Araştırma Makalesi / *Research Article* Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 9(4): 2048-2061, 2019
Fizik / Physics *Journal of the Institute of Science and Technology*, 9(4): 2048-2061, 2019

DOI: 10.21597/jist.549716 ISSN: 2146-0574, eISSN: 2536-4618

İkinci Mertebeden Lineer Diferansiyel Denklemlerin Runge-Kutta, Kuvvet Serisi ve Laplace Dönüşümü Yöntemleri ile Mathematica'da Çözümü

Halil MUTUK[1]*

ÖZET: Bu çalışmada mühendislik ve fen bilimleri alanlarındaki uygulamalarda sıkça yer bulan ikinci mertebeden lineer diferansiyel denklemlerin Runge-Kutta, Kuvvet Serisi ve Laplace Dönüşümü Yöntemleri ile çözümü ele alınmıştır. Bu yöntemlerin kısa bir özeti verildikten sonra Mathematica programlama dilinde algoritmalar oluşturularak örnek problemlerin çözümü yapılmıştır.

Anahtar Kelimeler: Runge-Kutta Yöntemi, Kuvvet Serisi Yöntemi, Laplace Dönüşümü Yöntemi, Mathematica

Solutions of Second Order Linear Differential Equations via Runge-Kutta, Power Series and Laplace Transformation in Mathematica

ABSTRACT: In this work, solutions of second order linear differential equations which frequently appears in science and engineering fields are given via Runge-Kutta, Power Series and Laplace Transformation methods. After giving short introductions about these methods, some problems are solved with the related algorithms in Mathematica software.

Keywords: Runge-Kutta Method, Power Series Method, Laplace Transformation Method, Mathematica

[1] Halil MUTUK (**Orcid ID:** 0000-0002-6794-0879), Ondokuz Mayıs Üniversitesi, Fen-Edebiyat Fakültesi, Fizik Bölümü, Samsun, Türkiye

*Sorumlu Yazar / Corresponding Author: Halil MUTUK, e-posta: halilmutuk@gmail.com

Geliş tarihi / *Received:* 05.04.2019
Kabul tarihi / *Accepted:* 15.06.2019

GİRİŞ

Teknolojinin gelişimiyle birlikte hayatımıza giren bilgisayarlar yaşamımızı doğrudan etkileyen araçlar olmuştur. Veri depolama ve bilgi işleme gibi süreçlerde sahip olduğu üstün özellikler sayesinde her yaştan insanın kullanımına açık olan bilgisayarlar, fen ve mühendislik alanında neredeyse vazgeçilmezdir.

Tipik bir bilgisayar aşağıdaki şu işlemleri gerçekleştirir:

Girdi işlemleri: Bilgisayara verilerin girilmesi bu aşamada olur. Konuya uygun olarak istenilen bilgiler/veriler bilgisayara girilebilir.

Aritmetik işlemler: Dört işlemi (çarpma, bölme, toplama, çıkarma) ve bunlarla ilintili olan işlemleri insana kıyasla daha hızlı bir şekilde yaparlar.

Çıktı işlemleri: Bilgisayara girilen verilerin/bilgilerin kullanılabilir hale gelmesi işlemidir.

Depolama işlemleri: Bilgisayarlar üstün depolama özelliğiyle ihtiyaç halinde veya sonradan kullanmak üzere programları/verileri/bilgileri depolar.

Altmışlı yıllardan itibaren bilgisayarların ortaya çıkması ile bilginin dijitalleşmesi söz konusu olmuştur. Gelişmiş ülkeler bilginin dijitalleşmesinin önemine varmış ve Bilgi ve İletişim Teknolojileri (BİT) olarak adlandırabilecek bilgi teknolojilerini sınıflara sokmaya ve öğretme-öğrenme etkinliklerine uyarlamaya çalışmışlardır (Şen ve Akdeniz, 2017). Bu süreç halen devam etmekle birlikte ülkemizin de arasında bulunduğu gelişmekte olan ülkeler, bu süreci uygulamaya çalışmaktadır.

Baş döndürücü bir gelişime sahip olan bilgisayarlar fen ve mühendislik alanlarında sıkça kullanılmasıyla birlikte özellikle eğitimle bütünleşmesi eğitici ve öğrenciler açısından son derece faydalıdır. Bilgisayar destekli eğitimle öğretmen-öğrenci etkileşimi, bilişsel ve öğretme süreçleri, öğretim yöntem ve ilkeleri ve uygulama alanı gibi temel süreçler çağın gerektirdiği normlara uygun hale getirilebilir (Gürkaynak, 2015).

Bilgisayar Destekli Eğitim (BDE), bilgisayar ve ilişkili araçların dersteki bir konu veya kavramı öğrencilere öğretmek veya öğrenileni uygulama yoluyla pekiştirmek olarak tanımlanabilir (Şen ve Akdeniz, 2017). Bu yolla eğitimcilerin sınıf ortamında yapamayacağı bazı öğretme aktivitelerinin bilgisayar ortamında yapılması sağlanır. Örneğin matematik derslerinde üç boyutlu şekillerin veya bazı grafiklerin tahtaya elle çizilmesi zor olabilir. İlgili matematik yazılımları ile bu işlem çok daha kısa ve etkin bir biçimde yapılabilir.

Birçok yazılım sembolik hesaplama ile matematiksel işlemleri yapmak üzere piyasaya sürülmüştür. Matlab, Maple ve Mathematica bunlardan akla gelen ilk örneklerdir. Dilimizde Matlab ve Maple ile ilgili kaydadeğer sayıda kitap bulunurken Mathematica ile ilgili çok fazla kitap bulunmamaktadır (Çınar ve Çalışkan, 1995; Çınar, 2000; Sınıksıran ve Aktütün, 2009). Mathematica, matematiksel, cebirsel ve sembolik hesaplamalar yapan genel bir sistemdir. Sembolik bir dil olduğundan veri girişi kolaydır. Mathematica ile türev, integral, denklem çözümleri, diferansiyel denklem çözümleri, matris işlemleri, grafik çizimleri vb. bir çok işlem rahatlıkla yapılabilir.

Bu çalışmanın ikinci bölümünde Mathematica'da yer alan diferansiyel denklem komutları tanıtılacak ve bir kaç örnekle pekiştirilecektir. Üçüncü bölümde bazı ikinci dereceden lineer diferansiyel denklemlerin Runge-Kutta, Seri Açılımı ve Laplace Dönüşümü ile çözümleri verilecektir. Üçüncü ve son bölümde bu sonuçlar tartışılacaktır.

MATERYAL VE YÖNTEM

Diferansiyel denklemler fen ve mühendislikteki birçok olgunun matematiksel olarak modellenmesinde kullanılırlar. Bu diferansiyel denklemlerin çözümleri Laplace dönüşümünde olduğu gibi bazen doğrudan yapılabilmekte, bazen de sayısal yöntemlere ihtiyaç duyulmaktadır. Euler, Runge-Kutta, Seri Açılımı ile çözümleme sayısal yöntemlere verilebilecek ilk örnektirler.

Mathematica'da yer alan DSolve komutu verilen diferansiyel denklemi çözer:

```
?DSolve
```

DSolve[*eqn*, *y*, *x*] solves a differential equation for the function *y*, with independent variable *x*.

DSolve[*eqn*, *y*, {*x*, x_{min}, x_{max}}] solves a differential equation for *x* between x_{min} and x_{max}.

DSolve[{eqn_1, eqn_2, ...}, {y_1, y_2, ...}, ...] solves a list of differential equations.

DSolve[*eqn*, *y*, {x_1, x_2, ...}] solves a partial differential equation. »

Şekil 1. Mathematica'da yerleşik DSolve komutu işlevleri

Şekil 1'den de görüleceği üzere Mathematica diferansiyel denklemi doğrudan çözebileceği gibi belirli bir x aralığında da çözebilir. Buna ek olarak birden fazla diferansiyel denklemi de aynı anda çözebilir. Şekildeki son satır kısmi diferansiyel denklemlerin çözümünü göstermektedir. Örnek bir uygulama olarak

$$\frac{dy}{dx} = xy + x \tag{1}$$

diferansiyel denkleminin Mathematica'da çözümü Şekil 2'de gösterilmiştir:

```
DSolve[y'[x] == x*y[x] + x , y[x] , x]
```

$$\left\{\left\{y[x] \to -1 + e^{\frac{x^2}{2}}\, C[1]\right\}\right\}$$

Şekil 2. Mathematica'da örnek bir diferansiyel denklem çözümü

Mathematica'da yer alan NDSolve komutu verilen diferansiyel denklemin belli sayısal değerler altında çözer. Şekil 3'te NDSolve komutunun işlevi görülmektedir.

```
? NDSolve
```

NDSolve[*eqns*, *u*, {*x*, x_{min}, x_{max}}] finds a numerical solution to the ordinary differential equations *eqns* for the function *u* with the independent variable *x* in the range x_{min} to x_{max}.

NDSolve[*eqns*, *u*, {*x*, x_{min}, x_{max}}, {*y*, y_{min}, y_{max}}] solves the partial differential equations *eqns* over a rectangular region.

NDSolve[*eqns*, *u*, {*x*, *y*} ∈ Ω] solves the partial differential equations *eqns* over the region Ω.

NDSolve[*eqns*, *u*, {*t*, t_{min}, t_{max}}, {*x*, *y*} ∈ Ω] solves the time-dependent partial differential equations *eqns* over the region Ω.

NDSolve[*eqns*, {u_1, u_2, ...}, ...] solves for the functions u_i. »

Şekil 3. Mathematica'da yerleşik NDSolve komutu işlevleri

İlk satırda *eqns* adı altında Mathematica'ya verilen diferansiyel denklem $u = u(x)$ fonksiyonu olarak belirli bir x aralığında çözer. Diğer satırdaki komutlar bu çalışmanın kapsamı dışındadır. Örneğin diferansiyel denklemler dersinde sıkça karşılaşılan başlangıç şartlı $\frac{dy}{dx} = f(x,y), y(x_0) = y_0$ diferansiyel denkleminin Mathematica'daki gösterimi Şekil 4'te gösterilmiştir.

```
NDSolve[{y'[x] == f[y[x], x], y[x0] == y0}, y[x], {x, xmin, xmax}]
```

Şekil 4. Mathematica'da NDSolve komutu ile diferansiyel denklem çözüm komutu

Bir örnek olması açısından

$$\frac{dy}{dx} = xy + x, \qquad y(0) = 1, \qquad 0 < x < 1 \tag{2}$$

diferansiyel denklemini çözelim:

```
NDSolve[{y'[x] == x*y[x] + x, y[0] == 1}, y[x], {x, 0, 1}]

{{y[x] → InterpolatingFunction[ Domain: {{0., 1.}} Output: scalar ][x]}}
```

Şekil 5. Mathematica'da NDSolve komutu ile diferansiyel denklem çözüm komutu

Şekil 5'te yer alan InterpolatingFunction komutu $y(x)$ fonksiyonuna $0 < x < 1$ bölgesinde yaklaşık bir değer sağlar. Sayısal bir çözüm elde etmek için şu işlem yapılmalıdır:

```
cozum = NDSolve[{y'[x] == x*y[x] + x, y[0] == 1}, y[x], {x, 0, 1}]

{{y[x] → InterpolatingFunction[ Domain: {{0., 1.}} Output: scalar ][x]}}

cozum /. x → 1

{{y[1] → 2.29744}}
```

Şekil 6. (2) denkleminin $x = 1$'de yaklaşık çözümü

Verilen diferansiyel denklem, *cozum* olarak adlandırdıktan sonra $x = 1$ noktasında yaklaşık çözüm elde edilmiştir.

Mathematica'nın sahip olduğu bu yerleşik komutlar sayısal hesap yapmayı büyük ölçüde kolaylaştırır. Sembolik bir dil olduğundan veri girişinin kolay olduğu daha önce belirtilmişti. Bununla birlikte Mathematica programlama açısından da uygun bir dildir. Problem çözümü için sahip olduğunuz algoritma doğru bir şekilde Mathematica'ya girildiğinde Mathematica size çözümü verecektir. Yerleşik komutlar hazır bulunuşluk açısından yararlı gözükse de öğrencilerin, öğretmenlerin ve akademisyenlerin ilgili problemler için bir algoritma oluşturup bunu bilgisayar diline dönüştürmesi önemlidir.

İkinci mertebeden diferansiyel denklemler lineer ve lineer olmayan olarak ikiye ayrılabilirler. Genellikle lineer diferansiyel denklemlerin çözümleri analitik olarak yapılabilmektedir. Lineer olmayan denklemlerin analitik çözümlerini bulmak neredeyse imkansızdır. Bu tip denklemler sayısal yöntemlerle çözülebilmektedir. Bu sayısal yöntemler lineer denklemlerin çözümleri için de uygulanabilir.

$$\frac{d^2y}{dx^2}+P(x)\frac{dy}{dx}+Q(x)y=R(x) \tag{3}$$

şeklindeki ikinci mertbeden diferansiyel denklemler $R(x)=0$ olması durumunda homojen, aksi halde homojen olmayan denklem olarak adlandırılır. $P(x)$ ve $Q(x)$ katsayıları değişken olabileceği gibi sabit de olabilir.

Runge-Kutta Yöntemi

$$\frac{d^2y}{dx^2}=f(y',y,x) \tag{4}$$

$$y(x_0)=y_0, \qquad y(x_1)=y_1$$

başlangıç değer problemi Runge-Kutta yöntemi ile çözülebilir. Runge-Kutta yöntemleri ikinci dereceden, üçüncü dereceden, dördüncü dereceden ve daha yüksek derecelerden oluşabilmektedir. Bu yöntemlerde Taylor serisi açılımı veya Euler yöntemindeki gibi fonksiyonun türevlerine gerek yoktur. Ele alınan alt aralıklardaki seçilen noktalarda fonksiyonun değerleri kullanılır (Akın, 1998).

$$y_{n+1}=y_n+c_1hf(x_n,y_n)+c_2hf(x_n+ah,y_n+bk_1) \tag{5}$$

indirgeme bağıntısı ele alınsın. Bu bağıntıda yer alan h adım miktarı olup, c_1,c_2,a,b sabitleri Taylor algoritması ile uyumlu olacak şekilde hesaplanması gerekli olan sabitlerdir. (x_n,y_n) komşuluğunda Taylor açılımı yazılırsa

$$y(x_{n+1})=y(x_n)+hy'(x_n)+\frac{h^2}{2}y''(x_n)+\frac{h^3}{3}y'''(x_n)$$

$$= y(x_n)+hf(x_n,y_n)+\frac{h^2}{2}\left(f_x(x_n,y_n)+f(x_n,y_n)f_y(x_n,y_n)\right)+$$

$$\frac{h^3}{6}\begin{bmatrix} f_{xx}(x_n,y_n)+2f(x_n,y_n)f_{xy}(x_n,y_n)+f_{yy}(x_n,y_n)f^2(x_n,y_n) \\ +f_x(x_n,y_n)f_y(x_n,y_n)+f_y^2(x_n,y_n)f(x_n,y_n) \end{bmatrix}+O(h^4) \tag{6}$$

$$\coloneqq y(x_n)+hf(x_n,y_n)+\frac{h^2}{2}(f_x+ff_y)|_{(x_n,y_n)}$$

$$+\frac{h^3}{6}\left(f_{xx}+2ff_{xy}+f_{yy}f^2+f_xf_y+f_y^2f\right)+O(h^4)$$

elde edilir (Akın, 1998). (5) nolu ifadede yer alan $f(x_n+ah, y_n+bk_1)$ yerine iki değişkenli fonksiyonların Taylor açılımı kullanılarak

$$f(x_n+ah, y_n+bk_1)=f(x_n,y_n)+ahf_x+bk_1f_y+\frac{a^2h^2}{2}f_{xx}+ahbk_1+f(x,y)$$
$$+\frac{b^2k_1^2}{2}f_{yy}+O(h^3) \tag{7}$$

yazılabilir. Böylece (5) nolu ifade

$$y_{n+1}=y_n+(c_1+c_2)hf+c_2h^2\left(af_x+bff_y\right)+c_2h^3\left(\frac{a^2}{2}f_{xx}+abff_{xy}+\frac{b^2}{2}f_{yy}\right)+O(h^4) \tag{8}$$

şeklinde yazılabilir. Bu son ifade (6) nolu eşitlikte yer alan h ve h^2'nin katsayıları ile eşitlenirse

$$c_1+c_2=1$$
$$ac_2=bc_2=\frac{1}{2} \tag{9}$$

bulunur. Bu sistemde üç denklem yer almasına rağmen dört bilinmeyen vardır bundan dolayı bilinmeyenlerden birinin keyfi seçimi halinde sistem çözülebilir. Bu çözümlerden bir tanesi $c_1=c_2=\frac{1}{2}$, $a=b=1$'dir. Böylece (5) nolu yaklaşım formülü

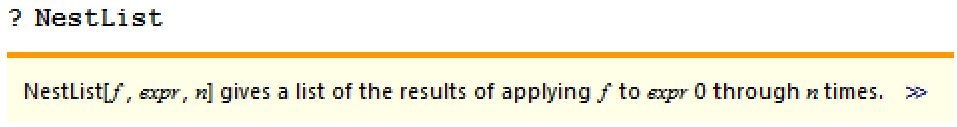
$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_n+h,y_n+hf(x_n,y_n))] \tag{10}$$

halini alır (Akın, 1998).

h adım aralığı olmak üzere 4. Mertebeden Runge-Kutta yöntemi şu şekildedir:

$$k_1=hf(x_n,y_n)$$
$$k_2=hf(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_1)$$
$$k_3=hf(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_2) \tag{11}$$
$$k_4=hf(x_n+h,y_n+k_3)$$
$$y_{n+1}\cong y_n+\frac{1}{6}(k_1+2k_2+2k_3+k_4).$$

Formüllerden de dikkat edileceği üzere her adımda $f(x,y)$ fonksiyonu hesaplanır ve bu değer bir sonraki adımda kullanılarak yeni değer elde edilir. Bu tip işlemler Mathematica'da NestList komutu ile yapılır. Şekil 7'de bu komutun işlevi görülmektedir.



Şekil 7. NestList komutu işlevi

NestList komutu f olarak tanıttığınız ifadeyi $expr$ olarak adlandırılan açıklama ifadesine n defa uygular. Şekil 8'de bir örnek gösterilmiştir.

```
NestList[f, x, 5]

{x, f[x], f[f[x]], f[f[f[x]]], f[f[f[f[x]]]], f[f[f[f[f[x]]]]]}
```

Şekil 8. x ifadesine f'nin 5 kez uygulanması

Runge-Kutta yöntemini ikinci mertebeden diferansiyel denklemlere uygulamak için öncelikle diferansiyel denklemin iki tane birinci mertebeden diferansiyel denklem haline getirmek gerekir. İkinci mertebeden lineer bir diferansiyel denklem mertebe indirme yöntemi uygulanarak birinci mertebeden iki tane diferansiyel denkleme dönüştürülebilir. $y'' = f(x, y, y')$ diferansiyel denkleminde $y' = z$ tanımlaması yapılırsa

$$y'' = z' = f(x, y, z) \tag{12}$$

elde edilir. Buradan hareketle

$$\begin{gathered} y' = z = g(x, y, z) \\ z' = f(x, y, z) \end{gathered} \tag{13}$$

denklem sistemi elde edilebilir.

Bu denklemler için Runge-Kutta formülleri şu şekli alır:

$$\begin{gathered} k_1 = hf(x_n, y_n, z_n) \\ k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, z_n + \frac{1}{2}m_1) \\ k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2, z_n + \frac{1}{2}m_2) \\ k_4 = hf(x_n + \frac{1}{2}h, y_n + k_3, z_n + m_3) \\ y_{n+1} \cong y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{gathered} \tag{14}$$

$$\begin{gathered} m_1 = hg(x_n, y_n, z_n) \\ m_2 = hg(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, z_n + \frac{1}{2}m_1) \\ m_3 = hg(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2, z_n + \frac{1}{2}m_2) \\ m_4 = hg(x_n + \frac{1}{2}h, y_n + k_3, z_n + m_3) \\ z_{n+1} \cong z_n + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4). \end{gathered} \tag{15}$$

Runge-Kutta yönteminin bir uygulaması için hem fiziksel olarak hem de mühendislik alanında önemli bir yere sahip olan kütle-yay sistemini tanımlayan diferansiyel denklem ele alınabilir. Kütle-yay sistemini temsil eden

$$y'' + 2y = 0, \qquad y(0) = 0, \qquad y'(0) = 2 \tag{16}$$

örnek bir diferansiyel denklem ele alınsın (Boyce ve DiPirima, 2001). Bu denklem iki tane birinci mertebeden diferansiyel denklem haline getirilebilir:

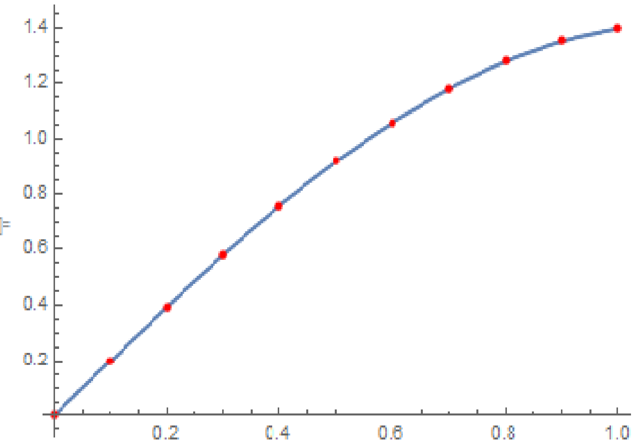
$$y' = z, \quad z' + 2y = 0; \quad y_0 = 0, \quad z_0 = 2 \tag{17}$$

Yöntemin programı ve sonucu Şekil 9'da görülmektedir.

```
In[1]:= {xsifir, ysifir, zsifir, xn} = {0, 0, 2, 1};
h = 0.1;
adim = IntegerPart[(xn - xsifir) / h];
f[x_, y_, z_] := z
g[x_, y_, z_] := -2 * y
k1[x_, y_, z_] := h * f[x, y, z]
k2[x_, y_, z_] := h * f[x + h / 2, y + (k1[x, y, z] / 2), z + (m1[x, y, z] / 2)]
k3[x_, y_, z_] := h * f[x + h / 2, y + (k2[x, y, z] / 2), z + (m2[x, y, z] / 2)]
k4[x_, y_, z_] := h * f[x + h, y + k3[x, y, z], z + m3[x, y, z]]
m1[x_, y_, z_] := h * g[x, y, z]
m2[x_, y_, z_] := h * g[x + h / 2, y + (k1[x, y, z] / 2), z + (m1[x, y, z] / 2)]
m3[x_, y_, z_] := h * g[x + h / 2, y + (k2[x, y, z] / 2), z + (m2[x, y, z] / 2)]
m4[x_, y_, z_] := h * g[x + h, y + k3[x, y, z], z + m3[x, y, z]]
rungekutta[{x_, y_, z_}] :=
 {x + h, y + 1/6 * (k1[x, y, z] + 2 * k2[x, y, z] + 2 * k3[x, y, z] + k4[x, y, z]),
  z + 1/6 * (m1[x, y, z] + 2 * m2[x, y, z] + 2 * m3[x, y, z] + m4[x, y, z])}
cozum1 = NestList[rungekutta, {xsifir, ysifir, zsifir}, adim]

Out[15]= {{0, 0, 2}, {0.1, 0.199333, 1.98003}, {0.2, 0.394687, 1.92053},
 {0.3, 0.582159, 1.82268}, {0.4, 0.758008, 1.68844},
 {0.5, 0.918723, 1.52049}, {0.6, 1.06109, 1.32218}, {0.7, 1.18228, 1.09747},
 {0.8, 1.27986, 0.850846}, {0.9, 1.35188, 0.587234}, {1., 1.39691, 0.311896}}

In[16]:= xdeger = cozum1[[All, 1]]

Out[16]= {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.}

In[17]:= ydeger = cozum1[[All, 2]]

Out[17]= {0, 0.199333, 0.394687, 0.582159, 0.758008, 0.918723, 1.06109, 1.18228, 1.27986, 1.35188, 1.39691}

In[18]:= TableForm[Join[Transpose[{xdeger, ydeger}]]]

Out[18]//TableForm=
0     0
0.1   0.199333
0.2   0.394687
0.3   0.582159
0.4   0.758008
0.5   0.918723
0.6   1.06109
0.7   1.18228
0.8   1.27986
0.9   1.35188
1.    1.39691

In[19]:= DSolve[{y''[x] + 2 * y[x] == 0, y[0] == 0, y'[0] == 2}, y[x], {x, 0, 1}]

Out[19]= {{y[x] → √2 Sin[√2 x]}}
```

```
In[20]:= Show[Plot[√2 Sin[√2 x], {x, 0, 1}],
  ListPlot[%18, Joined → True, Mesh → All, MeshStyle → Directive[PointSize[Medium], Red]]]
```

Out[20]=



Şekil 9. (16) nolu diferansiyel denklem için Runge-Kutta programı

Şekil 9'dan da görüleceği üzere ilk önce başlangıç değerleri ve adım aralığı tanımlanmıştır. Runge-Kutta yöntemi uygulanarak diferansiyel denklem çözülmüş ve $h = 0.1$ adım aralıklarıyla x değeri 0'dan başlayıp 1'e kadar artırılarak her bir x değeri için çözüm bulunmuştur. Daha sonrasında da gerçek çözüm ile birlikte grafiği verilmiştir.

Bu çalışmada kullanılan Runge-Kutta yöntemi 4. dereceden Runge-Kutta yöntemi olarak adlandırılır ve hata mertebesi 5. dereceden olduğundan hassas sonuç verir. 2. Dereceden Runge-Kutta yöntemi var olmakla birlikte hassasiyeti daha azdır (Hatun ve Vatansever, 2016).

Kuvvet Serisi Yöntemi

İkinci mertebeden lineer homojen diferansiyel denklemler mekanik, elektrodinamik, termodinamik ve kuantum mekaniği gibi fizik ve mühendislik dallarında sıkça karşılaşılır (Karaoğlu, 2009). İkinci dereceden sabit katsayılı lineer diferansiyel denklemlerin elle çözümü yapılabilmektedir. Fakat uygulamalı matematik ve fizikte karşılaşılan bazı diferansiyel denklemlerin kapalı form çözümleri elde edilememektedir. Legendre $((1-x)^2y'' - 2xy' + l(l+1)y = 0)$, Bessel $(x^2y'' + xy' + (x^2 - n^2)y = 0)$ ve Hermite $(xy'' + (1-x)y' + \alpha y = 0)$ diferansiyel denklemleri bu tip kapalı formda çözümü elde edilemeyen diferansiyel denklemlere örnektirler. Dolayısıyla ele alınan diferansiyel denklemin, denklemle ilgili bir nokta civarındaki çözümü kuvvet serisi yoluyla araştırılabilir (Pala, 2006).

Değişken katsayılı lineer diferansiyel denklemlerin standart çözüm yöntemi kuvvet serisidir. Çözüm bir $x = a$ noktası civarında fonksiyonu kuvvet serisi olarak yazılmasıyla başlar:

$$y(x) = \sum_{k=0}^{\infty} c_k (x-a)^k. \tag{18}$$

Bu yöntemde diferansiyel denklemi çözmek demek ilgili c_k katsayılarını bulmak demektir. Burada akla gelebilecek bir soru bir diferansiyel denklemin $x = a$ noktasında seri açılımının her zaman var olup olmayacağıdır. Genellikle $x = 0$ noktasında tekillik olmayacağı ve seriye açılabileceği düşünülür. Şekil 10'da (16) nolu denklemin çözümü için bir program gösterilmektedir.

```
In[1]:= cozum = NDSolve[{y''[x] + 2*y[x] == 0, y[0] == 0, y'[0] == 2}, y[x], {x, 0, 1}]

Out[1]= {{y[x] → InterpolatingFunction[ Domain: {{0., 1.}} Output: scalar ][x]}}
```

```
In[2]:= kuvvetserisi1 = Series[y[x], {x, 0, 4}]
```

$$\text{Out[2]= } y[0]+y'[0]\,x+\frac{1}{2}y''[0]\,x^2+\frac{1}{6}y^{(3)}[0]\,x^3+\frac{1}{24}y^{(4)}[0]\,x^4+O[x]^5$$

```
In[3]:= kuvvetserisi2 = Series[y[x], {x, 0, 8}]
```

$$\text{Out[3]= } y[0]+y'[0]\,x+\frac{1}{2}y''[0]\,x^2+\frac{1}{6}y^{(3)}[0]\,x^3+\frac{1}{24}y^{(4)}[0]\,x^4+\frac{1}{120}y^{(5)}[0]\,x^5+\frac{1}{720}y^{(6)}[0]\,x^6+\frac{y^{(7)}[0]\,x^7}{5040}+\frac{y^{(8)}[0]\,x^8}{40\,320}+O[x]^9$$

```
In[4]:= kuvvetserisi3 = Series[y[x], {x, 0, 12}]
```

$$\text{Out[4]= } y[0]+y'[0]\,x+\frac{1}{2}y''[0]\,x^2+\frac{1}{6}y^{(3)}[0]\,x^3+\frac{1}{24}y^{(4)}[0]\,x^4+\frac{1}{120}y^{(5)}[0]\,x^5+\frac{1}{720}y^{(6)}[0]\,x^6+\frac{y^{(7)}[0]\,x^7}{5040}+\frac{y^{(8)}[0]\,x^8}{40\,320}+\frac{y^{(9)}[0]\,x^9}{362\,880}+\frac{y^{(10)}[0]\,x^{10}}{3\,628\,800}+\frac{y^{(11)}[0]\,x^{11}}{39\,916\,800}+\frac{y^{(12)}[0]\,x^{12}}{479\,001\,600}+O[x]^{13}$$

```
In[5]:= difdenk = y''[x] + 2*y[x] == 0;
        bassart = {y[0] -> 0, y'[0] → 2};
```

```
In[7]:= difdenkseri1 = difdenk /. y[x] → kuvvetserisi1
```

$$\text{Out[7]= } (2y[0]+y''[0])+(2y'[0]+y^{(3)}[0])\,x+\left(y''[0]+\frac{1}{2}y^{(4)}[0]\right)x^2+\left(\frac{1}{3}y^{(3)}[0]+\frac{1}{6}y^{(5)}[0]\right)x^3+\left(\frac{1}{12}y^{(4)}[0]+\frac{1}{24}y^{(6)}[0]\right)x^4+O[x]^5==0$$

```
In[8]:= difdenkseri2 = difdenk /. y[x] → kuvvetserisi2
```

$$\text{Out[8]= } (2y[0]+y''[0])+(2y'[0]+y^{(3)}[0])\,x+\left(y''[0]+\frac{1}{2}y^{(4)}[0]\right)x^2+\left(\frac{1}{3}y^{(3)}[0]+\frac{1}{6}y^{(5)}[0]\right)x^3+\left(\frac{1}{12}y^{(4)}[0]+\frac{1}{24}y^{(6)}[0]\right)x^4+\left(\frac{1}{60}y^{(5)}[0]+\frac{1}{120}y^{(7)}[0]\right)x^5+\left(\frac{1}{360}y^{(6)}[0]+\frac{1}{720}y^{(8)}[0]\right)x^6+\left(\frac{y^{(7)}[0]}{2520}+\frac{y^{(9)}[0]}{5040}\right)x^7+\left(\frac{y^{(8)}[0]}{20\,160}+\frac{y^{(10)}[0]}{40\,320}\right)x^8+O[x]^9==0$$

```
In[9]:= difdenkseri3 = difdenk /. y[x] → kuvvetserisi3
```

$$\text{Out[9]= } (2y[0]+y''[0])+(2y'[0]+y^{(3)}[0])\,x+\left(y''[0]+\frac{1}{2}y^{(4)}[0]\right)x^2+\left(\frac{1}{3}y^{(3)}[0]+\frac{1}{6}y^{(5)}[0]\right)x^3+\left(\frac{1}{12}y^{(4)}[0]+\frac{1}{24}y^{(6)}[0]\right)x^4+\left(\frac{1}{60}y^{(5)}[0]+\frac{1}{120}y^{(7)}[0]\right)x^5+\left(\frac{1}{360}y^{(6)}[0]+\frac{1}{720}y^{(8)}[0]\right)x^6+\left(\frac{y^{(7)}[0]}{2520}+\frac{y^{(9)}[0]}{5040}\right)x^7+\left(\frac{y^{(8)}[0]}{20\,160}+\frac{y^{(10)}[0]}{40\,320}\right)x^8+\left(\frac{y^{(9)}[0]}{181\,440}+\frac{y^{(11)}[0]}{362\,880}\right)x^9+\left(\frac{y^{(10)}[0]}{1\,814\,400}+\frac{y^{(12)}[0]}{3\,628\,800}\right)x^{10}+\left(\frac{y^{(11)}[0]}{19\,958\,400}+\frac{y^{(13)}[0]}{39\,916\,800}\right)x^{11}+\left(\frac{y^{(12)}[0]}{239\,500\,800}+\frac{y^{(14)}[0]}{479\,001\,600}\right)x^{12}+O[x]^{13}==0$$

In[10]:=
```
denklem1 = LogicalExpand[difdenkseri1] /. bassart
```

Out[10]=
$$y''[0] == 0 \;\&\&\; 4 + y^{(3)}[0] == 0 \;\&\&\; y''[0] + \frac{1}{2} y^{(4)}[0] == 0 \;\&\&\; \frac{1}{3} y^{(3)}[0] + \frac{1}{6} y^{(5)}[0] == 0 \;\&\&\; \frac{1}{12} y^{(4)}[0] + \frac{1}{24} y^{(6)}[0] == 0$$

In[11]:=
```
denklem2 = LogicalExpand[difdenkseri2] /. bassart
```

Out[11]=
$$y''[0] == 0 \;\&\&\; 4 + y^{(3)}[0] == 0 \;\&\&\; y''[0] + \frac{1}{2} y^{(4)}[0] == 0 \;\&\&\; \frac{1}{3} y^{(3)}[0] + \frac{1}{6} y^{(5)}[0] == 0 \;\&\&\; \frac{1}{12} y^{(4)}[0] + \frac{1}{24} y^{(6)}[0] == 0 \;\&\&\; \frac{1}{60} y^{(5)}[0] + \frac{1}{120} y^{(7)}[0] == 0 \;\&\&\; \frac{1}{360} y^{(6)}[0] + \frac{1}{720} y^{(8)}[0] == 0 \;\&\&\; \frac{y^{(7)}[0]}{2520} + \frac{y^{(9)}[0]}{5040} == 0 \;\&\&\; \frac{y^{(8)}[0]}{20\,160} + \frac{y^{(10)}[0]}{40\,320} == 0$$

In[12]:=
```
denklem3 = LogicalExpand[difdenkseri3] /. bassart
```

Out[12]=
$$y''[0] == 0 \;\&\&\; 4 + y^{(3)}[0] == 0 \;\&\&\; y''[0] + \frac{1}{2} y^{(4)}[0] == 0 \;\&\&\; \frac{1}{3} y^{(3)}[0] + \frac{1}{6} y^{(5)}[0] == 0 \;\&\&\; \frac{1}{12} y^{(4)}[0] + \frac{1}{24} y^{(6)}[0] == 0 \;\&\&\; \frac{1}{60} y^{(5)}[0] + \frac{1}{120} y^{(7)}[0] == 0 \;\&\&\; \frac{1}{360} y^{(6)}[0] + \frac{1}{720} y^{(8)}[0] == 0 \;\&\&\; \frac{y^{(7)}[0]}{2520} + \frac{y^{(9)}[0]}{5040} == 0 \;\&\&\; \frac{y^{(8)}[0]}{20\,160} + \frac{y^{(10)}[0]}{40\,320} == 0 \;\&\&\; \frac{y^{(9)}[0]}{181\,440} + \frac{y^{(11)}[0]}{362\,880} == 0 \;\&\&\; \frac{y^{(10)}[0]}{1\,814\,400} + \frac{y^{(12)}[0]}{3\,628\,800} == 0 \;\&\&\; \frac{y^{(11)}[0]}{19\,958\,400} + \frac{y^{(13)}[0]}{39\,916\,800} == 0 \;\&\&\; \frac{y^{(12)}[0]}{239\,500\,800} + \frac{y^{(14)}[0]}{479\,001\,600} == 0$$

In[13]:=
```
serikatsayilari1 = First[Solve[denklem1]]
```

Out[13]=
$$\{y''[0] \to 0,\ y^{(3)}[0] \to -4,\ y^{(4)}[0] \to 0,\ y^{(5)}[0] \to 8,\ y^{(6)}[0] \to 0\}$$

In[14]:=
```
serikatsayilari2 = First[Solve[denklem2]]
```

Out[14]=
$$\{y''[0] \to 0,\ y^{(3)}[0] \to -4,\ y^{(4)}[0] \to 0,\ y^{(5)}[0] \to 8,\ y^{(6)}[0] \to 0,\ y^{(7)}[0] \to -16,\ y^{(8)}[0] \to 0,\ y^{(9)}[0] \to 32,\ y^{(10)}[0] \to 0\}$$

In[15]:=
```
serikatsayilari3 = First[Solve[denklem3]]
```

Out[15]=
$$\{y''[0] \to 0,\ y^{(3)}[0] \to -4,\ y^{(4)}[0] \to 0,\ y^{(5)}[0] \to 8,\ y^{(6)}[0] \to 0,\ y^{(7)}[0] \to -16,\ y^{(8)}[0] \to 0,\ y^{(9)}[0] \to 32,\ y^{(10)}[0] \to 0,\ y^{(11)}[0] \to -64,\ y^{(12)}[0] \to 0,\ y^{(13)}[0] \to 128,\ y^{(14)}[0] \to 0\}$$

In[16]:=
```
difdenkcozum1 = kuvvetserisi1 /. serikatsayilari1 /. bassart
```

Out[16]=
$$2x - \frac{2x^3}{3} + O[x]^5$$

In[17]:=
```
difdenkcozum2 = kuvvetserisi2 /. serikatsayilari2 /. bassart
```

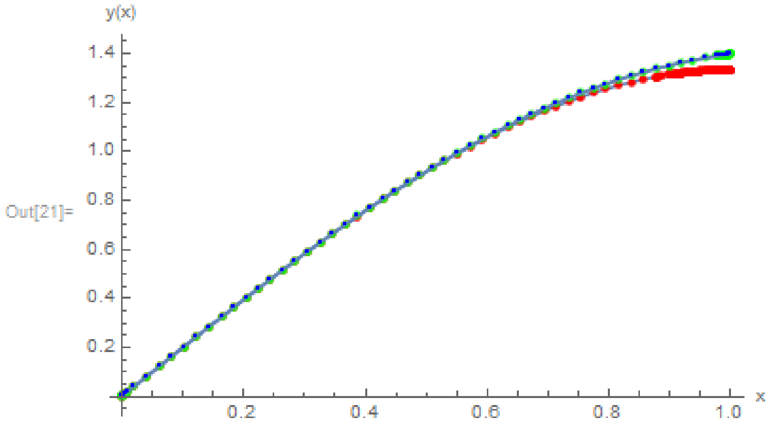
Out[17]=
$$2x - \frac{2x^3}{3} + \frac{x^5}{15} - \frac{x^7}{315} + O[x]^9$$

In[18]:=
```
difdenkcozum3 = kuvvetserisi3 /. serikatsayilari3 /. bassart
```

Out[18]=
$$2x - \frac{2x^3}{3} + \frac{x^5}{15} - \frac{x^7}{315} + \frac{x^9}{11\,340} - \frac{x^{11}}{623\,700} + O[x]^{13}$$

In[19]:=
```
cozumgrafigi1 = Plot[Evaluate[Normal[{difdenkcozum1}]], {x, 0, 1},
   PlotRange → All, AxesLabel → {"x", "y(x)"}, Mesh → All,
   MeshStyle → Directive[PointSize[Medium], Red],
   FrameLabel → {Style["x", 5, "Label"], Style["y(x)", 5, "Label"]}];
cozumgrafigi2 = Plot[Evaluate[Normal[{difdenkcozum2}]], {x, 0, 1},
  PlotRange → All, AxesLabel → {"x", "y(x)"}, Mesh → All,
  MeshStyle → Directive[PointSize[Medium], Green],
  FrameLabel → {Style["x", 5, "Label"], Style["y(x)", 5, "Label"]}];
cozumgrafigi3 = Plot[Evaluate[Normal[{difdenkcozum3}]], {x, 0, 1},
  PlotRange → All, AxesLabel → {"x", "y(x)"}, Mesh → All,
  MeshStyle → Directive[PointSize[Small], Blue],
  FrameLabel → {Style["x", 5, "Label"], Style["y(x)", 5, "Label"]}];
```

```
In[21]:= Show[cozumgrafigi1, cozumgrafigi2, cozumgrafigi3]
```



Şekil 10. (16) nolu denklemin Mathematica'da Kuvvet Serisi ile çözümü

Şekil 10'dan da görüleceği gibi eğer seriye açılan terim daha fazla olursa gerçek çözüme daha da yaklaşılmaktadır. Kırmızı grafikle elde edilen çözümde serinin ilk 4 terimi, yeşille gösterilen grafikte ilk 8 terimi ve maviyle gösterilen grafikte ise ilk 12 terimi kullanılarak çözüm elde edilmiştir.

Laplace Dönüşümü Yöntemi

Laplace dönüşümü bir integral dönüşümü olarak fen ve mühendislik dallarında diferansiyel denklem çözümlerinde en çok kullanılan çözüm yöntemlerinden biridir. Laplace dönüşümündeki amaç başlangıç ya da sınır şartlarına bağlı olan bir diferansiyel denklemi çözümü görece kolay olan cebirsel bir denkleme dönüştürmektir. Bu yöntemin en önemli avantajlarından biri ilgili diferansiyel denklemin keyfi sabitlere bağlı genel çözümünü bulmaktansa doğrudan başlangıç veya sınır değer probleminin çözümünü vermesidir (Pala, 2006).

Bir $f(x)$ fonksiyonunun Laplace dönüşümü, $\mathcal{L}[f]$ şeklinde gösterilir ve şöyle tanımlanır:

$$\mathcal{L}[f] = \int_0^\infty f(t)\, e^{-st} dt. \tag{19}$$

Her $f(t)$ fonksiyonunun Laplace dönüşümünün olup olmaması bazı şartlara bağlıdır. İlk şart, $t = t_0$ gibi bir değerde $f(t_0) = \infty$ oluyorsa (19) nolu denklemdeki integral sonsuz olur. Bunu önlemek için $f(t)$ fonksiyonunun $[0, \infty]$ kapalı aralığında sürekli veya sürekli parçalardan oluştuğu kabul edilir. İkinci şart verilen bu aralıkta fonksiyon sürekli olsa bile integral sonsuz olabilir. Örneğin e^{t^2} fonksiyonunun Laplace dönüşümü tanımsızdır. Dolayısıyla bunun önüne geçmek için her t değerinde fonksiyonun üstel fonksiyondan daha küçük olacağı varsayılır: $|f(t)| \leq e^{bt}$ (Pala, 2006).

Sık karşılaşılan bazı fonksiyonların Laplace dönüşümü elle yapılabilir, $f(t) = 1$, $f(t) = e^{at}$, $f(t) = \cos wt$ gibi. Bunlar genellikle bir tablo halinde bulunabilir. (3) nolu denkleme $P(x) = a$ ve $Q(x) = b$ olmak üzere Laplace dönüşümü uygulandığında

$$L[y''] + a[y'] + \mathrm{b}L[y] = L[r] \tag{20a}$$
$$[s^2Y(s) - sy(0) - y'(0)] + a[sY(s) - y(0)] + bY(s) = R(s) \tag{20b}$$

(25b) denkleminden $Y(s)$ çekilirse

$$Y(s) = \frac{R(s) + (s+a)y(0) + y'(0)}{s^2 + as + b} \tag{21}$$

bulunur. Başlangıç şartları bilindiği takdirde eşitliğin sağ tarafı biliniyor demektir. Bu takdirde çözüm fonksiyonu olan $y(t)$ yi bulmak için ters Laplace dönüşümü yapmak gerekir (Pala, 2006):

$$y(t) = L^{-1}[Y(s)]. \tag{22}$$

Mathematica'da yerleşik olarak LaplaceTransform komutu bulunmaktadır. Şekil 11'de bu komutun işlevi görünmektedir.

```
? LaplaceTransform

LaplaceTransform[expr, t, s] gives the Laplace transform of expr.
LaplaceTransform[expr, {t1, t2, ...}, {s1, s2, ...}] gives the multidimensional Laplace transform of expr. >>
```

Şekil 11. LaplaceTransform komutunun işlevi

Şekil 12'de (16) nolu diferansiyel denklemin Laplace dönüşüm yöntemi ile çözümü verilmektedir.

```
In[1]:= DSolve[{y''[x] + 2*y[x] == 0, y[0] == 0, y'[0] == 2}, y[x], x]
Out[1]= {{y[x] → √2 Sin[√2 x]}}

In[2]:= difdenk = y''[x] + 2*y[x] == 0
Out[2]= 2 y[x] + y''[x] == 0

In[3]:= bassart = {y[0] → 0, y'[0] → 2}
Out[3]= {y[0] → 0, y'[0] → 2}

In[4]:= lapdon = LaplaceTransform[difdenk, x, s] /. bassart
Out[4]= -2 + 2 LaplaceTransform[y[x], x, s] + s^2 LaplaceTransform[y[x], x, s] == 0

In[5]:= cebirdenk = lapdon /. LaplaceTransform[y[x], x, s] → Y[s]
Out[5]= -2 + 2 Y[s] + s^2 Y[s] == 0

In[6]:= cebirdenkcozum[s_] = Y[s] /. Solve[cebirdenk, Y[s]][[1]]
Out[6]= 2/(2 + s^2)

In[7]:= cebirdenkcozum[x_] = InverseLaplaceTransform[cebirdenkcozum[s], s, x]
Out[7]= √2 Sin[√2 x]
```

Şekil 12. (16) nolu denklemin Laplace dönüşüm yöntemi ile Mathematica'da çözümü

Şekil 12'den de görüleceği üzere ilk satırdaki DSolve komutu kullanılarak verilen diferansiyel denklem çözülmüştür. Üçüncü satırdan itibaren Laplace dönüşümü uygulanmış ve son satırda çözüme ulaşılmıştır.

SONUÇ

Bu çalışmada Runge-Kutta, Kuvvet Serisi ve Laplace dönüşüm yöntemleri kullanılarak ikinci mertebeden diferansiyel denklemlerin çözümü araştırıldı. Şüphesiz bu çalışmada yer alan programlardan farklı olarak okuyucu kendi programlarını yazabilirler.

Mathematica'nın sahip olduğu yerleşik komutlar ikinci mertebeden diferansiyel denklemleri çözebilmektedir. Fakat bazı problemlerde istenilen hassasiyette çözüme ulaşabilmek için sayısal çözümleme yöntemlerinin kullanılması gerekebilir. Ayrıca bazı programlarda görüldüğü üzere grafik çizmek çözümün nasıl davrandığı hakkında bilgi vermesi açısından önemlidir. Mathematica, bir, iki ve üç boyutlu grafik çizme konusunda son derece üstün özelliklere sahiptir. Dileyen okuyucu bunu hem çevrimiçi olarak (Wolfram, 2019a) hem de Mathematica kitabından (Wolfram, 1991) bakarak kendi yazılımında deneyebilirler.

Mathematica sembolik bir dil olduğu kadar programlama açısından da uygun bir yazılımdır. Sayısal çözümleme derslerinde Mathematica'nın kullanımı artırılabilir. Özellikle Mathematica'nın çevrimiçi kütüphanesinde birçok yaklaşık yöntemin, veri analizinin ve bulut işlemleri gibi birçok konu hakkında başka insanlarca yazılmış programlar mevcuttur (Wolfram, 2019b) Mathematica'nın sahip olduğu bu geniş kütüphane onu diğer sembolik dillerden ayırmaktadır.

KAYNAKLAR

Akın, Ö. 1998. Nümerik Analiz. Ankara Üniversitesi Yayınları, Ankara-Türkiye

Boyce, W.E. and DiPirima, R.C. 2001. Elemantary Differential Equations and Boundary Value Problems, John Wiley & Sons, New York, USA.

Çınar, M. ve Çalışkan, F. 1995. Mathematica ile Programlama, Beta Basım Yayım Dağıtım, İstanbul-Türkiye.

Çınar, M. Mathematica 3.0 ve 4.0 Sürümü, Seçkin Yayıncılık, Ankara-Türkiye, 2000.

Gürkaynak, G. 2015. Bilgisayar Destekli Matematik Dersinin Mathematica Yazılımı ile İşlenmesine Yönelik Durum Çalışması, Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi, Eğitim Bilimleri Enstitüsü, Yüksek Lisans Tezi (Basılmamış).

Hatun, M ve Vatansever, F. 2016. Differential Equation Solver Simulator For Runge-Kutta Methods, Uludağ University Journal of the Faculty of the Engineering, 21 (1), 145-162.

Karaoğlu, B. 2009. Fizik ve Mühendislikte Matematik Yöntemler, Seçkin Yayıncılık, Ankara-Türkiye.

Pala, Y. 2006. Modern Uygulamalı Diferansiyel Denklemler, Nobel Yayın Dağıtım, Ankara-Türkiye.

Sınıksıran, E. ve Aktütün, A. 2009. Matematik ve İstatistik Uygulamalarıyla Mathematica, Türkmen Kitapevi, İstanbul-Türkiye.

Şen, A. İ. ve Akdeniz, A. R. (Editörler). 2017.Fizik Öğretimi, Kuramsal Bilgiler ve Örnek Etkinlik Uygulamaları, Pegem Akademi, Ankara-Türkiye.

www.wolframalpha.com (Erişim tarihi: 27.03.2019a)

Wolfram, S. 1991. Mathematica: A System for Doing Mathematics by Computer, Addison Wesley Publishing Company, USA,

http://reference.wolfram.com/language/ (Erişim tarihi: 27.03.2019b)