadır.

Tersine, sabit noktaya sahip bir  $g$  fonksiyonu verilsin. Sıfırı  $p$  noktası olan bir  $f$  fonksiyonu  $f(x) = x - g(x)$  şeklinde tanımlanabilir. Fonksiyonda  $x$  yerine  $p$  yazılırsa  $g(p) = p$  olduğu bilindiğinden,  $f(p) = p - g(p) = p - p = 0$  elde edilir. Bulunan  $p$  noktası  $f$  fonksiyonunun sıfırı veya kökü olarak adlandırılır[6-7-8].

## 2.3 Newton-Raphson yöntemi

Newton ya da Newton-Raphson Yöntemi kök bulma problemlerini çözmek için kullanılan en güçlü ve en iyi bilinen sayısal analiz yöntemidir. Yöntem, Şekil 2. de gösterildiği gibi seçilen noktadaki teğetin eğiminden yararlanarak köke yakın bir başka noktanın bulunması esasına dayanır.

$f \in C^2[a, b]$  olsun.  $p_0 \in [a, b]$  sayısı  $f'(p) \neq 0$  ve  $|p - p_0|$  küçük bir sayı olmak üzere  $p$  ye yaklaşan bir sayı olsun. (Burada  $p_0$  ilgili probleme Newton-Raphson Yönteminin uygulanması için verilen başlangıç değeridir.)  $f(x)$  fonksiyonu  $x = p$  olacak şekilde  $p_0$  noktası civarında birinci mertebeden Taylor Serisine açılırsa,

$$f(x) = f(p_0) + (x - p_0)f'(p_0) + \frac{(x - p_0)^2}{2!}f''(\xi(x))$$

$$f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2!}f''(\xi(p)) \quad (x = p \text{ den}),$$

Burada  $\xi(p)$ ,  $p$  ile  $p_0$  arasında kalmaktadır.  $f(p) = 0$  olduğundan denklem şu hale gelir:

$$0 = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2!}f''(\xi(p)).$$

Newton Yönteminde  $|p - p_0|$  küçük bir sayı kabul edildiğinden,  $(p - p_0)^2$  çok daha küçüktür. Dolayısıyla,

$$0 \approx f(p_0) + (p - p_0)f'(p_0)$$

olur.  $p$  için çözüm yapılırsa,

$$p \approx p_1 \equiv p_0 - \frac{f(p_0)}{f'(p_0)} \quad (f'(p_0) \neq 0)$$

$$p_2 \equiv p_1 - \frac{f(p_1)}{f'(p_1)} \quad (f'(p_1) \neq 0)$$

$$\vdots$$

$$p_n \equiv p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \quad (1)$$

Newton-Raphson Formülü elde edilir.

Elde edilen her  $p_n$  değeri için,  $|f(p_n)| < \varepsilon$ ,  $|p_n - p_{n-1}| < \varepsilon$ ,  $\left| \frac{p_n - p_{n-1}}{p_n} \right| < \varepsilon$  ( $n = 1, 2, \dots, k$ ) durma koşullarından herhangi biri sağlanıncaya kadar işlemlere devam edilir[6-7-8].

## 2.4 Secant yöntemi (Kiriş yöntemi)

Newton-Raphson Yöntemi son derece güçlü bir yöntem olmasına karşın her yaklaşımda verilen  $f$  fonksiyonunun türevine ihtiyaç duyulacağından bu durum bazen çok karmaşık ve zor hale gelebilmektedir. Bu gibi durumlarda içinde türevi barındırmayan, Newton-Raphson Yönteminin bir varyasyonu olan Secant Yöntemi kullanılır. Secant Yöntemine ait formül şu şekilde elde edilir:

$$f'(p_{n-1}) = \lim_{x \rightarrow p_{n-1}} \frac{f(x) - f(p_{n-1})}{x - p_{n-1}}$$

Burada,  $p_{n-2}$ ,  $p_{n-1}$  e yakınsa,

$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}} = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}$$

Bulunan bu  $f'(p_{n-1})$  yaklaşımını (1) Newton-Raphson formülünde yerine yazarsak,

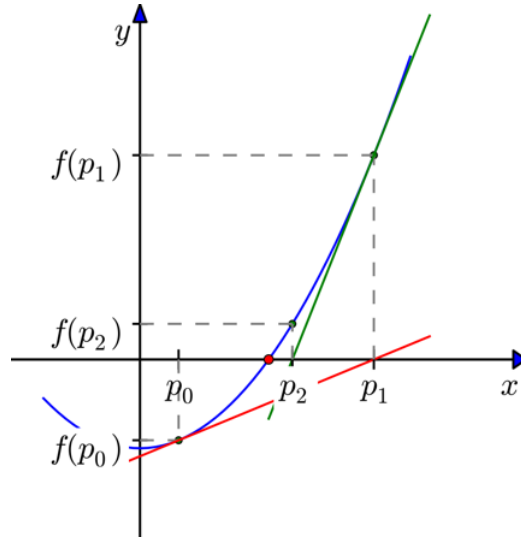
$$p_n = p_{n-1} - \frac{f(p_{n-1})}{\frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}}}$$

Sonuç olarak,

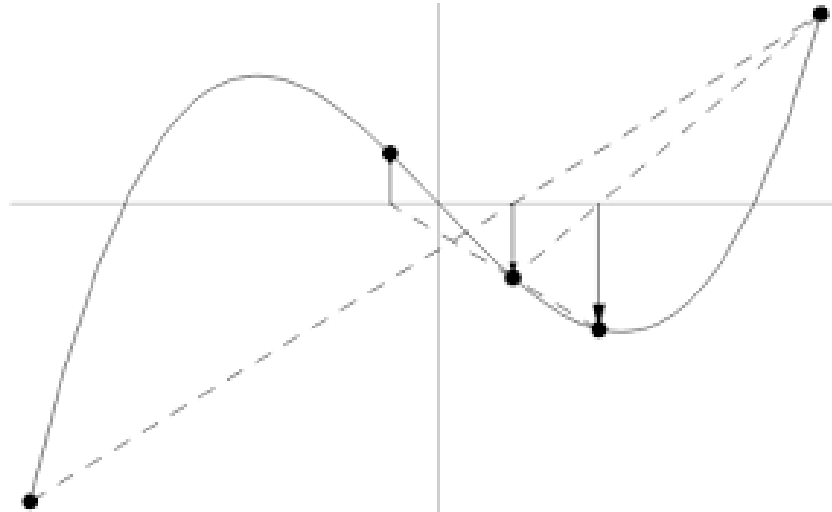
$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})} \quad (2)$$

Secant Formülü elde edilir.

Şekil 3. den de anlaşılacağı üzere Secant Yönteminin uygulanması için  $p_0$  ve  $p_1$  gibi iki başlangıç değerine ihtiyaç vardır. Elde edilen her  $p_n$  değeri için  $|f(p_n)| < \epsilon$  durma koşulu sağlanana kadar işlemlere devam edilir [6-7-8].



Şekil 2. Newton-Raphson yöntemi grafiği [11]



Şekil 3. Secant yöntemi grafiği [12]

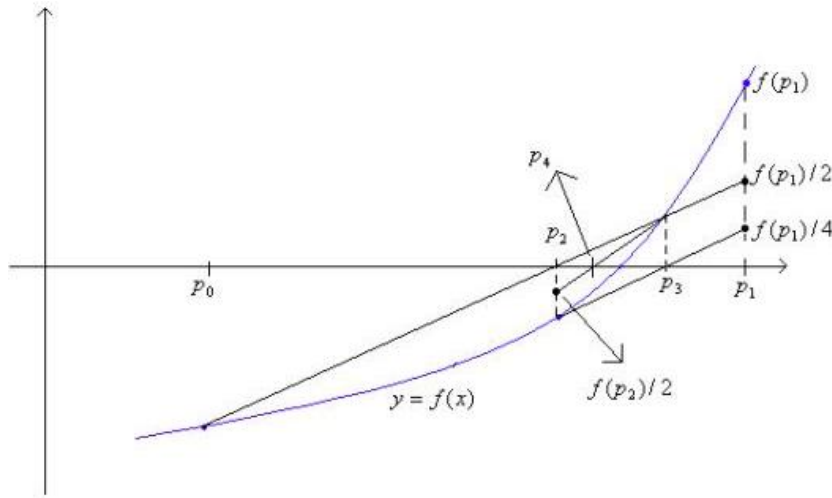
## 2.5 Regula Falsi yöntemi (Değişik işaret yöntemi)

Regula Falsi Yöntemi sıklıkla kullanılan bir yöntem olmamasıyla beraber Secant Yöntemiyle aynı şekilde tahminler üretir, ancak kökün verilen aralık içinde her zaman var olduğundan emin olmak için bazı kontroller yapar.

Bu yöntemde önce,  $f(p_0) \cdot f(p_1) < 0$  koşulunu sağlayan  $p_0$  ve  $p_1$  başlangıç değerleri seçilir. Daha sonra, Secant Yönteminde elde edilen iteratif formül kullanılarak  $(p_0, f(p_0))$  ve  $(p_1, f(p_1))$  noktalarından geçen doğrunun  $x$  eksenini kestiği nokta  $p_2$  olarak bulunur.  $p_3$  noktasını bulmak için  $f(p_0), f(p_1), f(p_2)$  değerlerinin işaretine bakılır.

- Eğer,  $f(p_1) \cdot f(p_2) < 0$  ise  $(p_1, f(p_1))$  ve  $(p_2, f(p_2))$  noktalarını birleştiren doğrunun  $x$  eksenini kestiği nokta  $p_3$  olarak elde edilir.
- Eğer,  $f(p_0) \cdot f(p_2) < 0$  ise  $(p_0, f(p_0))$  ve  $(p_2, f(p_2))$  noktalarını birleştiren doğrunun  $x$  eksenini kestiği nokta  $p_3$  olarak elde edilir [6-7-8].

Regula Falsi Yöntemi ile çözüm elde etmek için anlatılan bu adımlar Şekil 4.de gösterilmiştir.



Şekil 4. Regula Falsi yöntemi grafiği [13]

## 3 BULGULAR

### 3.1 Bisection yöntemi sonuçları

Şekil 5.de  $f(x) = x^2 - 3$  fonksiyonunun  $[a, b] = [1, 2]$  aralığında tanımlı olan kökünü Bisection Yöntemi kullanarak bulan Mathematica programı verilmiştir.

### 3.2 Sabit nokta yöntemi sonuçları

Şekil 6.da  $f(x) = 3x - 1.5^x + 2\sin(x)$  fonksiyonunun kökünü Sabit Nokta Yöntemi kullanarak bulan Mathematica programı verilmiştir.

### 3.3 Newton-Raphson yöntemi sonuçları

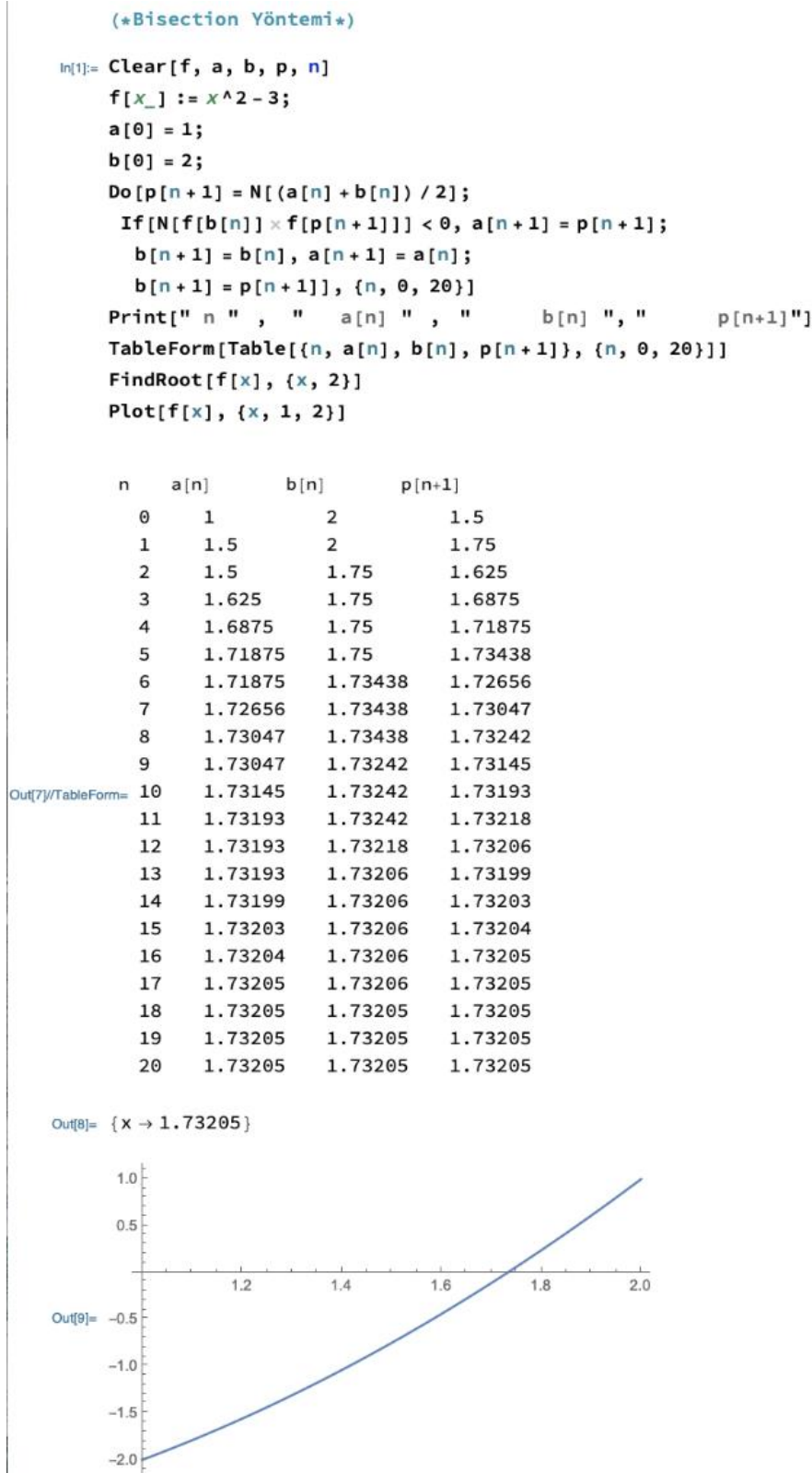
Şekil 7.de  $f(x) = x^3 + 4x^2 - 3$  fonksiyonunun kökünü  $p_0 = 0.7$  başlangıç değerinden başlayarak Newton-Raphson Yöntemi ile bulan Mathematica programı verilmiştir.

### 3.4 Secant yöntemi sonuçları

Şekil 8.de  $f(x) = x^3 - 5x + 1$  fonksiyonunun kökünü  $p_0 = 1$  ve  $p_1 = 2$  değerlerinden başlayarak Secant Yöntemi ile bulan Mathematica programı verilmiştir.

### 3.5 Regula Falsi yöntemi sonuçları

Şekil 9.da  $f(x) = x - \cos(x)$  fonksiyonunun kökünü  $p_0 = 1$  ve  $p_1 = 2$  değerlerinden başlayarak Regula Falsi Yöntemi ile bulan Mathematica programı verilmiştir.

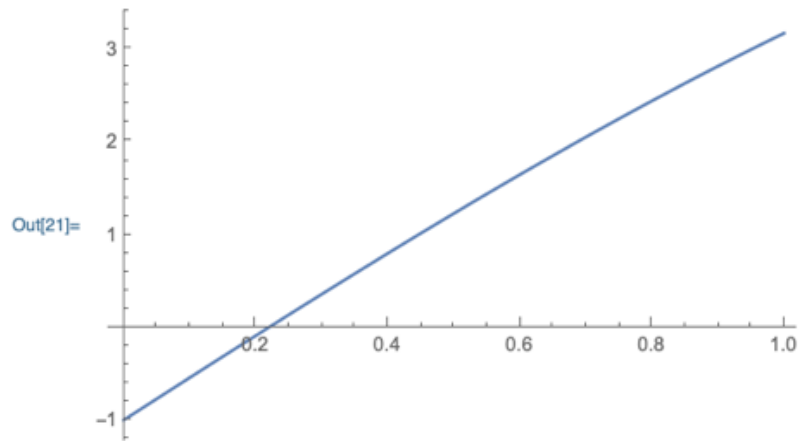


Şekil 5. Mathematica'da Bisection yöntemi

```
(*Sabit Nokta Yöntemi*)

In[1]:= Clear[f, g, x]
f[x_] := 3 x - 1.5 ^ x + 2 Sin[x];
g[x_] := (1.5 ^ x - 2 Sin[x]) / 3;
x0 = 1;
x1 = g[1];
x2 = g[x1];
x3 = g[x2];
x4 = g[x3];
x5 = g[x4];
x6 = g[x5];
x7 = g[x6];
x8 = g[x7];
x9 = g[x8];
x10 = g[x9];
x11 = g[x10];
x12 = g[x11];
x13 = g[x12];
x14 = g[x13];
x15 = g[x14];
FindRoot[f[x], {x, 0, 1}]
Plot[3 x - 1.5 ^ x + 2 Sin[x], {x, 0, 1}]

Out[20]= {x -> 0.2193}
```



Şekil 6. Mathematica 'da sabit nokta yöntemi

(\*Newton-Raphson Yöntemi\*)

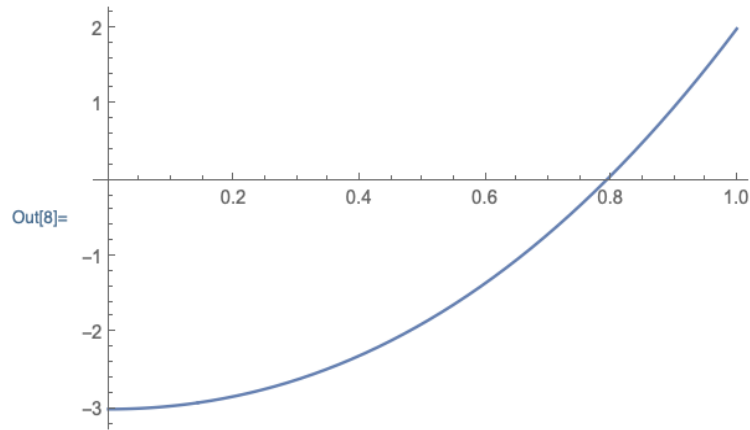
```
In[1]:= ClearAll
f[x_] := x^3 + 4 x^2 - 3;
p[0] = 0.7;
Do[p[n + 1] = N[(p[n] - (f[p[n]] / f'[p[n]]))], {n, 0, 10}]
Print["n " , " p[n] " ]
TableForm[Table[{n, p[n]}, {n, 0, 10}]]
FindRoot[f[x], {x, 2}]
Plot[f[x], {x, 0, 1}]
```

Out[1]= ClearAll

| n  | p[n]     |
|----|----------|
| 0  | 0.7      |
| 1  | 0.798586 |
| 2  | 0.791329 |
| 3  | 0.791288 |
| 4  | 0.791288 |
| 5  | 0.791288 |
| 6  | 0.791288 |
| 7  | 0.791288 |
| 8  | 0.791288 |
| 9  | 0.791288 |
| 10 | 0.791288 |

Out[6]/TableForm=

Out[7]= {x → 0.791288}



Şekil 7. Mathematica 'da Newton-Raphson yöntemi

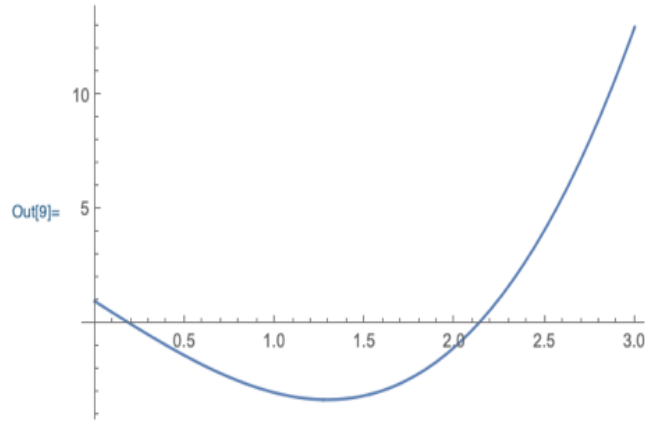
(\*Secant Yöntemi\*)

```
In[1]:= ClearAll
f[x_] := x^3 - 5 x + 1;
p[0] = 1;
p[1] = 2;
Do[p[n+1] = p[n] - ((p[n] - p[n-1]) / (f[p[n]] - f[p[n-1]])) f[p[n]], {n, 1, 10}]
Print[" n " , " p[n] " , " f[p[n]] "]
TableForm[Table[{n, N[p[n-1]], N[f[p[n-1]]}], {n, 1, 10}]
FindRoot[f[x], {x, 2}]
Plot[f[x], {x, 0, 3}]
```

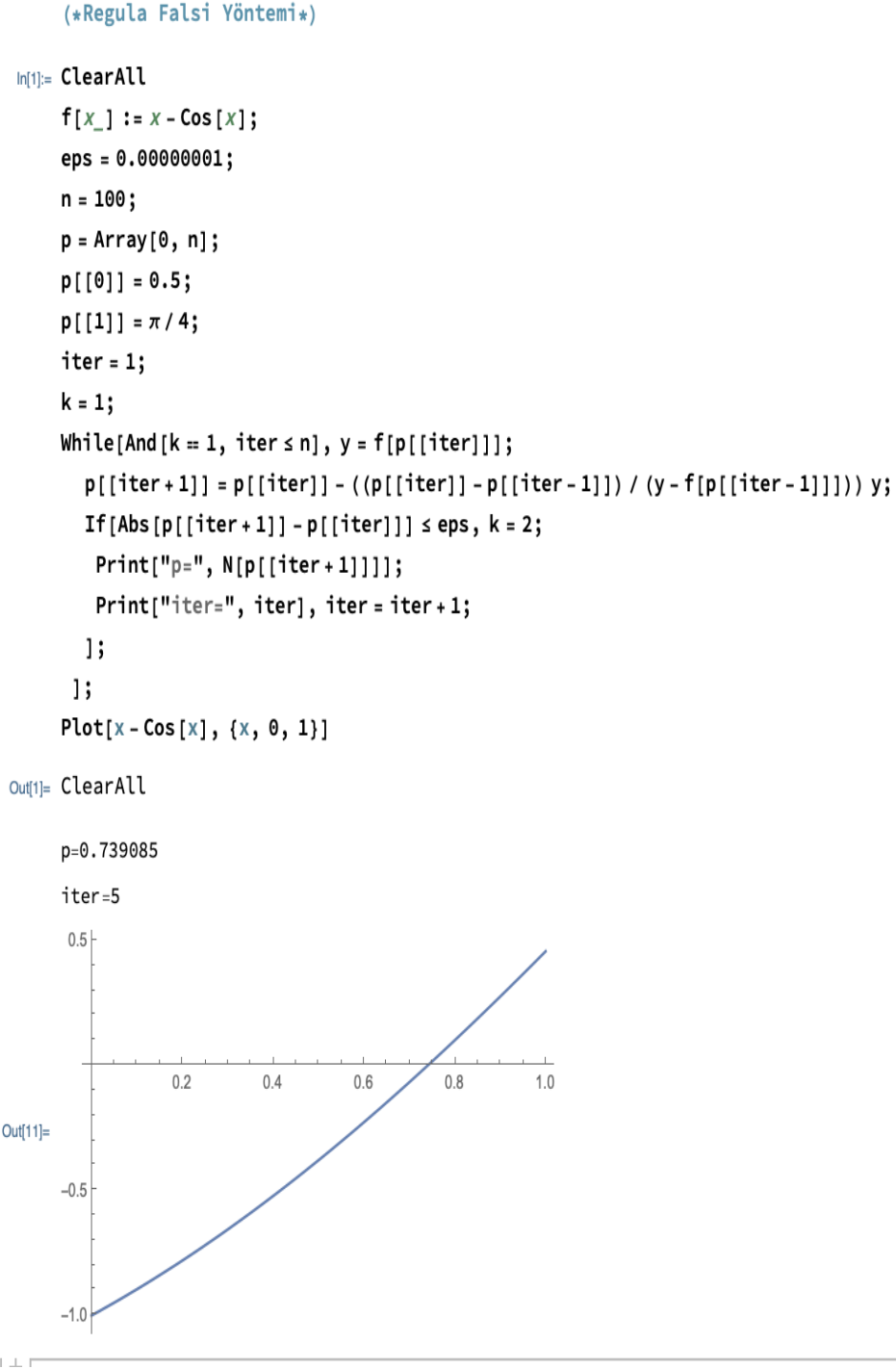
Out[1]= ClearAll

| n                   | p[n]    | f[p[n]]                    |
|---------------------|---------|----------------------------|
| 1                   | 1.      | -3.                        |
| 2                   | 2.      | -1.                        |
| 3                   | 2.5     | 4.125                      |
| 4                   | 2.09756 | -0.259036                  |
| 5                   | 2.12134 | -0.060497                  |
| Out[7]/TableForm= 6 | 2.12859 | 0.00142684                 |
| 7                   | 2.12842 | $-7.5379 \times 10^{-6}$   |
| 8                   | 2.12842 | $-9.30406 \times 10^{-10}$ |
| 9                   | 2.12842 | $6.06826 \times 10^{-16}$  |
| 10                  | 2.12842 | $-4.88515 \times 10^{-26}$ |

Out[8]= {x → 2.12842}



Şekil 8. Mathematica 'da Secant yöntemi



Şekil 9. Mathematica 'da Regula Falsi yöntemi

## 4 SONUÇLAR

Bu çalışmada lineer olmayan denklemlerin çözümü için kullanılan nümerik analiz yöntemleri açıklandı ve açıklanan yöntem ile alakalı örnek Mathematica kodu yazılıp, çıktısı gösterildi. Bu Mathematica kodlarında yer alan değişkenler değiştirilerek farklı örnekler oluşturulabilir, sonuçları incelenebilir. Mathematica yazılımının güçlü sembolik hesaplama özellikleri, türev alma, kök tahmini ve grafiksel gösterimler gibi işlemleri çalışmanın hızlı ve verimli bir şekilde gerçekleştirilmesini sağladı. Bu sayede nümerik analiz süreci daha anlaşılır hale geldi.

Gelecek çalışmalar kapsamında, araştırmının daha geniş bir yelpazeye yayılması amaçlanmaktadır. Bu amaçla öncelikle, çok değişkenli lineer olmayan denklem sistemleri, diferansiyel denklemler ve integral denklemler gibi daha karmaşık matematiksel yapılar üzerinde çalışmalar yapılacaktır. Daha sonra ise bu çalışmaların farklı

mühendislik problemlerine yönelik gerçek dünya uygulamaları ele alınarak nümerik analizlerin teorik boyuttan çıkarılıp pratik mühendislik problemlerine nasıl bir katkı sunduğu değerlendirilecektir.

### Yazar Katkıları

**Esin İBİŞ:** Kavramlaştırma, Metodoloji, Yazılım, Doğrulama, Veri Analizi, Araştırma, Materyaller / Kaynaklar, Veri İyileştirme, Yazım - Özgün Taslak, Yazım - Değerlendirme & Düzenleme, Görselleştirme, Süpervizyon, Proje yönetimi, Finansman temini

Yazarlar makalenin son halini okuyup onaylamışlardır.

### Çıkar Çatışması Beyanı

Yazarlar herhangi bir çıkar çatışması olmadığını beyan eder.

### Kaynakça

- [1] VİKİPEDİ, “Sayısal Analiz,” 2024. [Online]. Mevcut: [https://tr.wikipedia.org/wiki/Sayısal\\_analiz](https://tr.wikipedia.org/wiki/Sayısal_analiz) [Erişim Tarihi: 08.10.2024].
- [2] R. Tapramaz, *Sayısal Çözümleme*. Literatür Yayınları; İstanbul, 2002.
- [3] H. Mutuk, “Mathematica ile Sayısal Çözümleme Programları,” *Kilis 7 Aralık Üniversitesi Fen ve Mühendislik Dergisi*, vol. 2, pp. 37-55, 2018.
- [4] Z. Aktaş, H. Öncül, S. Ural, *Sayısal Çözümleme*. ODTÜ Yayınları; Ankara, 1991.
- [5] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*. Addison Wesley Publishing Company, 1991.
- [6] R.L. Burden, J.D. Faires, *Numerical Analysis*. Brooks-Cole, Boston: 9. Baskı, 2010.
- [7] F.B. Hildebrand, *Introduction to Numerical Analysis*. Mc Graw Hill: 2. Baskı, 1974.
- [8] B. Çağal, *Sayısal Analiz*. Seç Yayınları, İstanbul, 1990.
- [9] BİLKENT ÜNİVERSİTESİ, “Bisection Yöntemi Grafiği,” 2024. [Online]. Mevcut: <https://kilyos.ee.bilkent.edu.tr/~microwave/programs/utilities/numeric1/Bisection.htm> [Erişim Tarihi: 08.10.2024].
- [10] E. Kreyszig, *Introductory Functional Analysis with Applications*. John Willey & Sons, New York, 1978.
- [11] WIKIMEDIA, “Newton-Raphson Yöntemi Grafiği,” 2024. [Online]. Mevcut: [https://commons.wikimedia.org/wiki/File:Newton\\_method\\_scheme.svg](https://commons.wikimedia.org/wiki/File:Newton_method_scheme.svg) [Erişim Tarihi: 08.10.2024].
- [12] WOLFRAM MATHWORLD, “Secant Yöntemi Grafiği,” 2024. [Online]. Mevcut: <https://mathworld.wolfram.com/SecantMethod.html> [Erişim Tarihi: 08.10.2024].
- [13] BİLKENT ÜNİVERSİTESİ, “Regula Falsi Yöntemi Grafiği,” 2024. [Online]. Mevcut: <https://kilyos.ee.bilkent.edu.tr/~microwave/programs/utilities/numeric1/MRegula.htm> [Erişim Tarihi: 08.10.2024].