

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MATEMATİK ANABİLİM DALI

İKİNCİ MERTEBE KISMİ DİFERANSİYEL DENKLEMLERİN
NÜMERİK ÇÖZÜMLERİ İÇİN BESSEL SIRALAMA
(COLLOCATION) METODU

DOKTORA TEZİ

ŞUAYİP YÜZBAŞI

KASIM 2012
MUĞLA

MUGLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

ŞUAYİP YÜZBAŞI tarafından hazırlanan **İKİNCİ MERTEBE KİSMİ DİFERANSİYEL DENKLEMLERİN NÜMERİK ÇÖZÜMLERİ İÇİN BESSEL SIRALAMA (COLLOCATION) METODU** başlıklı tezinin, 20/11/2012 tarihinde aşağıdaki jüri tarafından Matematik Anabilim Dalı'nda doktora derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Prof. Dr. Mehmet SEZER (**Jüri Başkanı**)

Matematik Ana bilim Dalı,
Celal Bayar Üniversitesi, Manisa

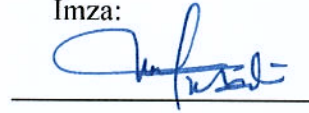
İmza:



Yrd. Doç. Dr. Niyazi ŞAHİN (**Danışman**)

Matematik Ana bilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

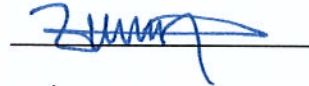
İmza:



Prof. Dr. Zeynep Fidan KOÇAK (**Üye**)

Matematik Ana Bilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Doç. Dr. Mustafa GÜLSU (**Üye**)

Matematik Ana bilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Doç. Dr. Salih YALÇINBAŞ (**Üye**)

Matematik Ana Bilim Dalı,
Celal Bayar Üniversitesi, Manisa

İmza:



ANA BİLİM DALI BAŞKANLIĞI ONAYI

Doç. Dr. Mustafa GÜLSU

Matematik Ana Bilim Dalı Başkan V.,
Muğla Sıtkı Koçman Üniversitesi, Muğla

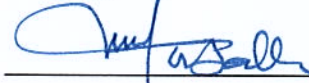
İmza:



Yrd. Doç. Dr. Niyazi ŞAHİN

Danışman, Matematik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:

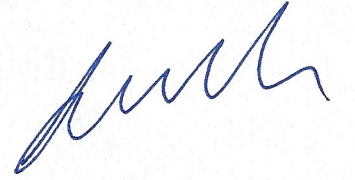


Savunma Tarihi: 20/11/2012

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Şuayip Yüzbaşı

20/11/2012



ÖZET
İKİNCİ MERTEBE KİSMİ DİFERANSİYEL DENKLEMLERİN NÜMERİK
ÇÖZÜMLERİ İÇİN BESSEL SIRALAMA (COLLOCATION) METODU

Şuayip YÜZBAŞI

Doktora Tezi

Fen Bilimleri Enstitüsü

Matematik Anabilim Dalı

Danışman: Yrd. Doç. Dr. Niyazi ŞAHİN

Kasım 2012, 204 sayfa

Bu çalışmada, Neuman, Dirichlet ve integral koşullu dalga problemi, Neuman, Dirichlet ve Robin koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi, başlangıç ve sınır koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi, singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon problemi ve iki boyutlu ikinci mertebeden lineer değişken katsayılı kısmi diferansiyel denklem problemi için Bessel sıralama yöntemi sunulur. Ayrıca, her bir problem için rezidüel hata fonksiyonunu temel alan yöntemin hata tahmini sunulmuştur. Tahmin edilen hata fonksiyonları aracılığıyla Bessel polinom çözümleri iyileştirilir. İlaveten, başlangıç ve sınır koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi için yöntemin karmaşıklığı üzerine çalışılmıştır.

Yöntemin etkinliğini ve tutarlılığını göstermek için, yöntemin pek çok örneğe uygulaması verilmiştir. Hata tahmini ve çözümlerin iyileştirilmeleri de örneklerde yapılmıştır. Tam çözüm bilinen problemlerde gerçek mutlak hatalar ile tahmin edilen mutlak hatalar ve gerçek ve tahmin edilen hata fonksiyonları karşılaştırılmıştır. Bizim sonuçlar ile daha başka metotların sonuçları arasında karşılaştırmalar yapılmıştır.

Anahtar Kelimeler: Kısmi Diferansiyel Denklemler, Bessel Sıralama Metodu, Birinci Tür Bessel Fonksiyonları, Sıralama noktaları

ABSTRACT

BESSEL COLLOCATION METHOD FOR NUMERICAL SOLUTIONS OF SECOND ORDER PARTIAL DIFFERENTIAL EQUATIONS

Şuayip YÜZBAŞI

Doctor of Philosophy (Ph.D.)

Graduate School of Natural and Applied Sciences

Department of Mathematics

Supervisor: Asst. Prof. Dr. Niyazi ŞAHİN

November 2012, 204 pages

In this study, Bessel collocation method is presented for wave problem with Neuman, Dirichlet and integral conditions, second order linear partial differential equation problem in general form under Neuman, Dirichlet and Robin conditions, second order linear partial differential equation problem in general form with initial and boundary conditions, singularly perturbed one-dimensional parabolic convection-diffusion problem and two-dimensional second order linear partial differential equation problem with variable coefficients. Also, the error estimation of the method, based residual error function, is presented for each problem. The Bessel polynomial solutions are improved by means of the estimated error functions. In addition, the complexity of the technique has been studied for second order linear partial differential equation problem in general form with initial and boundary conditions.

To demonstrate the effectiveness and consistency of the method, applications of the technique have been given to several examples. Also, error estimation and improvement of the solutions have been done in examples. In problems with the exact solutions, the actual absolute errors are compared with the estimated absolute errors, and the actual and the estimated error functions are compared. Comparisons between our results and the results of other methods are given.

Keywords: Partial Differential Equations, Bessel Collocation Method, Bessel Functions with First Kind, Collocation Points

Sevgili Eşim, Çocuğum ve Aileme

ÖNSÖZ

Bu çalışmanın ortaya çıkmasını sağlayan, gerek ders aşaması gerek tez aşaması boyunca bilgisini ve desteğini hiç esirgemeyen ve hep yanımda olan danışman hocam sayın Yrd. Doç. Dr. Niyazi ŞAHİN' e, her türlü yardımlarını hiç esirgemeyen sayın Prof. Dr. Mehmet SEZER' e ve her konuda bizlerin yanında olan sayın Doç. Dr. Mustafa GÜLSU' ya, Matematik bölümündeki hepsi birbirinden değerli tüm hocalarıma ve araştırma görevlisi arkadaşlarıma, hayatımın her döneminde benden sevgilerini ve desteklerini hiç esirgemeyen aileme, her türlü konuda her zaman yanımda olan eşim Esin' e ve biricik kızım Beren' e teşekkürlerimi bir borç bilirim.

İÇİNDEKİLER

ÖNSÖZ.....	vi
İÇİNDEKİLER	vii
ÇİZELGELER DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
SEMBOLLER VE KISALTMALAR DİZİNİ	xvi
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ	8
2.1. Kısmi Diferansiyel Denklemler.....	8
2.2. İkinci Mertebe Kısmi Diferansiyel Denklemlerin Sınıflandırılması.....	8
2.2.1. Hiperbolik tip denklemler.....	9
2.2.2. Parabolik tip denklemler.....	12
2.2.3. Eliptik tip denklemler	13
2.2.4. Karışık tip denklemler	15
2.3. Cauchy Problemi	15
2.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Problemi	16
2.5. Koşullar	17
2.5.1. Başlangıç koşulları.....	17
2.5.2. Sınır koşulları.....	17
2.6. Kayan Nokta Aritmetiği (Floating-Point Operation)	19
2.6.1. Vektör-vektör işlemleri.....	20
2.6.2. Matris-vektör çarpımı	21
2.6.3. Matris-matris çarpımı	21
2.6.4. Bir matrisin tersi	22
2.6.5. Gauss eliminasyon	23
2.7. Karmaşıklık (Complexity).....	23
3. MATERYAL VE YÖNTEM.....	24
3.1. Neuman, Dirichlet ve İntegral Koşulları Altında Dalga Denkleminin Yaklaşık Çözümü için Bessel Sıralama Metodu	25
3.1.1. Dalga Denklemi için Temel Matris Bağıntıları	26

3.1.2. Neuman, Dirichlet ve integral koşulları için matris bağıntıları	29
3.1.3. Dalga denklemi problemi için çözüm yöntemi.....	30
3.1.4. Dalga denklemi problemi için hata tahmini.....	33
3.1.5. Dalga denklemi probleminin çözümlerinin iyileştirilmesi	36
3.2. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu.....	36
3.2.1. Genel formda ikinci mertebe lineer kısmi diferansiyel denklem için temel matris bağıntıları	37
3.2.2. Neuman, Dirichlet ve Robin koşulları için matris bağıntıları.....	38
3.2.3. Problem için çözüm yöntemi	40
3.2.4. Hata tahmini.....	43
3.2.5. Çözümlerin iyileştirilmesi	45
3.3. Başlangıç ve Sınır Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu.....	46
3.3.1. Başlangıç ve sınır koşulları için matris bağıntıları	46
3.3.2. Problem için çözüm yöntemi	47
3.3.3. Hata tahmini.....	50
3.3.4. Çözümlerin iyileştirilmesi	52
3.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Probleminin Yaklaşık Çözümü için Bessel Sıralama Metodu	52
3.4.1. Başlangıç ve sınır koşulları için matris bağıntıları	53
3.4.2. Problem için çözüm yöntemi	53
3.4.3. Hata tahmini.....	56
3.4.4. Çözümlerin iyileştirilmesi	58
3.5. İki boyutlu İkinci Mertebeden Lineer Değişken Katsayılı Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu.....	58
3.5.1. İki boyutlu kısmi diferansiyel denklemler için temel matris bağıntıları...	60
3.5.2. Sınır koşulları için matris bağıntıları	62
3.5.3. İki boyutlu kısmi diferansiyel denklem problemi için çözüm yöntemi	63
3.5.4. Hata tahmini.....	70
3.5.5. Çözümlerin iyileştirilmesi	72
3.6. İkinci Mertebeden Lineer Kısmi diferansiyel denklemler için Bessel Sıralama (Collocation) Metodunun Karmaşıklığı	73
4. ARAŞTIRMA BULGULARI	79

4.1. Dalga Denklemi Problemi İçin Örnekler.....	79
4.2. Neuman, Dirichlet ve Robin Koşullu İkinci Mertebe Lineer Kısmi Diferansiyel Denklem Problemi İçin Örnekler	100
4.3. Başlangıç ve Sınır Koşullu İkinci Mertebe Lineer Kısmi Diferansiyel Denklem Problemi İçin Örnekler	107
4.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Problemi için Örnekler	124
4.5. İki Boyutlu İkinci Mertebeden Lineer Kısmi Diferansiyel Denklem Problemi için Örnekler	149
5. TARTIŞMA VE SONUÇLAR	160
KAYNAKLAR	163
EK A. Dalga Problemi İçin Matlab Kodu	171
EK B. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemler İçin Matlab Kodu	178
EK C. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemler İçin Maple Kodu.....	186
EK D. Başlangıç ve Sınır Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemler İçin Matlab Kodu.....	193
EK E. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Problemi İçin Maple Kodu Kısmi Diferansiyel Denklemler İçin Matlab Kodu	200
ÖZGEÇMİŞ.....	204

ÇİZELGELER DİZİNİ

Çizelge 3.1. Bazı N değerleri için metodun karmaşıklığının değerleri	79
Çizelge 4.1. Örnek 4.1.1' in L_2 ve L_∞ hataları	82
Çizelge 4.2. Örnek 4.1.1 için bazı (N, M) değerleri için gerçek ve tahmin edilen maksimum hataların karşılaştırılması	82
Çizelge 4.3. Örnek 4.1.2' nin $N = 3, 7, 10$ için bazı (x_i, t_j) noktalarındaki mutlak hataların maksimum hataların karşılaştırılması	96
Çizelge 4.4. Örnek 4.1.3' ün $u(x_i, 0.5)$ için sunulan yöntemin mutlak hatalarının sonlu farklar metodu ile karşılaştırılması	98
Çizelge 4.5. Örnek 4.2.1' in L_2 ve L_∞ hataları	102
Çizelge 4.6. Örnek 4.2.2' nin gerçek maksimum hataları (L_∞) ve tahmin edilen maksimum mutlak hataların karşılaştırılması	108
Çizelge 4.7. Örnek 4.3.1' in L_2 ve L_∞ hataları	109
Çizelge 4.8. Örnek 4.3.2' nin L_2 ve L_∞ hataları	118
Çizelge 4.9. Örnek 4.3.2' nin $u(x_i, 1)$ için mutlak hatalarının karşılaştırılması	122
Çizelge 4.10. Örnek 4.4.1' in L_2 hataları	126
Çizelge 4.11. Örnek 4.4.1' in L_∞ hataları	127
Çizelge 4.12. Örnek 4.4.1' in Bessel polinom çözümlerinin ve iyileştirilmiş Bessel polinom çözümlerinin maksimum hataları ve tahmin edilen maksimum hataların karşılaştırılması	127
Çizelge 4.13. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$ için $t = 1$ zamanındaki $ R_N(x_i, 1) $ hataları ...	137
Çizelge 4.14. Bizim metodun ve başka metodların N ve ε ' nun çeşitli değerleri için e_ε^N maksimum noktasal hatalarının karşılaştırılması	137
Çizelge 4.15. Örnek 4.5.1' in farklı zamanındaki maksimum hataları	151
Çizelge 4.16. Örnek 4.5.2' nin $N = 5$ için sunulan yöntem ve Taylor matris yönteminin L_2 ve L_∞ hatalarının karşılaştırılması	160

ŞEKİLLER DİZİNİ

Şekil 4.1. Örnek 4.1.1' in $N = 5, 6, 7, 8, 9, 10$ için $ e_N(x, t) $ mutlak hata fonksiyonlarının grafikleri	85
Şekil 4.2. Örnek 4.1.1' in $N = 3$ ve $M = 4$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	86
Şekil 4.3. Örnek 4.1.1' in $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	87
Şekil 4.4. Örnek 4.1.1' in $N = 3$ ve $M = 4$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	88
Şekil 4.5. Örnek 4.1.1' in $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	89
Şekil 4.6. Örnek 4.1.2' nin $N = 7$ ve $M = 8$ için $u(x, t)$ tam çözümü, $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması	92
Şekil 4.7. Örnek 4.1.2' nin $N = 4$ ve $M = 5$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	93
Şekil 4.8. Örnek 4.1.2' nin $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	94
Şekil 4.9. Örnek 4.1.2' nin $N = 4$ ve $M = 5$ için $ e_N(x, t) $ mutlak hata fonksiyonu fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	95
Şekil 4.10. Örnek 4.1.3' ün $u(x, t)$ tam çözümü, $u_7(x, t)$ Bessel polinom çözümü ve $u_{7,8}(x, t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması	98

Şekil 4.11. Örnek 4.1.3 için $ e_7(x,t) $ mutlak hata fonksiyonu ve $ e_{7,8}(x,t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması.....	99
Şekil 4.12. Örnek 4.1.3 için $ e_7(x,t) $ mutlak hata fonksiyonu ve $ E_{7,8}(x,t) $ iyileştirilmiş edilen mutlak hata fonksiyonunun karşılaştırılması	100
Şekil 4.13. Örnek 4.2.1' in $u(x,t)$ tam çözümü, $u_{12}(x,t)$ Bessel polinom çözümü ve $u_{12,13}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması	103
Şekil 4.14. Örnek 4.2.1' in $N = 12$, $M = 13$ ve $M = 14$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x,t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	105
Şekil 4.15. Örnek 4.2.1' in $N = 12$, $M = 13$ ve $M = 14$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x,t) $ iyileştirilmiş mutlak hata fonksiyonlarının karşılaştırılması	106
Şekil 4.16. Örnek 4.3.1' in $N = 3$ ve $M = 4$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x,t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	110
Şekil 4.17. Örnek 4.3.1' in $N = 5$ ve $M = 6$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x,t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	111
Şekil 4.18. Örnek 4.3.1' in $N = 6$ ve $M = 7$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x,t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	112
Şekil 4.19. Örnek 4.3.1' in $N = 8$ ve $M = 9$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x,t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	113
Şekil 4.20. Örnek 4.3.1' in $N = 3$ ve $M = 4$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x,t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	114
Şekil 4.21. Örnek 4.3.1' in $N = 5$ ve $M = 6$ için $ e_N(x,t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x,t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	115

Şekil 4.22. Örnek 4.3.1' in $N = 6$ ve $M = 7$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	116
Şekil 4.23. Örnek 4.3.1' in $N = 8$ ve $M = 9$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	117
Şekil 4.24. Örnek 4.3.2' nin $N = 9$ ve $M = 10$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	119
Şekil 4.25. Örnek 4.3.2' nin $N = 15$ ve $M = 16$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	120
Şekil 4.26. Örnek 4.3.2' nin $N = 9$ ve $M = 10$ için $ e_N(x, t) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	121
Şekil 4.27. Örnek 4.3.3' ün $N = 3$ için Taylor matris yöntemi ve sunulan yöntem ile elde edilen $ e_N(x, t) $ mutlak hata fonksiyonlarının karşılaştırılması	124
Şekil 4.28. Örnek 4.3.3' ün $N = 3$ için diğer Taylor matris yöntemi ve sunulan yöntem ile elde edilen $ e_N(x, t) $ mutlak hata fonksiyonlarının karşılaştırılması	125
Şekil 4.29. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 5$ ve $M = 6$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	128
Şekil 4.30. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	129
Şekil 4.31. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 5$ ve $M = 6$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	130
Şekil 4.32. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	131

Şekil 4.33. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 5$ ve $M = 6$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	132
Şekil 4.34. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	133
Şekil 4.35. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 5$ ve $M = 6$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	134
Şekil 4.36. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 7$ ve $M = 8$ için $ e_N(x, t) $ mutlak hata fonksiyonunun ve $ E_{N,M}(x, t) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	135
Şekil 4.37. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $N = 5$ ve $M = 6$ için için $u(x, t)$ tam çözümü, $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması	138
Şekil 4.38. Örnek 4.4.2' nin $\varepsilon = 2^{-14}$, $N = 5$ ve $M = 6$ için için $u(x, t)$ tam çözümü, $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması	139
Şekil 4.39. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (3, 4)$ ve $(N, M) = (3, 5)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	140
Şekil 4.40. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (4, 5)$ ve $(N, M) = (4, 6)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	141
Şekil 4.41. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (5, 6)$ ve $(N, M) = (5, 7)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	142
Şekil 4.42. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (6, 7)$ ve $(N, M) = (6, 8)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	143
Şekil 4.43. Örnek 4.4.2' nin $\varepsilon = 2^{-14}$, $(N, M) = (3, 4)$, $(N, M) = (4, 5)$, $(N, M) = (5, 6)$ ve $(N, M) = (6, 7)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	145

Şekil 4.44. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N, M) = (3, 4)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	146
Şekil 4.45. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N, M) = (4, 5)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	147
Şekil 4.46. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N, M) = (5, 6)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	148
Şekil 4.47. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N, M) = (6, 7)$ için $ e_{N,M}(x, t) $ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması	149
Şekil 4.48. Örnek 4.5.1' in $t = 1$ zamanındaki $u(x, y, 1)$ tam çözümü ve $N = 4$ için $u_N(x, y, 1)$ Bessel polinom çözümünün karşılaştırılması	152
Şekil 4.49. Örnek 4.5.1' in $t = 1$ zamanındaki $N = 4$ ve $M = 5$ için $ e_N(x, y, 1) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, y, 1) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	153
Şekil 4.50. Örnek 4.5.1' in $t = 1/2$ zamanındaki $N = 4$ ve $M = 5$ için $ e_N(x, y, 1/2) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, y, 1/2) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması.....	154
Şekil 4.51. Örnek 4.5.1' in $t = 1/2$ zamanındaki $N = 4$ ve $M = 5$ için $ e_N(x, y, 1/2) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, y, 1/2) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	155
Şekil 4.52. Örnek 4.5.1' in $t = 0$ zamanındaki $N = 7$ ve $M = 8$ için $ e_N(x, y, 0) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, y, 0) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	156
Şekil 4.53. Örnek 4.5.2' nin $t = 0$ zamanındaki $N = 5$ ve $M = 6$ için $ e_N(x, y, 0) $ mutlak hata fonksiyonu ve $ e_{N,M}(x, y, 0) $ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması	158
Şekil 4.54. Örnek 4.5.2' nin $t = 0$ zamanındaki $N = 5$ ve $M = 6$ için $ e_N(x, y, 0) $ mutlak hata fonksiyonu ve $ E_{N,M}(x, y, 0) $ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması	159

SEMBOLLER VE KISALTMALAR DİZİNİ

KDD	Kısmi diferansiyel denklem
Ω	Kısmi diferansiyel denklemin ele alındığı bölge
$\partial\Omega$	Ω Bölgesinin sınırı
∇	Gradyan vektörü
L	Lineer operatör
∇^2	Laplace operatörü
$\mathbf{n}$	$\partial\Omega$ sınırı için birim normal vektör
$\bar{\mathbf{n}}$	$\partial\Omega$ sınırı için birim normal vektör
$\frac{\partial u}{\partial \mathbf{n}}(x)$	Normal türev
$\Delta(x, t)$	Diskriminant fonksiyonu
$J_n(x)$	Birinci tür Bessel fonksiyonları
$a_{r,s}$	Bilinmeyen Bessel katsayıları
u_N	N kesme sınırı için Bessel polinom çözümü
$u_{N,M}$	İyileştirilmiş Bessel polinom çözümü
R_N	u_N Bessel polinom çözümü için Rezidüel fonksiyon
e_N	Bessel polinom çözümü için mutlak hata fonksiyonu
$e_{N,M}$	Bessel polinom çözümü için tahmin edilen mutlak hata fonksiyonu
$E_{N,M}$	İyileştirilmiş Bessel polinom çözümü için mutlak hata fonksiyonu
(x_i, t_j)	Sıralama noktaları
$ \cdot $	Mutlak değer
$[\mathbf{W}; \mathbf{G}]$	Arttırılmış matris

1. GİRİŞ

Kısmi diferansiyel denklemler titreşim, elastikiyet, elektrostatik, akışkanlar, ısı iletimi, elektrik ve manyetizma teorileri gibi fizik alanlarında modeller genişletilirken ortaya çıkmıştır. Kısmi diferansiyel denklemler konusunda yapılan ilk çalışmalar 18. yüzyılda Euler, d'Alembert, Lagrange ve Laplace' ın fizik bilimindeki modeller için bir araç olarak kısmi diferansiyel denklemleri kullanması ile başlar. 19. yüzyılın ortalarında başlayan Riemann' ın çalışmalarıyla kısmi diferansiyel denklemler matematiğin diğer dallarında da temel bir araç oldu.

19. ve 20. yüzyıl boyunca kısmi diferansiyel denklemlerin fizik, mühendislik ve diğer uygulama alanlarında modeller ile ilişkilere sahip oluşu ve matematiğin diğer dallarının gelişiminde bir araç olarak kullanılması kısmi diferansiyel denklemler için büyük bir önem taşımıştır. Kısmi diferansiyel denklemleri fizik ve mühendislik gibi uygulama alanlarında modeller ile ilişkilendiren ve matematiğin diğer dallarının gelişiminde bir araç olarak kullanan ilk kez 1890 yılında H. Poincare olmuştur.

Diğer taraftan, birinci mertebe kısmi diferansiyel denklemler sistemlerinin teorisinin 1870' deki S. Lie' nin orijinal çalışmasındaki Lie teorisi ile önemli bir ilişkisi olmuştur ve 1890' lı yıllarda E. Cartan' ın Lie teorisi ile olan çalışmaları ile de önemli bir ilişkisi olmuştur.

20. yüzyılın başlarından beri kısmi diferansiyel denklemler ve onların sınır değer problemlerinin çözümleri reel analiz ve fonksiyonel analizin gelişmesine katkıda bulunmuştur. Fonksiyonel analizin gelişimindeki bu perspektifi J. Dieudonne ortaya koymuştur. 1950 ve 1960' lı yıllarda başlayan lineer kısmi diferansiyel denklemler ve onların sınır değer problemleri üzerine yapılan çalışmalar Fourier analizdeki tekniklerin oldukça genişlemesini sağlamıştır. Kısmi diferansiyel denklemlerin tekil integral operatörler ile ilişkisi 1930' da başlamıştır. Pseudo-diferansiyel operatörler ve Fourier integral operatörleri sayesinde kısmi diferansiyel denklemler için Fourier analiz uygulamaları lineer kısmi diferansiyel denklemlerin teorisinin oldukça genişlemesini sağlamıştır.

Kısmi diferansiyel denklemler ve topoloji arasında da önemli bir etkileşim vardır. Bu etkileşim özellikle akışkanlar mekaniğinde artış gösteren lineer olmayan kısmi diferansiyel denklemlerin global çözümleri için ilk olarak 1920 ve 1930 yılları arasında başlamıştır. 1920' li yıllarda M. Morse ve Ljusternik-Schnirelman' ın varyasyonlar teorisi ve 1930' lu yıllarda klasik Brouwer derecesinin bir genişlemesi olan sonsuz boyutlu uzaylarda Leray-Schaude derecesi bu etkileşime örnektir.

Kısmi diferansiyel denklemler matematiğin diğer dallarında da önemli bir rol oynar. Örneğin, Korteweg-De Vries kısmi diferansiyel denklemi soliton teorisi ve cebirsel geometri arasında bir ilişki sağlamıştır. İlk kez 1896' da çalışılmaya başlanan bu denklem 1960' da M. Kruskal ve onun çalışma arkadaşları tarafından su dalgaları için bir model olarak kullanıldı.

Diferansiyel geometrideki problemler Monge-Ampere denklemi ve minimal yüzey denklemleri gibi lineer olmayan kısmi diferansiyel denklemler ile verilir.

Titreşimin bir modeli olarak 1752' de d'Alembert tarafından

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}$$

bir boyutlu dalga denklemi ortaya kondu ve analiz edildi. D'Alembert' in çalışmasını 1759' da Euler iki boyutlu dalga denklemine ve 1762' de D. Bernoulli üç boyutlu dalga denklemine genişletti. 1780 yılında Laplace çekimsel potansiyel alanındaki onun çalışmasında $\Delta u = 0$ Laplace denklemini çalışmıştır.

$$\frac{\partial u}{\partial t} = \Delta u$$

ısı denklemi 1810-1822 yılları arasında Fourier tarafından verilmiştir. 2. mertebe kısmi diferansiyel denklemlerin üç önemli örnekleri olan hiperbolik, eliptik ve parabolik denklemler ile ilgili çalışmalar ilk kez 19. yüzyılın ilk on yılında yapılmasına rağmen 20. yüzyıla kadar açık bir şekilde formüle edilememiştir.

1750 ve 1900 yılları arasında fiziksel olaylar sayesinde ortaya çıkan bazı denklemler şunlardır:

- Sıkıştırılmaz akışkanların Euler denklemi (1755).
- 1760' da Lagrange tarafından verilen minimal yüzey denklemi.

- 1755’ de Monge tarafından Monge-Ampere denklemi.
- Elektrik ve manyetik problemlere uygulanan Laplace denklemi (1780) ve Poisson denklemi (1813).
- 1822-1827 yılları arasında Navier tarafından akışkanlar için Navier Stokes denklemleri.
- 1864’ de elektromanyetik teoride Maxwell denklemi.
- 1860’ da ses bilimi ile ilişkili Laplace operatörü için öz değer problemi ve Helmholtz denklemi.
- 1840’ da sabun köpükleri için bir model olarak Plateau problemi.
- 1896’ da sadece su dalgaları için bir model olarak Korteweg-De Vries denklemi.

19. yüzyıl boyunca uygun sınır koşullarıyla kısmi diferansiyel denklemlerin çözümlerini bulmak için kullanılan bazı önemli metotlar aşağıdaki gibidir.

a) Değişkenlerine ayırma metodu:

Dalga denkleminin çözümleri için ilk kez 1747 yılında d’Alembert bu metodu çalışmaya başlamıştır. Ayrıca, dalga denklemi için değişkenlerine ayırma metodu üzerine 1748 yılında Euler’ de çalışmaya başlamıştır. 1782 yılında Laplace ve Legendre tarafından Laplace denklemi için değişkenlerine ayırma metodu kullanılmıştır. Isı denklemi için ise bu metot 1811-1824 yılları arasında Fourier tarafından çalışılmıştır.

b) Riemann’ ın 1851’ de yaptığı çalışmaya dayanan tek bir kompleks değişkenli analitik fonksiyonlar ve iki boyutlu reel harmonik fonksiyonlar çalışması arasındaki etkileşim 1870 yılı civarında C. Neumann, H. A. Schwarz ve E. B. Christoffel tarafından geliştirildi.

c) Laplace denklemi için 1835’ de Green’s fonksiyonlar metodu çalışılmaya başlandı. Bu çalışma Laplace denkleminin özel tekil çözümleri çalışmasını da içerir.

d) $\Omega \subset \mathbb{R}^3$ ’ nın $\partial\Omega$ sınırında verilen bir $u = \varphi$ sınır değeri için

$$\Delta u = 0$$

Laplace denkleminin bir çözümünün

$$\int_{\Omega} \sum_{i=1}^3 \left(\frac{\partial v}{\partial x_i} \right)^2$$

integralini $\partial\Omega$ sınırında $v = \varphi$ olacak şekilde tüm v fonksiyonları arasında minimize ettiği 1833 yılında G. Green tarafından gözlemlendi.

e) Kısmi diferansiyel denklemlerin özel çözümlerini elde etmek için Euler, d'Alembert, Laplace ve diğerleri tarafından kuvvet seri metotları kullanıldı.

1890 yılında Poincare her sürekli Dirichlet sınır koşulu için Laplace denkleminin bir çözümünün varlığının ve tekliğinin ispatını verdi.

Picard ve onun okulundakiler 1880 yılının başlarında lineer olmayan problemlerin çözümleri için ardışık yaklaşım metodunu uyguladı.

Picard Fuchsian fonksiyonlar teorisinde artış gösteren $\Delta u = e^u$ lineer olmayan denkleminin bir çözümünün varlığını 1898 yılında ispatladı. Bu problemin yardımıyla Picard süreklilik metodunun prensiplerini formüleştirdi.

1906 yılının başlarında S. Bernstein lineer olmayan eliptik denklemler için Dirichlet probleminin süreklilik metodu ile çözülebilirliğini gözlemledi.

Korteweg-De Vries (KDV) denkleminin soliton (geçici) çözümleri ilk kez 1834 yılında J. S. Russell tarafından fark edildi. 1960 yılında M. Kruskal ve çalışma arkadaşları KDV denkleminin soliton çözümlerinin bir teorisini çalışmıştır.

Kısmi diferansiyel denklemlerin çözümlerinin hesaplanması oldukça zordur. Poincare döneminde mevcut pratik teknikler olmasına rağmen kısmi diferansiyel denklemlerin çözümlerinin hesaplanmasının zor olduğundan 1890' da Poincare bahsetmiştir. Günümüzde ise, yüksek hızlı bilgisayarlar sayesinde hesaplamalar kısa zamanda yapılabilir.

Yukarıdaki bilgiler Brezis ve Browder (1998) tarafından sunulmuştur.

Bugüne kadar, kısmi diferansiyel denklemlerin yaklaşık çözümleri için oldukça çok çalışmalar yapılmış ve yöntemler verilmiştir. Telegraph denkleminin nümerik çözümleri için Adomian ayrıştırma metodu (Biazar ve Ebrahimi, 2007) ve varyasyonel iterasyon metodu (Biazar vd., 2009a) kullanıldı. Hiperbolik telegraph denklemini Chebyshev Tau metodu (Saadatmandi ve Dehghan, 2010), çekirdek yöntemi (Yao, 2011), Legendre multiwavelet Galerkin metodu (Yousefi, 2010) ve

radyal temel fonksiyonları (Dehghan ve Shokri, 2009a) aracılığıyla çözülmüştür. Isı dağılımı problemini çözmek için sonlu farklar metodu (Azari ve Zhang, 2010) ve üstel fonksiyon yöntemi (Wen vd., 2009) çalışıldı. Dehghan ve Shokri (2009b) ve Bratsos (2009) Klein-Gordon denkleminin nümerik çözümü üzerine çalışmıştır. Zakharov-Kuznetsov denklemi homotopi pertürbasyon metodu (Biazar vd., 2009b) ve yardımcı denklem metodu (Ma vd., 2009) ile çözüldü. Kawahara tipi denklemleri çözmek için radyal temel fonksiyonları (Dereli ve Dağ, 2012) kullanıldı. MRLW denklemi de radyal temel fonksiyonları (Dereli, 2012) kullanılarak çözülmüştür. Korteweg-de Vries Burgers denklemini çözmek için He' nin homotopi pertürbasyon metodu (İnc, 2010) kullanıldı. Fokker-Planck denklemi Cubic B-Spline fonksiyonları (Lakestani ve Dehghan, 2009) aracılığıyla çözülmüştür. KdV-Burgers-Kuramoto denklemlerini çözmek için He' nin varyasyonel iterasyon metodu ve Adomian ayrıştırma yöntemi (Safari vd., 2009) kullanıldı. KDV-Burger denklemleri homotopi pertürbasyon metodu (Yildirim ve Berberler, 2010) ile çözüldü. RLW denkleminin çözümleri için kosinüs genişleme tabanlı diferansiyel kareleme algoritması (Dağ vd., 2010) kullanıldı. Darvishi ve Khani (2009) Foam drainage denkleminin bir seri çözümü üzerine çalıştı. Dalga denklemi varyasyonel iterasyon metodu (Hemeda, 2008) ve Bernstein Ritz-Galerkin metodu (Yousefi vd., 2010) kullanılarak çözüldü. Kelleci ve Yıldırım (2011) homotopi pertürbasyon ve Pade tekniklerinin birleşimi ile Burger denklemini çözdü. Chebyshev spectral kollokasyon metoduyla Kuramoto-Sivashinsky denklemi Khater ve Temsah (2008) tarafından çözüldü. Murphy ve Evans (1981) iki boyutlu ısı denkleminin Chebyshev seri çözümleri üzerine çalıştı. Sezer (1989) Dirichlet problemi için bir Chebyshev polinom yaklaşımı sundu. İkinci ve dördüncü mertebe eliptik denklemler Chebyshev açılım metodu (Khalifa vd., 2003) ile çözüldü. Keşan (2003) ikinci mertebe lineer kısmi diferansiyel denklemleri Chebyshev polinom metodu ile çözdü. Parabolik ve hiperbolik kısmi diferansiyel denklemlerin çözümleri için wavelet Galerkin metodu (Kumar ve Mehra, 2006), sonlu farklar metodu (Benito vd., 2007) ve homotopi pertürbasyon metodu (Biazar ve Ghazvini, 2008) sunuldu. İkinci mertebe kısmi diferansiyel denklemler Kurulay ve Bayram (2009) tarafından kuvvet seri metodu ile çözüldü. Doha ve Al-Kholi (2001) parabolik ve eliptik kısmi diferansiyel denklemleri Legendre spectral metodunu kullanarak çözdü. Kısmi diferansiyel denklemleri çözmek için iki boyutlu diferansiyel dönüşüm metodu (Bildik ve Konuralp, 2006), Adomian ayrıştırma yöntemi (Bildik ve Konuralp, 2006), varyasyonel iterasyon yöntemi (Bildik ve

Konuralp, 2006), n-boyutlu diferansiyel dönüşüm metodu (Kurnaz vd., 2005), kollokasyon metodu (Liu ve Tai, 2006) ve Legendre wavelet metodu (Liu ve Lin 2010) kullanıldı. Tepki-difüzyon problemleri Revelli ve Ridolfi (2008) tarafından genelleştirilmiş kollokasyon metodu kullanılarak çözüldü. Cauchy tepki-difüzyon problemini çözmek için He' nin homotopi pertürbasyon metodu (Yıldırım, 2009) kullanıldı. Nanoteknolojide artış gösteren problemleri çözmek için homotopi pertürbasyon metodu (Zhu vd., 2009) çalışıldı. Değişken katsayılı adveksiyon-difüzyon modelleri için uygun bir nümerik metot (Ponsoda vd., 2008) kullanıldı. Gecikmeli difüzyon modellerinin analitik ve nümerik çözümleri üzerine çalışıldı (Reyes vd., 2008). Bülbül ve Sezer (2011) hiperbolik kısmi diferansiyel denklemlerin çözümleri için Taylor matris yöntemini uyguladı. Norhashidah ve Evans (2004) eliptik kısmi diferansiyel denklemleri çözmek için bazı iteratif metotlar çalıştı. Lineer olmayan kısmi diferansiyel denklemler Parand ve Rad (2011) tarafından üstel-fonksiyon yöntemi kullanılarak çözüldü. Akyüz-Daşcıoğlu (2009) yüksek mertebe kısmi diferansiyel denklemler için Chebyshev polinom yaklaşımını çalıştı. Lineer ve lineer olmayan kısmi diferansiyel denklemler için varyasyonel iterasyon metodu ve homotopi pertürbasyon metodu arasında karşılaştırma yapıldı (Ateş ve Yıldırım 2010). Biazar ve Aminikhah (2009) kısmi diferansiyel denklem sistemleri için homotopi pertürbasyon yönteminin yakınsaklığı üzerine çalıştı. Gecikmeli kısmi diferansiyel denklem sistemleri için Bashier ve Patidar (2011) tarafından bir nümerik metot verildi. Ayrıca, kısmi diferansiyel denklemlerin nümerik çözümleri değişik metotlar ile bulundu (Pinchover ve Rubinstein, 2005; Morton ve Mayers, 2005; Bernatz, 2010; Stavroulakis ve Tersian, 2004). Akışkanlar dinamiğinde karşılaşılan kısmi diferansiyel denklemler için hesaplama yöntemleri çalışılmıştır (Pozrikidis, 2009; Herron ve Foster, 2008; Chung, 2010; Ferziger ve Peric, 2002). Hava kirliliği ve türbülans için kullanılan kısmi diferansiyel denklem modelleri için (Moreira ve Vilhena, 2010) ve bilim ve mühendislikte karşılaşılan lineer olmayan kısmi diferansiyel denklemler için (Debnath, 2005) çeşitli hesaplama metotları kullanıldı.

İntegro-diferansiyel denklemlerin çözümleri için Taylor sıralama yöntemi (Karamete ve Sezer, 2002; Nas vd., 2000; Kurt ve Sezer, 2008) ve Taylor matris yöntemi (Sezer ve Gülsu, 2005), integro-diferansiyel denklem sistemlerinin çözümleri için Taylor sıralama yöntemi (Gülsu ve Sezer, 2006) ve Chebyshev sıralama yöntemi (Akyüz-Daşcıoğlu ve Sezer, 2005), hiperbolik kısmi diferansiyel denklemleri

özmek için Taylor matris metodu (Bölböl ve Sezer, 2011), lineer kısmi diferansiyel denklemlerin nümerik çözümleri için Chebyshev matris yöntemi (Yüksel, 2011) ve lineer fonksiyonel denklemlerin yaklaşık çözümleri için Bernstein polinomları ve rasyonel fonksiyonlara dayalı sıralama yöntemi (Işık, 2012) kullanıldı. İlâveten, Fredholm integro-diferansiyel-fark denklemleri (Şahin vd., 2011a), diferansiyel ve integro-diferansiyel denklemleri (Yüzbaşı, 2009), Volterra integral denklemlerin sistemleri (Şahin vd., 2011b) ve pantograph denklemleri (Yüzbaşı vd., 2012) çözmek için Bessel sıralama (collocation) metodu kullanıldı.

2. KAYNAK ÖZETLERİ

Bu bölümde, kısmi diferansiyel denklemlerin sınıflandırılması yapılacak, çalışmalarda çokça karşılaşılan denklemler tanıtılacak, kısmi diferansiyel denklem problemlerinde karşılaşılan bazı koşul tipleri verilecek ve karmaşıklık (complexity) hakkında bilgiler sunulacaktır.

2.1. Kısmi Diferansiyel Denklemler

Tanım 2.1: Bir veya daha çok bağımlı değişkenin birden çok bağımsız değişkene göre kısmi türevleri ile beraber bağımlı ve bağımsız değişkenleri içeren denkleme kısmi diferansiyel denklem denir.

u bağımlı, x ve y bağımsız değişkenler olmak üzere bir kısmi diferansiyel denklem genel olarak

$$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, \dots) = 0$$

biçimindedir.

2.2. İkinci Mertebe Kısmi diferansiyel Denklemlerin Sınıflandırılması

Adi diferansiyel denklemlerde sınıflandırma, diferansiyel denklemin mertebesine ve derecesine bağlı olarak lineer ya da lineer olmamasına göre yapılır. Kısmi diferansiyel denklemlerde ise sınıflandırma mertebe ve lineerliğin yanı sıra kısmi diferansiyel denklemin yapısına göre yapılır. Çünkü denklemin yapısı değiştikçe çözümlerin özellikleri değişir.

u bağımlı değişken, x ve t bağımsız değişkenler olmak üzere, ikinci mertebe hemen hemen lineer kısmi diferansiyel denklem genel olarak

$$Lz = A(x,t)\frac{\partial^2 u}{\partial x^2} + B(x,t)\frac{\partial^2 u}{\partial t\partial x} + C(x,t)\frac{\partial^2 u}{\partial t^2} + F(x,t,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial t}) = 0 \quad (2.1)$$

şeklindedir. Burada, A , B ve C ; xt -düzleminin bir Ω bölgesinde x ve t 'nin iki defa sürekli türetilbilir fonksiyonlarıdır ve aynı anda üçü birden sıfır değildir. $A(x,t)D_x^2 + B(x,t)D_x D_t + C(x,t)D_t^2$, ye L operatörünün esas kısmı denir. Denklemin çözümlerinin özelliklerini bu kısım tayin eder.

$$\Delta(x,t) = B^2(x,t) - 4A(x,t)C(x,t)$$

fonksiyonuna L operatörünün diskriminantı denir (Çağlıyan ve Çelebi, 2010).

Tanım 2.2: Eğer Ω 'nin bir (x_0, t_0) noktasında,

$\Delta(x_0, t_0) > 0$ ise (2.1) denkleminde bu noktada hiperboliktir,

$\Delta(x_0, t_0) = 0$ ise (2.1) denkleminde bu noktada paraboliktir,

$\Delta(x_0, t_0) < 0$ ise (2.1) denkleminde bu noktada eliptiktir,

denir. Bir Ω bölgesinin tüm noktalarında hiperbolik, parabolik veya eliptik olan bir denkleme Ω 'da sırasıyla hiperbolik, parabolik veya eliptiktir denir.

Genellikle bir denklem, katsayılarının tanımlı olduğu bir bölgede her üç tipten de olabilir. Özel olarak, A , B ve C katsayıları (yani ikinci mertebe türevlerin katsayıları) sabit olan bir denklem tüm düzlemde aynı tiptendir (Çağlıyan ve Çelebi, 2010).

Hiperbolik, parabolik ve eliptik tip denklemler, birçok fiziksel olayın modellenmesinde ortaya çıkan denklemlerdir. En çok karşılaşılan ve literatürde önemli yeri olan bu denklemlerin bazı örnekleri aşağıda verilmektedir.

2.2.1. Hiperbolik tip denklemler

Hiperbolik tip kısmi diferansiyel denklemler ile genellikle mekanik ve fizik alanlarındaki uygulamalarda karşılaşılır. Örneğin, makinelerin titreşimi problemleri ve elektromanyetik, hidrodinamik ve ses yayılması gibi modeller hiperbolik tip

denklemlerdir. Dalga denklemi, telgraf denklemi ve lineer Kleain-Gordon denklemleri fizik ve mühendislikte oldukça çok karşılaşılan hiperbolik tip kısmi diferansiyel denklemlerdir.

Dalga denklemi:

c , pozitif bir sabit ve F , dalgaya etki eden bilinen bir dış kuvvet olmak üzere,

$$u_{tt} - c^2 u_{xx} = 0$$

denklemine bir boyutlu homojen dalga denklemi,

$$u_{tt} - c^2 u_{xx} = F(x, t)$$

denklemine ise bir boyutlu homojen olmayan dalga denklemi denir. Bu denklemlerde x değişkeni yer t değişkeni zamanı göstermektedir. Dalga denklemi fizikte çok önemli yere sahip bir kısmi diferansiyel denklemdir. Bu denklemin çözümlerinden, ses, ışık ve su dalgalarının hareketlerini ortaya koyan fiziksel nicelikler çıkar. Dalga denklemi, en çok titreşen yay (vibrating-string) hareketinin modelinde kullanılır (Weinberger, 1965). Ayrıca elektrodinamik, elektromanyetik, akışkanlar mekaniği ve akustik gibi alanlardaki uygulamalarda da dalga denklemi ile karşılaşılır (Polyanin, 2002). Bir boyutlu dalga denkleminin çözümü bize l uzunluğunda titreşim halinde olan bir telden t saniye sonra x kadar uzaklıktaki enine yer değişimini verir. Bu tip denklemler başlangıç ve sınır koşulları ile ele alınır.

Son yıllarda,

$$u_{tt} - u_{xx} = q(x, t), \quad 0 \leq x \leq l, \quad 0 \leq t \leq T,$$

dalga denklemi

$$u(x, 0) = f_1(x), \quad 0 \leq x \leq l,$$

$$u(0, t) = g_1(t), \quad 0 \leq t \leq T,$$

Dirichlet sınır koşulları,

$$u_t(x, 0) = f_2(x), \quad 0 \leq x \leq l,$$

Neumann sınır koşulu ve

$$\int_0^l u(x, t) dx = g_2(t), \quad 0 \leq t \leq T,$$

lineer olmayan (nonlocal) integral koşulu ile alınarak değişik metotlar ile bu problemin nümerik çözümleri üzerine çalışmalar yapılmıştır (Dehghan, 2005; Yousefi vd., 2010). Burada, $(x, t) \in [0, l] \times [0, T]$ için q , f_1 , f_2 , g_1 ve g_2 tanımlı fonksiyonlardır. Ayrıca bu problemin çözümünün varlık ve teklik teoremi Bouziani (1999) tarafından verilmiştir.

Telgraf Denklemi:

α , β ve γ sırasıyla rezistans, kapasitans ve indüktans sabitleri olmak üzere, telgraf denklemi

$$\alpha u_{tt} + \beta u_t + \gamma u = u_{xx} + g(x, t),$$

şeklindedir.

Telgraf denklemi sinyal analizlerinde, özellikle elektrik sinyallerinin yayılımında ve elektrik dolaşımının olduğu bir sistemde, x konumu ve t anındaki voltaj hesaplamalarında kullanılmaktadır (Bülbül, 2011).

Klein-Gordon Denklemi:

Klein-Gordon denklemi lineer ve lineer olmayan olarak iki türe ayrılır ve bunlar farklı fiziksel olaylara karşılık gelir. Klein-Gordon denklemleri kuantum cisim, rölâivist ve plazma fizik teorilerinde kullanılır (Polyanin, 2002).

a ve $b > 0$ fiziksel sabitler olmak üzere,

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} - bu$$

olarak verilen denkleme lineer Klein-Gordon denklemi denir.

Lineer olmayan Klein-Gordon denklemine özellikle yüksek enerji fizikçilerinin çalışma alanı olan cisim teorisinde karşılaşılır (Greiner, 2000).

Bir boyutlu lineer olmayan Klein-Gordon denklemi

$$\frac{\partial^2 u}{\partial t^2} + \alpha \frac{\partial^2 u}{\partial x^2} + h(u) = g(x, t)$$

kısmi diferansiyel denklemi ile verilir (Dehghan, 2009).

2.2.2. Parabolik tip denklemler

Parabolik tip kısmi diferansiyel denklemlere zamana bağlı ısı ve madde yayılması problemlerinde karşılaşılır (Weinberger, 1965).

Isı denklemi:

Isının bir nesne üzerinde, belli bir konumda ve zamanda, nasıl dağılacığını tanımlayan kısmi diferansiyel denkleme ısı denklemi denir.

$k > 0$ ısı yayılım katsayısı ve $u = u(x, t)$, x noktasında t anındaki sıcaklık olmak üzere

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}$$

denklemi zamana bağlı ısı yayılımını gösteren bir boyutlu homojen ısı denklemi olarak ve

$$\frac{\partial u}{\partial t} - k \frac{\partial^2 u}{\partial x^2} = g(x, t)$$

denklemi bir boyutlu homojen olmayan ısı denklemi olarak adlandırılır.

Bu denklemin çözümü, termal olarak izole edilmiş bir çubuğun bir ucundan x kadar uzaklıkta ve t anındaki sıcaklığını verir. Bu tip denklemler başlangıç ve sınır koşulları ile ele alınır.

Dehghan (2007) yukarıdaki homojen olmayan ısı denklemini

$$u(x,0) = f(x), \quad 0 \leq x \leq 1,$$

başlangıç koşulu

$$u(1,t) = g_1(t), \quad 0 \leq t \leq T,$$

sınır koşulu ve

$$\int_0^b u(x,t)dx = g_2(t), \quad 0 \leq t \leq T, \quad 0 < b \leq 1$$

lineer olmayan (nonlocal) integral koşulu ile ele alarak sonlu farklar metodu ile nümerik olarak çözdü. Burada, b bir reel sabit ve $(x,t) \in [0,1] \times [0,T]$ için g , f , g_1 ve g_2 tanımlı fonksiyonlardır. Ayrıca, Bülbül (2011) bu integral koşullu ısı problemi için Taylor matris yöntemini sundu.

2.2.3. Eliptik tip denklemler

Eliptik tip kısmi diferansiyel denklemler çoğunlukla denge ya da kararlı hal problemlerinin modellenmesinde kullanılırlar. Uygulamalarda en çok Laplace, Poission ve Helmholtz gibi eliptik tip kısmi diferansiyel denklemler ile karşılaşılır.

Laplace denklemi:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial t^2} = 0$$

denklemi Laplace denklemi olarak adlandırılır. Elektromanyetizma, astronomi ve akışkanlar dinamiği gibi birçok bilim alanında Laplace denkleminin çözümleri önemli yer tutar. Laplace denkleminin çözümleri elektrik ve yer çekim potansiyeli ile akışkan potansiyelinin davranışını açıklar. Laplace denkleminin çözümlerinin genel teorisi aynı zamanda potansiyel teorisi olarak da bilinmektedir

Potansiyel teorisinin temel denklemlerinden biri olan Laplace denkleminin fiziksel uygulaması oldukça çoktur. Örneğin, yüzeyleri izole edilmiş bir ortamda zamandan bağımsız (ki buna kararlı ısı denir) herhangi bir (x,y) noktasındaki ısı miktarını veren $u(x,y)$ fonksiyonu Laplace denklemini sağlar. Ayrıca ısı kaynağı olmayan bir bölgede kararlı sıcaklık dağılımı, iletkenlerle çevrili yüksüz bir bölgede elektrostatik potansiyel, bir akışkanda hız dağılımı vs. ile ilgili problemlerde de Laplace denklemi ile karşılaşılır (Çilingir, 2005).

Poisson denklemi:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial t^2} = f(x,y)$$

denklemi Poisson denklemi olarak adlandırılır. Isının kararlı yayılımı problemlerinde Poisson denklemi ortaya çıkar. $f(x,y) \neq 0$ olması, problemde bir ısı kaynağının olduğunu gösterir. Dış kuvvetlerin etkisi altındaki bir telin zamana bağlı olarak denge konumuna gelmesi de poisson denklemi ile verilir. Poisson denklemi ile gösterilen fiziksel iki olguyu örnek olarak verecek olursak; birincisi, ρ yük yoğunluğunun varlığı negatif ϕ elektriksel potansiyelin oluşmasına sebep olur. Bu fiziksel olgu, ε sabit olmak üzere $\nabla^2 \phi = -\rho/\varepsilon$ ile gösterilir. İkincisi ise, akışkan içinde θ ısı kaynağından bir yayılımın varlığı, T sıcaklığının oluşmasına sebep olur. Böylece k sabit olmak üzere, $\nabla^2 T = k\theta$ eşitliği elde edilir (Çilingir, 2005).

Helmholtz denklemi:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial t^2} + \lambda u = -\theta(x)$$

denklemi Helmholtz denklemi olarak adlandırılır. Helmholtz denklemi zamanla harmonik değişim gösteren elektromanyetik veya akustik dalgalarla uyarılmış ortamlardaki alan dağılımını modellemek için kullanılır. Bu denklemin $\lambda < 0$ hali, madde transfer problemlerini ifade eder (Bülbül, 2011).

2.2.4. Karışık tip denklemler

İkinci mertebe hemen hemen lineer kısmi diferansiyel denkleminin diskriminantının durumuna göre hiperbolik, parabolik ve eliptik tip olabileceğini kesim (2.2)' de söylemiştik. Değişken katsayılı hemen hemen lineer kısmi diferansiyel denklemlerde diskriminantın işareti noktaya göre değişebilir. Bu nedenle, aynı denklem farklı bölgelerde farklı türden olabilir. Bu tip denklemlere karışık tip kısmi diferansiyel denklemler adı verilir. Sabit katsayılı hemen hemen lineer kısmi diferansiyel denklemlerin sadece bir türden olabileceğini not ediyoruz. En çok bilinen karışık tip denklem

$$\frac{\partial^2 u}{\partial x^2} + x \frac{\partial^2 u}{\partial t^2} = 0, (x, t) \in \mathbb{R}^2$$

Tricomi denklemidir (Hasanoğlu, 2010). Tricomi denklemi, $x < 0$ olduğunda hiperbolik, $x = 0$ olduğunda parabolik ve $x > 0$ durumunda eliptik tip denklem olur.

2.3. Cauchy Problemi

İkinci mertebeden kısmi diferansiyel denklemler için Cauchy problemini vereceğiz.

Tanım 2.3: x ve t bağımsız değişkenlerine bağımlı $u = u(x, t)$ fonksiyonu için ikinci mertebeden en genel bir kısmi diferansiyel denklem göz önüne alınsın ve denklemin

$$k = F(x, t, u, p, q, r, s) \quad (2.2)$$

şeklinde $u_{tt} = k$ türevine göre çözülebildiği varsayılınsın. Bir $t = t_0$ değeri için bilinmeyen fonksiyonu ve onun t' ye göre türevi,

$$\begin{cases} u(x, t_0) = f(x) \\ u_t(x, t_0) = g(x) \end{cases} \quad (2.3)$$

şeklinde tanımlansın. (2.3) koşulları altında (2.2) denkleminin çözümlerinin bulunması problemine (2.2) denklemi için Cauchy problemi denir. Burada, $p = u_x$, $q = u_t$, $r = u_{xx}$, $s = u_{xt}$, $k = u_{tt}$ şeklinde tanımlanmıştır.

Özellikle fizik ile ilgili uygulama problemlerinde t bağımsız değişkeni zamanı gösterir ve (2.3) koşulları $t_0 = 0$ noktasında tanımlanırsa o zaman bu probleme başlangıç değer problemi denir. Yani, başlangıç değer problemi Cauchy probleminin özel bir durumudur (Çağlıyan ve Çelebi, 2010).

Daha genel olarak,

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} - \frac{\partial^2 u}{\partial t^2} = 0 \quad (2.4)$$

denklemi ile bağlantılı olan temel problem, başlangıç değer problemi veya Cauchy problemi olarak adlandırılır. Bu problem; (2.4) denklemini ve verilen

$$\begin{cases} u(x_1, x_2, \dots, x_n, 0) = \phi(x_1, x_2, \dots, x_n) \\ \frac{\partial u}{\partial t}(x_1, x_2, \dots, x_n, 0) = \psi(x_1, x_2, \dots, x_n) \end{cases}$$

başlangıç koşullarını sağlayan $u(x_1, x_2, \dots, x_n, t)$ çözümünün aranması problemidir. ϕ ve ψ fonksiyonları, $t = 0$ da x uzayında tanımlanmış bilinen fonksiyonlardır. ϕ ve ψ ' ye başlangıç verileri denir. $\mathbb{R}^{n+1}$, (x, t) -uzayında $t = 0$ düzlemi, problemin başlangıç yüzeyi ya da başlangıç manifoldu olarak adlandırılır (Anar, 2005).

2.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Problemi

$$\frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + a(x) \frac{\partial u}{\partial x} + b(x)u = g(x, t), \quad 0 \leq x \leq L, \quad 0 \leq t \leq T$$

kısmi diferansiyel denklemi

$$u(x, 0) = f(x), \quad 0 \leq x \leq L < \infty,$$

başlangıç koşulu ve

$$u(0,t) = h(t), \quad 0 \leq t \leq T < \infty,$$

$$u(L,t) = s(t), \quad 0 \leq t \leq T < \infty$$

sınır koşulları ile birlikte singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon problemi olarak adlandırılır. Burada, $\varepsilon: 0 < \varepsilon \ll 1$ olacak şekilde bir difüzyon katsayısı ya da singüler pertürbasyon parametresi, $a(x):$ hız, $g(x,t) - b(x)u$: reaksiyon terimi, L ve T : uygun sabitler ve $(x,t) \in [0,L] \times [0,T]$ için $g(x,t)$, f , h ve s tanımlı fonksiyonlardır (Lenferink, 2002; Kadalbajoo ve Gupta, 2009).

2.5. Koşullar

Bu kısımda, kısmi diferansiyel denklem problemlerinde çokça karşılaşılan başlangıç ve sınır koşullarını vereceğiz.

2.5.1. Başlangıç koşulları

Herhangi bir kısmi diferansiyel denklem problemindeki bağımlı değişken u olsun. Bağımsız değişkenlerin tanımlı olduğu aralıkların başlangıç noktalarında u ve onun kısmi türevlerinin birleşimini içeren koşullara başlangıç koşulları adı verilir.

Bir fiziksel olgunun incelenmesinin başladığı anda tüm Ω bölgesinde sağlanması gereken koşullar zamana göre başlangıç koşullarıdır. Genellikle, uygulama problemleri zamana göre başlangıç koşulları ile verilir.

2.5.2. Sınır koşulları

Bunlar Kısmi diferansiyel denklemlerin ele alındığı Ω bölgesinin $\partial\Omega$ sınırının üzerindeki noktalarda sağlanması gereken koşullardır. Kısmi diferansiyel denklemler çeşitli tipte sınır şartlarıyla birlikte verilir. Sınır koşulu u cinsinden verilmişse

“Dirichlet tipi sınır koşulu” olarak, u ’ nun gradyanı cinsinden verilmişse “Neumann tipi sınır koşulu” olarak adlandırılır. u ve gradyanı birlikte verildiği takdirde “Robin sınır koşulu (karışık sınır şartı)” olarak isimlendirilir. $\partial\Omega$ sınırı için birim normal vektör ‘ $\mathbf{n}$ ’ olmak üzere, bazı sınır koşulları aşağıda verilmiştir.

Dirichlet sınır koşulu

Adi diferansiyel denklemler için $[a, b]$ aralığında Dirichlet sınır koşulları

$$y(a) = \alpha \text{ ve } y(b) = \beta$$

şeklindedir. Burada α ve β sabitlerdir ve y : adi diferansiyel denklemdeki bilinmeyen fonksiyondur.

Kısmi diferansiyel denklemler için bir $\Omega \subset \mathbb{R}^n$ bölgesinde, Dirichlet sınır koşulları

$$u(x) = f(x), \quad \forall x \in \partial\Omega$$

ile verilir. Burada, $f: \partial\Omega$ sınırında tanımlı bir fonksiyondur.

Neumann sınır koşulu

Adi diferansiyel denklemler için $[a, b]$ aralığında Neumann sınır koşulları

$$y'(a) = \alpha \text{ ve } y'(b) = \beta$$

şeklindedir. Burada α ve β sabitlerdir ve y : adi diferansiyel denklemdeki bilinmeyen fonksiyondur.

Kısmi diferansiyel denklemler için bir $\Omega \subset \mathbb{R}^n$ bölgesinde Neumann sınır koşulları

$$\frac{\partial u}{\partial \mathbf{n}}(x) = f(x), \quad \forall x \in \partial\Omega$$

şeklindedir. Burada, $\mathbf{n}: \partial\Omega$ sınırı için birim normal vektör ve f skaler bir fonksiyondur. Ayrıca normal türev aşağıdaki gibi tanımlıdır:

$$\frac{\partial u}{\partial \mathbf{n}}(x) = \nabla u(x) \cdot \mathbf{n}$$

burada, ∇ : gradyan vektörü gösterir ve ‘ $\cdot$ ’ ile iç çarpım gösterilir.

Robin sınır koşulu

α ve β sıfır olmayan sabitler, $g: \partial\Omega$ sınırında tanımlı bir fonksiyon, $u: \Omega$ bölgesinde tanımlı bilinmeyen çözüm fonksiyonu ve $\frac{\partial u}{\partial \mathbf{n}}$: sınırdaki normal türev olmak üzere $\partial\Omega$ sınırında Robin sınır koşulu

$$\alpha u + \beta \frac{\partial u}{\partial \mathbf{n}} = g$$

biçimindedir.

Nonlocal (yerel olmayan) sınır koşulu

$u: \Omega = [a, b] \times [\alpha, \beta]$ bölgesinde tanımlı bilinmeyen çözüm fonksiyonu ve $g(t): \Omega$ bölgesinde tanımlı bir fonksiyon olmak üzere nonlocal (yerel olmayan) sınır koşulu

$$\int_a^b u(x, t) dx = g(t), \quad a \leq x \leq b, \quad \alpha \leq t \leq \beta$$

olarak verilir.

2.6. Kayan Nokta Aritmetiği (Floating-Point Operation)

Kayan noktalı sayılar gerçel sayıların bilgisayar ortamındaki bir gösterim biçimidir. Yani, gerçel sayılar bilgisayar ortamında bir kayan nokta formatında hafızada tutulur (Watkins, 2002). Gerçek dünyada sayılar sonsuza kadar giderken, bilgisayar ortamında bilgisayar donanımının getirdiği sınırlamalardan dolayı bütün sayıların gösterilmesi mümkün değildir. Bununla birlikte gerçekte sonsuza kadar giden

birtakım değerler bilgisayar ortamında ortamın kapasitesine bağlı olarak yaklaşık değerlerle temsil edilirler. Bu sınırlamaların etkisini en aza indiren, sayıların maksimum miktarda ve gerçeğe en yakın şekilde temsilini sağlayan sisteme kayan-noktalı sayılar sistemi denir. Bir aritmetik işlemdeki, toplama, çıkarma, çarpma ve bölme işlemlerinin her biri bir kayan nokta olarak adlandırılır. Aşağıda vektör-vektör işlemleri, matris-vektör çarpımı, matris-matris çarpımı, bir matrisin tersi ve Gauss eliminasyon için ne kadar kayan nokta olduğundan bahsedeceğiz.

2.6.1. Vektör-vektör işlemleri

$x = [x_1 \ x_2 \ \cdots \ x_n]^T$, $y = [y_1 \ y_2 \ \cdots \ y_n]^T \in \mathbb{R}^n$ olan iki vektör olsun. O zaman,

$$x^T y = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

dir. Bu iki vektörün iç çarpımında; n tane çarpma ve $n-1$ tane toplama işlemi vardır. Yani, toplam işlem sayısı $2n-1$ ' dir. Böylece, iki vektörün iç çarpımında $2n-1$ kayan nokta vardır.

Eğer, $n \gg 1$ ise o zaman, iki vektörün iç çarpımında $2n$ kayan nokta vardır.

x ve y vektörlerinin toplanmasında n kayan nokta olur.

x ve y vektörlerinin çıkarılmasında da n kayan nokta vardır.

x vektörünün bir skaler ile çarpımı içinde n tane kayan nokta olur (Boyd ve Vandenberghe, 2004).

Örneğin;

$x, y \in \mathbb{R}^{10}$ olsun. O zaman bu iki vektörün iç çarpımı için

$$2n-1 = 2 \cdot 10 - 1 = 19$$

kayan nokta vardır. İlaveten, bu iki vektörün toplanması, çıkarılması ve bu vektörlerden birinin skaler ile çarpılması işlemlerinin her biri için

$$n = 10$$

tane kayan nokta olur.

2.6.2. Matris-vektör çarpımı

$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$ olan vektör ve $A \in \mathbb{R}^{m \times n}$ olan bir matris olsun. O zaman, A matrisi ve

x vektörünün çarpımında ($y = Ax$ ' de);

y ' nin her bir elemanı hesaplanırken $\mathbb{R}^n$ ' de iç çarpım yapılır ve $\mathbb{R}^n$ ' de bir iç çarpımı için $2n-1$ kayan nokta olduğunu yukarıda söylemiştik. Böylece y ' nin her bir elemanının hesaplanmasında $2n-1$ tane kayan nokta vardır. $y \in \mathbb{R}^m$ yani y ' nin m elemanı olduğundan A matrisi ve x vektörünün çarpımı için $(2n-1)m$ tane kayan nokta olur. Çok büyük n değerleri için A matrisi ve x vektörünün çarpımında yaklaşık olarak $2nm$ kayan nokta vardır (Boyd ve Vandenberghe, 2004).
Örneğin; $A \in \mathbb{R}^{15 \times 9}$ ve $x \in \mathbb{R}^9$ ise o zaman A matrisi ve x vektörünün çarpımı için

$$(2n-1)m = (2 \cdot 9 - 1)15 = 255$$

kayan nokta vardır.

Buradaki A matrisi ve x vektörünün bazı özel durumları için matris-vektör çarpımındaki kayan noktalar şöyledir (Boyd ve Vandenberghe, 2004):

- $m = n$ ve A köşegen matris olmak üzere A matrisi ve x vektörünün çarpımı n kayan noktadır.
- $m = n$ ve A alt üçgensel matris olmak üzere A matrisi ve x vektörünün çarpımı n^2 kayan noktadır.
- A matrisinin elemanlarında sıfırlar çok ve A matrisinin N elemanı sıfırdan farklı ise A matrisi ve x vektörünün çarpımı için $2N$ tane kayan nokta vardır.

2.6.3. Matris-matris çarpımı

$A \in \mathbb{R}^{m \times n}$ ve $B \in \mathbb{R}^{n \times p}$ olan iki matris olsun. O zaman, A matrisi ve B matrisinin çarpımında ($C = AB$ ' de);

C' 'nin her bir elemanı hesaplanırken $\mathbb{R}^n$ 'de iç çarpım yapılır ve $\mathbb{R}^n$ 'de bir iç çarpımında $2n-1$ kayan nokta olduğundan yukarıda bahsetmiştik. $C \in \mathbb{R}^{m \times p}$ yani C' 'nin mp elemanı olduğundan A matrisi ve B matrisinin çarpımı için $(2n-1)mp$ kayan nokta vardır. Çok büyük n değerleri için A matrisi ve x vektörünün çarpımında yaklaşık olarak $2nmp$ kayan nokta olur (Boyd ve Vandenberghe, 2004).

2.6.4. Bir matrisin tersi

$A \in \mathbb{R}^{n \times n}$ olan bir matris olsun.

Naive Gaussian eliminasyonu kullanılarak A matrisinin tersinin bulunması için

$$-\frac{n^2}{3} + n^3 + \frac{n^4}{3}$$

tane kayan nokta olur (Trahan vd., 2006).

Eğer A matrisinin tersi LU ayrıştırması kullanılarak hesaplanırsa o zaman A matrisinin tersinin bulunması için

$$-\frac{5n}{6} + \frac{3n^2}{2} + \frac{4n^3}{3}$$

kayan nokta vardır (Trahan vd., 2006).

Örneğin;

$n = 10$ için yani $A \in \mathbb{R}^{10 \times 10}$ tipinde bir matris olsun. O zaman,

Naive Gaussian eliminasyonu kullanılarak A matrisinin tersinin hesaplanmasında

$$-\frac{n^2}{3} + n^3 + \frac{n^4}{3} = -\frac{10^2}{3} + 10^3 + \frac{10^4}{3} = 4300$$

kayan nokta olur.

LU ayrıştırması kullanılarak A matrisinin tersinin hesaplanmasın için

$$-\frac{5n}{6} + \frac{3n^2}{2} + \frac{4n^3}{3} = -\frac{5 \cdot 10}{6} + \frac{3 \cdot 10^2}{2} + \frac{4 \cdot 10^3}{3} = 1475$$

kayan nokta vardır.

2.6.5. Gauss eliminasyon

$A \in \mathbb{R}^{m \times n}$ olan bir matris ve $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$ olan bir vektör olsun. O zaman Gauss

eliminasyon kullanılarak $Ax = b$ sistemi çözülürse toplam

$$n^2 \left(m - \frac{n}{3} \right)$$

kayan nokta olur (Higham, 2002).

2.7. Karmaşıklık (Complexity)

Bir metodun karmaşıklığını tahmin etmek için ilk olarak problemin boyutlarının polinom fonksiyonu olarak kayan noktaların sayısı ifade edilir. Daha sonra aynı dereceye sahip olan kuvvetler sadeleştirilir ve metodun karmaşıklığı polinomun en yüksek derecesi olarak belirlenir (Boyd ve Vandenberghe, 2004).

3. MATERYAL VE YÖNTEM

Önceki çalışmalarda birçok adi diferansiyel, integral ve integro-diferansiyel denklemler ve onların sistemlerinin nümerik çözümleri için Bessel sıralama metodu uygulandı. Örneğin, Fredholm integro-diferansiyel-fark denklemleri (Şahin vd., 2011a), diferansiyel ve integro-diferansiyel denklemleri (Yüzbaşı, 2009), Volterra integral denklemlerin sistemleri (Şahin vd., 2011b) ve pantograph denklemleri (Yüzbaşı vd., 2012) çözmek için Bessel sıralama metodu kullanıldı. Adi diferansiyel, integral and integro-diferansiyel denklemler için Bessel sıralam metodu uygulanarak,

$$y(x) = \sum_{n=0}^N a_n J_n(x)$$

kesilmiş Bessel seri formunda yaklaşık çözümler elde edilir. Burada, a_n ; $n = 0, \dots, N$ ' ler bilinmeyen Bessel katsayıları ve $J_n(x)$, $(n = 0, 1, \dots, N)$ ' ler

$$J_n(x) = \sum_{k=0}^{\left\lfloor \frac{N-n}{2} \right\rfloor} \frac{(-1)^k}{k!(k+n)!} \left(\frac{x}{2}\right)^{2k+n}, \quad n \in \mathbb{N}, \quad 0 \leq x \leq \infty$$

ile tanımlı birinci tip Bessel fonksiyonlarıdır.

Fakat bu zamana kadar kısmi diferansiyel denklemler için Bessel sıralama metodu uygulanmadı. Bu çalışmamızda biz Bessel sıralama metodunu bazı kısmi diferansiyel denklem problemleri için geliştirerek uygulayacağız. Ayrıca, Tau metodu için rezidüel hata tahmini (Shahmorad, 2005) ve rezidüel düzeltme metodu (Oliveira, 1980; Çelik, 2006) kullanılarak, metot için etkili bir hata tahmini sunacağız. Bu hata fonksiyonu yardımıyla Bessel polinom çözümlerini iyileştireceğiz.

3.1. Neuman, Dirichlet ve İntegral Koşulları Altında Dalga Denkleminin Yaklaşık Çözümü için Bessel Sıralama Metodu

Bu kısımda, (Dehghan, 2005; Yousefi vd., 2010)' de ele alınan

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = q(x, t), \quad 0 \leq x \leq l, \quad 0 \leq t \leq T, \quad (3.1)$$

dalga denklemini

$$u(x, 0) = f_1(x), \quad 0 \leq x \leq l, \quad (3.2)$$

$$u(0, t) = g_1(t), \quad 0 \leq t \leq T, \quad (3.3)$$

Dirichlet sınır koşulları,

$$u_l(x, 0) = f_2(x), \quad 0 \leq x \leq l, \quad (3.4)$$

Neumann sınır koşulu ve

$$\int_0^l u(x, t) dx = g_2(t), \quad 0 \leq t \leq T, \quad (3.5)$$

lineer olmayan (nonlocal) integral koşulu ile alacağız ve bu problemin Birinci tür Bessel fonksiyonlarını temel alan yaklaşık çözümlerini elde etmek için Bessel sıralama yöntemini uygulayacağız. Burada, $(x, t) \in [0, l] \times [0, T]$ için q , f_1 , f_2 , g_1 ve g_2 tanımlı fonksiyonlar.

Şimdi bu problem için yöntemi uygulayarak,

$$u(x, t) = \sum_{r=0}^N \sum_{s=0}^N a_{r,s} J_{r,s}(x, t) \quad (3.6)$$

kesilmiş Bessel seri formunda yaklaşık çözümler bulacağız. Burada, $a_{r,s}$; $r, s = 0, \dots, N$ ' ler bilinmeyen Bessel katsayıları,

$$J_{r,s}(x,t) = J_r(x)J_s(t)$$

ve $J_n(x)$, $(n = 0, 1, \dots, N)$ ' ler

$$J_n(x) = \sum_{k=0}^{\left\lfloor \frac{N-n}{2} \right\rfloor} \frac{(-1)^k}{k!(k+n)!} \left(\frac{x}{2}\right)^{2k+n}, \quad n \in \mathbb{N}, \quad 0 \leq x \leq \infty$$

ile tanımlı birinci tip Bessel fonksiyonlarıdır.

3.1.1. Dalga denklemi için temel matris bağıntıları

(3.1) ile ifade edilen dalga denkleminin (3.2), (3.3), (3.4) ve (3.5) koşulları altında (3.6) formunda yaklaşık çözümünü elde etmek için, bilinmeyen Bessel katsayılarının bulunması gerekmektedir. Bu amaç için ilk olarak (3.1) denklemindeki terimlerin matris formları oluşturulmalıdır. İlk olarak, (3.6) ile verilen çözüm

$$u(x,t) = \mathbf{J}(x)\mathbf{Q}(t)\mathbf{A} \quad (3.7)$$

olarak matris formunda yazılabilir. Burada,

$$\mathbf{J}(x) = [J_0(x) \quad J_1(x) \quad \cdots \quad J_N(x)]_{1 \times (N+1)}, \quad \mathbf{Q}(t) = \begin{bmatrix} \mathbf{J}(t) & 0 & \cdots & 0 \\ 0 & \mathbf{J}(t) & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{J}(t) \end{bmatrix}_{(N+1) \times (N+1)^2}$$

ve

$$\mathbf{A} = [a_{0,0} \quad a_{0,1} \quad \cdots \quad a_{0,N} \quad a_{1,0} \quad a_{1,1} \quad \cdots \quad a_{1,N} \quad \cdots \quad a_{N,0} \quad a_{N,1} \quad \cdots \quad a_{N,N}]^T.$$

Ayrıca, birinci tip Bessel fonksiyonları

$$\mathbf{J}(x) = \mathbf{X}(x)\mathbf{D}^T \quad (3.8)$$

olarak matris formunda ifade edilebilir (Yüzbaşı, 2009; Şahin vd., 2011a; Şahin vd., 2011b, Yüzbaşı vd., 2012). Burada,

$$\mathbf{X}(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^N \end{bmatrix}$$

ve N tek ise,

$$\mathbf{D} = \begin{bmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & \frac{(-1)^{\frac{N-1}{2}}}{\left(\frac{N-1}{2}\right)! \left(\frac{N-1}{2}\right)! 2^{N-1}} & 0 \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{\left(\frac{N-1}{2}\right)! \left(\frac{N+1}{2}\right)! 2^N} \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & \frac{(-1)^{\frac{N-3}{2}}}{\left(\frac{N-3}{2}\right)! \left(\frac{N+1}{2}\right)! 2^{N-1}} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{0!(N-1)! 2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{0!N! 2^N} \end{bmatrix}_{(N+1) \times (N+1)},$$

N çift ise,

$$\mathbf{D} = \begin{bmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N}{2}}}{\left(\frac{N}{2}\right)! \left(\frac{N}{2}\right)! 2^N} \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & \frac{(-1)^{\frac{N-2}{2}}}{\left(\frac{N-2}{2}\right)! \left(\frac{N}{2}\right)! 2^{N-1}} & 0 \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N-2}{2}}}{\left(\frac{N-2}{2}\right)! \left(\frac{N+2}{2}\right)! 2^N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{0!(N-1)! 2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{0!N! 2^N} \end{bmatrix}_{(N+1) \times (N+1)}.$$

İlaveten, $\mathbf{X}(x)$ ve onun $\mathbf{X}^{(1)}(x)$ ve $\mathbf{X}^{(2)}(x)$ türevleri arasında ilişki

$$\mathbf{X}^{(1)}(x) = \mathbf{X}(x)\mathbf{C} \quad \text{ve} \quad \mathbf{X}^{(2)}(x) = \mathbf{X}(x)\mathbf{C}^2 \quad (3.9)$$

şeklindedir (Yüzbaşı, 2009; Şahin vd., 2011a; Şahin vd., 2011b, Yüzbaşı vd., 2012).

Burada,

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & N \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

(3.8) ve (3.9) ilişkileri kullanılarak,

$$\mathbf{J}^{(1)}(x) = \mathbf{X}(x)\mathbf{C}\mathbf{D}^T \quad \text{ve} \quad \mathbf{J}^{(2)}(x) = \mathbf{X}(x)\mathbf{C}^2\mathbf{D}^T. \quad (3.10)$$

matris ilişkilerini elde ederiz.

(3.8) denkleminde $\mathbf{D}^T$ matrisi tersinir olduğu için,

$$\mathbf{X}(x) = \mathbf{J}(x)(\mathbf{D}^T)^{-1} \quad (3.11)$$

yazılabilir. (3.11) ilişkisi (3.10)' da verilen ilişkilerde yerine konularak,

$$\mathbf{J}^{(1)}(x) = \mathbf{J}(x)\mathbf{P} \quad \text{ve} \quad \mathbf{J}^{(2)}(x) = \mathbf{J}(x)\mathbf{P}^2 \quad (3.12)$$

matris formları elde edilir. Burada,

$$\mathbf{P}^k = (\mathbf{D}^T)^{-1}\mathbf{C}^k\mathbf{D}^T, \quad k = 1, 2.$$

Benzer şekilde, $\mathbf{Q}(t)$ ve onun $\mathbf{Q}^{(1)}(t)$ ve $\mathbf{Q}^{(2)}(t)$ türevleri arasında ilişki

$$\mathbf{Q}^{(1)}(t) = \mathbf{Q}(t)\bar{\mathbf{P}} \quad \text{ve} \quad \mathbf{Q}^{(2)}(t) = \mathbf{Q}(t)\bar{\mathbf{P}}^2 \quad (3.13)$$

olur. Burada,

$$\bar{\mathbf{P}} = \begin{bmatrix} \mathbf{P} & 0 & \cdots & 0 \\ 0 & \mathbf{P} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{P} \end{bmatrix}_{(N+1)^2 \times (N+1)^2}.$$

3.1.2. Neuman, Dirichlet ve integral koşulları için matris bağıntıları

Bu kısımda, (3.2)-(3.3) Dirichlet sınır koşullarının, (3.4) Neumann sınır koşulunun ve (3.5) lineer olmayan (nonlocal) integral koşulunun matris formlarını oluşturacağız. İlk olarak, (3.2) Dirichlet sınır koşulunun matris formunu bulmak için, çözüm fonksiyonunun (3.7) matris formunda t yerine 0 yazdığımızda

$$u(x, 0) = \mathbf{J}(x)\mathbf{Q}(0)\mathbf{A} = f_1(x), \quad 0 \leq x \leq l \quad (3.14)$$

matris formunu elde ederiz.

Benzer şekilde, (3.7) çözüm fonksiyonunda x yerine 0 yazıldığında

$$u(0, t) = \mathbf{J}(0)\mathbf{Q}(t)\mathbf{A} = g_1(t), \quad 0 \leq t \leq T, \quad (3.15)$$

matris formuna sahip oluruz.

(3.4) Neumann sınır koşulunun matris formunu bulmak için, ilk olarak (3.7) çözüm fonksiyonunun t' ye göre birinci mertebe kısmi türevinin matris formunu bulalım. (3.7) çözüm fonksiyonunun t' ye göre birinci mertebe kısmi türevi

$$u_t(x, t) = \mathbf{J}(x)\mathbf{Q}^{(1)}(t)\mathbf{A}$$

olur ve (3.13) ilişkisi kullanılarak,

$$u_t(x, t) = \mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}\mathbf{A} \quad (3.16)$$

biçiminde daha açık olarak ifade edilebilir. Böylece, (3.16) denkleminde t yerine 0 yazılarak, (3.4) başlangıç koşulunun matris formu

$$u_t(x, 0) = \mathbf{J}(x)\mathbf{Q}(0)\bar{\mathbf{P}}\mathbf{A} = f_2(x), \quad a \leq x \leq b \quad (3.17)$$

olarak elde edilir.

Son olarak, (3.5) lineer olmayan integral koşulunun matris formunu bulalım. Bu amaç için, (3.7) matris formu (3.5) koşulunun sol tarafında yerine yerleştirilir ve

$$\int_0^l \mathbf{J}(x) \mathbf{Q}(t) \mathbf{A} dx = \left\{ \int_0^l \mathbf{J}(x) dx \right\} \mathbf{Q}(t) \mathbf{A} = \left\{ \int_0^l \mathbf{X}(x) dx \right\} \mathbf{D}^T \mathbf{Q}(t) \mathbf{A}, \quad 0 \leq t \leq T$$

matris formu elde edilir. Böylece,

$$\mathbf{L} = \begin{bmatrix} l & \frac{l^2}{2} & \frac{l^3}{3} & \cdots & \frac{l^{N+1}}{N+1} \end{bmatrix}$$

olmak üzere (3.5) koşulunun matris formu

$$\mathbf{L} \mathbf{D}^T \mathbf{Q}(t) \mathbf{A} = g_2(t), \quad 0 \leq t \leq T \quad (3.18)$$

olarak elde edilir.

3.1.3. Dalga denklemi problemi için çözüm yöntemi

Öncelikle, (3.1) denklemindeki $u_{xx}(x,t)$ ve $u_{tt}(x,t)$ terimlerinin matris formlarını oluşturalım. (3.7) çözüm fonksiyonunun x ' e göre ikinci mertebe kısmi türevinin matris formu

$$u_{xx}(x,t) = \mathbf{J}^{(2)}(x) \mathbf{Q}(t) \mathbf{A}$$

dır ve burada, (3.12) ilişkisi kullanılarak,

$$u_{xx}(x,t) = \mathbf{J}(x) \mathbf{P}^2 \mathbf{Q}(t) \mathbf{A} \quad (3.19)$$

elde edilir.

Benzer şekilde, (3.13) ilişkisi kullanılarak, (3.7) çözüm fonksiyonunun t ' ye göre ikinci mertebe kısmi türevinin matris formu

$$u_{tt}(x,t) = \mathbf{J}(x) \mathbf{Q}^{(2)}(t) \mathbf{A}$$

$$= \mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2\mathbf{A} \quad (3.20)$$

olarak bulunur.

(3.19) ve (3.20) matris formları (3.1) denkleminde yerine konularak,

$$\{\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 - \mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t)\}\mathbf{A}=q(x,t) \quad (3.21)$$

matris denklemi elde edilir. Kısaca, (3.21) denklemini

$$\mathbf{W}(x,t)\mathbf{A}=q(x,t) \quad (3.22)$$

olarak ifade edebiliriz. Burada,

$$\mathbf{W}(x,t)=[w_{i,k}]_{1 \times (N+1)^2} = \mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 - \mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t), \quad k = 0,1,\dots,(N+1)^2.$$

$0 \leq x \leq l$ ve $0 \leq t \leq T$ olmak üzere,

$$x_i = \frac{l}{N}i, \quad t_j = \frac{T}{N}j, \quad i = 0,1,\dots,N, \quad j = 0,1,\dots,N \quad (3.23)$$

ile tanımlı sıralama noktalarını (3.22) denkleminde kullanarak,

$$\mathbf{W}(x_i,t_j)\mathbf{A}=q(x_i,t_j), \quad i = 0,1,\dots,N, \quad j = 0,1,\dots,N \quad (3.24)$$

matris denklemlerin bir sistemini elde ederiz ve bu sistem

$$\mathbf{W}\mathbf{A}=\mathbf{Q} \text{ ya da } [\mathbf{W};\mathbf{Q}] \quad (3.25)$$

olarak kısaca yazılabilir. O zaman,

$$\mathbf{W}=[\mathbf{W}(x_0,t_0) \mathbf{W}(x_0,t_1) \cdots \mathbf{W}(x_0,t_N) \mathbf{W}(x_1,t_0) \mathbf{W}(x_1,t_1) \cdots \mathbf{W}(x_1,t_N) \cdots \mathbf{W}(x_N,t_0) \mathbf{W}(x_N,t_1) \cdots \mathbf{W}(x_N,t_N)]_{(N+1)^2 \times (N+1)^2}^T,$$

$$\mathbf{A}=[a_{0,0} \ a_{0,1} \cdots a_{0,N} \ a_{1,0} \ a_{1,1} \cdots a_{1,N} \cdots a_{N,0} \ a_{N,1} \cdots a_{N,N}]_{(N+1)^2 \times 1}^T$$

ve

$$\mathbf{Q} = \left[q(x_0, t_0) q(x_0, t_1) \cdots q(x_0, t_N) q(x_1, t_0) q(x_1, t_1) \cdots q(x_1, t_N) \cdots q(x_N, t_0) q(x_N, t_1) \cdots q(x_N, t_N) \right]_{(N+1)^2 \times 1}^T.$$

(3.2)-(3.5) koşullarına karşılık gelen (3.14), (3.15), (3.17) ve (3.18) matris ilişkilerinde (3.23) sıralama noktaları kullanılarak

$$\mathbf{U}\mathbf{A} = [\mathbf{F}_1] \text{ ya da } [\mathbf{U}; \mathbf{F}_1], \quad (3.26)$$

$$\bar{\mathbf{U}}\mathbf{A} = [\mathbf{F}_2] \text{ ya da } [\bar{\mathbf{U}}; \mathbf{F}_2], \quad (3.27)$$

$$\mathbf{V}\mathbf{A} = [\mathbf{G}_1] \text{ ya da } [\mathbf{V}; \mathbf{G}_1], \quad (3.28)$$

$$\bar{\mathbf{V}}\mathbf{A} = [\mathbf{G}_2] \text{ ya da } [\bar{\mathbf{V}}; \mathbf{G}_2], \quad (3.29)$$

matris formları elde edilir. Burada,

$$\mathbf{U} = [\mathbf{U}_0 \quad \mathbf{U}_1 \quad \cdots \quad \mathbf{U}_N]^T, \quad \bar{\mathbf{U}} = [\bar{\mathbf{U}}_0 \quad \bar{\mathbf{U}}_1 \quad \cdots \quad \bar{\mathbf{U}}_N]^T, \quad \mathbf{V} = [\mathbf{V}_0 \quad \mathbf{V}_1 \quad \cdots \quad \mathbf{V}_N]^T,$$

$$\bar{\mathbf{V}} = [\bar{\mathbf{V}}_0 \quad \bar{\mathbf{V}}_1 \quad \cdots \quad \bar{\mathbf{V}}_N]^T, \quad \mathbf{F}_1 = [f_1(x_0) \quad f_1(x_1) \quad \cdots \quad f_1(x_N)]^T,$$

$$\mathbf{F}_2 = [f_2(x_0) \quad f_2(x_1) \quad \cdots \quad f_2(x_N)]^T, \quad \mathbf{G}_1 = [g_1(t_0) \quad g_1(t_1) \quad \cdots \quad g_1(t_N)]^T,$$

$$\mathbf{G}_2 = [g_2(t_0) \quad g_2(t_1) \quad \cdots \quad g_2(t_N)]^T, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

$$\mathbf{U}_i = \mathbf{J}(x_i)\mathbf{Q}(0) = \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} & \cdots & u_{i(N+1)^2} \end{bmatrix},$$

$$\bar{\mathbf{U}}_i = \mathbf{J}(x_i)\mathbf{Q}(0)\bar{\mathbf{P}} = \begin{bmatrix} \bar{u}_{i1} & \bar{u}_{i2} & \bar{u}_{i3} & \cdots & \bar{u}_{i(N+1)^2} \end{bmatrix},$$

$$\mathbf{V}_j = \mathbf{J}(0)\mathbf{Q}(t_j) = \begin{bmatrix} v_{j1} & v_{j2} & v_{j3} & \cdots & v_{j(N+1)^2} \end{bmatrix}$$

ve

$$\bar{\mathbf{V}}_j = \mathbf{L}\mathbf{D}^T\mathbf{Q}(t_j) = \begin{bmatrix} \bar{v}_{j1} & \bar{v}_{j2} & \bar{v}_{j3} & \dots & \bar{v}_{j(N+1)^2} \end{bmatrix}.$$

(3.1) denkleminin (3.2)-(3.5) koşulları altında yaklaşık çözümünü elde etmek için, (3.25) denklemindeki $[\mathbf{W};\mathbf{Q}]$ artırılmış matrisine koşulların (3.26)-(3.29) artırılmış matris formlarının eklenmesiyle

$$[\tilde{\mathbf{W}};\tilde{\mathbf{Q}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F}_1 \\ \bar{\mathbf{U}} & ; & \mathbf{F}_2 \\ \mathbf{V} & ; & \mathbf{G}_1 \\ \bar{\mathbf{V}} & ; & \mathbf{G}_2 \\ \mathbf{W} & ; & \mathbf{Q} \end{bmatrix} \quad (3.30)$$

yeni artırılmış matrisi elde edilir. Böylece bilinmeyen Bessel katsayılar matrisi

$$\mathbf{A} = (\tilde{\mathbf{W}})^{-1} \tilde{\mathbf{Q}}$$

olarak elde edilir. Burada $[\tilde{\mathbf{W}};\tilde{\mathbf{Q}}]$ artırılmış matrisi, $[\tilde{\mathbf{W}};\tilde{\mathbf{Q}}]$ artırılmış matrisine Gauss eliminasyon uygulandıktan sonra sıfır satırları atılarak elde edilen matristir. Matlab’ de (3.30) sisteminin çözümü ‘ $\mathbf{A} = \tilde{\mathbf{W}} \setminus \tilde{\mathbf{Q}}$ ’ komutu ile kolayca hesaplanabilir.

Son olarak, belirlenen $a_{r,s}$; $(r,s = 0,...,N)$ katsayılarını (3.6) denkleminde yerine yerleştirerek

$$u_N(x,t) = \sum_{r=0}^N \sum_{s=0}^N a_{r,s} J_{r,s}(x,t) \quad (3.31)$$

yaklaşık çözümünü elde ederiz.

3.1.4. Dalga denklemleri problemi için hata tahmini

Bu kesimde, Tau metodu için rezidüel hata tahmini (Shahmorad, 2005) ve rezidüel düzeltme metodu (Oliveira, 1980; Çelik, 2006) kullanılarak, yöntem için etkili bir hata tahmini sunulacaktır. İlk olarak, (3.1) denklemini

$$L[u(x,t)] = q(x,t), \quad L[u(x,t)] = \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2}, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T, \quad (3.32)$$

olarak ifade edebiliriz. Burada, L : lineer operatör.

Bizim metot için,

$$R_N(x,t) = L[u_N(x,t)] - q(x,t) \quad (3.33)$$

rezidüel fonksiyonunu ele alalım. Burada, $u_N(x,t)$: (3.1) denkleminin (3.31) ile verilen yaklaşık çözümünü gösterir. Böylece, $u_N(x,t)$,

$$L[u_N(x,t)] = \frac{\partial^2 u_N}{\partial t^2} - \frac{\partial^2 u_N}{\partial x^2} = q(x,t) + R_N(x,t) \quad (3.34)$$

denklemini ve

$$\begin{cases} u_N(x,0) = f_1(x), \quad u_{Nt}(x,0) = f_2(x), \quad 0 \leq x \leq l, \\ u_N(0,t) = g_1(t), \quad \int_0^l u_N(x,t) dx = g_2(t), \quad 0 \leq t \leq T. \end{cases} \quad (3.35)$$

koşullarını sağlar.

$u(x,t)$: (3.1)-(3.5) probleminin tam çözümü olmak üzere,

$$e_N(x,t) = u(x,t) - u_N(x,t) \quad (3.36)$$

olarak hata fonksiyonunu tanımlayalım.

(3.1)-(3.5) denklemleri ve (3.34)-(3.35) denklemleri aracılığıyla,

$$L[e_N(x,t)] = L[u(x,t)] - L[u_N(x,t)] = -R_N(x,t)$$

hata diferansiyel denklemini ve

$$\begin{cases} e_N(x,0) = 0, \quad e_{Nt}(x,0) = 0, \quad 0 \leq x \leq l, \\ e_N(0,t) = 0, \quad \int_0^l e_N(x,t) dx = 0, \quad 0 \leq t \leq T \end{cases} \quad (3.37)$$

homojen koşullarını elde ederiz.

Açık olarak,

$$\begin{cases} \frac{\partial^2 e_N}{\partial t^2} - \frac{\partial^2 e_N}{\partial x^2} = -R_N(x, t), \\ e_N(x, 0) = 0, \quad e_{N_t}(x, 0) = 0, \quad 0 \leq x \leq l, \\ e_N(0, t) = 0, \quad \int_0^l e_N(x, t) dx = 0, \quad 0 \leq t \leq T \end{cases} \quad (3.38)$$

hata problemini elde etmiş oluruz.

Biz, (3.2)-(3.5) ve (3.35) homojen olmayan koşullarını kullanarak (3.37) homojen koşullarını elde ettiğimizi not ediyoruz.

Bölüm (3.1.3)' de sunulan yöntem ile (3.38) problemini çözerek, $e_N(x, t)$ hata fonksiyonu için $e_{N,M}(x, t)$ hata yaklaşımını elde ederiz. Burada, M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını gösterir.

Hata probleminin çözümü için metodu uygularken, orijinal problemin çözümünü elde ederken kullandığımız matrislerde sadece değişecek olanlar $q(x, t)$ yerine $-R_N(x, t)$ ve koşulların sağ taraflarını sıfır almaktır.

Sonuç olarak, eğer problemin tam çözümü bilinmiyorsa, o zaman $e_{N,M}(x, t)$ yaklaşımı kullanılarak hata fonksiyonu tahmin edilebilir.

Diğer taraftan, eğer problemin tam çözümü biliniyor ve $u(x, t)$ ise o zaman metodun etkinliği ve güvenilirliği,

$$L_2 = \|u(x, t) - u_N(x, t)\|_2 = \left(\int_0^T \int_0^l (u(x, t) - u_N(x, t))^2 dx dt \right)^{1/2}$$

ve

$$L_\infty = \|u(x, t) - u_N(x, t)\|_\infty = \max \{ |u(x, t) - u_N(x, t)|, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T \}.$$

hataları ile de test edilebilir.

3.1.5. Dalga denklemi probleminin çözümlerinin iyileştirilmesi

Bu kesimde, (3.1)-(3.5) dalga denklemi probleminin (3.31) yaklaşık çözümleri rezidüel hata fonksiyonu tahmini aracılığıyla iyileştirilecek. Bu iyileştirme yapılırken, ilk olarak, bölüm (3.1.4)' de sunulan prosedür takip edilerek $e_N(x,t)$ hata fonksiyonu için $e_{N,M}(x,t)$ yaklaşımı elde edilir. Daha sonra, (3.31) ile bu hata fonksiyonunun yaklaşımı toplanarak,

$$u_{N,M}(x,t) = u_N(x,t) + e_{N,M}(x,t)$$

iyileştirilmiş yaklaşık çözümü elde edilir. Burada, N : orijinal problemin (3.31) yaklaşık çözümü elde edilirken kullanılan kesme sınırını gösterir ve M : hata fonksiyonu için Bessel polinom yaklaşımı elde edilirken kullanılan kesme sınırını gösterir. İyileştirme yapılırken $M > N$ seçilirse daha iyi çözümler elde edilebilir.

3.2. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu

Bu kesimde,

$$A(x,t)\frac{\partial^2 u}{\partial x^2} + B(x,t)\frac{\partial^2 u}{\partial t \partial x} + C(x,t)\frac{\partial^2 u}{\partial t^2} + D(x,t)\frac{\partial u}{\partial x} + E(x,t)\frac{\partial u}{\partial t} + F(x,t)u = G(x,t), \quad (3.39)$$

ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denkleminin

$$u(x,\phi) = f_1(x), \quad a \leq x \leq b, \quad \phi = c \text{ ya da } \phi = d, \quad (3.40)$$

$$u(\phi,t) = g_1(t), \quad c \leq t \leq d, \quad \phi = a \text{ ya da } \phi = b, \quad (3.41)$$

Dirichlet sınır koşulları,

$$\frac{\partial u}{\partial \mathbf{n}}(\phi,t) = g_2(t), \quad c \leq t \leq d, \quad \phi = a \text{ ya da } \phi = b, \quad (3.42)$$

Neumann sınır koşulu ve

$$\alpha u(x, \mu) + \beta \frac{\partial u}{\partial \mathbf{n}}(x, \eta) = f_2(x), \quad a \leq x \leq b, \quad \mu = c \text{ ya da } \mu = d, \quad \eta = c \text{ ya da } \eta = d \quad (3.43)$$

Robin sınır koşulu altında Bessel polinom çözümlerini elde etmek için Bessel sıralama metodunu uygulayacağız. Burada, $(x, t) \in [a, b] \times [c, d]$ için $A, B, C, D, E, F, G, f_1, f_2, g_1$ ve g_2 tanımlı fonksiyonlar. $\mathbf{n}$ ve $\bar{\mathbf{n}}$: $\partial\Omega$ sınırı için birim normal vektörler ve ayrıca normal türev aşağıdaki gibi tanımlıdır:

$$\frac{\partial u}{\partial \mathbf{n}}(x, t) = \nabla u(x, t) \cdot \mathbf{n},$$

burada, ∇ : gradyan vektörü gösterir ve ‘ $\cdot$ ’ ile iç çarpım gösterilir.

Önerilen metodu (3.39)-(3.43) problemi için uygulayarak (3.6) formunda yaklaşık çözümler elde edeceğiz.

3.2.1. Genel formda ikinci mertebe lineer kısmi diferansiyel denklem için temel matris bağıntıları

Bu kısımda, (3.6) yaklaşık çözümün matris formu olan (3.7) denklemi aracılığıyla (3.39) denklemindeki $u_x(x, t)$, $u_t(x, t)$, $u_{xx}(x, t)$, $u_{xt}(x, t)$ ve $u_{tt}(x, t)$ terimlerinin matris formlarını oluşturacağız.

(3.12) denklemi kullanılarak, (3.7) çözüm fonksiyonunun x ’ e göre birinci mertebe kısmi türevinin matris formu

$$u_x(x, t) = \mathbf{J}(x) \mathbf{PQ}(t) \mathbf{A} \quad (3.44)$$

olarak elde edilir.

(3.7) çözüm fonksiyonunun t ’ ye göre birinci mertebe kısmi türevinin matris formu (3.16)’ da

$$u_t(x, t) = \mathbf{J}(x) \mathbf{Q}(t) \bar{\mathbf{P}} \mathbf{A} \quad (3.45)$$

şeklinde bulunmuştu.

(3.7) çözüm fonksiyonunun x' e ve t' ye göre ikinci mertebe kısmi türevlerinin matris formları sırasıyla (3.19) ve (3.20) denklemlerinde

$$u_{xx}(x, t) = \mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t)\mathbf{A} \quad (3.46)$$

ve

$$u_{tt}(x, t) = \mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2\mathbf{A} \quad (3.47)$$

olarak bulunmuştu.

(3.13) ilişkisi aracılığıyla, (3.44) denkleminin t' ye göre birinci mertebe kısmi türevinin matris formu

$$u_{xt}(x, t) = \mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}}\mathbf{A} \quad (3.48)$$

olur.

3.2.2. Neuman, Dirichlet ve Robin koşulları için matris bağıntıları

Bu kısımda, (3.40)-(3.43) koşullarının matris formlarını oluşturacağız.

İlk olarak, (3.40) koşulu için (3.7) matris formunda t yerine ϕ yazarak,

$$u(x, \phi) = \mathbf{J}(x)\mathbf{Q}(\phi)\mathbf{A} = f_1(x) \quad (3.49)$$

matris ilişkisi elde edilir.

Benzer şekilde, (3.41) koşulu için (3.7) matris formunda x yerine φ yazarak,

$$u(\varphi, t) = \mathbf{J}(\varphi)\mathbf{Q}(t)\mathbf{A} = g_1(t) \quad (3.50)$$

matris denklemine sahip oluruz.

(3.42) ile verilen Neumann sınır koşulunun matris formunu elde etmek için ilk olarak bu koşulu açık olarak

$$n_1 \frac{\partial u}{\partial x}(\varphi, t) + n_2 \frac{\partial u}{\partial t}(\varphi, t) = g_2(t), \quad c \leq t \leq d \quad (3.51)$$

şeklinde yazalım. Burada,

$$\frac{\partial u}{\partial \mathbf{n}}(x, t) = \nabla u(x, t) \cdot \mathbf{n} \quad \text{ve} \quad \mathbf{n} = (n_1, n_2).$$

(3.44) ve (3.45)' de x yerine φ konularak,

$$u_x(\varphi, t) = \mathbf{J}(\varphi) \mathbf{PQ}(t) \mathbf{A}$$

ve

$$u_t(\varphi, t) = \mathbf{J}(\varphi) \mathbf{Q}(t) \bar{\mathbf{P}} \mathbf{A}$$

matris ilişkileri elde edilir. Böylece, bu matris denklemleri (3.51) denkleminde yerine yerleştirilerek,

$$\{n_1 \mathbf{J}(\varphi) \mathbf{PQ}(t) + n_2 \mathbf{J}(\varphi) \mathbf{Q}(t) \bar{\mathbf{P}}\} \mathbf{A} = g_2(t) \quad (3.52)$$

matris formu elde edilir.

Benzer şekilde, $\bar{\mathbf{n}} = (\bar{n}_1, \bar{n}_2)$ ve $\frac{\partial u}{\partial \bar{\mathbf{n}}}(x, t) = \nabla u(x, t) \cdot \bar{\mathbf{n}}$ kullanılarak (3.43) ile verilen

Robin sınır koşulu kısaca

$$\alpha u(x, \mu) + \beta \left(\bar{n}_1 \frac{\partial u}{\partial x}(x, \eta) + \bar{n}_2 \frac{\partial u}{\partial t}(x, \eta) \right) = f_2(x), \quad a \leq x \leq b, \quad (3.53)$$

olarak yazılır.

(3.7)' de t yerine μ yazılarak ve (3.44) ve (3.45)' de t yerine η konularak,

$$u(x, \mu) = \mathbf{J}(x) \mathbf{Q}(\mu) \mathbf{A},$$

$$u_x(x, \eta) = \mathbf{J}(x) \mathbf{PQ}(\eta) \mathbf{A}$$

ve

$$u_i(x, \eta) = \mathbf{J}(x)\mathbf{Q}(\eta)\bar{\mathbf{P}}\mathbf{A}$$

matris ilişkilerine sahip oluruz ve bu matris ilişkileri (3.53)' de yerine konularak

$$\left\{ \alpha \mathbf{J}(x)\mathbf{Q}(\mu) + \beta \left(\bar{n}_1 \mathbf{J}(x)\mathbf{P}\mathbf{Q}(\eta) + \bar{n}_2 \mathbf{J}(x)\mathbf{Q}(\eta)\bar{\mathbf{P}} \right) \right\} \mathbf{A} = f_2(x) \quad (3.54)$$

matris formunu elde ederiz.

3.2.3. Problem için çözüm yöntemi

Bu kısımda, (3.39) ile verilen ikinci mertebe lineer kısmi diferansiyel denkleminin (3.40)-(3.43) koşulları ile birlikte (3.6) formunda yaklaşık çözümünü elde etmek için metodu uygulayacağız.

Öncelikle, (3.7) ve (3.44)-(3.48) matris formlarını (3.39) denkleminde yerine yazarak,

$$\mathbf{W}(x, t)\mathbf{A} = G(x, t) \quad (3.55)$$

elde ederiz. Burada,

$$\begin{aligned} \mathbf{W}(x, t) = & A(x, t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t) + B(x, t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}} + C(x, t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 + D(x, t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t) \\ & + E(x, t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}} + F(x, t)\mathbf{J}(x)\mathbf{Q}(t). \end{aligned}$$

$a \leq x \leq b$ ve $c \leq t \leq d$ olmak üzere,

$$x_i = a + \frac{(b-a)}{N}i, \quad t_j = c + \frac{(d-c)}{N}j, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N, \quad (3.56)$$

ile tanımlı sıralama noktaları (3.55) denkleminde yerleştirilerek,

$$\mathbf{W}(x_i, t_j)\mathbf{A} = G(x_i, t_j), \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

olarak matris denklemlerin bir sistemi elde edilir.

$$\mathbf{W} = \left[\mathbf{W}(x_0, t_0) \mathbf{W}(x_0, t_1) \cdots \mathbf{W}(x_0, t_N) \mathbf{W}(x_1, t_0) \mathbf{W}(x_1, t_1) \cdots \mathbf{W}(x_1, t_N) \cdots \mathbf{W}(x_N, t_0) \mathbf{W}(x_N, t_1) \cdots \mathbf{W}(x_N, t_N) \right]_{(N+1)^2 \times (N+1)^2}^T,$$

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N} & a_{1,0} & a_{1,1} & \cdots & a_{1,N} & \cdots & a_{N,0} & a_{N,1} & \cdots & a_{N,N} \end{bmatrix}_{(N+1)^2 \times 1}^T$$

ve

$$\mathbf{G} = \begin{bmatrix} G(x_0, t_0) & G(x_0, t_1) & \cdots & G(x_0, t_N) & G(x_1, t_0) & G(x_1, t_1) & \cdots & G(x_1, t_N) & \cdots & G(x_N, t_0) & G(x_N, t_1) & \cdots & G(x_N, t_N) \end{bmatrix}_{(N+1)^2 \times 1}^T$$

olmak üzere, yukarıdaki sistem

$$\mathbf{WA} = \mathbf{G} \text{ ya da } [\mathbf{W}; \mathbf{G}] \quad (3.57)$$

olarak kısaca ifade edilebilir.

Şimdi (3.40)-(3.43) koşullarının temel matris formlarını oluşturmak için (3.56) sıralama noktalarını (3.49), (3.50), (3.52) ve (3.54) matris denklemlerinde yerine koyarak,

$$u(x_i, \phi) = \mathbf{J}(x_i) \mathbf{Q}(\phi) \mathbf{A} = f_1(x_i), \quad i = 0, 1, \dots, N,$$

$$u(\phi, t_j) = \mathbf{J}(\phi) \mathbf{Q}(t_j) \mathbf{A} = g_1(t_j), \quad j = 0, 1, \dots, N,$$

$$\{n_1 \mathbf{J}(\phi) \mathbf{P} \mathbf{Q}(t_j) + n_2 \mathbf{J}(\phi) \mathbf{Q}(t_j) \bar{\mathbf{P}}\} \mathbf{A} = g_2(t_j), \quad j = 0, 1, \dots, N$$

ve

$$\{\alpha \mathbf{J}(x_i) \mathbf{Q}(\mu) + \beta (\bar{n}_1 \mathbf{J}(x_i) \mathbf{P} \mathbf{Q}(\eta) + \bar{n}_2 \mathbf{J}(x_i) \mathbf{Q}(\eta) \bar{\mathbf{P}})\} \mathbf{A} = f_2(x_i), \quad i = 0, 1, \dots, N$$

sistemlerini elde ederiz.

Kısaca, yukarıdaki sistemleri sırasıyla aşağıdaki gibi

$$\mathbf{UA} = [\mathbf{F}_1] \text{ ya da } [\mathbf{U}; \mathbf{F}_1], \quad (3.58)$$

$$\bar{\mathbf{U}} \mathbf{A} = [\mathbf{F}_2] \text{ ya da } [\bar{\mathbf{U}}; \mathbf{F}_2], \quad (3.59)$$

$$\mathbf{VA} = [\mathbf{G}_1] \text{ ya da } [\mathbf{V}; \mathbf{G}_1], \quad (3.60)$$

$$\bar{\mathbf{V}}\mathbf{A} = [\mathbf{G}_2] \text{ ya da } [\bar{\mathbf{V}}; \mathbf{G}_2] \quad (3.61)$$

matris formlarında yazabiliriz. Burada,

$$\mathbf{U} = [\mathbf{U}_0 \quad \mathbf{U}_1 \quad \cdots \quad \mathbf{U}_N]^T, \quad \bar{\mathbf{U}} = [\bar{\mathbf{U}}_0 \quad \bar{\mathbf{U}}_1 \quad \cdots \quad \bar{\mathbf{U}}_N]^T, \quad \mathbf{V} = [\mathbf{V}_0 \quad \mathbf{V}_1 \quad \cdots \quad \mathbf{V}_N]^T,$$

$$\bar{\mathbf{V}} = [\bar{\mathbf{V}}_0 \quad \bar{\mathbf{V}}_1 \quad \cdots \quad \bar{\mathbf{V}}_N]^T, \quad \mathbf{F}_1 = [f_1(x_0) \quad f_1(x_1) \quad \cdots \quad f_1(x_N)]^T,$$

$$\mathbf{F}_2 = [f_2(x_0) \quad f_2(x_1) \quad \cdots \quad f_2(x_N)]^T, \quad \mathbf{G}_1 = [g_1(t_0) \quad g_1(t_1) \quad \cdots \quad g_1(t_N)]^T,$$

$$\mathbf{G}_2 = [g_2(t_0) \quad g_2(t_1) \quad \cdots \quad g_2(t_N)]^T, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

$$\mathbf{U}_i = \mathbf{J}(x_i)\mathbf{Q}(c) = \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} & \cdots & u_{i(N+1)^2} \end{bmatrix},$$

$$\bar{\mathbf{U}}_i = \alpha \mathbf{J}(x_i)\mathbf{Q}(d) + \beta (\bar{n}_1 \mathbf{J}(x_i)\mathbf{P}\mathbf{Q}(c) + \bar{n}_2 \mathbf{J}(x_i)\mathbf{Q}(c)\bar{\mathbf{P}}) = \begin{bmatrix} \bar{u}_{i1} & \bar{u}_{i2} & \bar{u}_{i3} & \cdots & \bar{u}_{i(N+1)^2} \end{bmatrix},$$

$$\mathbf{V}_j = \mathbf{J}(a)\mathbf{Q}(t_j) = \begin{bmatrix} v_{j1} & v_{j2} & v_{j3} & \cdots & v_{j(N+1)^2} \end{bmatrix},$$

$$\bar{\mathbf{V}}_j = n_1 \mathbf{J}(a)\mathbf{P}\mathbf{Q}(t_j) + n_2 \mathbf{J}(a)\mathbf{Q}(t_j)\bar{\mathbf{P}} = \begin{bmatrix} \bar{v}_{j1} & \bar{v}_{j2} & \bar{v}_{j3} & \cdots & \bar{v}_{j(N+1)^2} \end{bmatrix}.$$

(3.39) denkleminin (3.40)-(3.43) koşulları altında yaklaşık çözümünü bulmak için, (3.57) denklemindeki $[\mathbf{W}; \mathbf{G}]$ artırılmış matrisine koşulların (3.58)-(3.61) artırılmış matris formlarının eklenmesiyle

$$[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F}_1 \\ \bar{\mathbf{U}} & ; & \mathbf{F}_2 \\ \mathbf{V} & ; & \mathbf{G}_1 \\ \bar{\mathbf{V}} & ; & \mathbf{G}_2 \\ \mathbf{W} & ; & \mathbf{G} \end{bmatrix} \quad (3.62)$$

yeni artırılmış matrisi elde edilir. Son olarak, bu sistem çözülerek bilinmeyen Bessel katsayılar matrisi

$$\mathbf{A} = \left(\tilde{\mathbf{W}} \right)^{-1} \tilde{\mathbf{G}} \quad (3.63)$$

olarak elde edilir. Burada $\left[\tilde{\mathbf{W}}; \tilde{\mathbf{G}} \right]$ artırılmış matrisi, $\left[\tilde{\mathbf{W}}; \tilde{\mathbf{G}} \right]$ artırılmış matrisine Gauss eliminasyon uygulandıktan sonra sıfır satırları atılarak elde edilen matristir. Matlab’ de (3.62) sisteminin çözümü ‘ $\mathbf{A} = \tilde{\mathbf{W}} \setminus \tilde{\mathbf{G}}$ ’ komutu ile kolayca bulunabilir.

Son olarak, belirlenen $a_{r,s}$; $(r, s = 0, \dots, N)$ katsayıları (3.6) denkleminde yerine yerleştirilerek yaklaşık çözüm elde edilir.

3.2.4. Hata tahmini

Bu kısımda, bölüm (3.2.1)’ de dalga denklem problemi için sunulan hata tahmini yöntemi Neuman, Dirichlet ve Robin koşullu genel formdaki ikinci mertebe lineer kısmi diferansiyel denklem problemi için kullanılacak.

Öncelikle, (3.39) denklemini

$$L[u(x, t)] = G(x, t), \quad a \leq x \leq b, \quad c \leq t \leq d$$

olarak yazalım. Burada, L : lineer operatör ve

$$L[u(x, t)] = A(x, t) \frac{\partial^2 u}{\partial x^2} + B(x, t) \frac{\partial^2 u}{\partial t \partial x} + C(x, t) \frac{\partial^2 u}{\partial t^2} + D(x, t) \frac{\partial u}{\partial x} + E(x, t) \frac{\partial u}{\partial t} + F(x, t) u.$$

$u_N(x, t)$: (3.39) denkleminin Bessel polinom çözümünü göstermek üzere, bu problem için bizim metodun rezidüel fonksiyonu

$$R_N(x, t) = L[u_N(x, t)] - G(x, t) \quad (3.64)$$

olur. Böylece, $u_N(x, t)$,

$$L[u_N(x, t)] = G(x, t) + R_N(x, t) \quad (3.65)$$

denklemini,

$$u_N(x, \phi) = f_1(x), \quad a \leq x \leq b, \quad \phi = c \text{ ya da } \phi = d, \quad (3.66)$$

$$u_N(\varphi, t) = g_1(t), \quad c \leq t \leq d, \quad \varphi = a \text{ ya da } \varphi = b, \quad (3.67)$$

Dirichlet sınır koşullarını,

$$\frac{\partial u_N}{\partial \mathbf{n}}(\varphi, t) = g_2(t), \quad c \leq t \leq d, \quad \varphi = a \text{ ya da } \varphi = b, \quad (3.68)$$

Neumann sınır koşulunu ve

$$\alpha u_N(x, \mu) + \beta \frac{\partial u_N}{\partial \mathbf{n}}(x, \eta) = f_2(x), \quad a \leq x \leq b, \quad \mu = c \text{ ya da } \mu = d, \quad \eta = c \text{ ya da } \eta = d \quad (3.69)$$

Robin sınır koşulunu sağlar.

$u(x, t)$: (3.39)-(3.43) probleminin tam çözümü olmak üzere,

$$e_N(x, t) = u(x, t) - u_N(x, t) \quad (3.70)$$

olarak hata fonksiyonunu tanımlayalım.

Böylece, (3.40)-(3.43) ve (3.66)-(3.69) homojen olmayan koşulları

$$\begin{cases} e_N(x, \phi) = 0, \quad \alpha e_N(x, \mu) + \beta \frac{\partial e_N}{\partial \mathbf{n}}(x, \eta) = 0, \quad a \leq x \leq b, \\ e_N(\varphi, t) = 0, \quad \frac{\partial e_N}{\partial \mathbf{n}}(\varphi, t) = 0, \quad c \leq t \leq d \end{cases} \quad (3.71)$$

homojen koşullarına indirgenir.

(3.39) ve (3.65) denklemlerinden,

$$L[e_N(x, t)] = L[u(x, t)] - L[u_N(x, t)] = -R_N(x, t)$$

hata diferansiyel denklemini elde ederiz. Açık olarak, hata diferansiyel denklemi

$$A(x, t) \frac{\partial^2 e_N}{\partial x^2} + B(x, t) \frac{\partial^2 e_N}{\partial t \partial x} + C(x, t) \frac{\partial^2 e_N}{\partial t^2} + D(x, t) \frac{\partial e_N}{\partial x} + E(x, t) \frac{\partial e_N}{\partial t} + F(x, t) e_N = -R_N(x, t) \quad (3.72)$$

olur.

Bölüm (3.2.3)' de sunulan yöntem ile (3.71) Dirichlet, Neumann ve Robin sınır koşulları altında (3.72) hata diferansiyel denklemi çözülerek, $e_N(x,t)$ hata fonksiyonu için $e_{N,M}(x,t)$ hata yaklaşımı elde edilir. Burada, M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını gösterir.

Hata probleminin Bessel polinom çözümü bulunurken, orijinal problemin çözümünü elde ederken kullandığımız matrislerde $G(x,t)$ yerine $-R_N(x,t)$ ve koşulların sağ taraflarını sıfır alarak hesaplanacağını not ediyoruz. Yani, orijinal problemin çözümü için kullanılan bilgisayar programı hata problemi içinde kullanılabilir.

Sonuç olarak, eğer problemin tam çözümü bilinmiyorsa, o zaman $e_{N,M}(x,t)$ yaklaşımı ile hata fonksiyonu için etkili bir tahmin yapılabilir.

3.2.5. Çözümlerin iyileştirilmesi

Bu kesimde, (3.39)-(3.43) Dirichlet, Neumann ve Robin sınır koşullu ikinci mertebe lineer kısmi diferansiyel denklem probleminin Bessel polinom çözümleri rezidüel hata fonksiyonu tahmini aracılığıyla iyileştirilecek. Bu iyileştirme yapılırken, ilk olarak, bölüm (3.2.4)' de sunulan hata fonksiyonu tahmini ile $e_N(x,t)$ hata fonksiyonu için $e_{N,M}(x,t)$ yaklaşımı elde edilir. Daha sonra, bu hata fonksiyonu yaklaşımı ile bölüm (3.2.3)' de elde edilen Bessel polinom çözümü toplanarak,

$$u_{N,M}(x,t) = u_N(x,t) + e_{N,M}(x,t)$$

iyileştirilmiş yaklaşık çözümü elde edilir. Burada, $e_{N,M}(x,t)$: hata probleminin Bessel polinom çözümü, N : orijinal problemin Bessel polinom çözümü elde edilirken kullanılan kesme sınırı ve M : hata fonksiyonu için Bessel polinom yaklaşımı elde edilirken kullanılan kesme sınırındır. Çözümler iyileştirilirken, $M > N$ seçilirse daha iyi çözümler elde edilebilir.

3.3. Başlangıç ve Sınır Koşulları Altında Genel Formda İkinci Mertebe Lineer Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu

Bu kesimde, $a \leq x \leq b$ ve $c \leq t \leq d$ olmak üzere,

$$A(x,t)\frac{\partial^2 u}{\partial x^2} + B(x,t)\frac{\partial^2 u}{\partial t \partial x} + C(x,t)\frac{\partial^2 u}{\partial t^2} + D(x,t)\frac{\partial u}{\partial x} + E(x,t)\frac{\partial u}{\partial t} + F(x,t)u = G(x,t), \quad (3.73)$$

ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklemini

$$u(x,c) = f_1(x), \quad a \leq x \leq b, \quad (3.74)$$

$$\frac{\partial u(x,c)}{\partial t} = f_2(x), \quad a \leq x \leq b, \quad (3.75)$$

başlangıç koşulları ve

$$u(a,t) = g_1(t), \quad c \leq t \leq d, \quad (3.76)$$

$$u(b,t) = g_2(t), \quad c \leq t \leq d \quad (3.77)$$

sınır koşulları ile ele alıyoruz. Burada, $(x,t) \in [a,b] \times [c,d]$ için $A, B, C, D, E, F, G, f_1, f_2, g_1$ ve g_2 tanımlı fonksiyonlar.

Bu problemin (3.6) formunda Bessel polinom çözümlerini elde etmek için Bessel sıralama metodunu uygulayacağız.

3.3.1. Başlangıç ve sınır koşulları için matris bağıntıları

Bu kesimde, (3.74)-(3.77) koşullarının matris formlarını oluşturacağız.

(3.74) koşulunun matris formu (3.49)' dan

$$u(x,c) = \mathbf{J}(x)\mathbf{Q}(c)\mathbf{A} = f_1(x) \quad (3.78)$$

olarak yazılır.

(3.75) koşulunun matris formu için (3.45)' de t yerine c yazılarak,

$$u_t(x, c) = \mathbf{J}(x)\mathbf{Q}(c)\bar{\mathbf{P}}\mathbf{A} = f_2(x) \quad (3.79)$$

elde edilir.

(3.76) ve (3.77) ile verilen sınır koşullarının matris formları için (3.7) denkleminde x yerine sırayla a ve b yazılarak,

$$u(a, t) = \mathbf{J}(a)\mathbf{Q}(t)\mathbf{A} = g_1(t) \quad (3.80)$$

ve

$$u(b, t) = \mathbf{J}(b)\mathbf{Q}(t)\mathbf{A} = g_2(t) \quad (3.81)$$

matris formlarını elde ederiz.

3.3.2. Problem için çözüm yöntemi

Bu kısımda, (3.73) ile verilen ikinci mertebe lineer kısmi diferansiyel denkleminin (3.74)-(3.77) sınır koşulları ile birlikte (3.6) formunda yaklaşık çözümünü elde etmek için yöntemi uygulayacağız.

(3.7) ve (3.44)-(3.48) matris formları (3.39) denkleminde yerine yazılarak ve (3.56) sıralama noktaları kullanılarak, genel formdaki ikinci mertebe lineer kısmi diferansiyel denklem için (3.57)' de

$$\mathbf{W}\mathbf{A} = \mathbf{G} \text{ ya da } [\mathbf{W}; \mathbf{G}] \quad (3.82)$$

sistemi elde edilmişti. Burada,

$$\mathbf{W} = [\mathbf{W}(x_0, t_0) \mathbf{W}(x_0, t_1) \cdots \mathbf{W}(x_0, t_N) \mathbf{W}(x_1, t_0) \mathbf{W}(x_1, t_1) \cdots \mathbf{W}(x_1, t_N) \cdots \mathbf{W}(x_N, t_0) \mathbf{W}(x_N, t_1) \cdots \mathbf{W}(x_N, t_N)]^T_{(N+1)^2 \times (N+1)^2},$$

$$\mathbf{A} = [a_{0,0} \ a_{0,1} \cdots a_{0,N} \ a_{1,0} \ a_{1,1} \cdots a_{1,N} \cdots a_{N,0} \ a_{N,1} \cdots a_{N,N}]^T_{(N+1)^2 \times 1},$$

$$\mathbf{G} = [G(x_0, t_0) G(x_0, t_1) \cdots G(x_0, t_N) G(x_1, t_0) G(x_1, t_1) \cdots G(x_1, t_N) \cdots G(x_N, t_0) G(x_N, t_1) \cdots G(x_N, t_N)]^T_{(N+1)^2 \times 1},$$

$$\begin{aligned}\mathbf{W}(x,t) = & A(x,t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t) + B(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}} + C(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 \\ & + D(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t) + E(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}} + F(x,t)\mathbf{J}(x)\mathbf{Q}(t).\end{aligned}$$

Şimdi, (3.74)-(3.77) sınır koşullarının temel matris formlarını oluşturmak için (3.56) sıralama noktalarını (3.78)-(3.81) matris denklemlerinde kullanarak,

$$u(x_i, c) = \mathbf{J}(x_i)\mathbf{Q}(c)\mathbf{A} = f_1(x_i), \quad i = 0, 1, \dots, N,$$

$$u_t(x_i, c) = \mathbf{J}(x_i)\mathbf{Q}(c)\bar{\mathbf{P}}\mathbf{A} = f_2(x_i), \quad i = 0, 1, \dots, N,$$

$$u(a, t_j) = \mathbf{J}(a)\mathbf{Q}(t_j)\mathbf{A} = g_1(t_j), \quad j = 0, 1, \dots, N,$$

$$u(b, t_j) = \mathbf{J}(b)\mathbf{Q}(t_j)\mathbf{A} = g_2(t_j), \quad j = 0, 1, \dots, N$$

sistemlerine sahip oluruz ve bu sistemleri sırasıyla

$$\mathbf{U}\mathbf{A} = [\mathbf{F}_1] \text{ ya da } [\mathbf{U}; \mathbf{F}_1], \quad (3.83)$$

$$\bar{\mathbf{U}}\mathbf{A} = [\mathbf{F}_2] \text{ ya da } [\bar{\mathbf{U}}; \mathbf{F}_2], \quad (3.84)$$

$$\mathbf{V}\mathbf{A} = [\mathbf{G}_1] \text{ ya da } [\mathbf{V}; \mathbf{G}_1], \quad (3.85)$$

$$\bar{\mathbf{V}}\mathbf{A} = [\mathbf{G}_2] \text{ ya da } [\bar{\mathbf{V}}; \mathbf{G}_2] \quad (3.86)$$

biçiminde matris formlarında kısaca ifade edebiliriz. Burada,

$$\mathbf{U} = [\mathbf{U}_0 \quad \mathbf{U}_1 \quad \dots \quad \mathbf{U}_N]^T, \quad \bar{\mathbf{U}} = [\bar{\mathbf{U}}_0 \quad \bar{\mathbf{U}}_1 \quad \dots \quad \bar{\mathbf{U}}_N]^T, \quad \mathbf{V} = [\mathbf{V}_0 \quad \mathbf{V}_1 \quad \dots \quad \mathbf{V}_N]^T,$$

$$\bar{\mathbf{V}} = [\bar{\mathbf{V}}_0 \quad \bar{\mathbf{V}}_1 \quad \dots \quad \bar{\mathbf{V}}_N]^T, \quad \mathbf{F}_1 = [f_1(x_0) \quad f_1(x_1) \quad \dots \quad f_1(x_N)]^T,$$

$$\mathbf{F}_2 = [f_2(x_0) \quad f_2(x_1) \quad \dots \quad f_2(x_N)]^T, \quad \mathbf{G}_1 = [g_1(t_0) \quad g_1(t_1) \quad \dots \quad g_1(t_N)]^T,$$

$$\mathbf{G}_2 = [g_2(t_0) \quad g_2(t_1) \quad \dots \quad g_2(t_N)]^T, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

$$\mathbf{U}_i = \mathbf{J}(x_i)\mathbf{Q}(c) = \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} & \dots & u_{i(N+1)^2} \end{bmatrix},$$

$$\bar{\mathbf{U}}_i = \mathbf{J}(x_i)\mathbf{Q}(c)\bar{\mathbf{P}} = \begin{bmatrix} \bar{u}_{i1} & \bar{u}_{i2} & \bar{u}_{i3} & \dots & \bar{u}_{i(N+1)^2} \end{bmatrix},$$

$$\mathbf{V}_j = \mathbf{J}(a)\mathbf{Q}(t_j) = \begin{bmatrix} v_{j1} & v_{j2} & v_{j3} & \dots & v_{j(N+1)^2} \end{bmatrix},$$

$$\bar{\mathbf{V}}_j = \mathbf{J}(b)\mathbf{Q}(t_j) = \begin{bmatrix} \bar{v}_{j1} & \bar{v}_{j2} & \bar{v}_{j3} & \dots & \bar{v}_{j(N+1)^2} \end{bmatrix}.$$

(3.83)-(3.86) denklemleri (3.74)-(3.77) koşullarına karşılık gelen temel matris denklemleridir.

(3.73) denkleminin (3.74)-(3.77) koşulları altında Bessel polinom çözümünü bulmak için, (3.82) denklemindeki $[\mathbf{W};\mathbf{G}]$ artırılmış matrisine koşulların (3.83)-(3.86) artırılmış matris formlarının eklenmesiyle

$$[\tilde{\mathbf{W}};\tilde{\mathbf{G}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F}_1 \\ \bar{\mathbf{U}} & ; & \mathbf{F}_2 \\ \mathbf{V} & ; & \mathbf{G}_1 \\ \bar{\mathbf{V}} & ; & \mathbf{G}_2 \\ \mathbf{W} & ; & \mathbf{G} \end{bmatrix}$$

yeni artırılmış matrisi elde edilir. Önceki bölümlerdeki gibi, bu sistem çözülerek bilinmeyen Bessel katsayıları hesaplanır ve belirlenen $a_{r,s}$; ($r,s = 0,...,N$) katsayıları (3.6) denkleminde yerine yerleştirilerek yaklaşık çözüm elde edilir.

Başlangıç ve sınır koşullu kısmi diferansiyel denklem problemleri (3.75) koşulu yerine bazen

$$u(x,d) = f_2(x), \quad a \leq x \leq b \quad (3.87)$$

başlangıç koşulu ile ele alınır.

Bu durumda yapılması gereken sadece (3.75) koşulunun matris formu olan (3.79) yerine (3.87) koşulunun matris formu alınması gerekir. (3.87) koşulunu matris formu (3.78) matris formunda $c \rightarrow d$ yazılarak

$$u(x,d) = \mathbf{J}(x)\mathbf{Q}(d)\mathbf{A} = f_2(x)$$

olarak bulunur.

Benzer şekilde, prosedür uygulanarak yaklaşık çözüm elde edilir.

3.3.3. Hata tahmini

Bu kısımda, önceki bölümlerdeki problemler için sunulan hata tahmini yöntemini başlangıç ve sınır koşullu genel formdaki ikinci mertebe lineer kısmi diferansiyel denklem problemi için uygulayacağız.

$u_N(x, t)$: (3.73) denkleminin Bessel polinom çözümü olmak üzere, (3.73) formundaki lineer ikinci mertebe kısmi diferansiyel denklemin bizim metot için rezidüel fonksiyonunun

$$R_N(x, t) = L[u_N(x, t)] - G(x, t)$$

olduğunu (3.64)' de söylemiştik. Burada, L : lineer operatör ve

$$L[u(x, t)] = A(x, t) \frac{\partial^2 u}{\partial x^2} + B(x, t) \frac{\partial^2 u}{\partial t \partial x} + C(x, t) \frac{\partial^2 u}{\partial t^2} + D(x, t) \frac{\partial u}{\partial x} + E(x, t) \frac{\partial u}{\partial t} + F(x, t)u .$$

O zaman, $u_N(x, t)$,

$$L[u_N(x, t)] = G(x, t) + R_N(x, t) \quad (3.88)$$

denklemini,

$$u_N(x, c) = f_1(x), \quad a \leq x \leq b, \quad (3.89)$$

$$\frac{\partial u_N(x, c)}{\partial t} = f_2(x), \quad a \leq x \leq b, \quad (3.90)$$

başlangıç koşullarını ve

$$u_N(a, t) = g_1(t), \quad c \leq t \leq d, \quad (3.91)$$

$$u_N(b, t) = g_2(t), \quad c \leq t \leq d \quad (3.92)$$

sınır koşullarını sağlar.

$u(x, t) : (3.73)-(3.77)$ probleminin tam çözümü olmak üzere,

$$e_N(x, t) = u(x, t) - u_N(x, t) \quad (3.93)$$

hata fonksiyonunu gösterebilir.

O zaman, (3.74)-(3.77) ve (3.89)-(3.92) homojen olmayan başlangıç ve sınır koşulları

$$\begin{cases} e_N(x, c) = 0, \quad \frac{\partial e_N(x, c)}{\partial t} = 0, \quad a \leq x \leq b, \\ e_N(a, t) = 0, \quad e_N(b, t) = 0, \quad c \leq t \leq d \end{cases} \quad (3.94)$$

homojen koşullarına indirgenir.

(3.73) ve (3.88) denklemlerinden,

$$L[e_N(x, t)] = L[u(x, t)] - L[u_N(x, t)] = -R_N(x, t)$$

hata diferansiyel denklemi elde edilir. Açık olarak, hata diferansiyel denklemi

$$A(x, t) \frac{\partial^2 e_N}{\partial x^2} + B(x, t) \frac{\partial^2 e_N}{\partial t \partial x} + C(x, t) \frac{\partial^2 e_N}{\partial t^2} + D(x, t) \frac{\partial e_N}{\partial x} + E(x, t) \frac{\partial e_N}{\partial t} + F(x, t) e_N = -R_N(x, t) \quad (3.95)$$

olur. Böylece, (3.94) başlangıç ve sınır koşulları altında (3.95) hata diferansiyel denklemi (3.3.2)' de sunulan prosedür takip edilerek hata probleminin $e_{N,M}(x, t)$

Bessel polinom çözümü elde edilir. Bu Bessel polinom çözümü $e_N(x, t)$ hata fonksiyonuna bir yaklaşımdır. Burada, M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını gösterir.

Hata probleminin Bessel polinom çözümü için yöntem uygulanırken, orijinal problemin çözümünü elde ederken kullandığımız matrislerde sadece farklı olarak $G(x, t)$ yerine $-R_N(x, t)$ ve koşulların sağ taraflarını sıfır almamız yeterlidir. Yani, orijinal problemin çözümü için kullanılan bilgisayar programı hata problemi içinde kullanılabilir.

3.3.4. Çözümlerin iyileştirilmesi

Bu kesimde, (3.73)-(3.77) başlangıç ve sınır koşullu ikinci mertebe lineer kısmi diferansiyel denklem probleminin Bessel polinom çözümleri rezidüel hata fonksiyonu tahmini aracılığıyla iyileştirilecek. Önceki bölümlerde olduğu gibi, bu iyileştirme orijinal problemin Bessel polinom çözümü ile kesim (3.3.3)' de elde edilen hata fonksiyonunun yaklaşımı olan $e_{N,M}(x,t)$ ile toplanarak,

$$u_{N,M}(x,t) = u_N(x,t) + e_{N,M}(x,t)$$

şeklinde iyileştirilmiş Bessel polinom çözümü elde edilir. Burada, N : orijinal problemin Bessel polinom çözümü elde edilirken kullanılan kesme sınırı ve M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını temsil eder. Çözümler iyileştirilirken, $M > N$ seçilmesi durumunda daha iyi çözümler elde edilebileceğini not ediyoruz.

3.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Probleminin Yaklaşık Çözümü için Bessel Sıralama Metodu

Bu kesimde,

$$\frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + a(x) \frac{\partial u}{\partial x} + b(x)u = G(x,t), \quad 0 \leq x \leq L, \quad 0 \leq t \leq T \quad (3.96)$$

singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon denklemini

$$u(x,0) = f(x), \quad 0 \leq x \leq L < \infty, \quad (3.97)$$

başlangıç koşulu ve

$$u(0,t) = h(t), \quad 0 \leq t \leq T < \infty, \quad (3.98)$$

$$u(L,t) = s(t), \quad 0 \leq t \leq T < \infty \quad (3.99)$$

sınır koşulları ile ele alıyoruz (Lenferink, 2002; Kadalbajoo ve Gupta, 2009).

$\varepsilon : 0 < \varepsilon \ll 1$ olacak şekilde bir difüzyon katsayısı ya da singüler pertürbasyon parametresi, $a(x)$: hız, $G(x,t) - b(x)u$: reaksiyon terimi, L ve T : uygun sabitler ve $(x,t) \in [0,L] \times [0,T]$ için a, b, G, f, h ve s tanımlı fonksiyonlardır.

Bu problemin (3.6) formunda Bessel polinom çözümlerini elde etmek için Bessel sıralama metodunu uygulayacağız.

3.4.1. Başlangıç ve sınır koşulları için matris bağıntıları

Bu kesimde, (3.97)-(3.99) koşullarının matris formlarını oluşturacağız.

(3.97) koşulunun matris formu için (3.78) denkleminde c yerine 0 ve $f_1(x)$ yerine $f(x)$ yazarak,

$$u(x,0) = \mathbf{J}(x)\mathbf{Q}(0)\mathbf{A} = f(x) \quad (3.100)$$

matris ilişkisi elde edilir.

Benzer şekilde, (3.80) ve (3.81) denklemlerinde, a yerine 0, b yerine L , $g_1(t)$ yerine $h(t)$ ve $g_2(t)$ yerine $s(t)$ yazarak,

$$u(0,t) = \mathbf{J}(0)\mathbf{Q}(t)\mathbf{A} = h(t) \quad (3.101)$$

ve

$$u(L,t) = \mathbf{J}(L)\mathbf{Q}(t)\mathbf{A} = s(t) \quad (3.102)$$

matris formlarına sahip oluruz.

3.4.2. Problem için çözüm yöntemi

Önceki bölümlerde olduğu gibi, (3.96) denkleminin karşılık gelen matris denklemini bulalım. Bu amaç için, çözüm fonksiyonu ve kısmi türevlerin matris formları olan (3.7), (3.19), (3.44) ve (3.45) matris formlarını (3.96) denkleminde yerine koyarak,

$$\left\{ \mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}} - \varepsilon \mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t) + a(x)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t) + b(x)\mathbf{J}(x)\mathbf{Q}(t) \right\} \mathbf{A} = G(x,t)$$

matris denklemini elde ederiz. Kısaca, bu matris denklemi

$$\mathbf{W}(x,t)\mathbf{A}=G(x,t) \quad (3.103)$$

olarak yazılabilir. Burada,

$$\begin{aligned} \mathbf{W}(x,t) = & A(x,t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t) + B(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}} + C(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 \\ & + D(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t) + E(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}} + F(x,t)\mathbf{J}(x)\mathbf{Q}(t) \end{aligned}$$

ve

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N} & a_{1,0} & a_{1,1} & \cdots & a_{1,N} & \cdots & a_{N,0} & a_{N,1} & \cdots & a_{N,N} \end{bmatrix}_{(N+1)^2 \times 1}^T.$$

Şimdi, (3.23) sıralama noktalarını (3.103) matris denkleminde kullanarak,

$$\mathbf{W}(x_i, t_j)\mathbf{A} = G(x_i, t_j), \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N$$

biçiminde matris denklemlerin bir sistemini elde ederiz ve bu sistem

$$\mathbf{W}\mathbf{A}=\mathbf{G} \text{ ya da } [\mathbf{W};\mathbf{G}] \quad (3.104)$$

olarak kısaca yazılabilir. Burada,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}(x_0, t_0) & \mathbf{W}(x_0, t_1) & \cdots & \mathbf{W}(x_0, t_N) & \mathbf{W}(x_1, t_0) & \mathbf{W}(x_1, t_1) & \cdots & \mathbf{W}(x_1, t_N) & \cdots & \mathbf{W}(x_N, t_0) & \mathbf{W}(x_N, t_1) & \cdots & \mathbf{W}(x_N, t_N) \end{bmatrix}_{(N+1)^2 \times (N+1)^2}^T,$$

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N} & a_{1,0} & a_{1,1} & \cdots & a_{1,N} & \cdots & a_{N,0} & a_{N,1} & \cdots & a_{N,N} \end{bmatrix}_{(N+1)^2 \times 1}^T,$$

$$\mathbf{G} = \begin{bmatrix} G(x_0, t_0) & G(x_0, t_1) & \cdots & G(x_0, t_N) & G(x_1, t_0) & G(x_1, t_1) & \cdots & G(x_1, t_N) & \cdots & G(x_N, t_0) & G(x_N, t_1) & \cdots & G(x_N, t_N) \end{bmatrix}_{(N+1)^2 \times 1}^T.$$

(3.23) sıralama noktalarını koşulların matris formları olan (3.100)-(3.102) denklemlerinde yerleştirerek,

$$u(x_i, 0) = \mathbf{J}(x_i)\mathbf{Q}(0)\mathbf{A} = f(x_i), \quad i = 0, 1, \dots, N,$$

$$u(0, t_j) = \mathbf{J}(0)\mathbf{Q}(t_j)\mathbf{A} = h(t_j), \quad j = 0, 1, \dots, N,$$

ve

$$u(L, t_j) = \mathbf{J}(L)\mathbf{Q}(t_j)\mathbf{A} = s(t_j), \quad j = 0, 1, \dots, N$$

matris denklemlerin sistemleri elde edilir ve bu sistemler sırasıyla

$$\mathbf{U}\mathbf{A} = [\mathbf{F}] \text{ ya da } [\mathbf{U}; \mathbf{F}], \quad (3.105)$$

$$\bar{\mathbf{U}}\mathbf{A} = [\mathbf{H}] \text{ ya da } [\bar{\mathbf{U}}; \mathbf{H}] \quad (3.106)$$

ve

$$\mathbf{V}\mathbf{A} = [\mathbf{S}] \text{ ya da } [\mathbf{V}; \mathbf{S}] \quad (3.107)$$

şeklinde kapalı olarak matris formlarında yazılabilir. Burada,

$$\mathbf{U} = [\mathbf{U}_0 \quad \mathbf{U}_1 \quad \dots \quad \mathbf{U}_N]^T, \quad \bar{\mathbf{U}} = [\bar{\mathbf{U}}_0 \quad \bar{\mathbf{U}}_1 \quad \dots \quad \bar{\mathbf{U}}_N]^T, \quad \mathbf{V} = [\mathbf{V}_0 \quad \mathbf{V}_1 \quad \dots \quad \mathbf{V}_N]^T,$$

$$\mathbf{F} = [f(x_0) \quad f(x_1) \quad \dots \quad f(x_N)]^T, \quad \mathbf{H} = [h(x_0) \quad h(x_1) \quad \dots \quad h(x_N)]^T,$$

$$\mathbf{S} = [s(t_0) \quad s(t_1) \quad \dots \quad s(t_N)]^T, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

$$\mathbf{U}_i = \mathbf{J}(x_i)\mathbf{Q}(0) = [u_{i1} \quad u_{i2} \quad u_{i3} \quad \dots \quad u_{i(N+1)^2}]$$

$$\bar{\mathbf{U}}_i = \mathbf{J}(0)\mathbf{Q}(t_j) = [\bar{u}_{i1} \quad \bar{u}_{i2} \quad \bar{u}_{i3} \quad \dots \quad \bar{u}_{i(N+1)^2}],$$

$$\mathbf{V}_j = \mathbf{J}(L)\mathbf{Q}(t_j) = [v_{j1} \quad v_{j2} \quad v_{j3} \quad \dots \quad v_{j(N+1)^2}].$$

(3.105)-(3.107) denklemleri (3.97)-(3.99) koşullarına karşılık gelen temel matris denklemleridir.

(3.96) denkleminin (3.97)-(3.99) başlangıç ve sınır koşulları altında Bessel polinom çözümünü elde etmek için, (3.104) denklemindeki $[\mathbf{W}; \mathbf{G}]$ artırılmış matrisine koşulların (3.105)-(3.107) artırılmış matris formlarının eklenmesiyle

$$[\tilde{\mathbf{W}}, \tilde{\mathbf{G}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F} \\ \bar{\mathbf{U}} & ; & \mathbf{H} \\ \mathbf{V} & ; & \mathbf{S} \\ \mathbf{W} & ; & \mathbf{G} \end{bmatrix}$$

yeni artırılmış matrisi elde edilir. Önceki bölümlerdeki gibi, bu sistem çözülerek bilinmeyen Bessel katsayıları hesaplanır ve belirlenen $a_{r,s}$; ($r, s = 0, \dots, N$) katsayıları (3.6) denkleminde yerine yerleştirilerek Bessel polinom çözümü elde edilir.

3.4.3. Hata tahmini

Bu kesimde, singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon problemi için önceki bölümlerde kullanılan rezidüel hata tahmini verilecektir.

İlk olarak, (3.96) denklemini

$$L[u(x, t)] = G(x, t), \quad 0 \leq x \leq L, \quad 0 \leq t \leq T \quad (3.108)$$

olarak yazalım. Burada, L : lineer operatör ve

$$L[u(x, t)] = \frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + a(x) \frac{\partial u}{\partial x} + b(x)u.$$

$u_N(x, t)$: (3.96) denkleminin Bessel polinom çözümünü göstermek üzere, bu problem için bizim metodun rezidüel fonksiyonu

$$R_N(x, t) = L[u_N(x, t)] - G(x, t)$$

olur. O zaman, $u_N(x, t)$,

$$L[u_N(x, t)] = G(x, t) + R_N(x, t) \quad (3.109)$$

denklemini,

$$u_N(x, 0) = f(x), \quad 0 \leq x \leq L, \quad (3.110)$$

başlangıç koşulunu ve

$$u_N(0,t) = h(t), \quad 0 \leq t \leq T, \quad (3.111)$$

$$u_N(L,t) = s(t), \quad 0 \leq t \leq T \quad (3.112)$$

sınır koşullarını sağlar.

Şimdi,

$$e_N(x,t) = u(x,t) - u_N(x,t)$$

olarak hata fonksiyonunu tanımlayalım. Burada, $u(x,t)$: (3.96)-(3.99) probleminin tam çözümüdür. Böylece, (3.97)-(3.99) ve (3.110)-(3.112) homojen olmayan başlangıç ve sınır koşullarından $e_N(x,t)$ hata fonksiyonu için

$$e_N(x,0) = 0, \quad 0 \leq x \leq L, \quad (3.113)$$

homojen başlangıç koşulunu ve

$$e_N(0,t) = 0, \quad 0 \leq t \leq T, \quad (3.114)$$

$$e_N(L,t) = 0, \quad 0 \leq t \leq T \quad (3.115)$$

homojen sınır koşullarını elde ederiz.

(3.96) ve (3.109) denklemlerinden

$$L[e_N(x,t)] = L[u(x,t)] - L[u_N(x,t)] = -R_N(x,t)$$

hata diferansiyel denklemini elde ederiz. Bu hata diferansiyel denklemi

$$\frac{\partial e_N}{\partial t} - \varepsilon \frac{\partial^2 e_N}{\partial x^2} + a(x) \frac{\partial e_N}{\partial x} + b(x) e_N = -R_N(x,t) \quad (3.116)$$

şeklinde açık bir şekilde yazılır.

(3.113)-(3.115) homojen başlangıç ve sınır koşulları ile birlikte (3.116) hata diferansiyel denklemi önceki bölümde sunulan teknik kullanılarak çözülerek hata

problemi için bir yaklaşım olan $e_{N,M}(x,t)$ Bessel polinom çözümü elde edilir. Burada, M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırınıdır.

3.4.4. Çözümlerin iyileştirilmesi

Bu kesimde, (3.96)-(3.99) singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon probleminin Bessel polinom çözümleri önceki bölümlerde olduğu gibi reziduel hata yaklaşımı kullanılarak iyileştirilecektir.

Bu iyileştirme, singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon probleminin Bessel polinom çözümü ile kesim (3.4.3)' de elde edilen hata probleminin Bessel polinom çözümü olan $e_{N,M}(x,t)$ toplanarak,

$$u_{N,M}(x,t) = u_N(x,t) + e_{N,M}(x,t)$$

iyileştirilmiş Bessel polinom çözümünü elde ederiz. Burada, N : singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırı ve M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını gösterir. Çözümler iyileştirilirken, $M > N$ seçilerek daha iyi çözümler elde edilebilir.

3.5. İki boyutlu İkinci Mertebeden Lineer Değişken Katsayılı Kısmi Diferansiyel Denklemlerin Yaklaşık Çözümleri için Bessel Sıralama Metodu

Bu kısımda,

$$\left\{ \begin{aligned} & B_1(x,y,t) \frac{\partial^2 u(x,y,t)}{\partial x^2} + B_2(x,y,t) \frac{\partial^2 u(x,y,t)}{\partial y \partial x} + B_3(x,y,t) \frac{\partial^2 u(x,y,t)}{\partial y^2} \\ & + B_4(x,y,t) \frac{\partial^2 u(x,y,t)}{\partial t \partial x} + B_5(x,y,t) \frac{\partial^2 u(x,y,t)}{\partial t^2} + B_6(x,y,t) \frac{\partial u(x,y,t)}{\partial x} \\ & + B_7(x,y,t) \frac{\partial u(x,y,t)}{\partial y} + B_8(x,y,t) \frac{\partial u(x,y,t)}{\partial t} + B_9(x,t) u(x,y,t) = G(x,y,t), \end{aligned} \right. \quad (3.117)$$

iki boyutlu ikinci mertebeden lineer deęişken katsayılı lineer kısmi diferansiyel denkleminin

$$u(x, y, 0) = f_1(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.118)$$

$$u(x, y, T) = f_2(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.119)$$

$$u(x, c, t) = g_1(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.120)$$

$$u(x, d, t) = g_2(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.121)$$

$$u(a, y, t) = h_1(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.122)$$

ve

$$u(b, y, t) = h_2(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T, \quad (3.123)$$

sınır koşulları altında Bessel sıralama yöntemi uygulanarak Bessel polinom çözümleri elde edilecektir. Burada, $(x, y, t) \in [a, b] \times [c, d] \times [0, T]$ için B_i ($i = 1, 2, \dots, 9$), G , f_1 , f_2 , g_1 , g_2 , h_1 ve h_2 tanımlı fonksiyonlardır.

Bu problem için, önerilen yöntem uygulanarak,

$$u(x, y, t) = \sum_{r=0}^N \sum_{s=0}^N \sum_{k=0}^N a_{r,s,k} J_{r,s,k}(x, y, t) \quad (3.124)$$

kesilmiş Bessel seri formunda yaklaşık çözümler bulacağız. Burada, $a_{r,s,k}$; $r, s, k = 0, 1, \dots, N$ ' ler bilinmeyen Bessel katsayıları,

$$J_{r,s,k}(x, y, t) = J_r(x) J_s(y) J_k(t)$$

ve $J_n(x)$, ($n = 0, 1, \dots, N$) ' ler

$$J_n(x) = \sum_{k=0}^{\left\lfloor \frac{N-n}{2} \right\rfloor} \frac{(-1)^k}{k!(k+n)!} \left(\frac{x}{2} \right)^{2k+n}, \quad n \in \mathbb{N}, \quad 0 \leq x \leq \infty$$

ile tanımlı birinci tip Bessel fonksiyonlarıdır. Bülbul (2011) sabit katsayılı ikinci mertebeden iki boyutlu kısmi diferansiyel denklemlerin Taylor polinom çözümleri üzerine çalışmıştır.

Bu çalışmada, biz sabit katsayılı ve değişken katsayılı ikinci mertebeden kısmi diferansiyel denklemlerin Bessel polinom çözümlerini elde etmek için yöntemi uygulayacağız.

3.5.1. İki boyutlu kısmi diferansiyel denklemler için temel matris bağıntıları

Bu kısımda, (3.117)-(3.123) probleminin (3.124) formunda yaklaşık çözümlerini elde etmek için ana matris denklemlerini oluşturacağız.

Bu amaç için, ilk olarak, (3.124) çözüm fonksiyonunu

$$u(x, y, t) = \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A} \quad (3.125)$$

olarak matris formunda yazalım. Burada,

$$\mathbf{J}(x) = [J_0(x) \ J_1(x) \ \cdots \ J_N(x)]_{1 \times (N+1)},$$

$$\mathbf{Q}(y) = \begin{bmatrix} \mathbf{J}(y) & 0 & \cdots & 0 \\ 0 & \mathbf{J}(y) & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{J}(y) \end{bmatrix}_{(N+1) \times (N+1)^2}, \quad \mathbf{R}(t) = \begin{bmatrix} \mathbf{Q}(t) & 0 & \cdots & 0 \\ 0 & \mathbf{Q}(t) & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{Q}(t) \end{bmatrix}_{(N+1)^2 \times (N+1)^3},$$

$$\mathbf{A} = [a_{0,0,0} \ a_{0,0,1} \ \cdots \ a_{0,0,N} \ a_{0,1,0} \ a_{0,1,1} \ \cdots \ a_{0,1,N} \ \cdots \ a_{0,N,0} \ a_{0,N,1} \ \cdots \ a_{0,N,N} \ \cdots \ a_{N,N,0} \ \cdots \ a_{N,N,N}]^T.$$

Denklem (3.8)'de, $J_n(x)$ ($n = 0, 1, \dots, N$) birinci tip Bessel fonksiyonları

$$\mathbf{J}(x) = \mathbf{X}(x)\mathbf{D}^T \quad (3.126)$$

olarak matris formunda yazılmıştı. Burada,

$$\mathbf{X}(x) = [1 \ x \ x^2 \ \cdots \ x^N]$$

ve N tek ise,

$$\mathbf{D} = \begin{bmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & \frac{(-1)^{\frac{N-1}{2}}}{\left(\frac{N-1}{2}\right)!\left(\frac{N-1}{2}\right)!2^{N-1}} & 0 \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{\left(\frac{N-1}{2}\right)!\left(\frac{N+1}{2}\right)!2^N} \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & \frac{(-1)^{\frac{N-3}{2}}}{\left(\frac{N-3}{2}\right)!\left(\frac{N+1}{2}\right)!2^{N-1}} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{0!(N-1)!2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{0!N!2^N} \end{bmatrix}_{(N+1) \times (N+1)},$$

N çift ise,

$$\mathbf{D} = \begin{bmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N}{2}}}{\left(\frac{N}{2}\right)!\left(\frac{N}{2}\right)!2^N} \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & \frac{(-1)^{\frac{N-2}{2}}}{\left(\frac{N-2}{2}\right)!\left(\frac{N}{2}\right)!2^{N-1}} & 0 \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N-2}{2}}}{\left(\frac{N-2}{2}\right)!\left(\frac{N+2}{2}\right)!2^N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{0!(N-1)!2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{0!N!2^N} \end{bmatrix}_{(N+1) \times (N+1)}$$

(3.12) denkleminde,

$$\mathbf{J}^{(1)}(x) = \mathbf{J}(x)\mathbf{P} \text{ ve } \mathbf{J}^{(2)}(x) = \mathbf{J}(x)\mathbf{P}^2 \quad (3.127)$$

matris formları oluşturulmuştu. Burada,

$$\mathbf{P}^k = (\mathbf{D}^T)^{-1} \mathbf{C}^k \mathbf{D}^T, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & N \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad k=1,2.$$

Ayrıca, (3.13) denkleminde $\mathbf{Q}(y)$ ve onun $\mathbf{Q}^{(1)}(y)$ ve $\mathbf{Q}^{(2)}(y)$ türevleri arasında ilişki

$$\mathbf{Q}^{(1)}(y) = \mathbf{Q}(y) \bar{\mathbf{P}} \text{ ve } \mathbf{Q}^{(2)}(y) = \mathbf{Q}(y) \bar{\mathbf{P}}^2 \quad (3.128)$$

olarak verilmişti. Burada,

$$\bar{\mathbf{P}} = \begin{bmatrix} \mathbf{P} & 0 & \cdots & 0 \\ 0 & \mathbf{P} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{P} \end{bmatrix}_{(N+1)^2 \times (N+1)^2}.$$

Benzer şekilde, $\mathbf{R}(t)$ ve onun $\mathbf{R}^{(1)}(t)$ ve $\mathbf{R}^{(2)}(t)$ türevleri arasında ilişki

$$\mathbf{R}^{(1)}(t) = \mathbf{R}(t) \bar{\bar{\mathbf{P}}} \text{ ve } \mathbf{R}^{(2)}(t) = \mathbf{R}(t) \bar{\bar{\mathbf{P}}}^2 \quad (3.129)$$

olarak oluşturulur. Burada,

$$\mathbf{R}(t) = \begin{bmatrix} \mathbf{Q}(t) & 0 & \cdots & 0 \\ 0 & \mathbf{Q}(t) & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{Q}(t) \end{bmatrix}_{(N+1)^2 \times (N+1)^3}, \quad \bar{\bar{\mathbf{P}}} = \begin{bmatrix} \bar{\mathbf{P}} & 0 & \cdots & 0 \\ 0 & \bar{\mathbf{P}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{\mathbf{P}} \end{bmatrix}_{(N+1)^3 \times (N+1)^3}.$$

3.5.2. Sınır koşulları için matris bağıntıları

Bu kısımda, (3.118)-(3.123) sınır koşullarının Bessel polinomlarını temel alan matris formlarını oluşturacağız. Öncelikle, (3.118) başlangıç koşulunun matris formunu bulmak için, (3.124) çözüm fonksiyonunun (3.125) matris formunda $t \rightarrow 0$ konularak,

$$u(x, y, 0) = \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(0)\mathbf{A} = f_1(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.130)$$

matris formunu elde ederiz.

(3.124) çözüm fonksiyonunun (3.125) matris formunda $t \rightarrow T$ yazılarak,

$$u(x, y, T) = \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(T)\mathbf{A} = f_2(x, y) \quad (3.131)$$

olarak (3.126) sınır koşulunun matris formu elde edilir.

(3.125) matris formunda $y \rightarrow c$ ve $y \rightarrow d$ yerleştirilerek,

$$u(x, c, t) = \mathbf{J}(x)\mathbf{Q}(c)\mathbf{R}(t)\mathbf{A} = g_1(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T \quad (3.132)$$

ve

$$u(x, d, t) = \mathbf{J}(x)\mathbf{Q}(d)\mathbf{R}(t)\mathbf{A} = g_2(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T \quad (3.133)$$

olarak (3.120) ve (3.121) sınır koşullarının matris formlarını oluştururuz.

Benzer şekilde, (3.125) matris formunda $x \rightarrow a$ ve $x \rightarrow b$ yazarak,

$$u(a, y, t) = \mathbf{J}(a)\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A} = h_1(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.134)$$

ve

$$u(b, y, t) = \mathbf{J}(b)\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A} = h_2(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.135)$$

şeklinde (3.122) ve (3.123) sınır koşullarının matris formlarına sahip oluruz.

3.5.3. İki boyutlu kısmi diferansiyel denklem problemi için çözüm yöntemi

İlk olarak, (3.117) iki boyutlu ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denkleminin her bir teriminin matris formunu bulacağız. (3.127) denkleminde, (3.125) denkleminin x' e göre birinci mertebe kısmi türevinin matris formu

$$u_x(x, y, t) = \mathbf{J}^{(1)}(x)\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A},$$

$$= \mathbf{J}(x)\mathbf{PQ}(y)\mathbf{R}(t)\mathbf{A} \quad (3.136)$$

elde edilir.

Benzer şekilde, (3.125) denkleminin y' e göre birinci mertebe kısmi türevinin matris formunda (3.128) ilişkisi kullanılarak,

$$\begin{aligned} u_y(x, y, t) &= \mathbf{J}(x)\mathbf{Q}^{(1)}(y)\mathbf{R}(t)\mathbf{A} \\ &= \mathbf{J}(x)\mathbf{Q}(y)\bar{\mathbf{P}}\mathbf{R}(t)\mathbf{A} \end{aligned} \quad (3.137)$$

matris formuna sahip oluruz.

(3.129) denklemi kullanılarak, (3.125) denkleminin t' ye göre birinci mertebe kısmi türevinin matris formu

$$\begin{aligned} u_t(x, y, t) &= \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}^{(1)}(t)\mathbf{A} \\ &= \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}}\mathbf{A} \end{aligned} \quad (3.138)$$

olur.

Şimdi, (3.124) çözüm fonksiyonunun (3.125) matris formunun x' e göre ikinci mertebe kısmi türevinde (3.127) yerleştirilerek,

$$\begin{aligned} u_{xx}(x, y, t) &= \mathbf{J}^{(2)}(x)\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A} \\ &= \mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(y)\mathbf{R}(t)\mathbf{A} \end{aligned} \quad (3.139)$$

matris formunu elde ederiz.

(3.136) denkleminin t' ye göre birinci mertebe kısmi türevi

$$u_{xt}(x, y, t) = \mathbf{J}(x)\mathbf{PQ}(y)\mathbf{R}^{(1)}(t)\mathbf{A} \quad (3.140)$$

dır ve (3.140) denkleminde (3.129) yerine konularak

$$u_{xt}(x, y, t) = \mathbf{J}(x)\mathbf{PQ}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}}\mathbf{A} \quad (3.141)$$

matris ilişkisine sahip oluruz.

Benzer şekilde, (3.136) denkleminin y' ye göre birinci mertebe kısmi türevi

$$u_{xy}(x, y, t) = \mathbf{J}(x)\mathbf{PQ}^{(1)}(y)\mathbf{R}(t)\mathbf{A}$$

olur ve (3.128) yerine yerleştirilerek

$$u_{xy}(x, y, t) = \mathbf{J}(x)\mathbf{PQ}(y)\bar{\mathbf{P}}\mathbf{R}(t)\mathbf{A} \quad (3.142)$$

olarak daha açık bir şekilde yazılır.

(3.128) ilişkisi kullanılarak, (3.125) denkleminin y' e göre ikinci mertebe kısmi türevi

$$u_{yy}(x, y, t) = \mathbf{J}(x)\mathbf{Q}(y)\bar{\mathbf{P}}^2\mathbf{R}(t)\mathbf{A} \quad (3.143)$$

olarak elde edilir.

(3.129) denklemi aracılığıyla, (3.125) denkleminin t' ye göre ikinci mertebe kısmi türevinin matris formu

$$u_{tt}(x, y, t) = \mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}}^2\mathbf{A} \quad (3.144)$$

olur.

(3.125) ve (3.136)-(3.139) ve (3.141)-(3.144) matris formları (3.117) denkleminde yerine yerleştirilerek,

$$\left\{ \begin{aligned} &B_1(x, y, t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(y)\mathbf{R}(t) + B_2(x, y, t)\mathbf{J}(x)\mathbf{PQ}(y)\bar{\mathbf{P}}\mathbf{R}(t) + B_3(x, y, t)\mathbf{J}(x)\mathbf{Q}(y)\bar{\mathbf{P}}^2\mathbf{R}(t) \\ &+ B_4(x, y, t)\mathbf{J}(x)\mathbf{PQ}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}} + B_5(x, y, t)\mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}}^2 + B_6(x, y, t)\mathbf{J}(x)\mathbf{PQ}(y)\mathbf{R}(t) \\ &+ B_7(x, y, t)\mathbf{J}(x)\mathbf{Q}(y)\bar{\mathbf{P}}\mathbf{R}(t) + B_8(x, y, t)\mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t)\bar{\bar{\mathbf{P}}}\mathbf{A} + B_9(x, t)\mathbf{J}(x)\mathbf{Q}(y)\mathbf{R}(t) \end{aligned} \right\} \mathbf{A} = \mathbf{G}(x, y, t)$$

matris denklemi elde edilir ve bu denklem

$$\mathbf{W}(x, y, t)\mathbf{A} = \mathbf{G}(x, y, t) \quad (3.145)$$

olarak kısaca yazılabilir. Burada,

$$\mathbf{W}(x, y, t) = [w_{1,k}]_{1 \times (N+1)^3}$$

$$= \left\{ \begin{aligned} &B_1(x, y, t) \mathbf{J}(x) \mathbf{P}^2 \mathbf{Q}(y) \mathbf{R}(t) + B_2(x, y, t) \mathbf{J}(x) \mathbf{P} \mathbf{Q}(y) \bar{\mathbf{P}} \mathbf{R}(t) + B_3(x, y, t) \mathbf{J}(x) \mathbf{Q}(y) \bar{\mathbf{P}}^2 \mathbf{R}(t) \\ &+ B_4(x, y, t) \mathbf{J}(x) \mathbf{P} \mathbf{Q}(y) \mathbf{R}(t) \bar{\bar{\mathbf{P}}} + B_5(x, y, t) \mathbf{J}(x) \mathbf{Q}(y) \mathbf{R}(t) \bar{\bar{\mathbf{P}}}^2 + B_6(x, y, t) \mathbf{J}(x) \mathbf{P} \mathbf{Q}(y) \mathbf{R}(t) \\ &+ B_7(x, y, t) \mathbf{J}(x) \mathbf{Q}(y) \bar{\mathbf{P}} \mathbf{R}(t) + B_8(x, y, t) \mathbf{J}(x) \mathbf{Q}(y) \mathbf{R}(t) \bar{\bar{\mathbf{P}}} \mathbf{A} + B_9(x, y, t) \mathbf{J}(x) \mathbf{Q}(y) \mathbf{R}(t) \end{aligned} \right\},$$

$$k = 0, 1, \dots, (N+1)^3.$$

$a \leq x \leq b$, $c \leq t \leq d$ ve $0 \leq r \leq T$ olmak üzere,

$$x_i = a + \frac{(b-a)}{N} i, \quad y_j = c + \frac{(d-c)}{N} j, \quad t_r = \frac{T}{N} r, \quad i, j, r = 0, 1, \dots, N \quad (3.146)$$

ile tanımlı sıralama noktaları (3.145) denkleminde kullanılarak,

$$\mathbf{W}(x_i, y_j, t_r) \mathbf{A} = G(x_i, y_j, t_r), \quad i, j, r = 0, 1, \dots, N$$

matris denklemlerin bir sistemini elde edilir ve bu sistem

$$\mathbf{W} \mathbf{A} = \mathbf{G} \text{ ya da } [\mathbf{W}; \mathbf{G}] \quad (3.147)$$

olarak kısaca ifade edilebilir. Burada,

$$\mathbf{W} = \left[\begin{aligned} &\mathbf{W}(x_0, y_0, t_0) \cdots \mathbf{W}(x_0, y_0, t_N) \mathbf{W}(x_0, y_1, t_0) \cdots \mathbf{W}(x_0, y_1, t_N) \cdots \\ &\mathbf{W}(x_0, y_N, t_0) \cdots \mathbf{W}(x_0, y_N, t_N) \mathbf{W}(x_1, y_N, t_N) \cdots \mathbf{W}(x_N, y_N, t_N) \end{aligned} \right]_{(N+1)^3 \times (N+1)^3}^T,$$

$$\mathbf{A} = \left[\begin{aligned} &a_{0,0,0} \quad a_{0,0,1} \quad \cdots \quad a_{0,0,N} \quad a_{0,1,0} \quad a_{0,1,1} \quad \cdots \quad a_{0,1,N} \quad \cdots \\ &a_{0,N,0} \quad a_{0,N,1} \quad \cdots \quad a_{0,N,N} \quad \cdots \quad a_{N,N,0} \quad \cdots \quad a_{N,N,N} \end{aligned} \right]^T$$

ve

$$\mathbf{G} = \left[\begin{aligned} &G(x_0, y_0, t_0) \cdots G(x_0, y_0, t_N) G(x_0, y_1, t_0) \cdots G(x_0, y_1, t_N) \cdots \\ &G(x_0, y_N, t_0) \cdots G(x_0, y_N, t_N) G(x_1, y_N, t_N) \cdots G(x_N, y_N, t_N) \end{aligned} \right]_{(N+1)^3 \times 1}^T.$$

(3.118)-(3.125) koşullarına karşılık gelen (3.130)-(3.135) matris denklemlerinde (3.146) sıralama noktaları yerleştirilerek,

$$\mathbf{U}\mathbf{A} = [\mathbf{F}_1] \text{ ya da } [\mathbf{U}; \mathbf{F}_1], \quad (3.148)$$

$$\bar{\mathbf{U}}\mathbf{A} = [\mathbf{F}_2] \text{ ya da } [\bar{\mathbf{U}}; \mathbf{F}_2], \quad (3.149)$$

$$\mathbf{V}\mathbf{A} = [\mathbf{G}_1] \text{ ya da } [\mathbf{V}; \mathbf{G}_1], \quad (3.150)$$

$$\bar{\mathbf{V}}\mathbf{A} = [\mathbf{G}_2] \text{ ya da } [\bar{\mathbf{V}}; \mathbf{G}_2], \quad (3.151)$$

$$\mathbf{Z}\mathbf{A} = [\mathbf{H}_1] \text{ ya da } [\mathbf{Z}; \mathbf{H}_1], \quad (3.152)$$

$$\bar{\mathbf{Z}}\mathbf{A} = [\mathbf{H}_2] \text{ ya da } [\bar{\mathbf{Z}}; \mathbf{H}_2], \quad (3.153)$$

koşulların ana matris denklemleri elde edilir. Burada,

$$\mathbf{U} = [\mathbf{U}_{00} \quad \mathbf{U}_{01} \quad \cdots \quad \mathbf{U}_{0N} \quad \mathbf{U}_{10} \quad \mathbf{U}_{11} \quad \cdots \quad \mathbf{U}_{1N} \quad \cdots \quad \mathbf{U}_{N0} \quad \mathbf{U}_{N1} \quad \cdots \quad \mathbf{U}_{NN}]^T,$$

$$\bar{\mathbf{U}} = [\bar{\mathbf{U}}_{00} \quad \bar{\mathbf{U}}_{01} \quad \cdots \quad \bar{\mathbf{U}}_{0N} \quad \bar{\mathbf{U}}_{10} \quad \bar{\mathbf{U}}_{11} \quad \cdots \quad \bar{\mathbf{U}}_{1N} \quad \cdots \quad \bar{\mathbf{U}}_{N0} \quad \bar{\mathbf{U}}_{N1} \quad \cdots \quad \bar{\mathbf{U}}_{NN}]^T,$$

$$\mathbf{V} = [\mathbf{V}_{00} \quad \mathbf{V}_{01} \quad \cdots \quad \mathbf{V}_{0N} \quad \mathbf{V}_{10} \quad \mathbf{V}_{11} \quad \cdots \quad \mathbf{V}_{1N} \quad \cdots \quad \mathbf{V}_{N0} \quad \mathbf{V}_{N1} \quad \cdots \quad \mathbf{V}_{NN}]^T,$$

$$\bar{\mathbf{V}} = [\bar{\mathbf{V}}_{00} \quad \bar{\mathbf{V}}_{01} \quad \cdots \quad \bar{\mathbf{V}}_{0N} \quad \bar{\mathbf{V}}_{10} \quad \bar{\mathbf{V}}_{11} \quad \cdots \quad \bar{\mathbf{V}}_{1N} \quad \cdots \quad \bar{\mathbf{V}}_{N0} \quad \bar{\mathbf{V}}_{N1} \quad \cdots \quad \bar{\mathbf{V}}_{NN}]^T,$$

$$\mathbf{Z} = [\mathbf{Z}_{00} \quad \mathbf{Z}_{01} \quad \cdots \quad \mathbf{Z}_{0N} \quad \mathbf{Z}_{10} \quad \mathbf{Z}_{11} \quad \cdots \quad \mathbf{Z}_{1N} \quad \cdots \quad \mathbf{Z}_{N0} \quad \mathbf{Z}_{N1} \quad \cdots \quad \mathbf{Z}_{NN}]^T,$$

$$\bar{\mathbf{Z}} = [\bar{\mathbf{Z}}_{00} \quad \bar{\mathbf{Z}}_{01} \quad \cdots \quad \bar{\mathbf{Z}}_{0N} \quad \bar{\mathbf{Z}}_{10} \quad \bar{\mathbf{Z}}_{11} \quad \cdots \quad \bar{\mathbf{Z}}_{1N} \quad \cdots \quad \bar{\mathbf{Z}}_{N0} \quad \bar{\mathbf{Z}}_{N1} \quad \cdots \quad \bar{\mathbf{Z}}_{NN}]^T,$$

$$\mathbf{F}_1 = [f_1(x_0, y_0) \quad \cdots \quad f_1(x_0, y_N) \quad f_1(x_1, y_0) \quad \cdots \quad f_1(x_1, y_N) \quad \cdots \quad f_1(x_N, y_0) \quad \cdots \quad f_1(x_N, y_N)]^T,$$

$$\mathbf{F}_2 = [f_2(x_0, y_0) \cdots f_2(x_0, y_N) \ f_2(x_1, y_0) \cdots f_2(x_1, y_N) \cdots f_2(x_N, y_0) \cdots f_2(x_N, y_N)]^T,$$

$$\mathbf{G}_1 = [g_1(x_0, y_0) \cdots g_1(x_0, y_N) \ g_1(x_1, y_0) \cdots g_1(x_1, y_N) \cdots g_1(x_N, y_0) \cdots g_1(x_N, y_N)]^T,$$

$$\mathbf{G}_2 = [g_2(x_0, y_0) \cdots g_2(x_0, y_N) \ g_2(x_1, y_0) \cdots g_2(x_1, y_N) \cdots g_2(x_N, y_0) \cdots g_2(x_N, y_N)]^T,$$

$$\mathbf{H}_1 = [h_1(x_0, y_0) \cdots h_1(x_0, y_N) \ h_1(x_1, y_0) \cdots h_1(x_1, y_N) \cdots h_1(x_N, y_0) \cdots h_1(x_N, y_N)]^T,$$

$$\mathbf{H}_2 = [h_2(x_0, y_0) \cdots h_2(x_0, y_N) \ h_2(x_1, y_0) \cdots h_2(x_1, y_N) \cdots h_2(x_N, y_0) \cdots h_2(x_N, y_N)]^T,$$

$$i = 0, 1, \dots, N, \ j = 0, 1, \dots, N, \ r = 0, 1, \dots, N,$$

$$\mathbf{U}_{ij} = \mathbf{J}(x_i)\mathbf{Q}(y_j)\mathbf{R}(0) = [u_{i1}^j \ u_{i2}^j \ u_{i3}^j \ \dots u_{i(N+1)^3}^j],$$

$$\bar{\mathbf{U}}_{ij} = \mathbf{J}(x_i)\mathbf{Q}(y_j)\mathbf{R}(0)\bar{\mathbf{P}} = [\bar{u}_{i1}^j \ \bar{u}_{i2}^j \ \bar{u}_{i3}^j \ \dots \bar{u}_{i(N+1)^3}^j],$$

$$\mathbf{V}_{ir} = \mathbf{J}(x_i)\mathbf{Q}(c)\mathbf{R}(t_r) = [v_{i1}^r \ v_{i2}^r \ v_{i3}^r \ \dots v_{i(N+1)^3}^r],$$

$$\bar{\mathbf{V}}_{ir} = \mathbf{J}(x_i)\mathbf{Q}(d)\mathbf{R}(t_r) = [\bar{v}_{i1}^r \ \bar{v}_{i2}^r \ \bar{v}_{i3}^r \ \dots \bar{v}_{i(N+1)^3}^r],$$

$$\mathbf{Z}_{jr} = \mathbf{J}(a)\mathbf{Q}(y_j)\mathbf{R}(t_r) = [z_{i1}^r \ z_{i2}^r \ z_{i3}^r \ \dots z_{i(N+1)^3}^r]$$

ve

$$\bar{\mathbf{Z}}_{jr} = \mathbf{J}(b)\mathbf{Q}(y_j)\mathbf{R}(t_r) = [\bar{z}_{i1}^r \ \bar{z}_{i2}^r \ \bar{z}_{i3}^r \ \dots \bar{z}_{i(N+1)^3}^r].$$

(3.117) denkleminin (3.118)-(3.125) başlangıç ve sınır koşulları altında yaklaşık çözümünü elde etmek için, (3.147) denklemindeki $[\mathbf{W}; \mathbf{G}]$ artırılmış matris koşulların (3.148)-(3.153) artırılmış matris formlarını ekleyerek

$$[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F}_1 \\ \bar{\mathbf{U}} & ; & \mathbf{F}_2 \\ \mathbf{V} & ; & \mathbf{G}_1 \\ \bar{\mathbf{V}} & ; & \mathbf{G}_2 \\ \mathbf{Z} & ; & \mathbf{H}_1 \\ \bar{\mathbf{Z}} & ; & \mathbf{H}_2 \\ \mathbf{W} & ; & \mathbf{G} \end{bmatrix} \quad (3.154)$$

yeni artırılmış matrise sahip oluruz. Böylece bilinmeyen Bessel katsayılar matrisi $\mathbf{A} = (\tilde{\mathbf{W}})^{-1} \tilde{\mathbf{G}}$ olarak elde edilir. Burada $[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}]$ artırılmış matrisi, $[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}]$ artırılmış matrisine Gauss eliminasyon uygulandıktan sonra sıfır satırları atılarak elde edilen matristir. Matlab’ de (3.154) sisteminin çözümü ‘ $\mathbf{A} = \tilde{\mathbf{W}} \setminus \tilde{\mathbf{G}}$ ’ komutu ile kolayca hesaplanabilir.

Son olarak, belirlenen $a_{r,s,k}$; $(r, s, k = 0, 1, \dots, N)$ katsayıları (3.124) denkleminde yerine konularak

$$u_N(x, y, t) = \sum_{r=0}^N \sum_{s=0}^N \sum_{k=0}^N a_{r,s,k} J_{r,s,k}(x, y, t) \quad (3.155)$$

yaklaşık çözümü elde edilir.

(3.117) formundaki bazı iki boyutlu ikinci mertebe lineer kısmi diferansiyel denklem problemleri, (3.119) sınır koşulu yerine

$$u_t(x, y, 0) = f_2(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d$$

başlangıç koşulu ile verilir.

Bu durumda yapılması gereken sadece çözüm fonksiyonun t ’ ye göre birinci mertebe kısmi türevinin matris formu olan (3.138) denkleminde $t \rightarrow 0$ yazarak

$$u_t(x, y, 0) = \mathbf{J}(x) \mathbf{Q}(y) \mathbf{R}(0) \bar{\mathbf{P}} \mathbf{A}$$

olarak yeni koşulun matris formu elde edilir.

Benzer şekilde, prosedür takip edilerek yaklaşık çözüm elde edilir.

3.5.4. Hata tahmini

Bu kesimde, önceki bölümlerde kullanılan rezidüel hata tahmini (3.117)-(3.123) iki boyutlu ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklem problemi için uygulanacaktır.

İlk olarak, (3.117) denklemini

$$L[u(x, y, t)] = G(x, y, t), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.156)$$

biçiminde ifade edelim. Burada, L : lineer operatör ve

$$\begin{aligned} L[u(x, y, t)] = & B_1(x, y, t) \frac{\partial^2 u(x, y, t)}{\partial x^2} + B_2(x, y, t) \frac{\partial^2 u(x, y, t)}{\partial y \partial x} + B_3(x, y, t) \frac{\partial^2 u(x, y, t)}{\partial y^2} \\ & + B_4(x, y, t) \frac{\partial^2 u(x, y, t)}{\partial t \partial x} + B_5(x, y, t) \frac{\partial^2 u(x, y, t)}{\partial t^2} + B_6(x, y, t) \frac{\partial u(x, y, t)}{\partial x} \\ & + B_7(x, y, t) \frac{\partial u(x, y, t)}{\partial y} + B_8(x, y, t) \frac{\partial u(x, y, t)}{\partial t} + B_9(x, t) u(x, y, t) \end{aligned}$$

$u_N(x, y, t)$: (3.117) denkleminin Bessel polinom çözümü olsun, o zaman bu problem için sunulan yöntemin rezidüel fonksiyonu

$$R_N(x, y, t) = L[u_N(x, y, t)] - G(x, y, t)$$

olur. Böylece, $u_N(x, t)$,

$$L[u_N(x, y, t)] = G(x, y, t) + R_N(x, y, t) \quad (3.157)$$

denklemini ve

$$u_N(x, y, 0) = f_1(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.158)$$

$$u_N(x, y, T) = f_2(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.159)$$

$$u_N(x, c, t) = g_1(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.160)$$

$$u_N(x, d, t) = g_2(x, t), \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.161)$$

$$u_N(a, y, t) = h_1(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.162)$$

$$u_N(b, y, t) = h_2(y, t), \quad c \leq y \leq d, \quad 0 \leq t \leq T, \quad (3.163)$$

sınır koşullarını sağlar.

Bu problem için, hata fonksiyonunu

$$e_N(x, y, t) = u(x, y, t) - u_N(x, y, t)$$

olarak tanımlayalım. Burada, $u(x, t)$: (3.117)-(3.123) probleminin tam çözümünü gösterir.

Böylece, (3.118)-(3.123) ve (3.158)-(3.163) homojen olmayan sınır koşulları kullanılarak, $e_N(x, y, t)$ hata fonksiyonu için

$$e_N(x, y, 0) = 0, \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.164)$$

$$e_N(x, y, T) = 0, \quad a \leq x \leq b, \quad c \leq y \leq d, \quad (3.165)$$

$$e_N(x, c, t) = 0, \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.166)$$

$$e_N(x, d, t) = 0, \quad a \leq x \leq b, \quad 0 \leq t \leq T, \quad (3.167)$$

$$e_N(a, y, t) = 0, \quad c \leq y \leq d, \quad 0 \leq t \leq T \quad (3.168)$$

$$e_N(b, y, t) = 0, \quad c \leq y \leq d, \quad 0 \leq t \leq T, \quad (3.169)$$

homojen sınır koşulları elde edilir.

İlaveten, (3.117) ve (3.157) denklemleri aracılığıyla,

$$L[e_N(x, y, t)] = L[u(x, y, t)] - L[u_N(x, y, t)] = -R_N(x, y, t)$$

hata diferansiyel denklemini elde ederiz. Daha açık olarak, bu hata diferansiyel denklemi

$$\begin{aligned}
& B_1(x, y, t) \frac{\partial^2 e_N(x, y, t)}{\partial x^2} + B_2(x, y, t) \frac{\partial^2 e_N(x, y, t)}{\partial y \partial x} + B_3(x, y, t) \frac{\partial^2 e_N(x, y, t)}{\partial y^2} \\
& + B_4(x, y, t) \frac{\partial^2 e_N(x, y, t)}{\partial t \partial x} + B_5(x, y, t) \frac{\partial^2 e_N(x, y, t)}{\partial t^2} + B_6(x, y, t) \frac{\partial e_N(x, y, t)}{\partial x} \\
& + B_7(x, y, t) \frac{\partial e_N(x, y, t)}{\partial y} + B_8(x, y, t) \frac{\partial e_N(x, y, t)}{\partial t} + B_9(x, t) e_N(x, y, t) = -R_N(x, y, t).
\end{aligned} \tag{3.170}$$

olarak yazılabilir.

(3.164)-(3.169) sınır koşulları ile birlikte (3.170) hata diferansiyel denklemi önceki bölümde sunulan yöntem kullanılarak çözülerek hata probleminin $e_{N,M}(x, y, t)$ Bessel polinom çözümü elde edilir. Burada, M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırıdır.

3.5.5. Çözümlerin İyileştirilmesi

Bu kısımda, (3.117)-(3.123) iki boyutlu ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklem probleminin Bessel polinom çözümleri reziduel hata fonksiyonu yardımıyla iyileştirilecektir.

Bu iyileştirme, iki boyutlu ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklem probleminin Bessel polinom çözümü ile kesim (3.5.4)' de hata problemi için elde edilen $e_{N,M}(x, y, t)$ Bessel plinom çözümünün

$$u_{N,M}(x, y, t) = u_N(x, y, t) + e_{N,M}(x, y, t)$$

şeklinde toplanmasıyla elde edilir. Bu yaklaşık çözüm iyileştirilmiş Bessel polinom çözümü olarak adlandırılır. Burada, N : iki boyutlu ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklem probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırı ve M : hata probleminin Bessel polinom çözümü elde edilirken kullanılan kesme sınırını temsil eder. $M > N$ seçilerek, orjinal problem için daha iyi yaklaşık çözümler elde edilebilir.

3.6. İkinci Mertebeden Lineer Kısmi diferansiyel denklemler için Bessel Sıralama (Collocation) Metodunun Karmaşıklığı

Kesim (3.3)' de, genel formda olan başlangıç ve sınır koşulları altında ikinci mertebeden kısmi diferansiyel denklemler için Bessel sıralama yöntemini uygulamıştık. Bu kesimde, genel formda olan (3.73)-(3.77) başlangıç ve sınır koşullu ikinci mertebeden kısmi diferansiyel denklem problemi için Bessel sıralama yönteminin karmaşıklığı üzerine çalışacağız.

Yöntemin karmaşıklığını hesaplamak için ilk olarak, metodun uygulanmasında ne kadar kayan nokta olduğunu hesaplayacağız.

Kesim (3.3)' de

$$\mathbf{W}(x,t) = A(x,t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t) + B(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}} + C(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2 \\ + D(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t) + E(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}} + F(x,t)\mathbf{J}(x)\mathbf{Q}(t)$$

bulunurken ne kadar kayan nokta olduğunu hesaplayalım. Bunun için, $\mathbf{W}(x,t)$ içindeki her bir toplam için oluşan kayan nokta sayısını bulalım. Burada,

$A(x,t): 1 \times 1$ boyutunda,

$B(x,t): 1 \times 1$ boyutunda,

$C(x,t): 1 \times 1$ boyutunda,

$D(x,t): 1 \times 1$ boyutunda,

$E(x,t): 1 \times 1$ boyutunda

ve

$F(x,t): 1 \times 1$ boyutundadır.

İlk olarak, $\mathbf{W}(x,t)$ 'nin $D(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)$ terimini ele alalım. Burada,

$D(x,t): 1 \times 1$ boyutunda,

$\mathbf{J}(x): 1 \times (N+1)$ boyutunda,

$\mathbf{P} = (\mathbf{D}^T)^{-1}\mathbf{C}\mathbf{D}^T : (N+1) \times (N+1)$ boyutunda,

$\mathbf{D}^T : (N+1) \times (N+1)$ boyutunda,

$\mathbf{C} : (N+1) \times (N+1)$ boyutunda,

$(\mathbf{D}^T)^{-1} : (N+1) \times (N+1)$ boyutunda

ve

$\mathbf{Q}(t) : (N+1) \times (N+1)^2$ boyutundadır.

Şimdi, $\mathbf{P} = (\mathbf{D}^T)^{-1} \mathbf{C} \mathbf{D}^T$ için kaç kayan nokta olduğunu bulalım.

$(\mathbf{D}^T)^{-1}$ için $-\frac{(N+1)^2}{3} + (N+1)^3 + \frac{(N+1)^4}{3}$ kayan nokta,

$(\mathbf{D}^T)^{-1} \mathbf{C}$ çarpımı için $(2n+1)(N+1)^2 - \frac{(N+1)^2}{3} + (N+1)^3 + \frac{(N+1)^4}{3}$ kayan nokta

ve

$(\mathbf{D}^T)^{-1} \mathbf{C} \mathbf{D}^T$ çarpımı için $2(2N+1)(N+1)^2 - \frac{(N+1)^2}{3} + (N+1)^3 + \frac{(N+1)^4}{3}$ kayan

nokta olur.

Böylece, $\mathbf{P} = (\mathbf{D}^T)^{-1} \mathbf{C} \mathbf{D}^T$ hesaplanırken toplam

$$2(2N+1)(N+1)^2 - \frac{(N+1)^2}{3} + (N+1)^3 + \frac{(N+1)^4}{3} \quad (3.171)$$

kayan nokta vardır.

$\mathbf{J}(x) \mathbf{P}$ çarpımı için $(2N+1)(N+1)$ kayan nokta olduğundan, $\mathbf{J}(x) \mathbf{P} \mathbf{Q}(t)$ çarpımı için

$$(2N+1)(N+1) + (2N+1)(N+1)^2 \quad (3.172)$$

kayan nokta olur.

Böylece, $D(x, t) \mathbf{J}(x) \mathbf{P} \mathbf{Q}(t)$ çarpımında

$$(2N+1)(N+1) + (2N+1)(N+1)^2 + (N+1)^2 \quad (3.173)$$

kayan nokta vardır.

İkinci olarak, $A(x,t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t)$ terimini ele alalım.

$\mathbf{P}^2 = \mathbf{P}\mathbf{P}$ için $(2N+1)(N+1)^2$ kayan nokta olduğundan $\mathbf{J}(x)\mathbf{P}^2$ çarpımında $(2N+1)(N+1) + (2N+1)(N+1)^2$ tane nokta kayar.

Benzer şekilde, $\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t)$ çarpımı için $(2N+1)(N+1) + 2(2N+1)(N+1)^2$ kayan nokta vardır ve $A(x,t)\mathbf{J}(x)\mathbf{P}^2\mathbf{Q}(t)$ çarpımında

$$(2N+1)(N+1) + 2(2N+1)(N+1)^2 + (N+1)^2 \quad (3.174)$$

tane nokta kayar.

Şimdi, $B(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}}$ terimini ele alalım.

(3.172) denkleminde $\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)$ çarpımı için

$$(2N+1)(N+1) + (2N+1)(N+1)^2$$

kayan nokta olduğunu söylemiştik.

$\bar{\mathbf{P}}$: $(N+1)^2 \times (N+1)^2$ boyutunda olduğundan $\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}}$ çarpımı için

$$(2N+1)(N+1) + (2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^2$$

kayan nokta vardır. O zaman, $B(x,t)\mathbf{J}(x)\mathbf{P}\mathbf{Q}(t)\bar{\mathbf{P}}$ çarpımı için

$$(2N+1)(N+1) + (2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^2 + (N+1)^2 \quad (3.175)$$

tane kayan nokta olur.

$\mathbf{J}(x)\mathbf{Q}(t)$ çarpımında $(2N+1)(N+1)^2$ kayan nokta var olup, $F(x,t)\mathbf{J}(x)\mathbf{Q}(t)$ çarpımında

$$(2N+1)(N+1)^2 + (N+1)^2 \quad (3.176)$$

nokta kayar.

$\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}$ çarpımı için $(2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^2$ kayan nokta olduğundan $E(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}$ çarpımında

$$(2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^2 + (N+1)^2 \quad (3.177)$$

kayan nokta vardır.

$\bar{\mathbf{P}}^2 = \bar{\mathbf{P}}\bar{\mathbf{P}}$ için $(2(N+1)^2 - 1)(N+1)^4$ tane nokta kaydığından $\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2$ için

$$(2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^4 + (2(N+1)^2 - 1)(N+1)^2$$

tane kayan nokta vardır ve $C(x,t)\mathbf{J}(x)\mathbf{Q}(t)\bar{\mathbf{P}}^2$ çarpımında

$$(2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^4 + (2(N+1)^2 - 1)(N+1)^2 + (N+1)^2 \quad (3.178)$$

tane kayan nokta olur.

Ayrıca, $\mathbf{W}(x,t)$ içinde 6 tane terim toplandığından bu terimlerin toplanması için

$$5 \quad (3.178)$$

toplama işlemi yapılır.

(3.74) koşulunun sol tarafına karşılık gelen matris formundaki $\mathbf{J}(x)\mathbf{Q}(c)$ çarpımı için

$$(2N+1)(N+1)^2 \quad (3.179)$$

kayan nokta vardır.

(3.75) koşulunun sol tarafına karşılık gelen matris formundaki $\mathbf{J}(x)\mathbf{Q}(c)\bar{\mathbf{P}}$ çarpımı için

$$(2N+1)(N+1)^2 + (2(N+1)^2 - 1)(N+1)^2 \quad (3.180)$$

nokta kayar.

(3.76) koşulunun sol tarafına karşılık gelen matris formundaki $\mathbf{J}(a)\mathbf{Q}(t)$ çarpımı için

$$(2N+1)(N+1)^2 \quad (3.181)$$

kayan nokta vardır.

(3.77) koşulunun sol tarafına karşılık gelen matris formundaki $\mathbf{J}(b)\mathbf{Q}(t)$ çarpımında

$$(2N+1)(N+1)^2 \quad (3.182)$$

nokta kayar.

$$x_i = a + \frac{(b-a)}{N}i, \quad t_j = c + \frac{(d-c)}{N}j, \quad i = 0, 1, \dots, N, \quad j = 0, 1, \dots, N,$$

sıralama noktaları hesaplanırken

$$8(N+1) \quad (3.183)$$

nokta kayar.

Kesim (3.3.2)' de problem için metot uygulanırken koşulları temel alan matris sistemi

$$[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}] = \begin{bmatrix} \mathbf{U} & ; & \mathbf{F}_1 \\ \bar{\mathbf{U}} & ; & \mathbf{F}_2 \\ \mathbf{V} & ; & \mathbf{G}_1 \\ \bar{\mathbf{V}} & ; & \mathbf{G}_2 \\ \mathbf{W} & ; & \mathbf{G} \end{bmatrix}$$

olarak bulunmuştu. Burada,

$$\tilde{\mathbf{W}}: [(N+1)^2 + 4(N+1)] \times (N+1)^2 \text{ boyutunda}$$

ve

$$\tilde{\mathbf{G}}: [(N+1)^2 + 4(N+1)] \times 1 \text{ boyutundadır.}$$

$[\tilde{\tilde{\mathbf{W}}}; \tilde{\tilde{\mathbf{G}}}]$: $[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}]$ artırılmış matrisine gauss eliminasyon yapılarak elde edilen matristir.

$[\tilde{\mathbf{W}}; \tilde{\mathbf{G}}]$ sistemi Gauss eliminasyon yapılarak çözülürken

$$\left((N+1)^2\right)^2 \left((N+1)^2 + 4(N+1) - \frac{(N+1)^2}{3} \right) \quad (3.184)$$

tane kayan nokta olur.

Böylece, (3.171) ve (3.173)- (3.184) denklemlerindeki sayıların toplamı kesim (3.3)' de ele alınan problemin Bessel polinom çözümü hesaplanırken oluşan toplam kayan noktaların sayısını verir. (3.171) ve (3.173)-(3.184) denklemlerindeki sayılar toplandıktan sonra basitleştirilerek toplam kayan noktaların sayısı

$$\frac{137}{3} + \frac{422}{3}N + \frac{599}{3}N^2 + \frac{449}{3}N^3 + \frac{202}{3}N^4 + 20N^5 + \frac{8}{3}N^6$$

olarak elde edilir. Burada, N : problemin Bessel polinom çözümü elde edilirken alınan kesme sınırındır. Böylece, metodun karmaşıklığı

$$f(N) = \frac{137}{3} + \frac{422}{3}N + \frac{599}{3}N^2 + \frac{449}{3}N^3 + \frac{202}{3}N^4 + 20N^5 + \frac{8}{3}N^6$$

fonksiyonundaki en yüksek derece olan N^6 dır.

Bazı N kesme sınırları için metodun karmaşıklığının değerleri Çizelge 3.1' de verilmiştir.

Çizelge 3.1. Bazı N değerleri için metodun karmaşıklığının değerleri

N	2	3	4	5	6	7	8
Karmaşıklık	32	729	4096	15625	46656	117649	262144
N	9	10	11	12	13	14	15
Karmaşıklık	531441	1000000	1771561	2985984	4826809	7529536	11390625
N	16	17	18	19	20	21	
Karmaşıklık	16777216	24137569	34012224	47045881	64000000	85766121	

4. ARAŞTIRMA BULGULARI

Bu bölümde, dalga denklemi problemi, Neuman, Dirichlet ve Robin koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi, başlangıç ve sınır koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi, singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon problemi ve iki boyutlu ikinci mertebeden lineer kısmi diferansiyel denklem problemi için üçüncü bölümde sunulan Bessel sıralama yöntemi bazı örneklerle uygulanacak. Bu problemler için önceki bölümde sunulan hata tahmininin uygulamaları yapılacak ve birde örneklerde

$$L_2 = \|u(x,t) - u_N(x,t)\|_2 = \left(\int_0^T \int_a^b (u(x,t) - u_N(x,t))^2 dx dt \right)^{1/2}$$

ve

$$L_\infty = \|u(x,t) - u_N(x,t)\|_\infty = \max \{ |u(x,t) - u_N(x,t)|, a \leq x \leq b, 0 \leq t \leq T \}$$

hataları ile de çözümlerin doğruluğu için ölçüm yapılacak. Gerçek ve tahmin edilen hatalar karşılaştırılacak. Sunulan örneklerde, önceki bölümde anlatıldığı gibi Bessel polinom çözümleri iyileştirilecek. Ayrıca, bu çalışmada sunulan yöntem başka metotların sonuçları ile karşılaştırılacak. Nümerik hesaplamaların Maple ve Matlab' de yazılan bilgisayar programları ile hesaplandığını not ediyoruz.

4.1. Dalga Denklemi Problemi İçin Örnekler

Örnek 4.1.1: İlk olarak,

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = -2(x-t)e^{-x-t}, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1,$$

dalga denklemini

$$u(x,0) = 0, \quad 0 \leq x \leq 1,$$

$$u(0,t) = 0, \quad 0 \leq t \leq 1,$$

Dirichlet sınır koşulları,

$$u_t(x,0) = xe^{-x}, \quad 0 \leq x \leq 1,$$

Neumann sınır koşulu ve

$$\int_0^1 u(x,t)dx = -2te^{-t-1} + te^{-t}, \quad 0 \leq t \leq T$$

lineer olmayan (nonlocal) integral koşulu ile ele alalım (Shakeri ve Dehghan, 2008).

Bu problemin tam çözümü $u(x,t) = xte^{-x-t}$ ' dir.

Kesim (3.1)' deki prosedür takip edilerek bu problemin $N = 3, 4, 5, 6, 7, 8, 9, 10$ için Bessel polinom çözümleri Matlab' de yazılan bilgisayar programı ile hesaplandı. M ' nin bazı değerleri için hata fonksiyonu tahmin edilir ve yaklaşık çözümler iyileştirilir. Çizelge 4.1' de Bessel polinom çözümleri için L_2 ve L_∞ hataları verilir. Çizelge 4.2' de yaklaşık çözümlerin gerçek maksimum hataları ile tahmini maksimum hatalar karşılaştırılır. Şekil 4.1' de, $N = 5, 6, 7, 8, 9, 10$ için mutlak hata fonksiyonlarının grafiklerini veriyoruz. Şekil 4.2 ve Şekil 4.3 $|e_N(x,t)|$ gerçek mutlak hata fonksiyonları ile $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılmalarını gösterir. Şekil 4.4 ve Şekil 4.5 iyileştirilmiş ve iyileştirilmemiş mutlak hata fonksiyonlarının karşılaştırılmalarını gösterir.

$N = 3$ ve $M = 4$ için,

$$\begin{aligned}
u_3(x,t) = & -(0.3714723102e-3) + (0.1021547155e-2)t + 0.9976309517xt \\
& + (0.2260430798e-3)x^2 - (0.8551278475e-2)x^3 \\
& + (0.1991513187e-3)t^2 - 0.9415186229xt^2 + 0.6832244539x^2t^2 \\
& - 0.9115686742x^2t^3 - 0.1306117772x^2t^3 - 0.1307680271x^3t^2 \\
& + (0.4172140460e-1)x^3t^3 + (0.1491456031e-2)t^3
\end{aligned}$$

Bessel polinom çözümü ve

$$\begin{aligned}
u_{3,4}(x,t) = & -(0.7084352702e-3) + (0.8192041970e-2)x - (0.1572567319e-2)t \\
& + 1.035422149xt - 1.093307519x^2t - 1.132354428xt^2 + (0.1365444044e-1)x^2 \\
& - (0.4615065578e-1)x^3 + (0.2265097408e-1)x^4 + (0.1365334596e-1)t^2 \\
& - (0.3308716998e-1)t^3 + (0.2246091893e-1)t^4 + 1.095234454x^2t^2 \\
& - 0.4086582031x^2t^3 - 0.4069683301x^3t^2 + (0.9226524694e-1)x^3t^3 \\
& + 0.5517084200xt^3 + 0.5512445602x^3t - (0.1902988594e-1)x^3t^4 \\
& - (0.1857733389e-1)x^4t^3 + (0.6222039318e-2)x^4t^4 + (0.7430526415e-1)x^2t^4 \\
& + (0.7361048604e-1)x^4t^2 - 0.1245780819x^4t - 0.1244317250xt^4
\end{aligned}$$

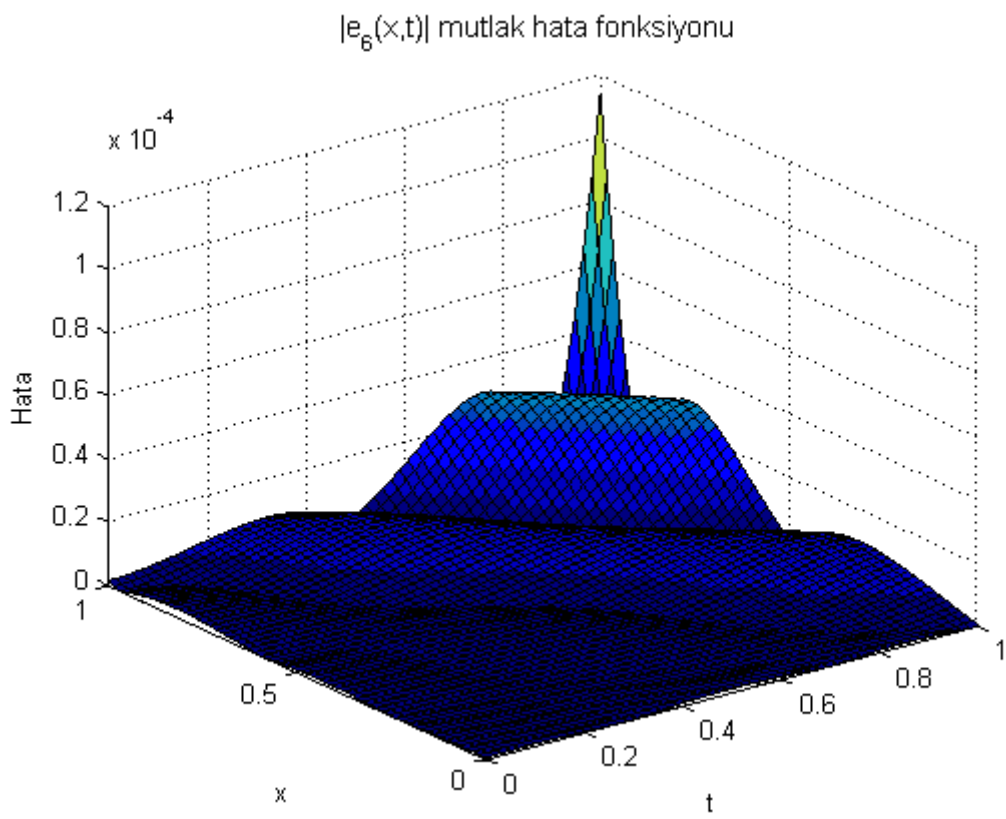
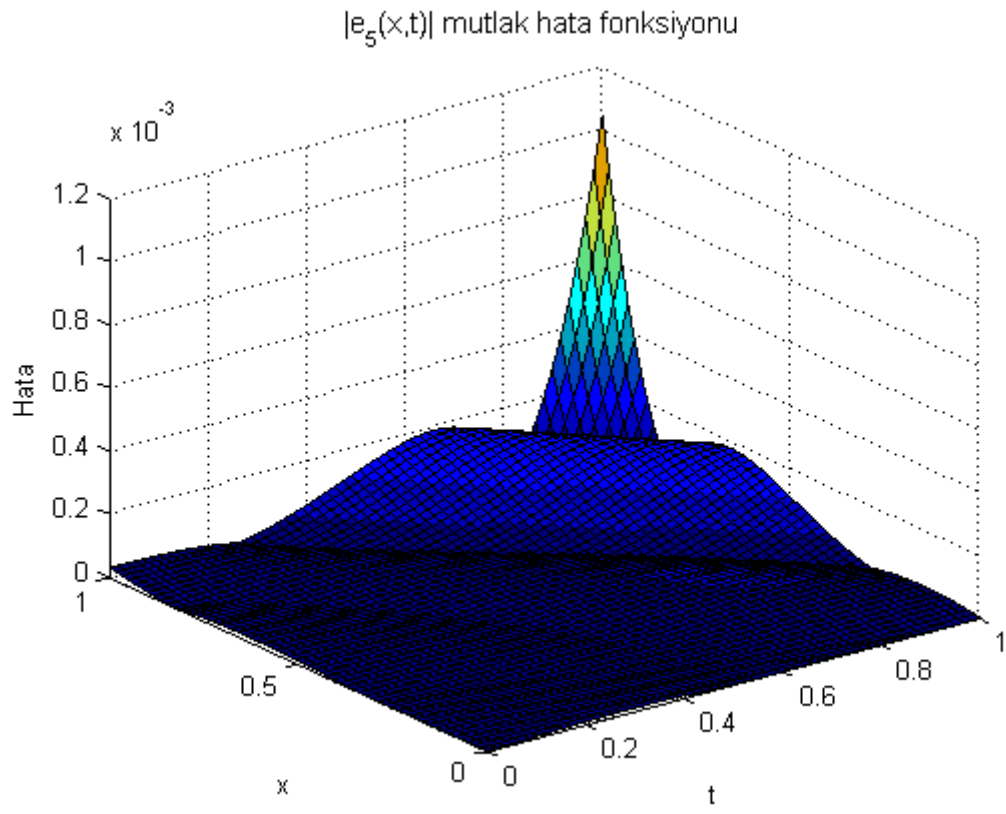
iyileştirilmiş Bessel polinom çözümü elde edilir.

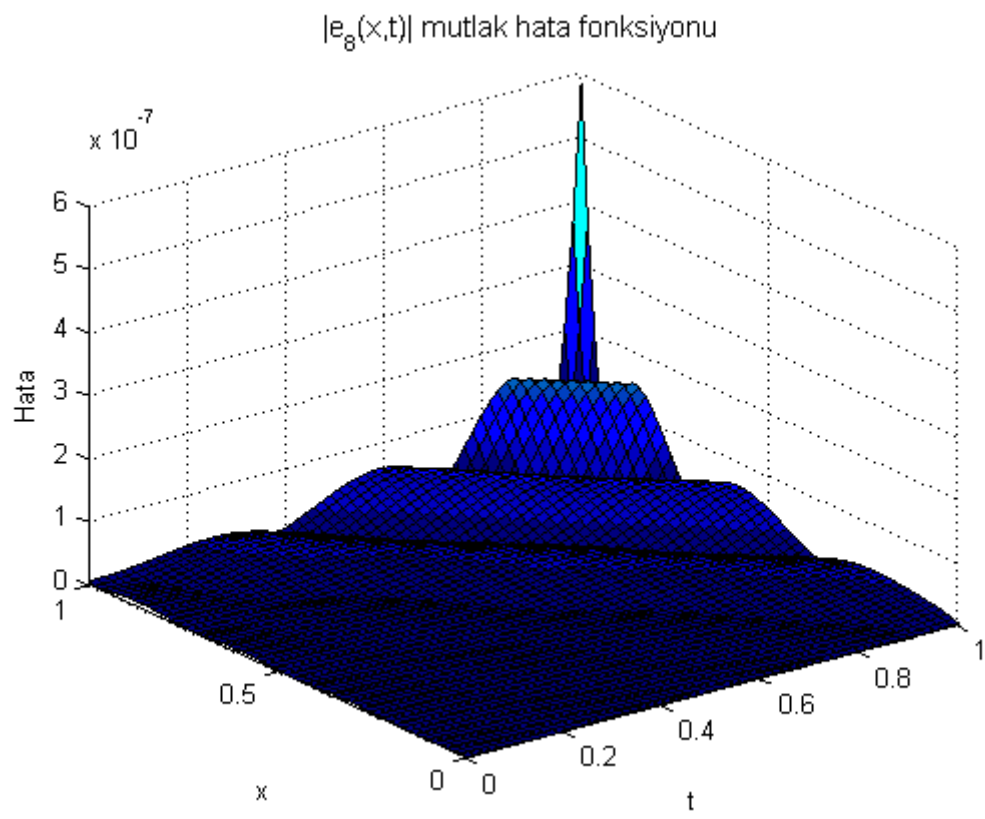
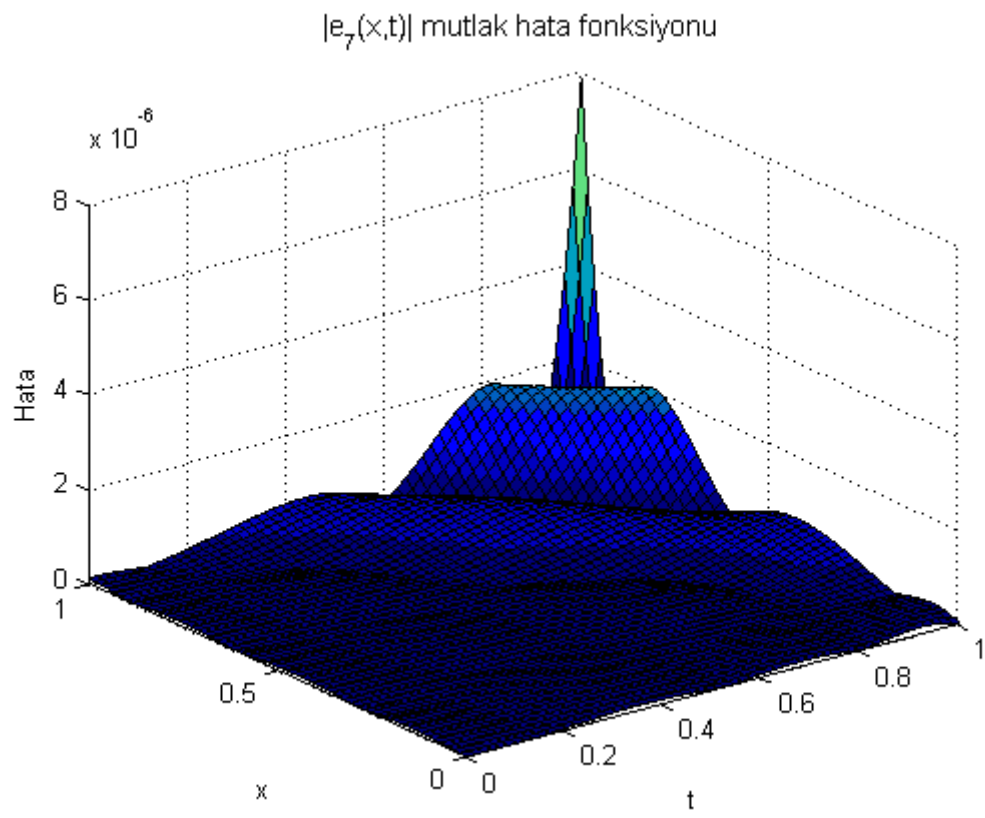
Çizelge 4.1. Örnek 4.1.1' in L_2 ve L_∞ hataları

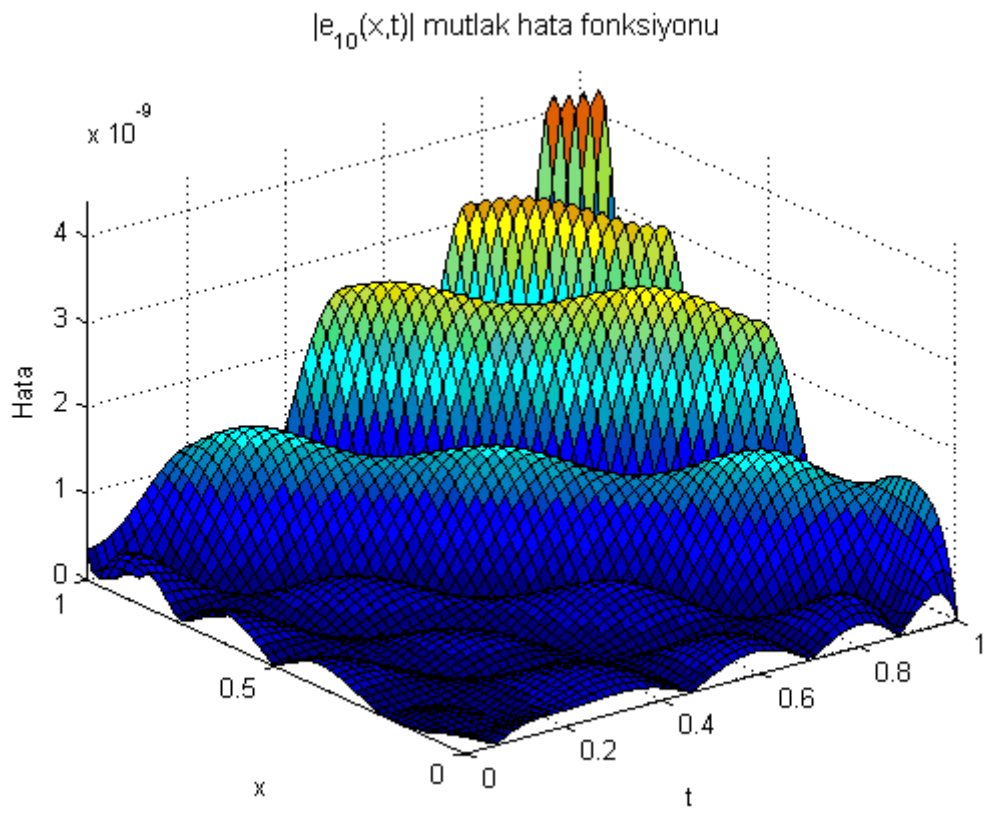
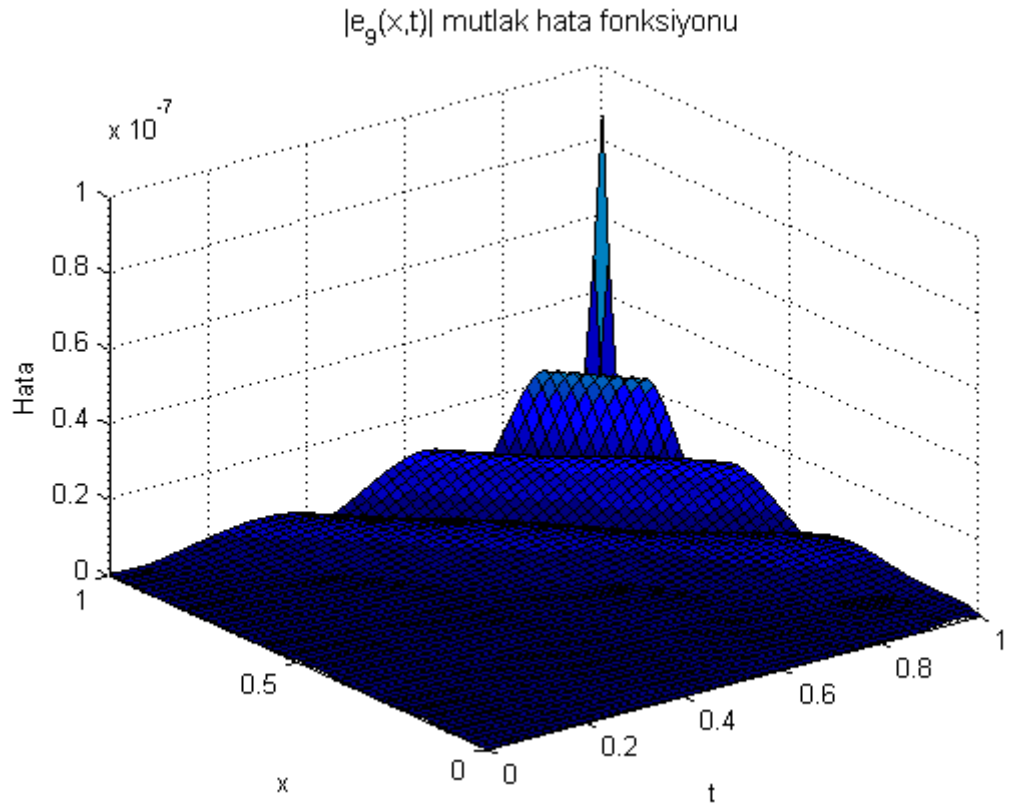
N	3	5	7	9
L_2	3.89×10^{-3}	6.89×10^{-5}	4.19×10^{-7}	1.06×10^{-9}
L_∞	3.58×10^{-2}	1.05×10^{-3}	7.9×10^{-6}	4.2×10^{-9}

Çizelge 4.2. Örnek 4.1.1 için bazı (N, M) değerleri için gerçek ve tahmin edilen maksimum hataların karşılaştırılması

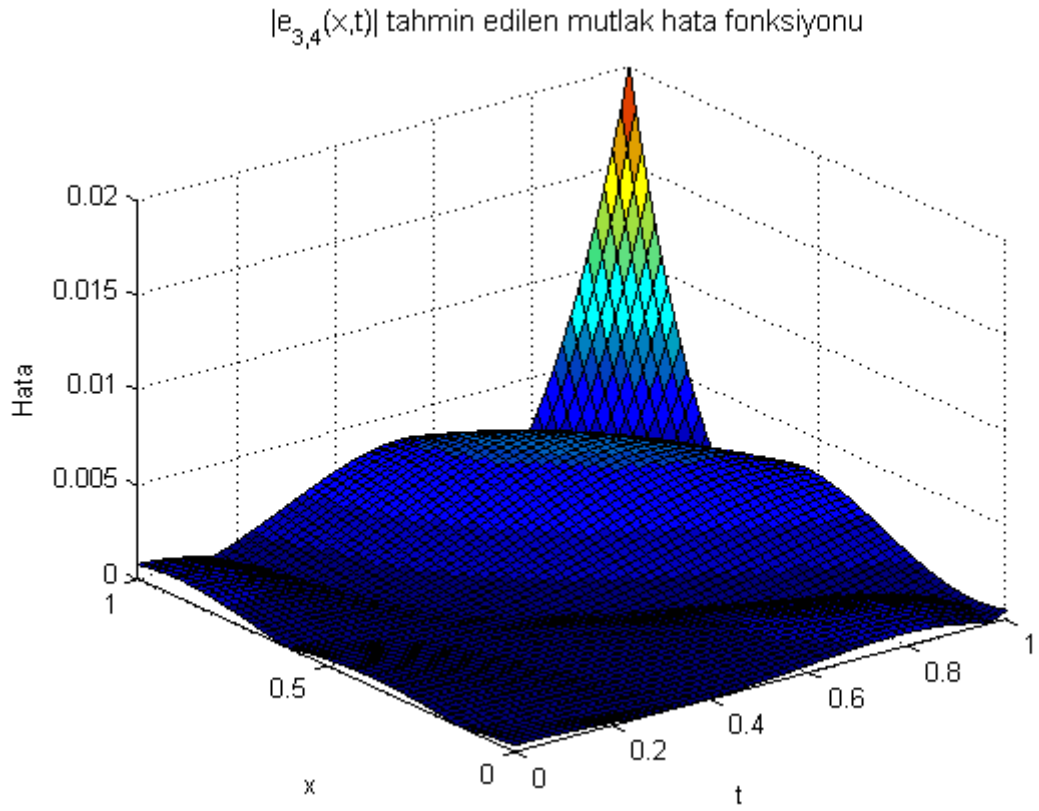
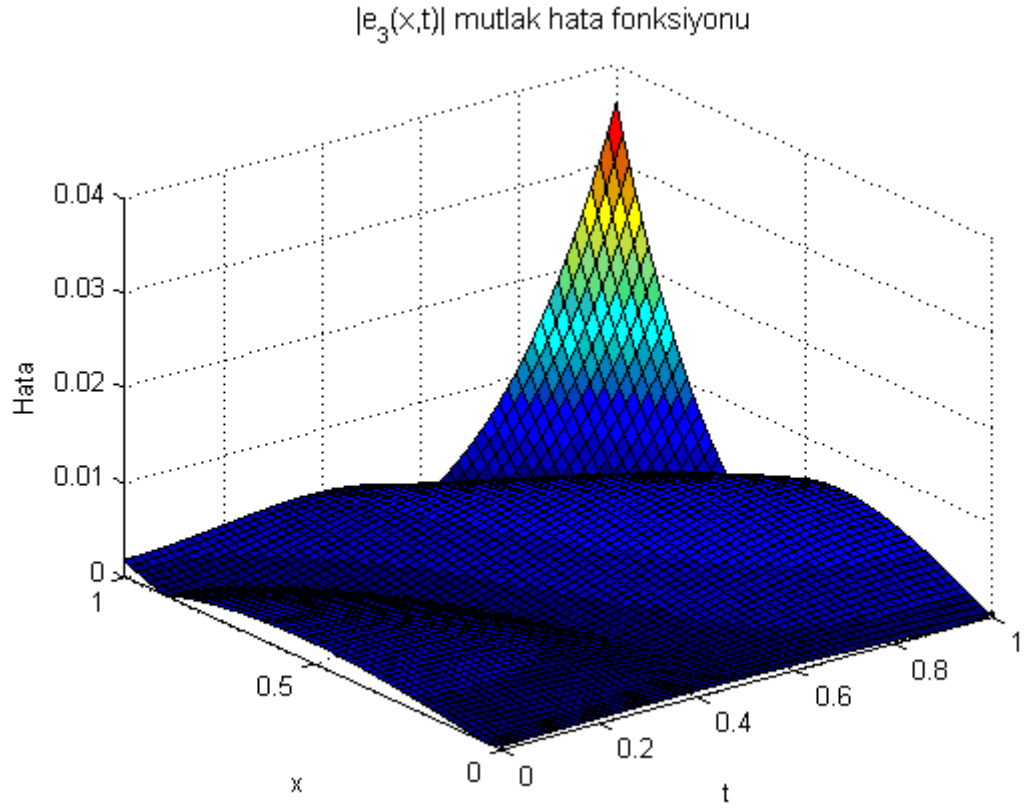
(N, M)	Gerçek maksimum hatalar L_∞	Tahmini maksimum hatalar
(3, 4)	3.5809×10^{-2}	1.9944×10^{-2}
(4, 5)	8.1408×10^{-3}	6.2206×10^{-4}
(4, 6)	8.1408×10^{-3}	3.6640×10^{-4}
(5, 7)	1.0435×10^{-3}	2.5119×10^{-4}
(7, 8)	7.8252×10^{-6}	1.3249×10^{-6}
(8, 9)	5.8052×10^{-7}	2.7452×10^{-7}
(9, 10)	8.6434×10^{-8}	6.9753×10^{-8}
(10, 11)	4.2730×10^{-9}	3.3826×10^{-8}



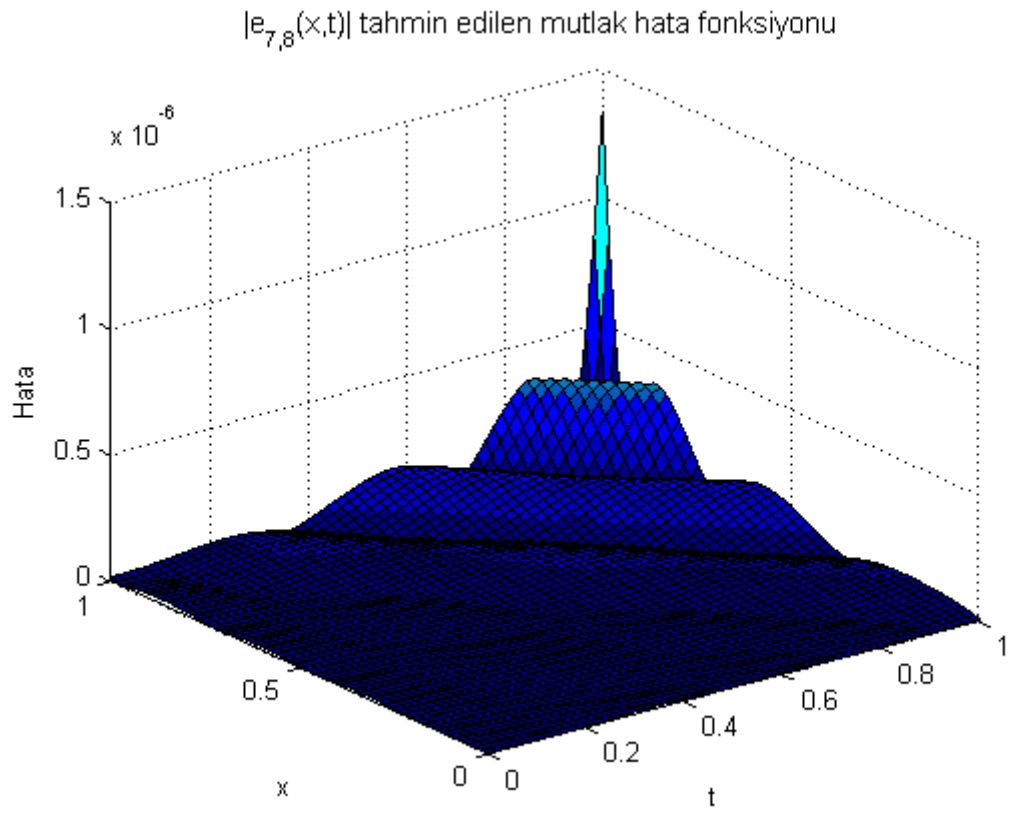
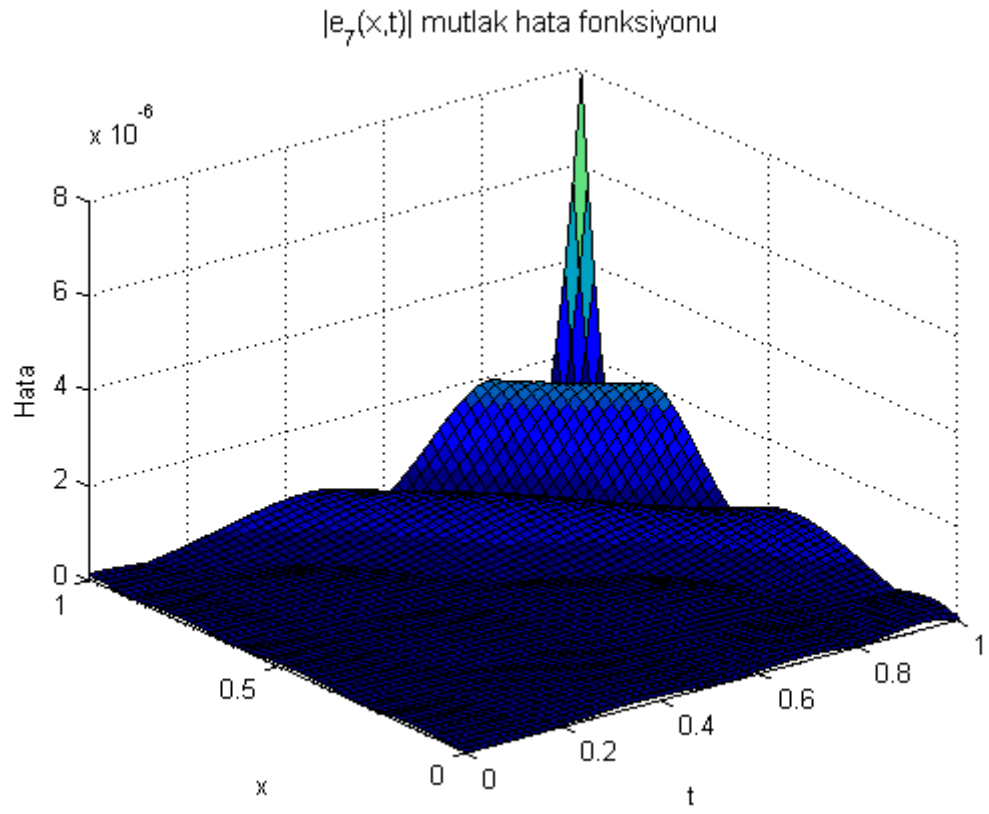




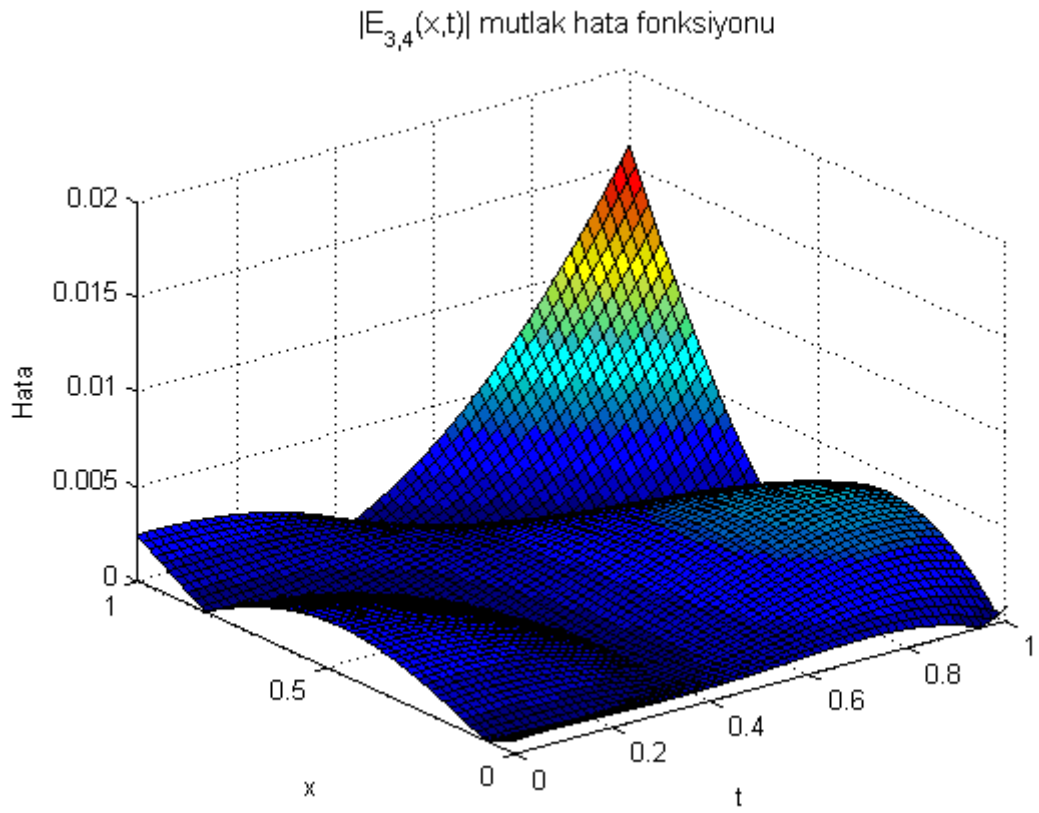
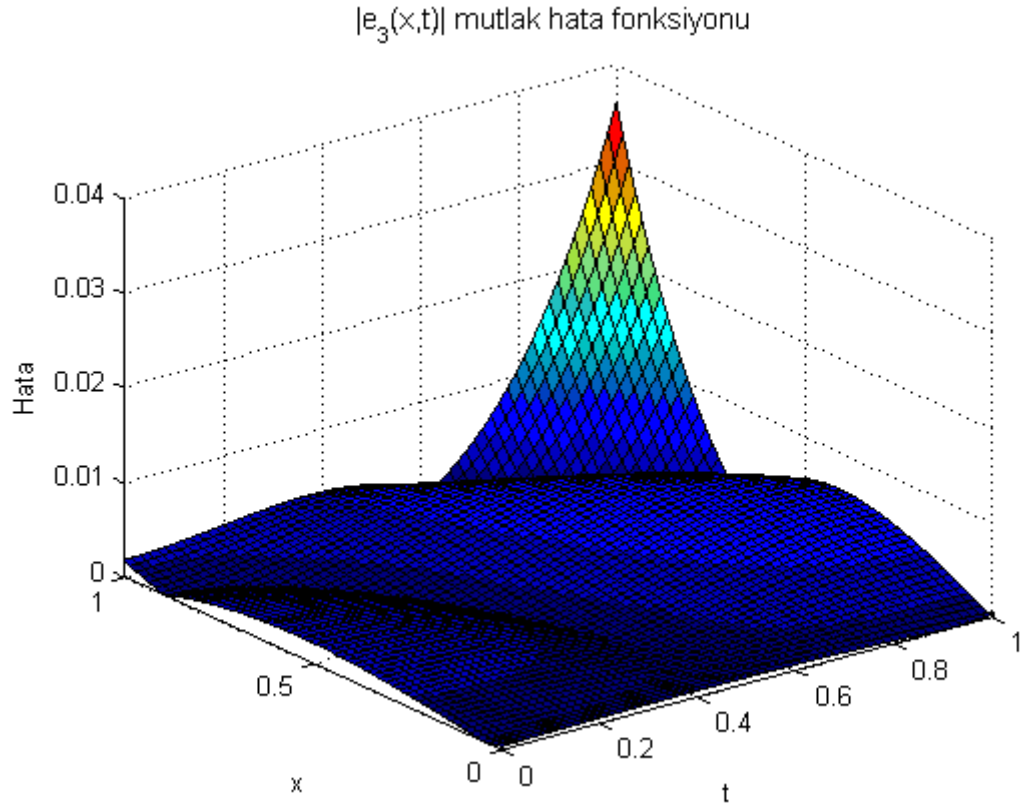
Şekil 4.1. Örnek 4.1.1' in $N = 5, 6, 7, 8, 9, 10$ için $|e_N(x,t)|$ mutlak hata fonksiyonlarının grafikleri



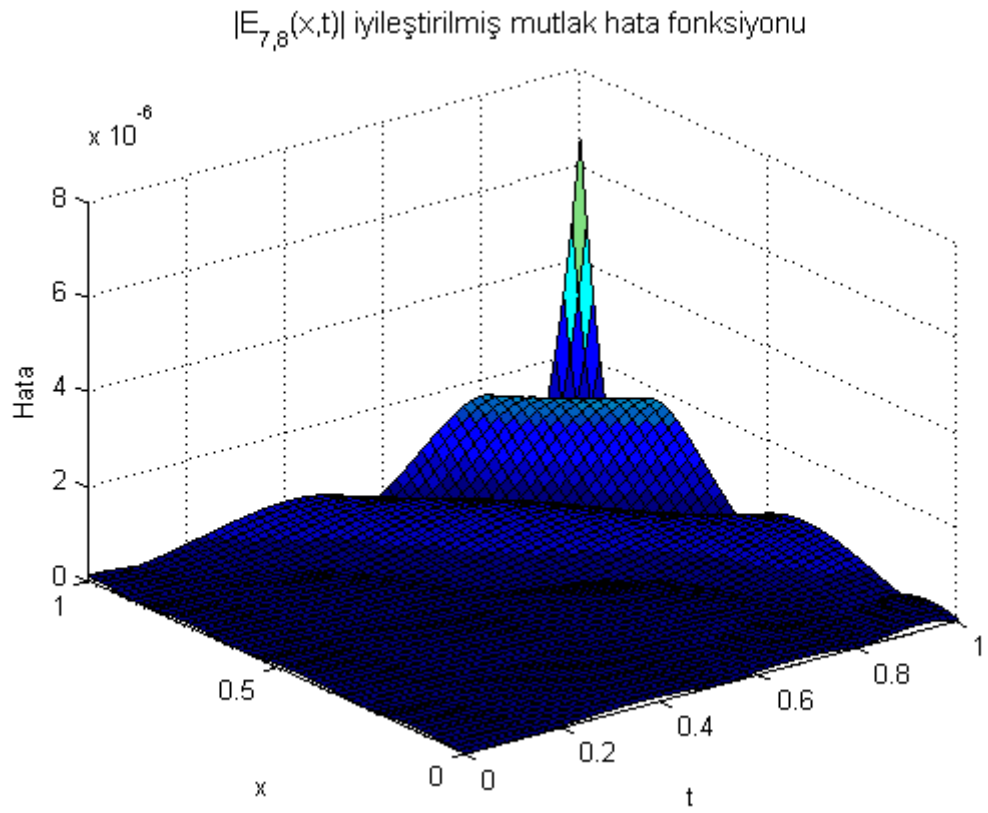
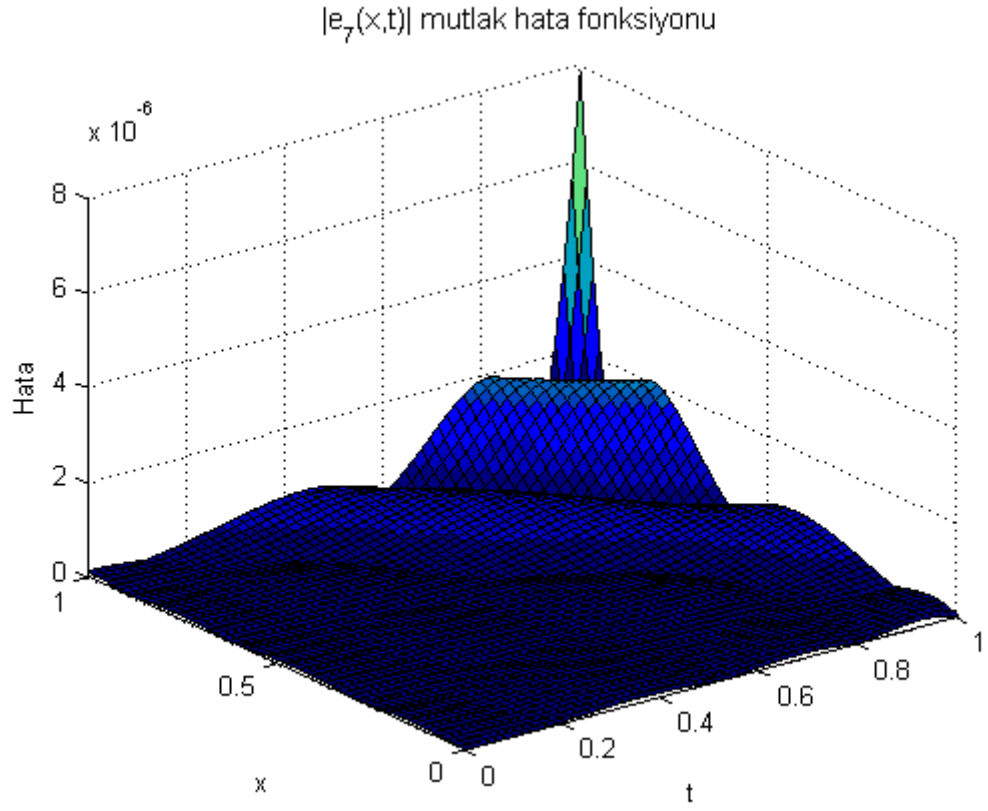
Şekil 4.2. Örnek 4.1.1' in $N = 3$ ve $M = 4$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.3. Örnek 4.1.1' in $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.4. Örnek 4.1.1' in $N = 3$ ve $M = 4$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.5. Örnek 4.1.1' in $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

Örnek 4.1.2: İkinci olarak,

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = (2x - 3x^2) \left(\frac{-4t^2}{(1+t^2)^2} + \frac{2}{1+t^2} \right) + 6 \ln(1+t^2), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1,$$

dalga denklemini

$$u(x, 0) = 0, \quad 0 \leq x \leq 1,$$

$$u(0, t) = 0, \quad 0 \leq t \leq 1,$$

Dirichlet sınır koşulları,

$$u_t(x, 0) = 0, \quad 0 \leq x \leq 1,$$

Neumann sınır koşulu ve

$$\int_0^1 u(x, t) dx = 0, \quad 0 \leq t \leq T$$

integral koşulu ile ele alalım (Yousefi vd., 2010).

$u(x, t) = \ln(1+t^2)(2x-3x^2)$ problemin tam çözümüdür.

Bu problem için sunulan yöntem $N=3, 7, 10$ için uygulanarak Bessel polinom çözümleri elde edildi. Yaklaşık çözümler Matlab' de yazılan bilgisayar kodu ile hesaplandı. Bazı (x_i, t_j) noktalarındaki mutlak hatalar Çizelge 4.3' de sunulur.

$N=7$ ve $M=8$ için $u(x, t)$ tam çözümü, $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümlerini Şekil 4.6' da karşılaştırıyoruz.

Şekil 4.7 ve Şekil 4.8' de $|e_N(x, t)|$ gerçek mutlak hata fonksiyonları ile $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonları karşılaştırılır. $N=4$ ve $M=5$ için $|e_N(x, t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x, t)|$ iyileştirilmiş mutlak hata fonksiyonu Şekil 4.9' da karşılaştırılır. N kesme sınırı artarken hataların azaldığı Çizelge 4.3, Şekil 4.7, Şekil 4.8 ve Şekil 4.9' dan gözlemlenir.

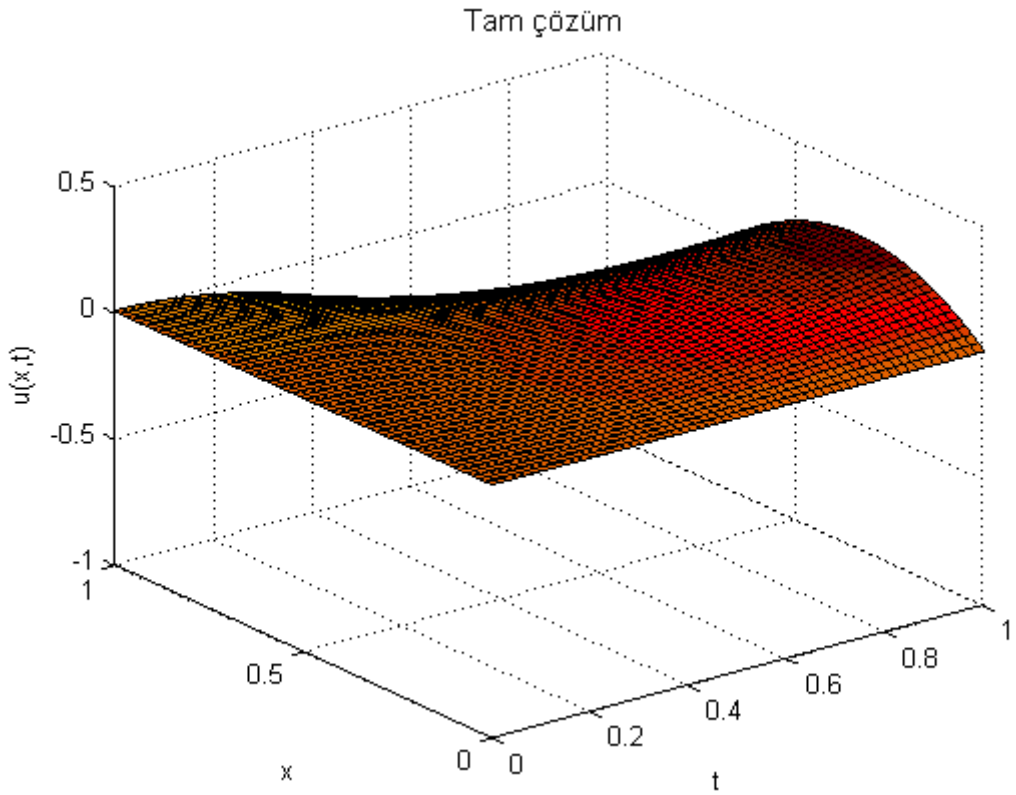
$N=3$ ve $M=4$ için Bessel polinom çözümü

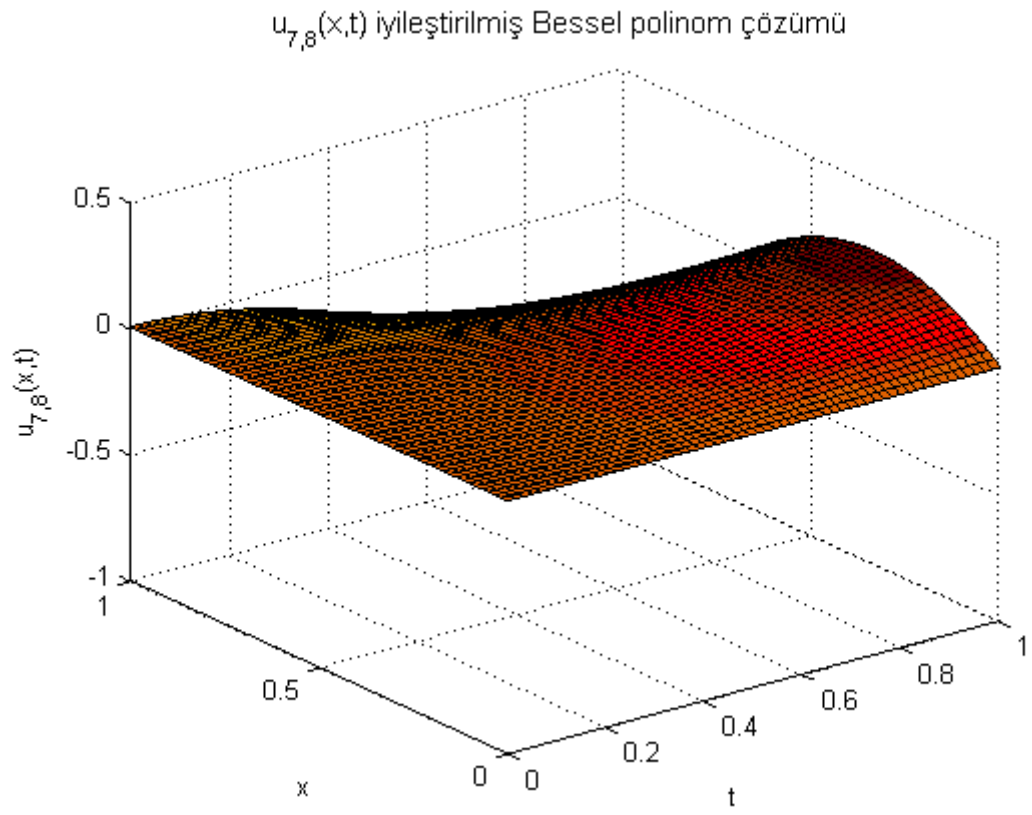
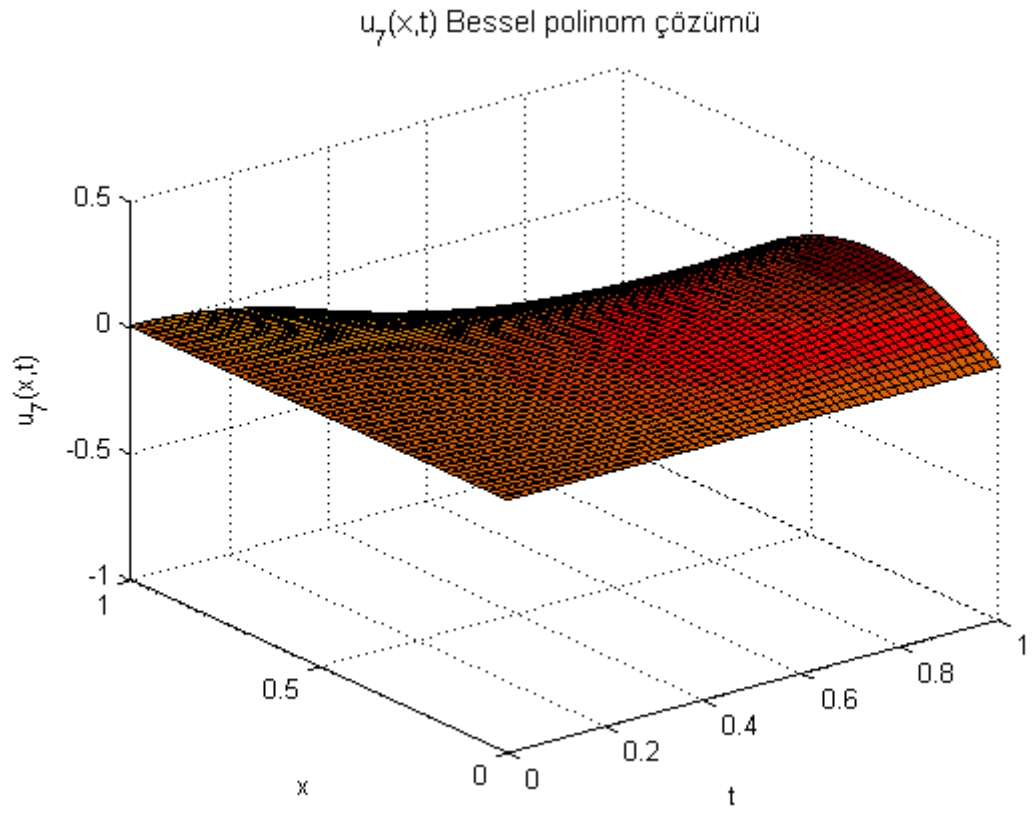
$$\begin{aligned}
u_3(x,t) = & (0.13889366e-1)x - (0.20870994e-2)t - (0.15329126e-1)x^2 \\
& - (0.17930239e-2)x^3 - (0.51377994e-4) - 2.4783137x^2t^2 + 0.65180409x^2t^3 \\
& - 0.39371562x^3t^2 + 0.26928065x^3t^3 + (0.22669235e-1)t^3 - (0.18895915e-1)t^2 \\
& - 0.19614006x^2t + 1.8488089xt^2 + (0.74352299e-1)xt - 0.57659236xt^3 \\
& + 0.12352758x^3t
\end{aligned}$$

olarak ve iyileştirilmiş Bessel polinom çözümü

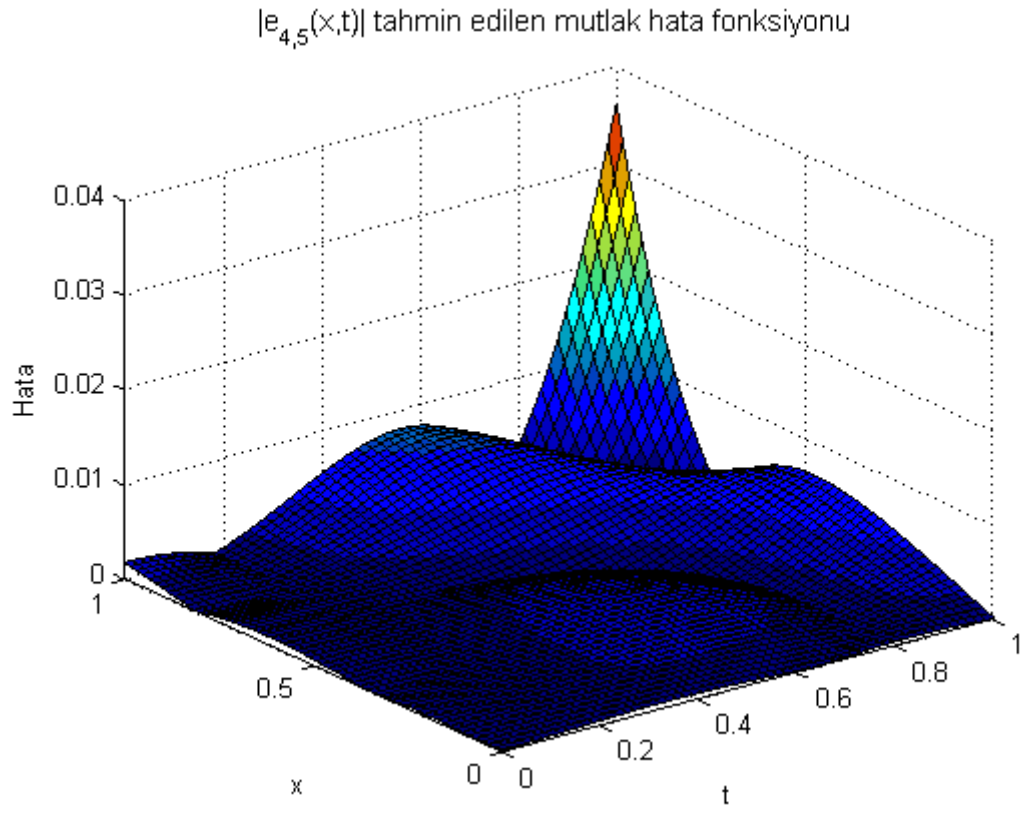
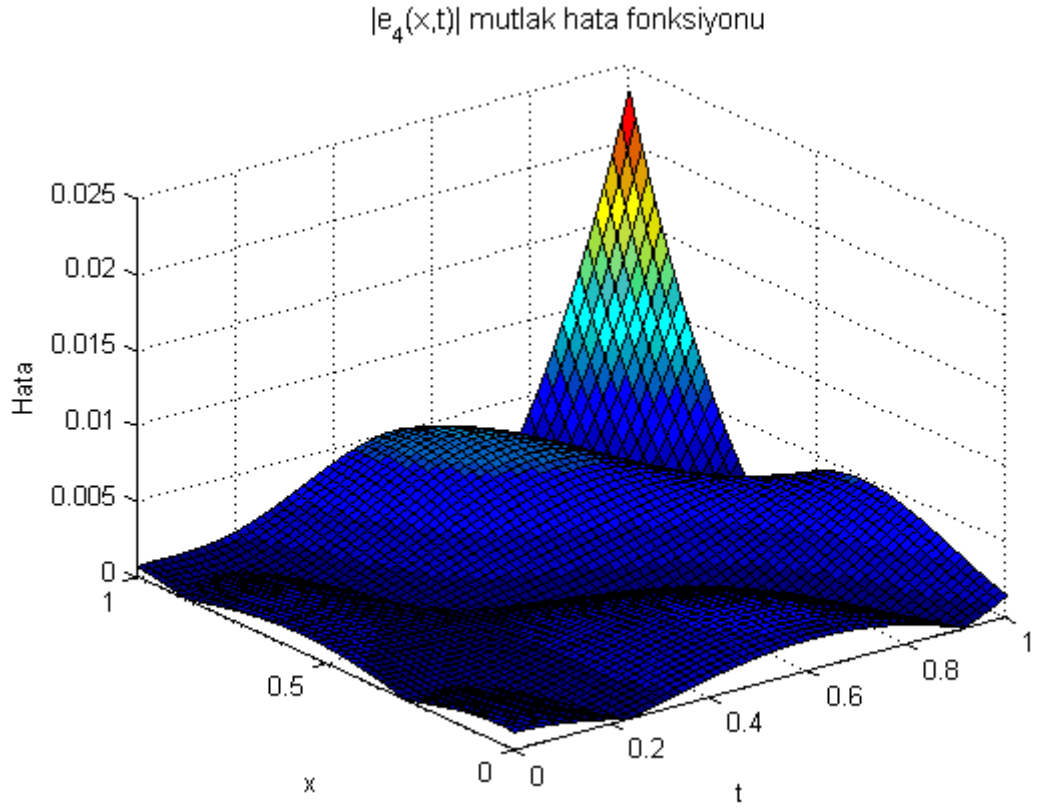
$$\begin{aligned}
u_{3,4}(x,t) = & (0.15835743e-1)x - (0.14818163e-2)t - (0.30489923e-1)x^2 \\
& + (0.22883290e-1)x^3 - 4.3425697x^2t^2 + 2.4317875x^2t^3 + 2.7075309x^3t^2 \\
& + 0.81129894x^3t^4 + (0.10081779e-3) - (0.11156841e-1)x^4 \\
& + (0.42496206e-1)t^4x - 1.5749044x^4t^2 - 0.41721906x^2t^4 + 0.46134843x^4t \\
& - (0.17537422e-1)t^2 + (0.83912908e-2)t^4 + (0.12330434e-1)t^3 \\
& + 1.5953367x^4t^3 - 0.43182028x^4t^4 - 2.8171553x^3t^3 + 0.36106807x^2t \\
& + 2.1778815xt^2 - (0.26131622e-1)xt - 0.85875346xt^3 - 0.79436954x^3t
\end{aligned}$$

şeklinde elde edilir.

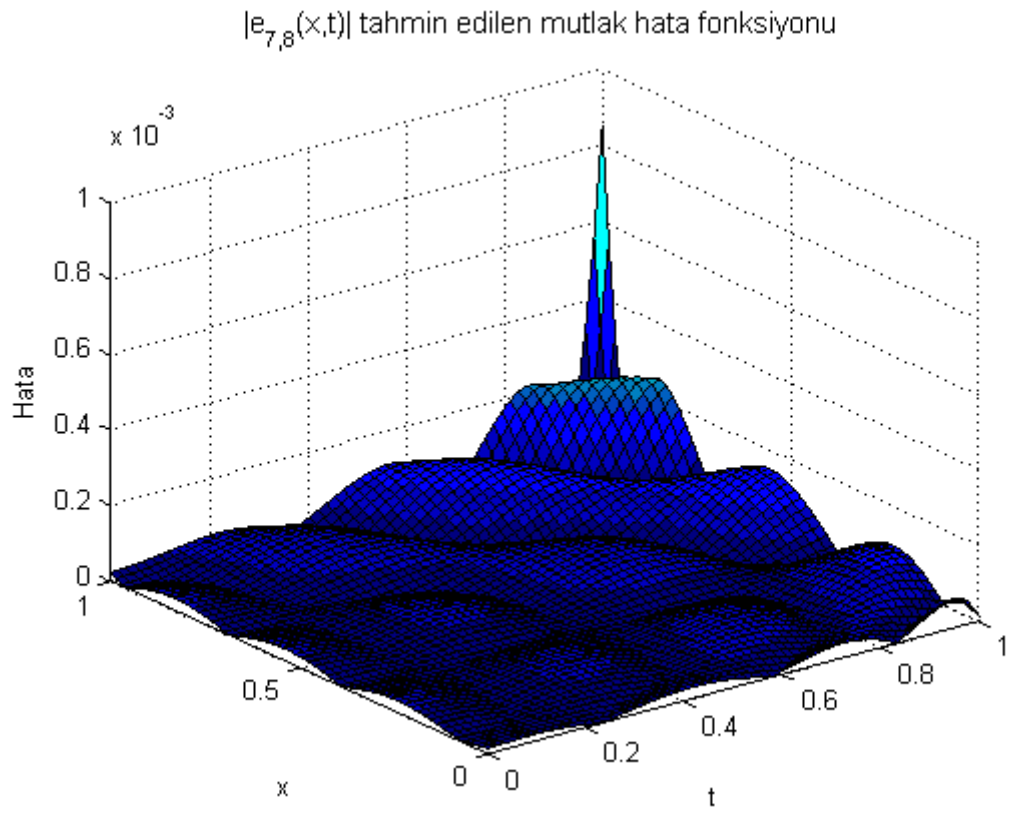
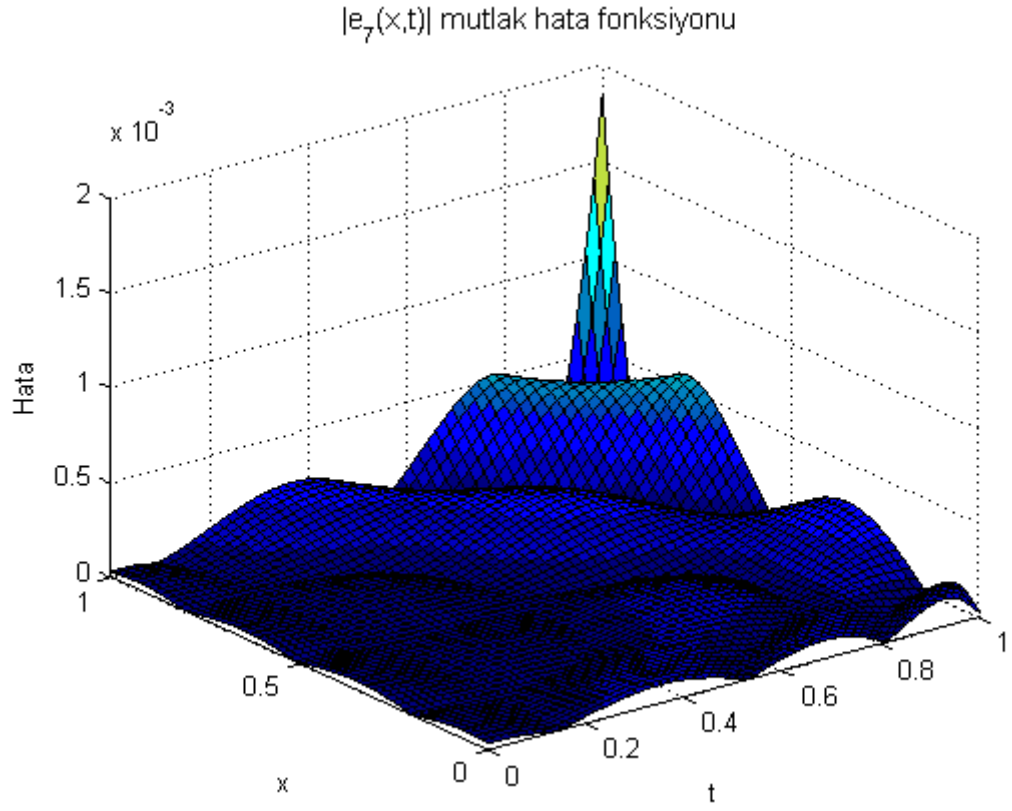




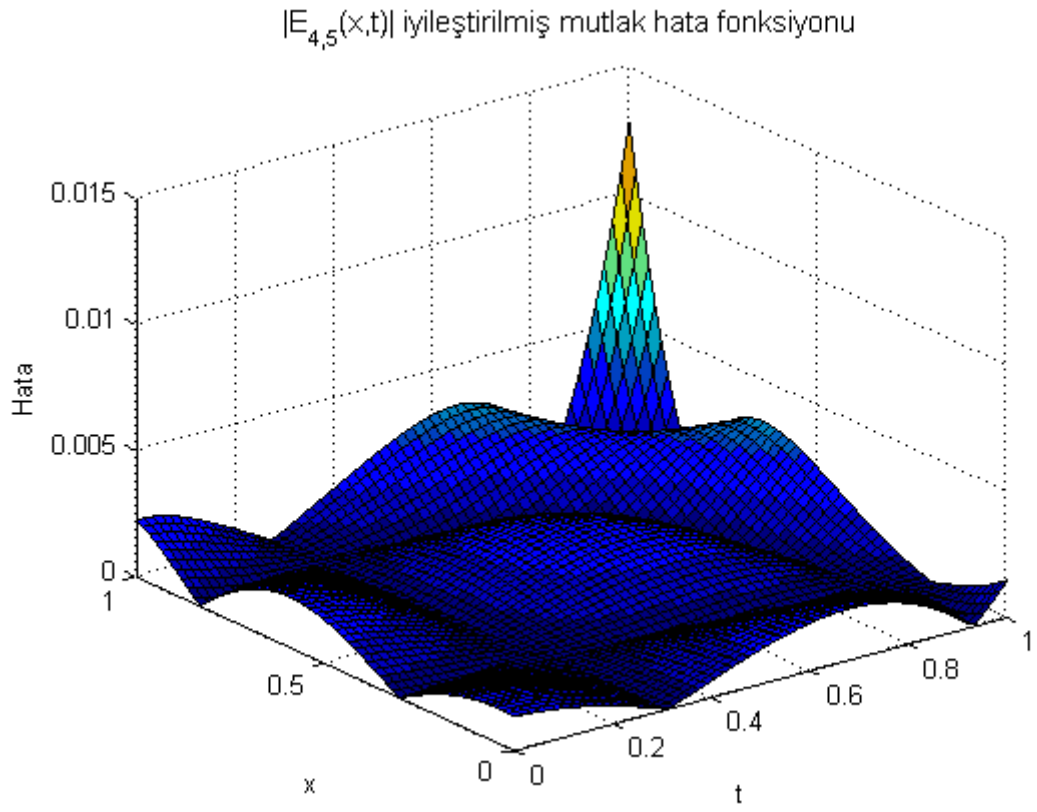
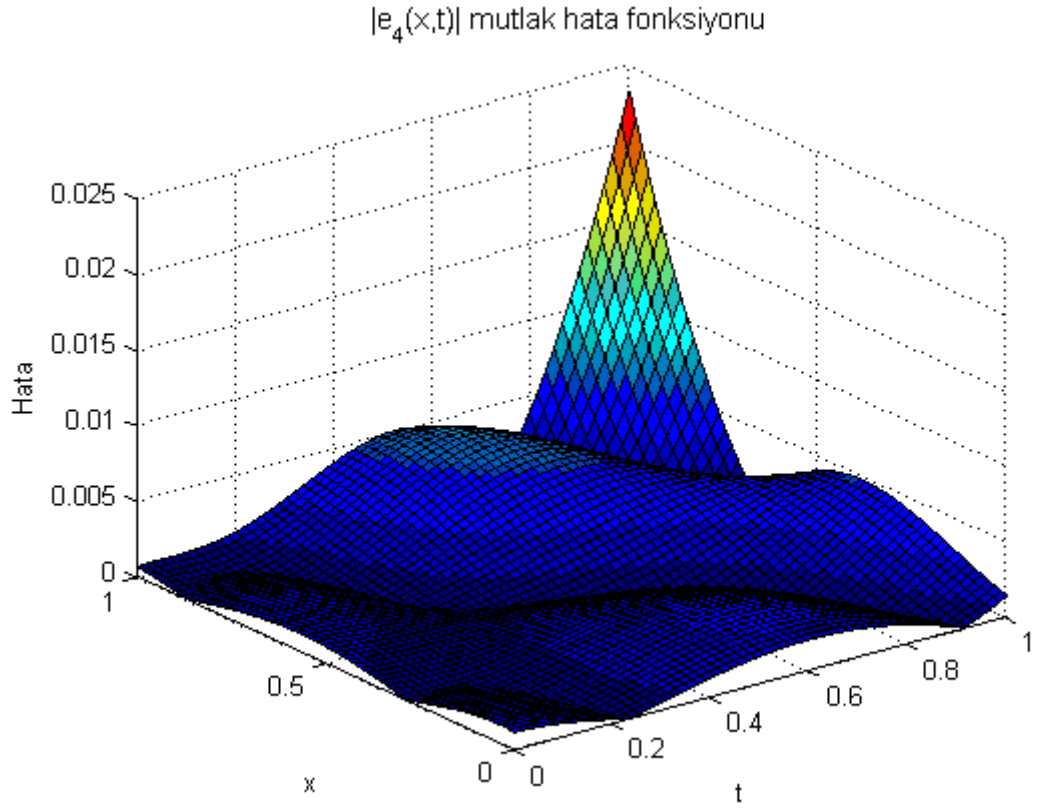
Şekil 4.6. Örnek 4.1.2' nin $N = 7$ ve $M = 8$ için $u(x,t)$ tam çözümü, $u_N(x,t)$ Bessel polinom çözümü ve $u_{N,M}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması



Şekil 4.7. Örnek 4.1.2' nin $N = 4$ ve $M = 5$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.8. Örnek 4.1.2' nin $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



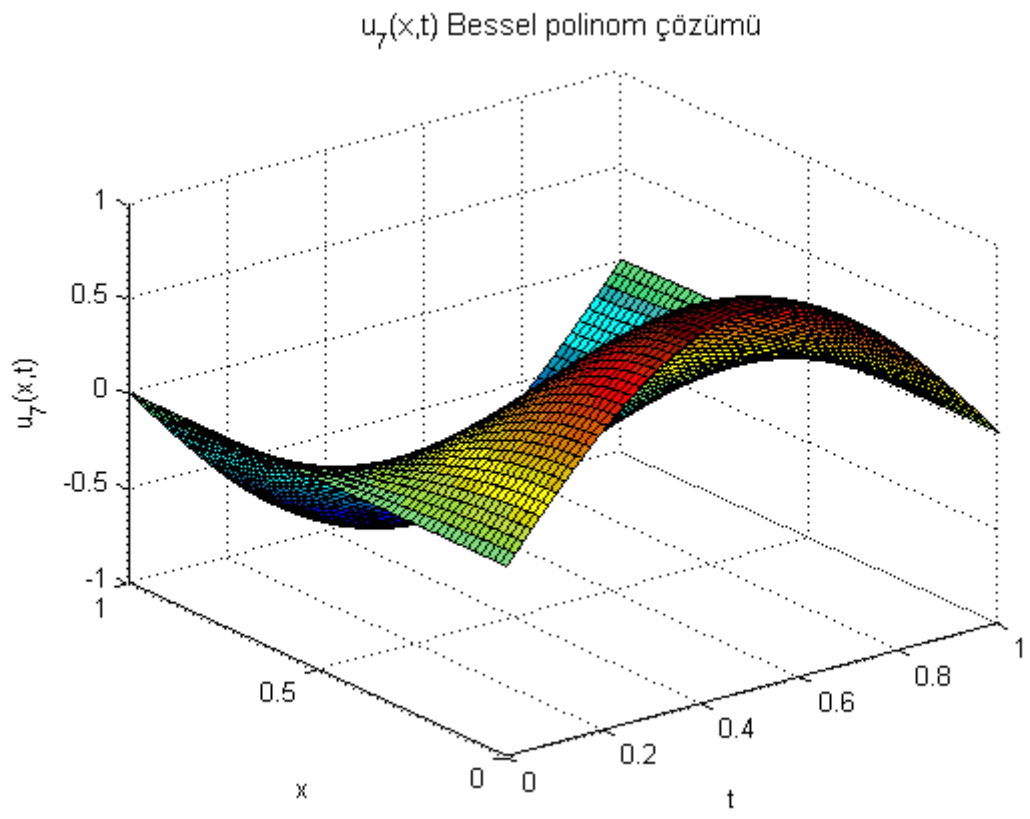
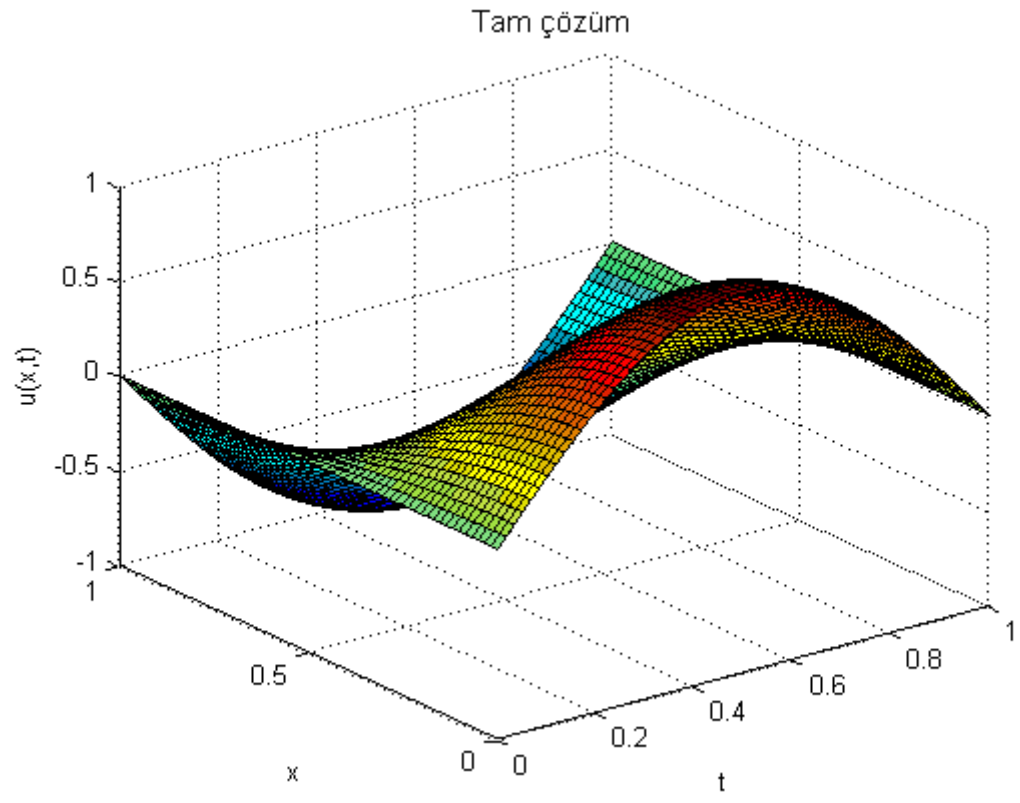
Şekil 4.9. Örnek 4.1.2' nin $N = 4$ ve $M = 5$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{NM}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

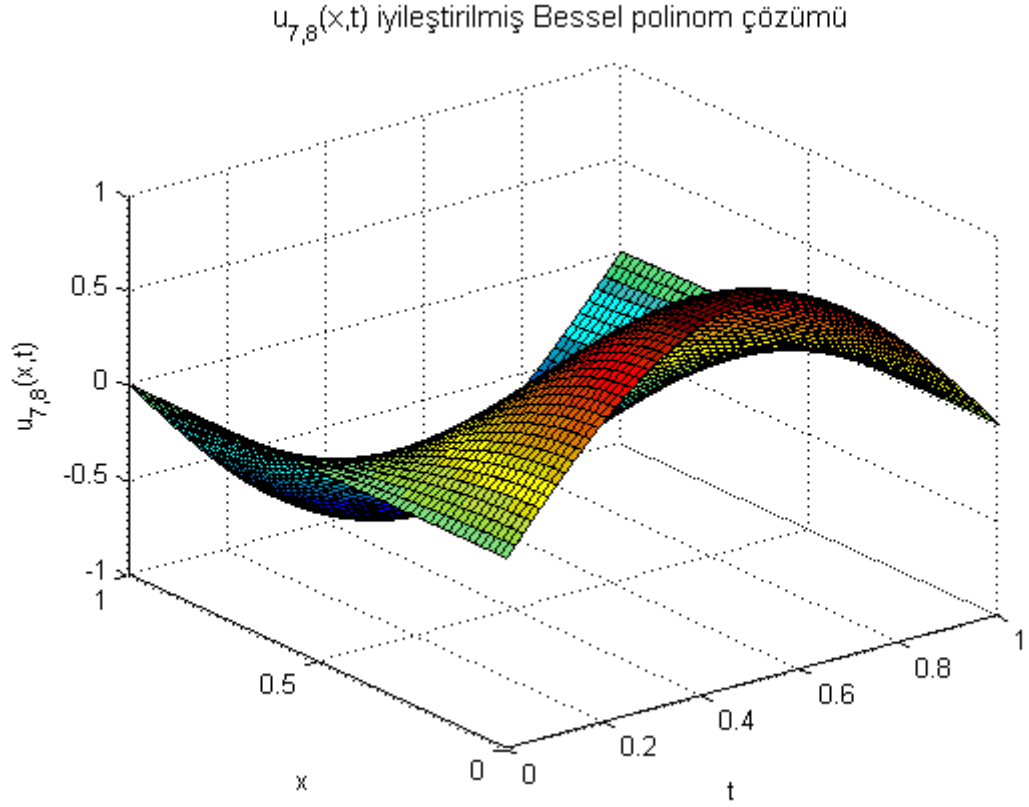
Çizelge 4.3. Örnek 4.1.2' nin $N = 3, 7, 10$ için bazı (x_i, t_j) noktalarındaki mutlak hataları

(x_i, t_j)	$N = 3, e_3(x_i, t_j)$	$N = 7, e_7(x_i, t_j)$	$N = 10, e_{10}(x_i, t_j)$
(0,0)	5.1378e-005	3.4366e-005	1.8486e-006
(0.1,0.1)	1.2218e-003	2.3335e-005	4.2913e-007
(0.2,0.2)	1.7303e-003	2.2185e-006	5.6527e-007
(0.3,0.3)	9.2713e-004	3.2261e-005	1.9086e-006
(0.4,0.4)	5.9512e-004	1.7129e-005	6.2070e-007
(0.5,0.5)	1.6496e-003	1.7738e-007	1.7617e-006
(0.6,0.6)	1.2108e-003	1.1823e-004	7.7937e-006
(0.7,0.7)	1.0464e-003	2.2970e-004	8.7023e-007
(0.8,0.8)	4.6978e-003	8.0386e-005	1.3817e-005
(0.9,0.9)	9.1120e-003	4.8088e-004	2.3497e-005
(1,1)	1.4561e-002	1.8319e-003	1.0376e-004

Örnek 4.1.3: Son olarak, (3.1)-(3.5) problemini $f_1(x)=0$, $f_2(x)=\pi \cos(\pi x)$, $l=T=1$, $g_1(t)=\sin(\pi t)$, $g_2(t)=0$ ve $q(x,t)=0$ ile ele alalım (Dehghan, 2005). Bu problemin tam çözümü $u(x,t)=\cos(\pi x)\sin(\pi t)$ ’ dir.

$N=7$ için sunulan yöntem uygulanarak Bessel polinom çözümü Matlab’ de yazılan kod ile hesaplandı. $u(x,t)$ tam çözüm, $u_7(x,t)$ Bessel polinom çözümü ve $u_{7,8}(x,t)$ iyileştirilmiş Bessel polinom çözümünü Şekil 4.10’ da karşılaştırıyoruz. $u(x_i,0.5)$ için sunulan yöntemin ve sonlu farklar metodunun (Dehghan, 2005) mutlak hataları Çizelge 4.4’ de karşılaştırılır. $|e_7(x,t)|$ gerçek mutlak hata fonksiyonu ve $|e_{7,8}(x,t)|$ tahmin edilen mutlak hata fonksiyonu Şekil 4.11’ de karşılaştırılır. Şekil 4.12 $|e_7(x,t)|$ mutlak hata fonksiyonu ve $|E_{7,8}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılmasını gösterir.

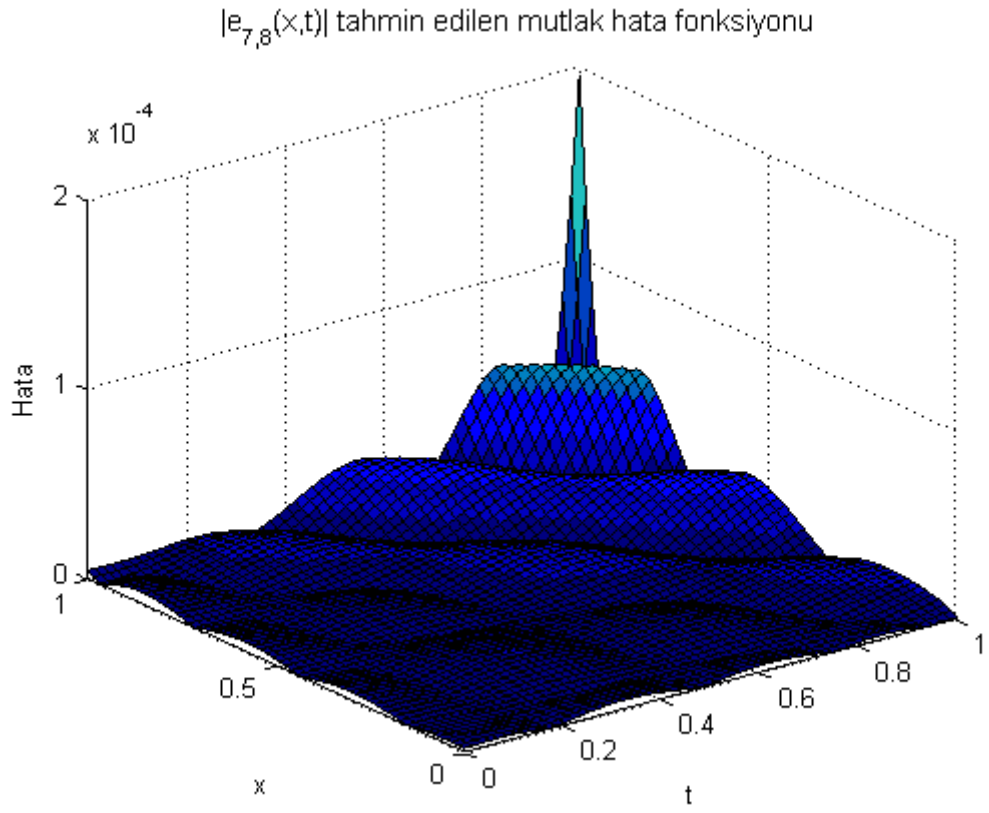
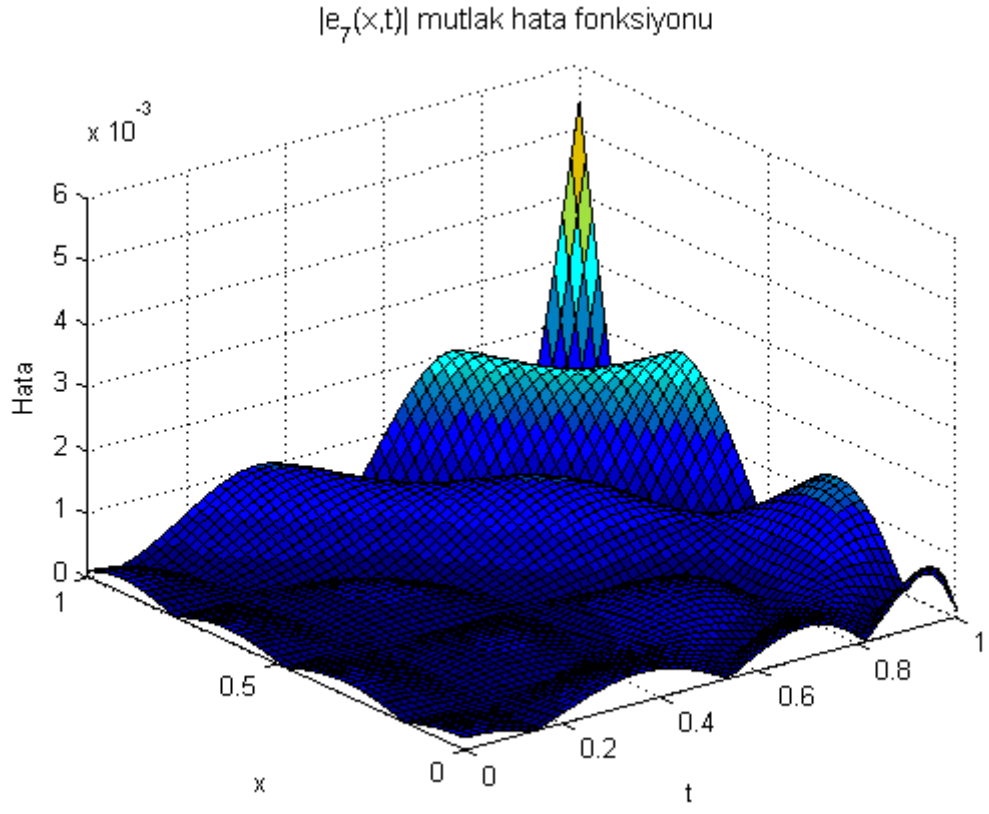




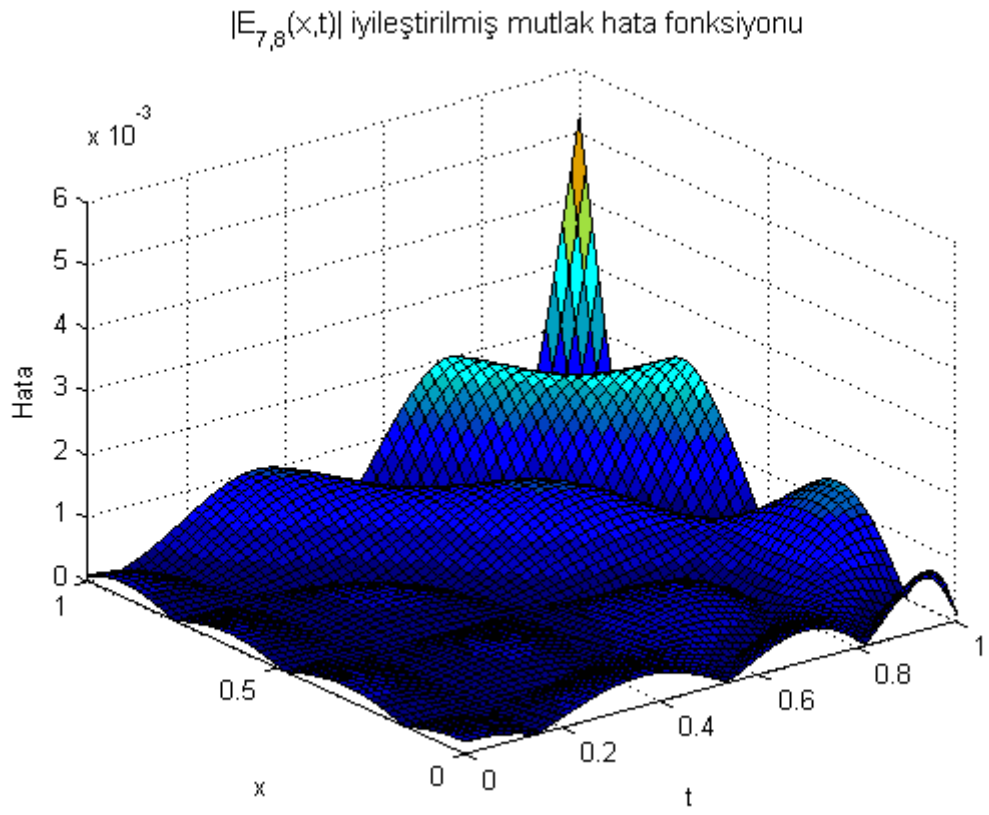
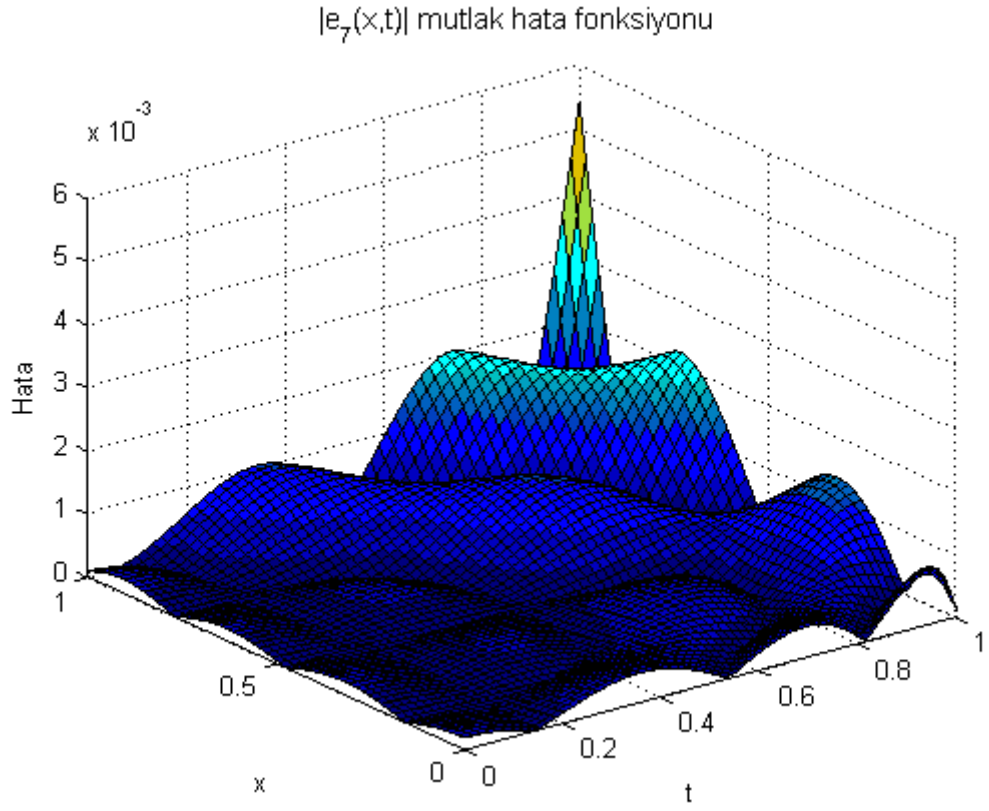
Şekil 4.10. Örnek 4.1.3' ün $u(x,t)$ tam çözümü, $u_7(x,t)$ Bessel polinom çözümü ve $u_{7,8}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması

Çizelge 4.4. Örnek 4.1.3' ün $u(x_i, 0.5)$ için sunulan yöntemin mutlak hatalarının sonlu farklar metodu ile karşılaştırılması

x_i	Sonlu farklar metodu (Dehghan, 2005)	Sunulan yöntem $N = 7, e_7(x_i, 0.5)$
0.1	1.5e-003	2.8149e-004
0.2	1.4e-003	2.5674e-004
0.3	1.7e-003	1.6368e-004
0.4	1.6e-003	6.4767e-005
0.5	1.5e-003	5.5188e-005
0.6	1.5e-003	2.5172e-004
0.7	1.9e-003	5.3196e-004
0.8	1.8e-003	7.8042e-004
0.9	1.7e-003	7.6521e-004
1	1.6e-003	2.6242e-004



Şekil 4.11. Örnek 4.1.3 için $|e_7(x,t)|$ mutlak hata fonksiyonu ve $|e_{7,8}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.12. Örnek 4.1.3 için $|e_7(x,t)|$ mutlak hata fonksiyonu ve $|E_{7,8}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

4.2. Neuman, Dirichlet ve Robin Koşullu İkinci Mertebe Lineer Kısmi Diferansiyel Denklem Problemi İçin Örnekler

Örnek 4.2.1: İlk olarak,

$$\frac{\partial^2 u}{\partial x^2} + 2xt \frac{\partial^2 u}{\partial t \partial x} - x \frac{\partial^2 u}{\partial t^2} + tu = (4 + 4xt - x + t)e^{2x+t}, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1$$

ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklemini $\mathbf{n} = (1, 0)$ ve $\bar{\mathbf{n}} = (0, 1)$ birim normal vektörler olmak üzere

$$u(x, 0) = e^{2x}, \quad 0 \leq x \leq 1,$$

$$u(0, t) = e^t, \quad 0 \leq t \leq 1,$$

Dirichlet sınır koşulları,

$$\frac{\partial u}{\partial \mathbf{n}}(0, t) = 2e^t, \quad 0 \leq t \leq 1,$$

Neumann sınır koşulu ve

$$2u(x, 1) + \frac{\partial u}{\partial \mathbf{n}}(x, 0) = 2e^{2x+1} + e^{2x}, \quad 0 \leq x \leq 1$$

Robin sınır koşulu ile ele alıyoruz. Bu problemin tam çözümü $u(x, t) = e^{2x+t}$ dir.

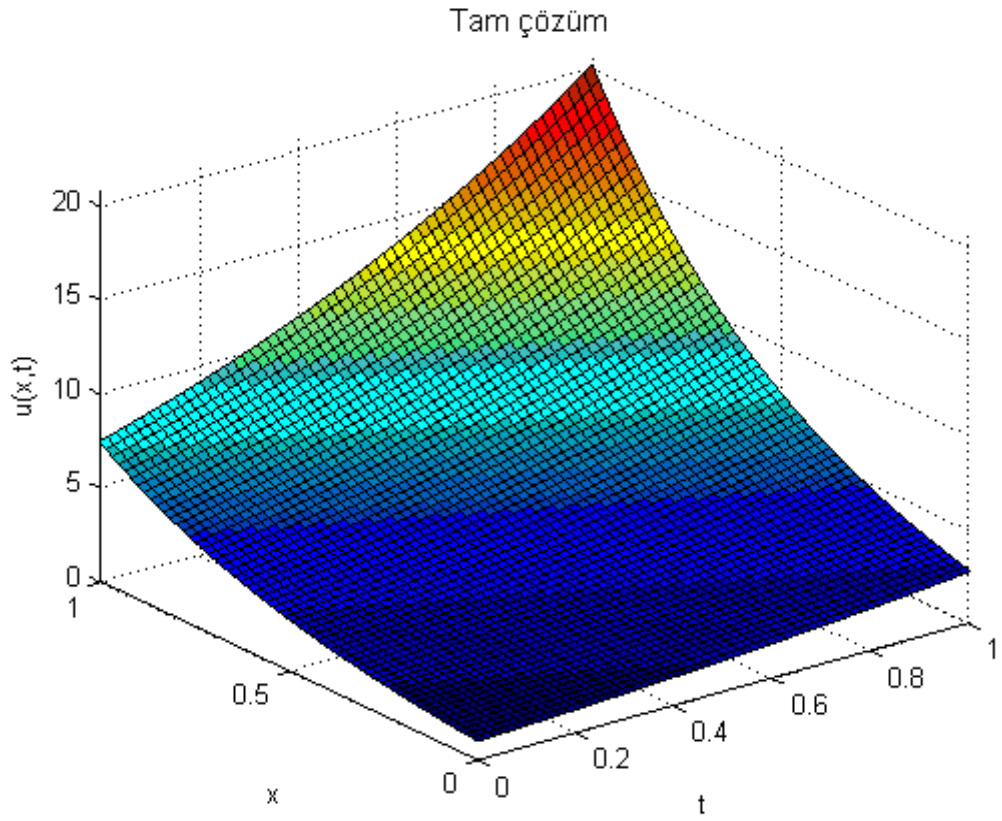
Bu problem için Bessel sıralama metodu farklı N değerleri için uygulanarak yaklaşık çözümler elde edildi. Yaklaşık çözümler Maple' da yazılan bilgisayar kodunda virgülden sonra 30 hane kullanılarak hesaplandı. $u(x, t)$ tam çözümü, $u_{12}(x, t)$ Bessel polinom çözümü ve $u_{12,13}(x, t)$ iyileştirilmiş Bessel polinom çözümü Şekil 4.13' de karşılaştırılır. Şekil 4.14 $|e_{12}(x, t)|$ gerçek mutlak hata fonksiyonu, $|e_{12,13}(x, t)|$ ve $|e_{12,14}(x, t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılmasını gösterir. $|e_{12}(x, t)|$ mutlak hata fonksiyonu, $|E_{12,13}(x, t)|$ ve

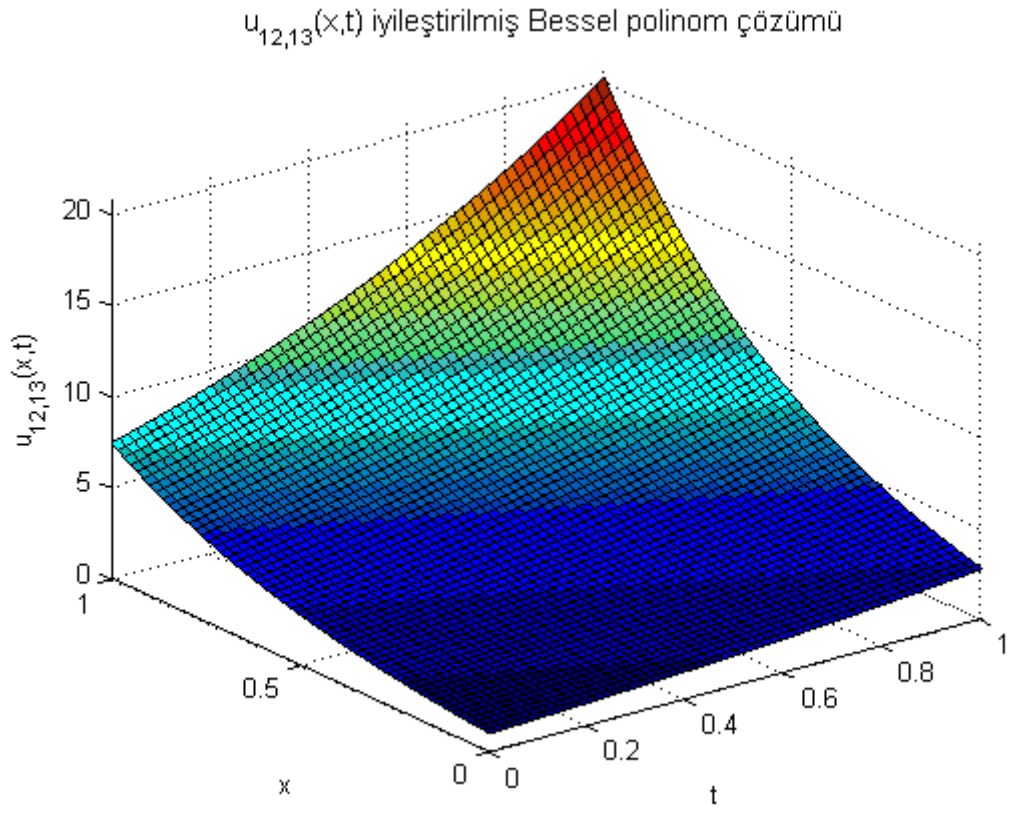
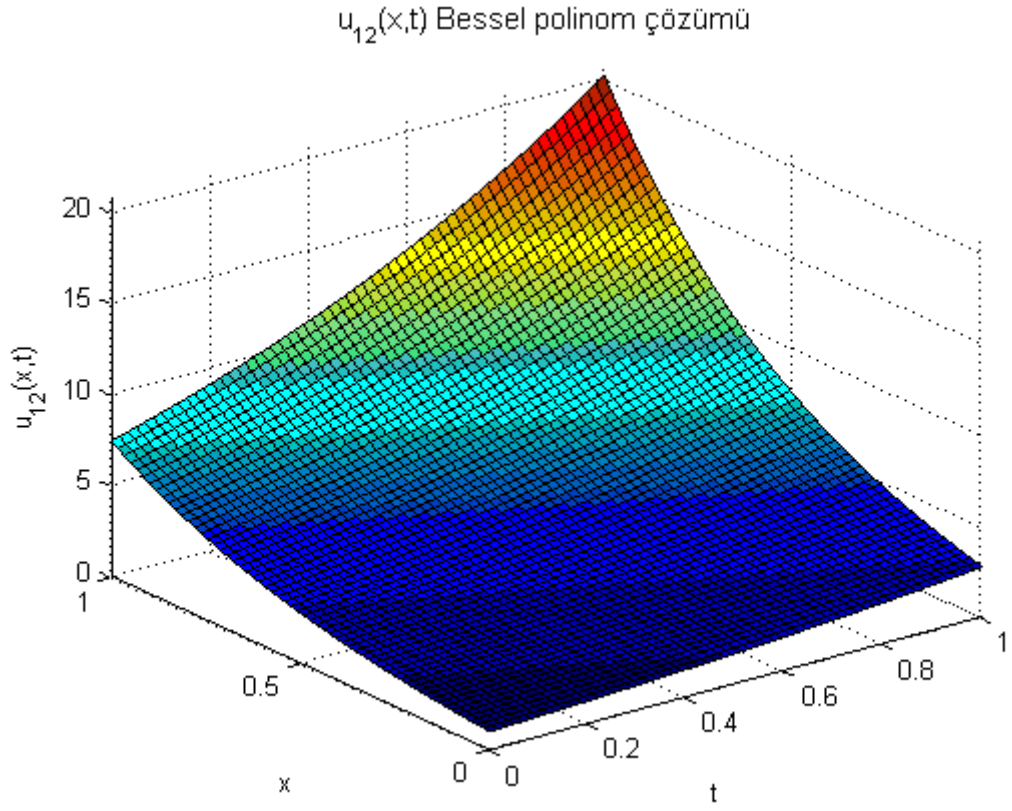
$|E_{12,14}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonları Şekil 4.15’ de karşılaştırılır.

Çizelge 4.5 $N = 5, 6, 7, 8, 9, 10$ için L_2 ve L_∞ hatalarını gösterir. N değeri artarken L_2 ve L_∞ hatalarının azaldığı Çizelge 4.5’ den görülür. Ayrıca N ve M değerleri artarken mutlak hataların azaldığı Şekil 13, Şekil 14 ve Şekil 15’ den gözlemlenir.

Çizelge 4.5. Örnek 4.2.1’ in L_2 ve L_∞ hataları

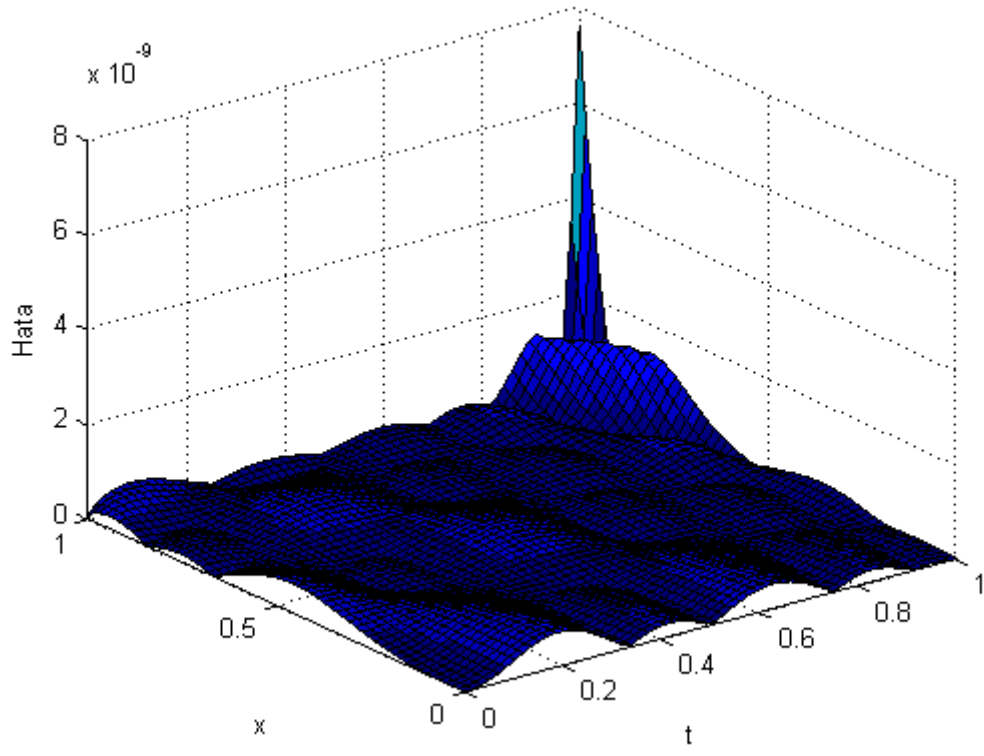
N	5	6	7	8	9	10
L_2	1.5052×10^{-2}	3.0753×10^{-3}	1.6777×10^{-4}	2.5707×10^{-5}	1.0080×10^{-6}	1.0080×10^{-6}
L_∞	9.3530×10^{-2}	3.2845×10^{-2}	1.9521×10^{-3}	3.3274×10^{-4}	7.0386×10^{-6}	5.4454×10^{-6}



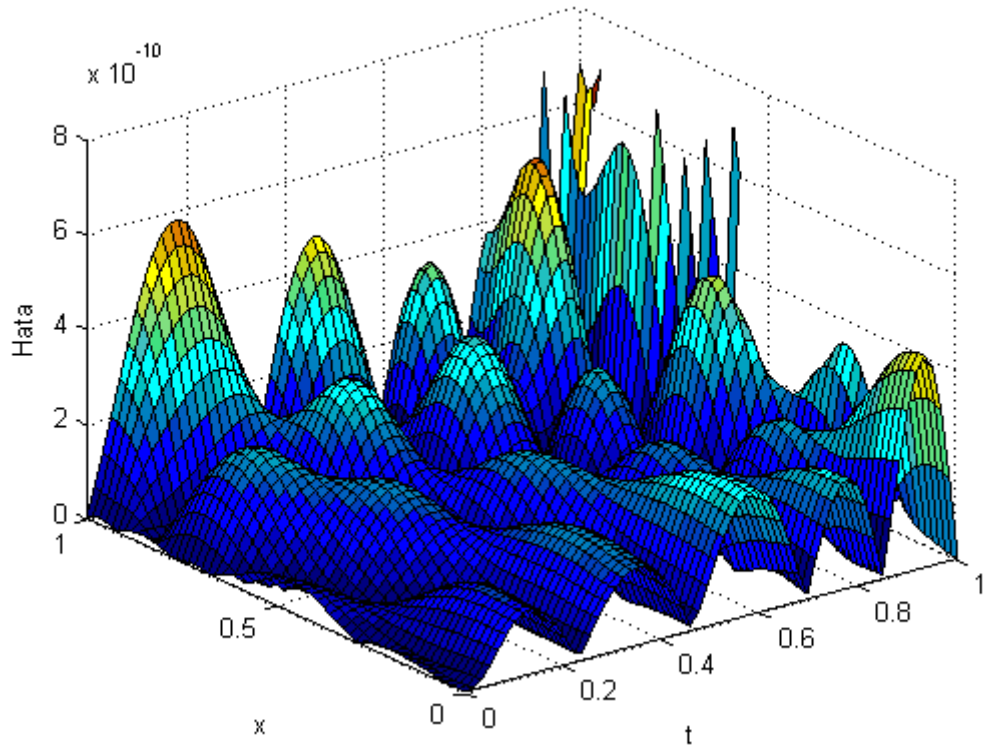


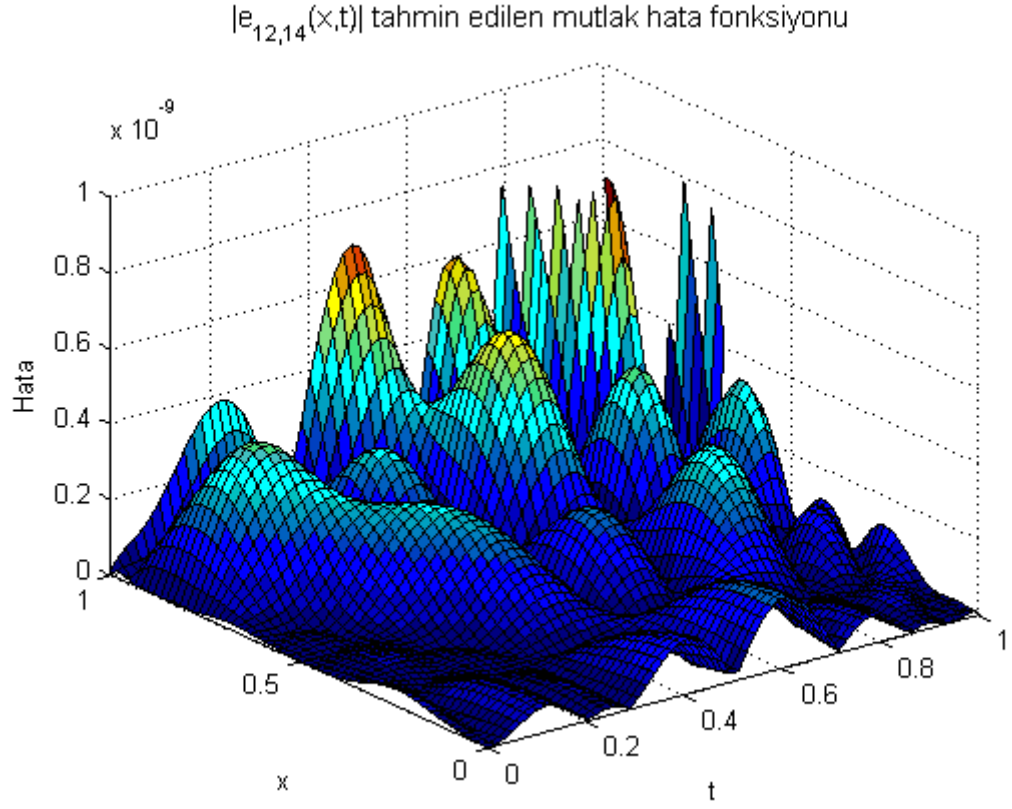
Şekil 4.13. Örnek 4.2.1' in $u(x,t)$ tam çözümü, $u_{12}(x,t)$ Bessel polinom çözümü ve $u_{12,13}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması

$|e_{12}(x,t)|$ mutlak hata fonksiyonu

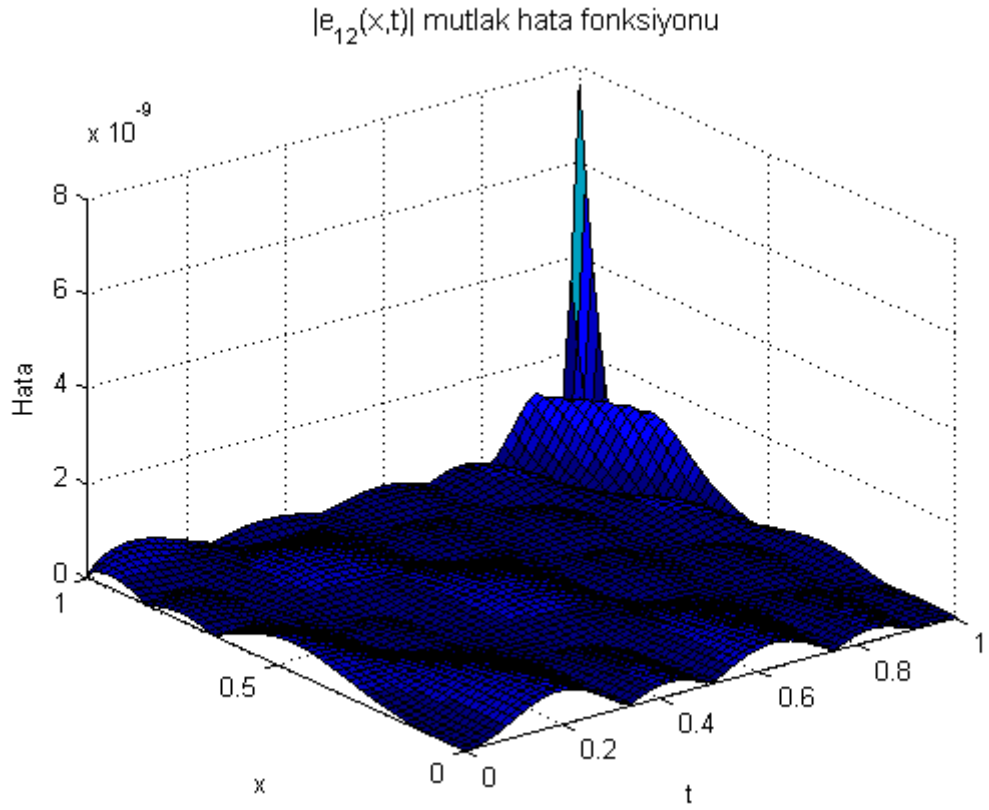


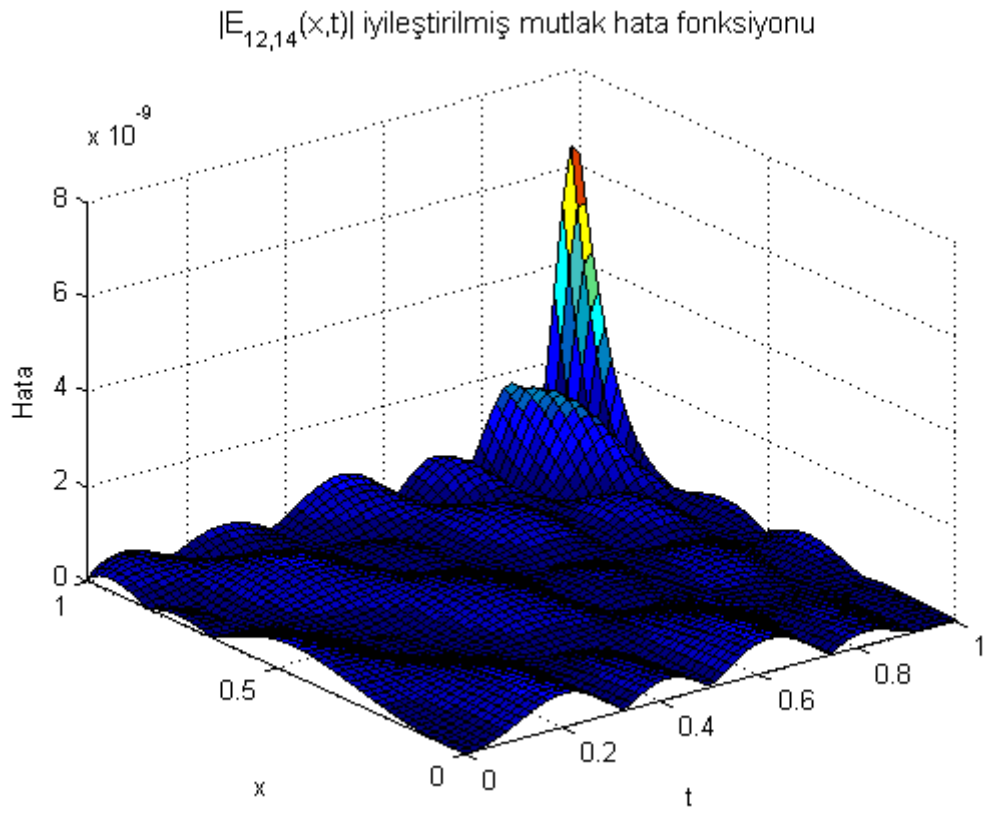
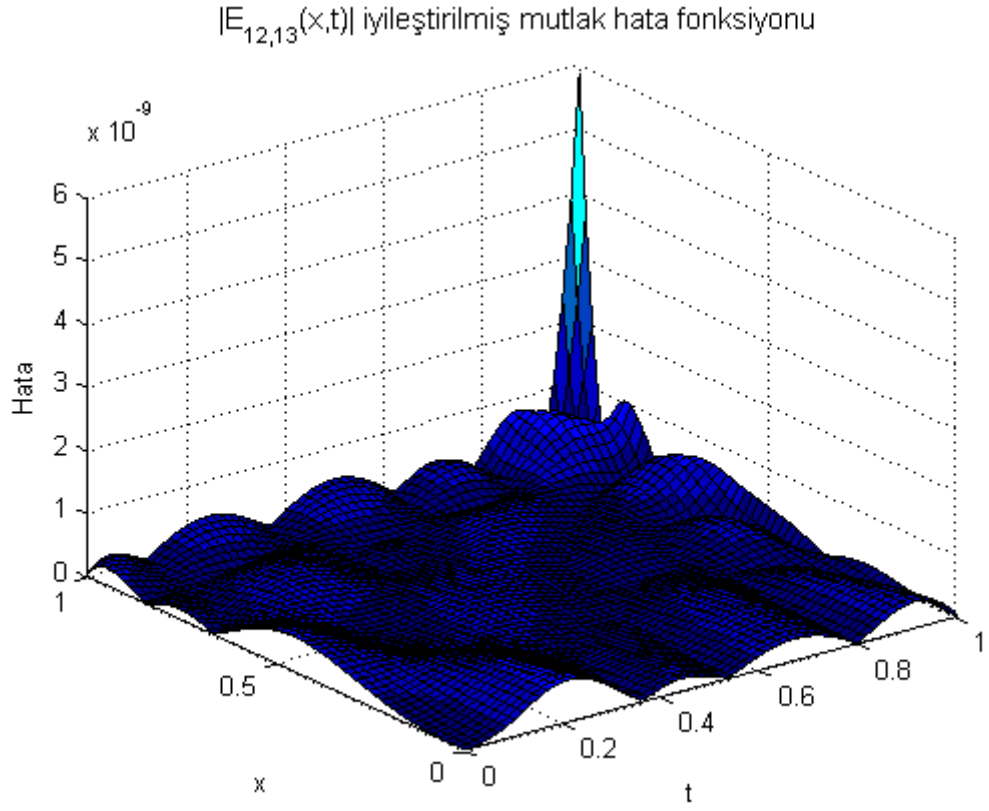
$|e_{12,13}(x,t)|$ tahmin edilen mutlak hata fonksiyonu





Şekil 4.14. Örnek 4.2.1' in $N = 12$, $M = 13$ ve $M = 14$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması





Şekil 4.15. Örnek 4.2.1' in $N = 12$, $M = 13$ ve $M = 14$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{NM}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonlarının karşılaştırılması

Örnek 4.2.2: Şimdi,

$$\frac{\partial^2 u}{\partial x^2} + 2 \frac{\partial^2 u}{\partial t \partial x} + 2x e^t \frac{\partial^2 u}{\partial t^2} - x \frac{\partial u}{\partial x} - t \sin(x) u = e^t \cos(x) [-1 + 2x e^t] + e^t \sin(x) [x - 2 - t \cos(x)],$$

$$0 \leq x \leq 2, \quad 0 \leq t \leq 3,$$

ikinci mertebeden değişken katsayılı lineer kısmi diferansiyel denklemini $\mathbf{n} = (0, -1)$

ve $\bar{\mathbf{n}} = (1, 0)$ birim normal vektörler olmak üzere

$$u(x, 0) = \cos(x), \quad 0 \leq x \leq 2,$$

$$u(2, t) = \cos(2) e^t, \quad 0 \leq t \leq 3,$$

Dirichlet sınır koşulları,

$$\frac{\partial u}{\partial \mathbf{n}}(0, t) = -e^t, \quad 0 \leq t \leq 3$$

Neumann sınır koşulu ve

$$u(x, 3) - \frac{\partial u}{\partial \bar{\mathbf{n}}}(x, 0) = e^3 \cos(x) + \sin(x), \quad 0 \leq x \leq 2$$

Robin sınır koşulu altında çözelim. $u(x, t) = \cos(x) e^t$ problemin tam çözümüdür.

N ve M 'nin çeşitli değerleri için yöntem uygulanarak Bessel polinom çözümleri hesaplanır ve bu çözümler için hata fonksiyonları tahmin edilir. Çizelge 4.6' da (N, M) 'nin farklı değerleri için gerçek maksimum mutlak hatalar (L_∞) ve tahmin edilen maksimum mutlak hatalar karşılaştırılır. N ve M değerleri artarken mutlak hataların azaldığı Çizelge 4.6' dan gözlemlenir. Yaklaşık çözümler ve hata fonksiyonu Maple' da yazılan bilgisayar kodu ile hesaplanırken virgülden sonra 30 hanede duyarlılık kullanıldı.

Çizelge 4.6. Örnek 4.2.2' nin gerçek maksimum hataları (L_{∞}) ve tahmin edilen maksimum mutlak hatalarının karşılaştırılması

Bessel polinom çözümleri için maksimum mutlak hatalar (L_{∞})							
N	6	7	8	9	10	11	12
Hata	1.68×10^{-1}	1.4×10^{-2}	7.2×10^{-4}	1.8×10^{-4}	4.8×10^{-6}	2.2×10^{-6}	1.28×10^{-7}
Tahmin edilen maksimum mutlak hatalar							
(N, M)	(6, 7)	(7, 8)	(8, 10)	(9, 10)	(10, 11)	(11, 12)	(12, 13)
Hata	1.8×10^{-1}	5.6×10^{-3}	3.4×10^{-4}	7.7×10^{-5}	5.0×10^{-6}	7.6×10^{-7}	3.1×10^{-7}
Bessel polinom çözümleri için maksimum mutlak hatalar (L_{∞})							
N	13	14	15	16	17	18	19
Hata	5.0×10^{-10}	1.52×10^{-10}	1.8×10^{-11}	2.2×10^{-13}	2.75×10^{-14}	8.8×10^{-15}	8.2×10^{-16}
Tahmin edilen maksimum mutlak hatalar							
(N, M)	(13, 14)	(14, 15)	(15, 16)	(16, 17)	(17, 18)	(18, 19)	(19, 20)
Hata	2.3×10^{-9}	6.8×10^{-10}	1.6×10^{-12}	1.2×10^{-13}	2.2×10^{-14}	6.4×10^{-16}	7.6×10^{-17}

4.3. Başlangıç ve Sınır Koşullu İkinci Mertebe Lineer Kısmi Diferansiyel Denklem Problemi İçin Örnekler

Örnek 4.3.1: İlk olarak,

$$\frac{\partial^2 u}{\partial x^2} - 2xt \frac{\partial^2 u}{\partial t \partial x} + x^2 t^2 \frac{\partial^2 u}{\partial t^2} + x^2 t \frac{\partial u}{\partial x} - 2 \frac{\partial u}{\partial t} + xu = e^{x+t} (x^2 t^2 - 2xt + x^2 t + x - 1)$$

parabolik tip kısmi diferansiyel denklemini

$$\begin{cases} u(x, 0) = e^x \\ \frac{\partial u(x, 0)}{\partial t} = e^x \end{cases}$$

başlangıç koşulları ve

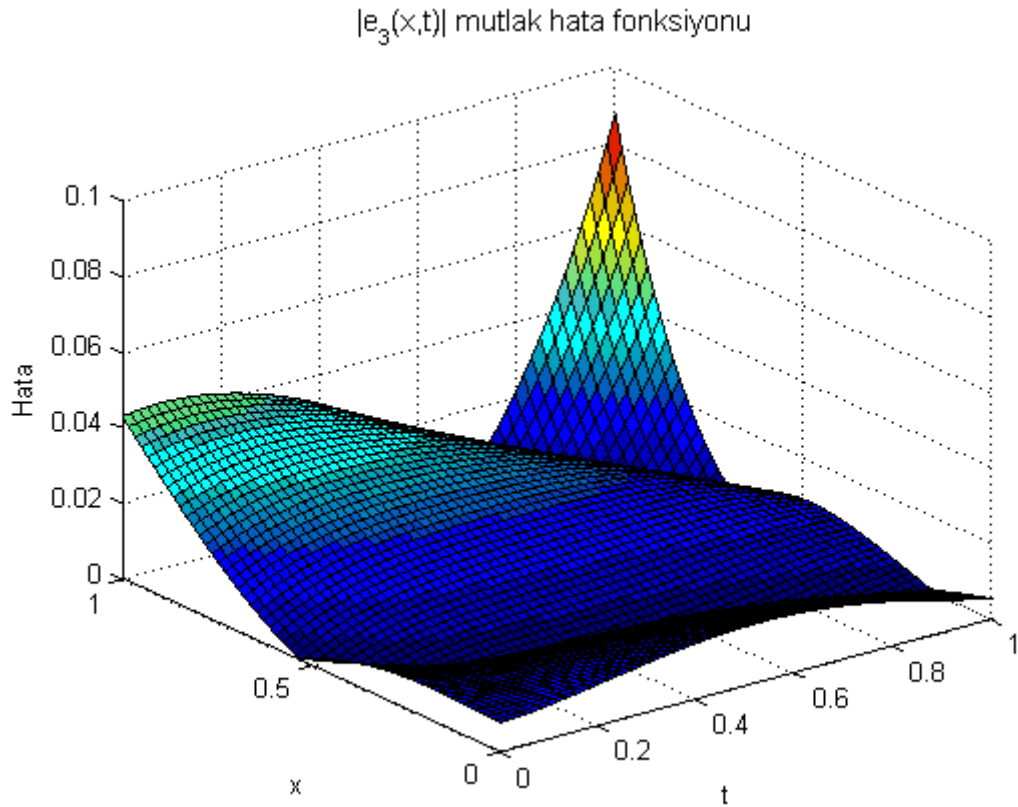
$$\begin{cases} u(0, t) = e^t \\ u(1, t) = ee^t \end{cases}$$

sınır koşulları ile $0 \leq x \leq 1$, $0 \leq t \leq 1$ bölgesinde ele alalım. Problemin tam çözümü $u(x, t) = e^{x+t}$ ' dir.

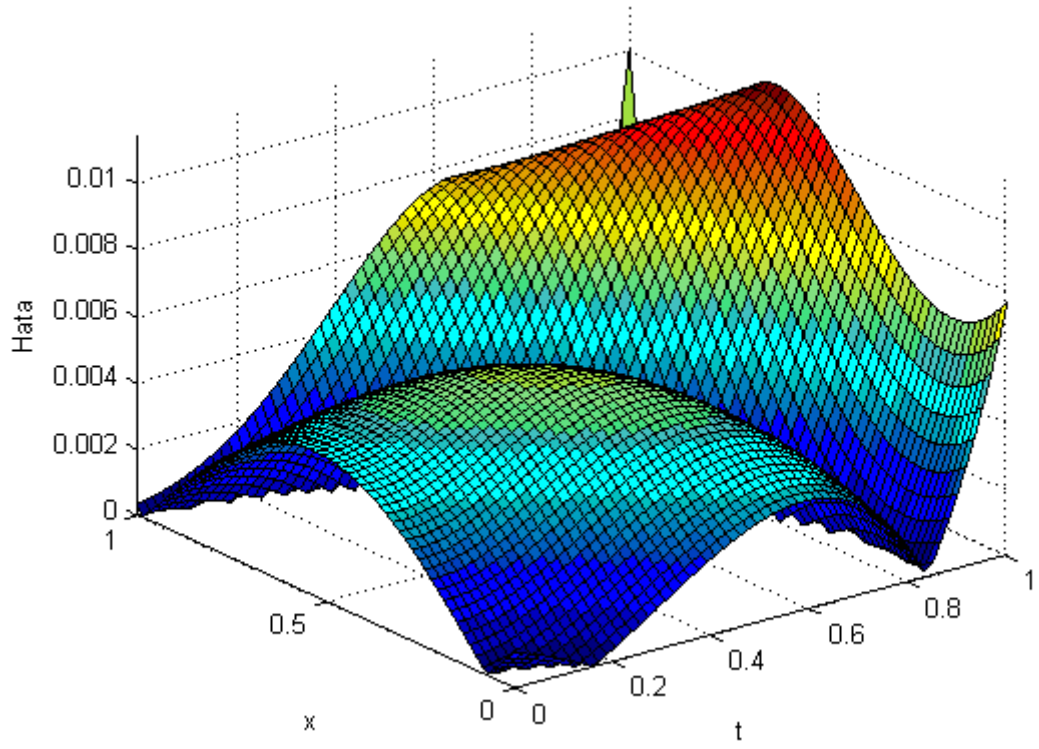
Sunulan metodu $N=3,5,6,8$ için uygulayarak yaklaşık çözümleri hesapladık ve metot Maple’ da yazılan kodda virgülden sonra 30 hane alınarak uygulandı. Çizelge 4.7’ de, L_2 ve L_∞ hataları verilmiştir. Şekil 4.16, Şekil 4.17, Şekil 4.18 ve Şekil 4.19’ da farklı N değerleri için $|e_N(x,t)|$ gerçek mutlak hata fonksiyonları ve farklı (N,M) değerleri için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonları karşılaştırılır. N ve M ’ nin çeşitli değerleri için $|e_N(x,t)|$ mutlak hata fonksiyonları ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonlarının karşılaştırılmaları Şekil 4.20, Şekil 4.21, Şekil 4.22 ve Şekil 4.23’ de sunulur. N değeri artarken hatalarının azaldığı Çizelge 4.7 ve Şekil 4.16-Şekil 4.23’ den görülebilir.

Çizelge 4.7. Örnek 4.3.1’ in L_2 ve L_∞ hataları

	Sunulan metot				
N	3	5	6	8	9
L_2	1.6239×10^{-2}	1.4771×10^{-4}	7.8336×10^{-6}	4.5073×10^{-7}	4.1336×10^{-10}
L_∞	8.88×10^{-2}	5.4×10^{-4}	4.9×10^{-5}	1.65×10^{-7}	4.5×10^{-9}

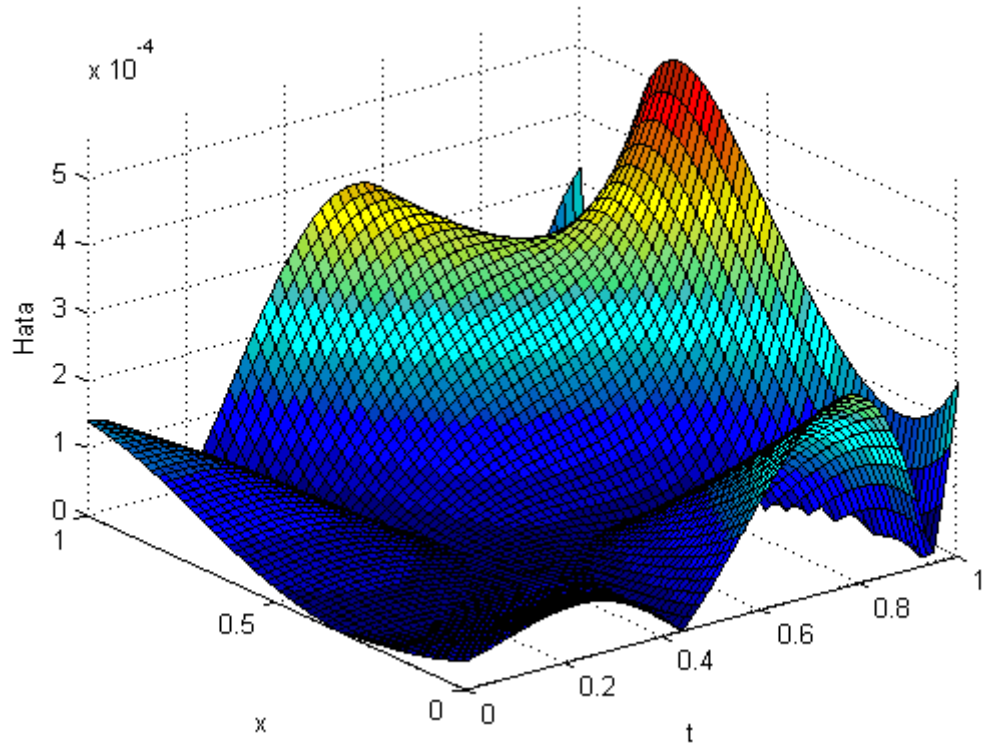


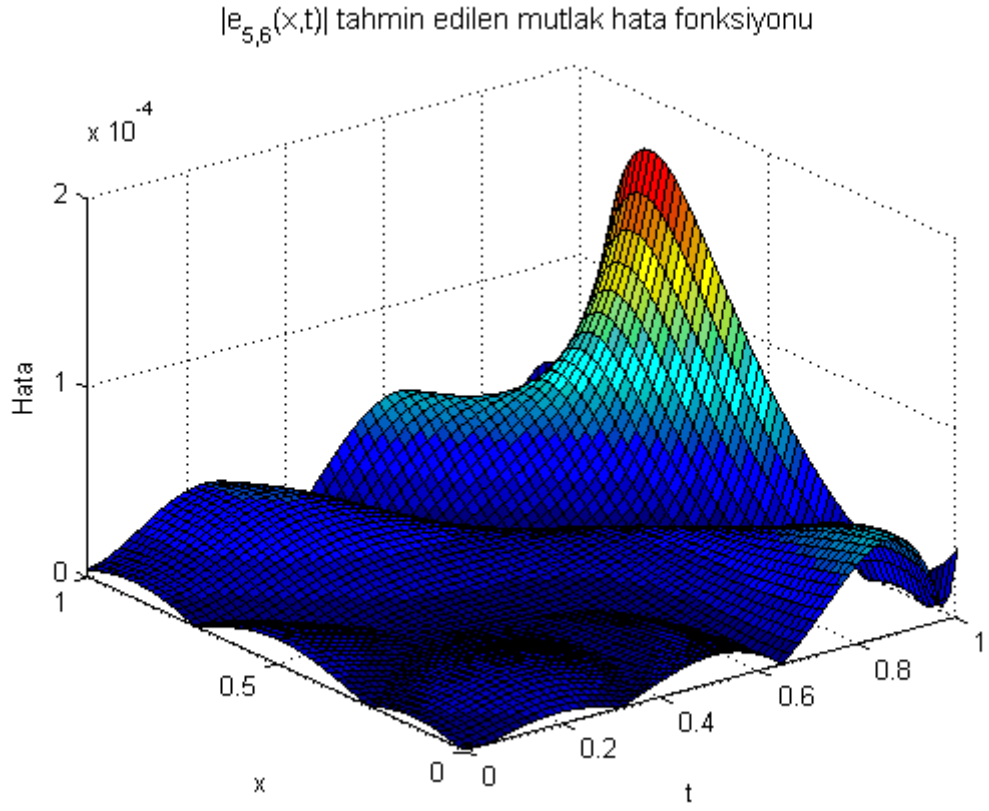
$|e_{3,4}(x,t)|$ tahmin edilen mutlak hata fonksiyonu



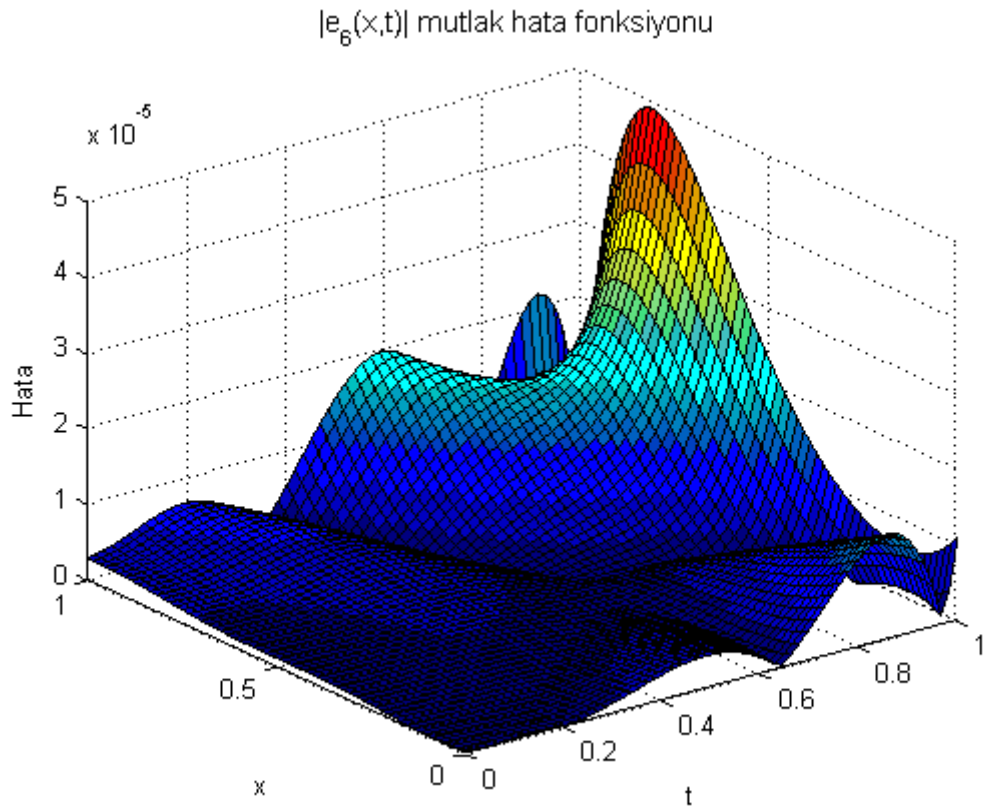
Şekil 4.16. Örnek 4.3.1' in $N = 3$ ve $M = 4$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

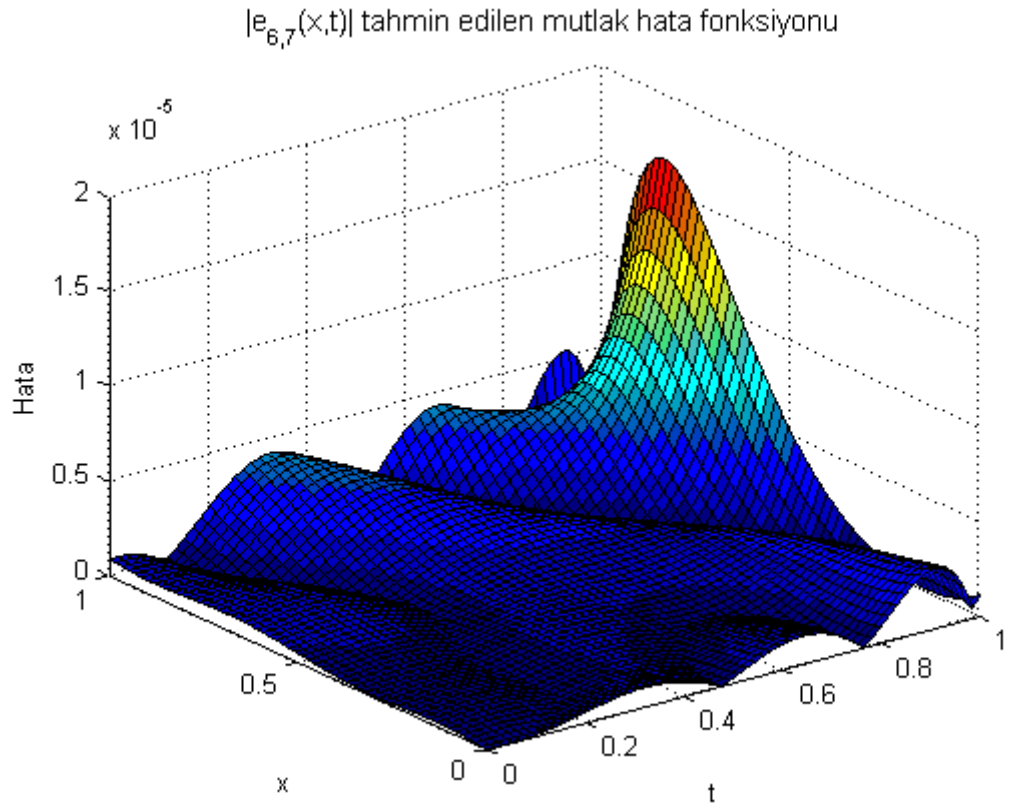
$|e_5(x,t)|$ mutlak hata fonksiyonu



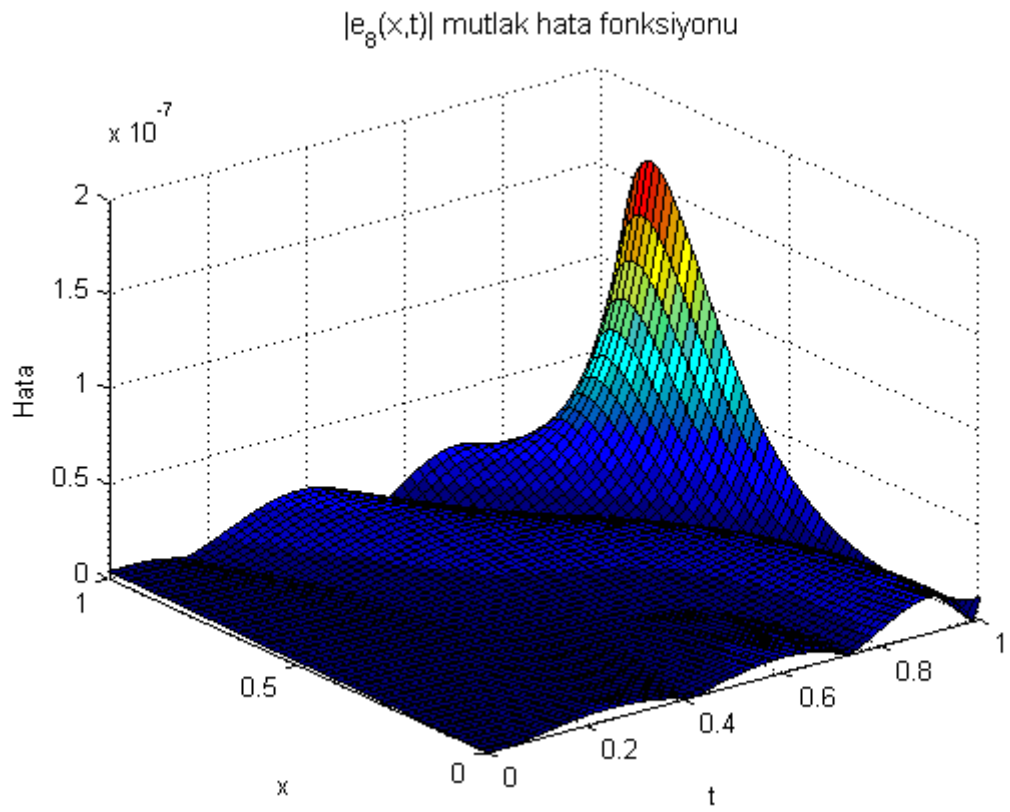


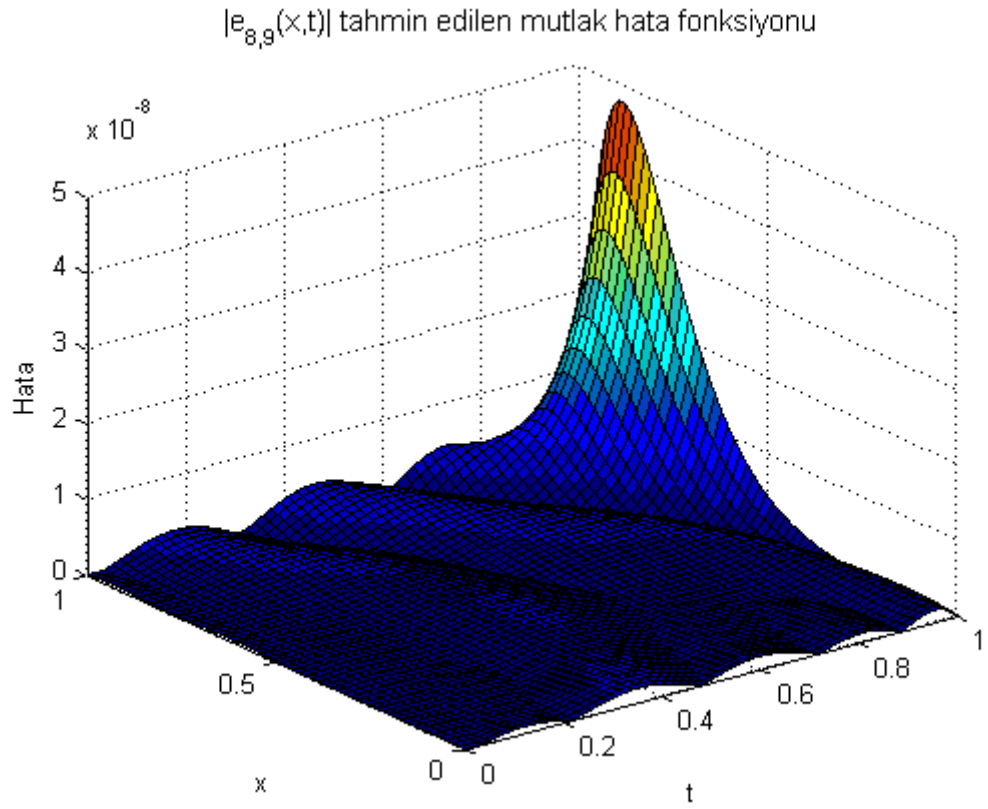
Şekil 4.17. Örnek 4.3.1' in $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{NM}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması.



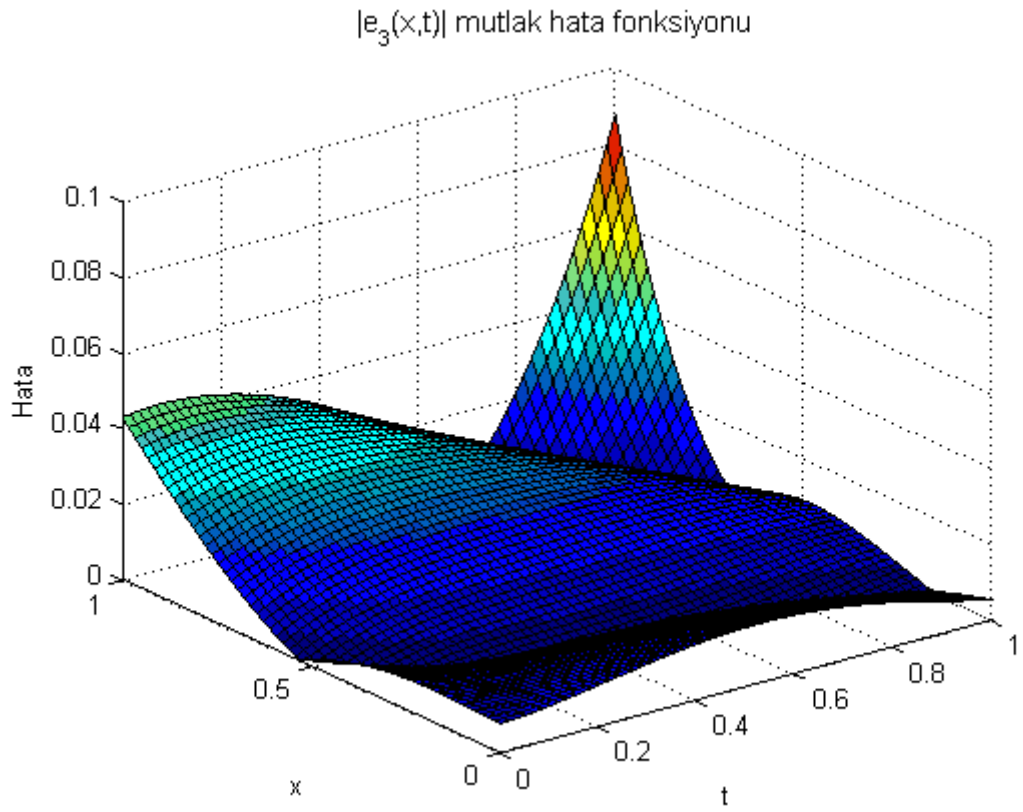


Şekil 4.18. Örnek 4.3.1' in $N = 6$ ve $M = 7$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

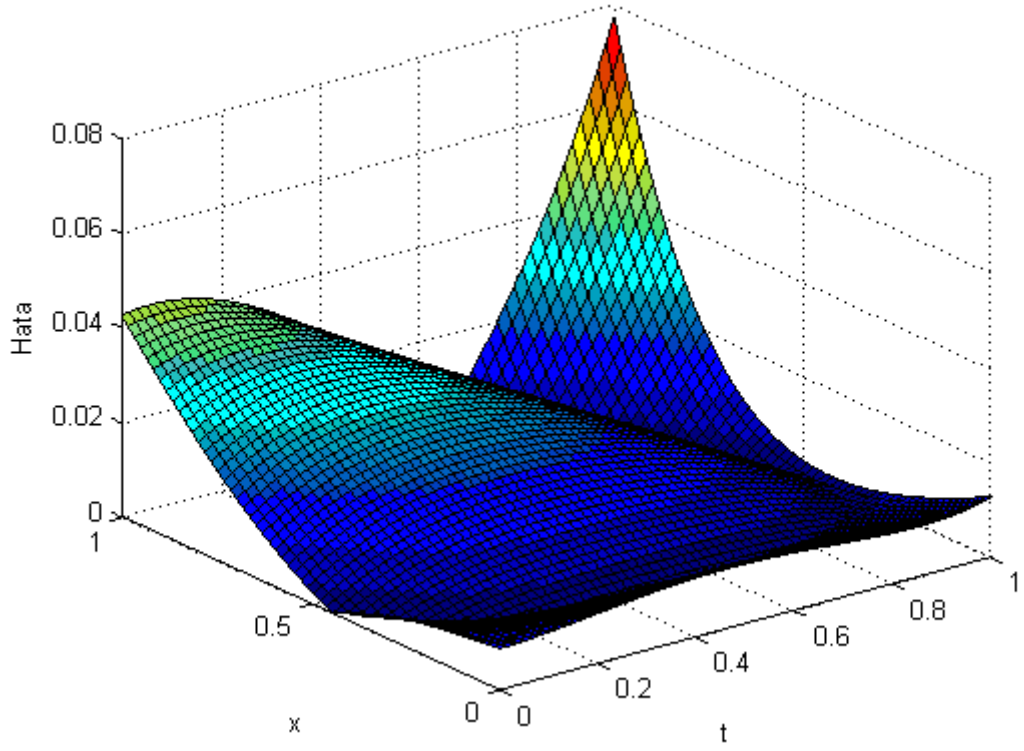




Şekil 4.19. Örnek 4.3.1' in $N = 8$ ve $M = 9$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{NM}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

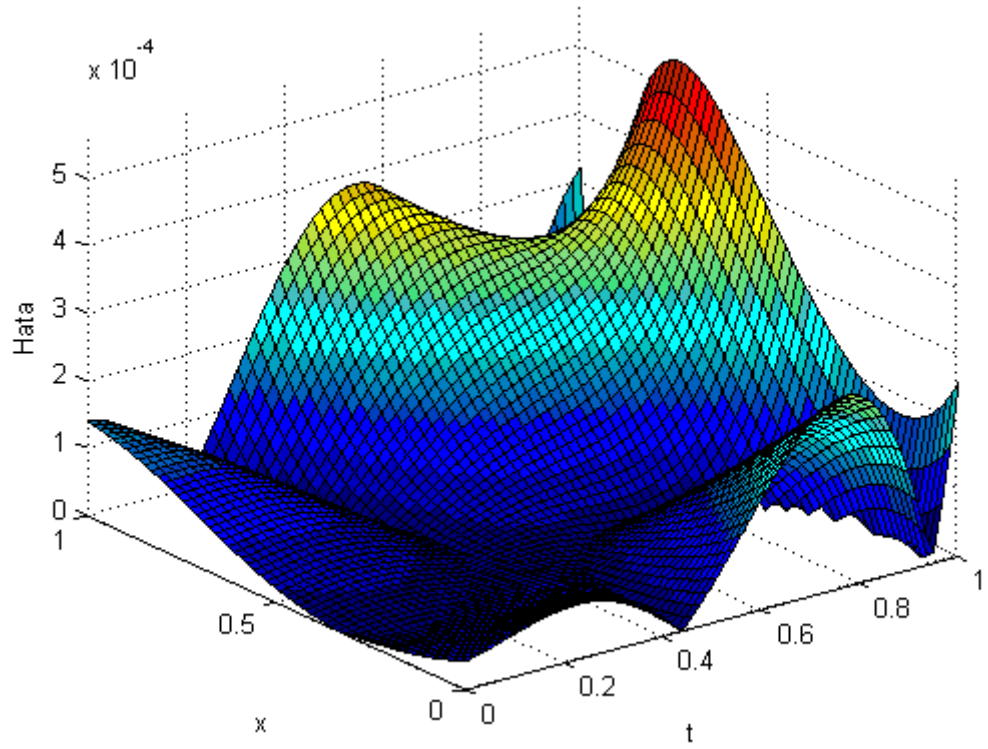


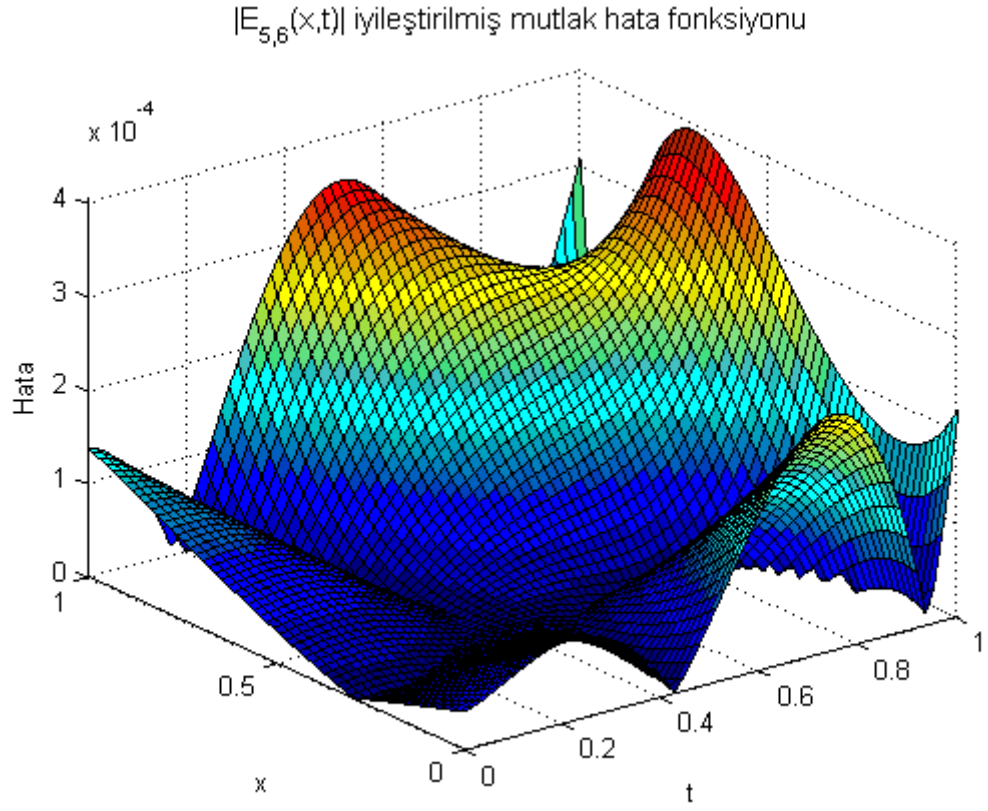
$|E_{3,4}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonu



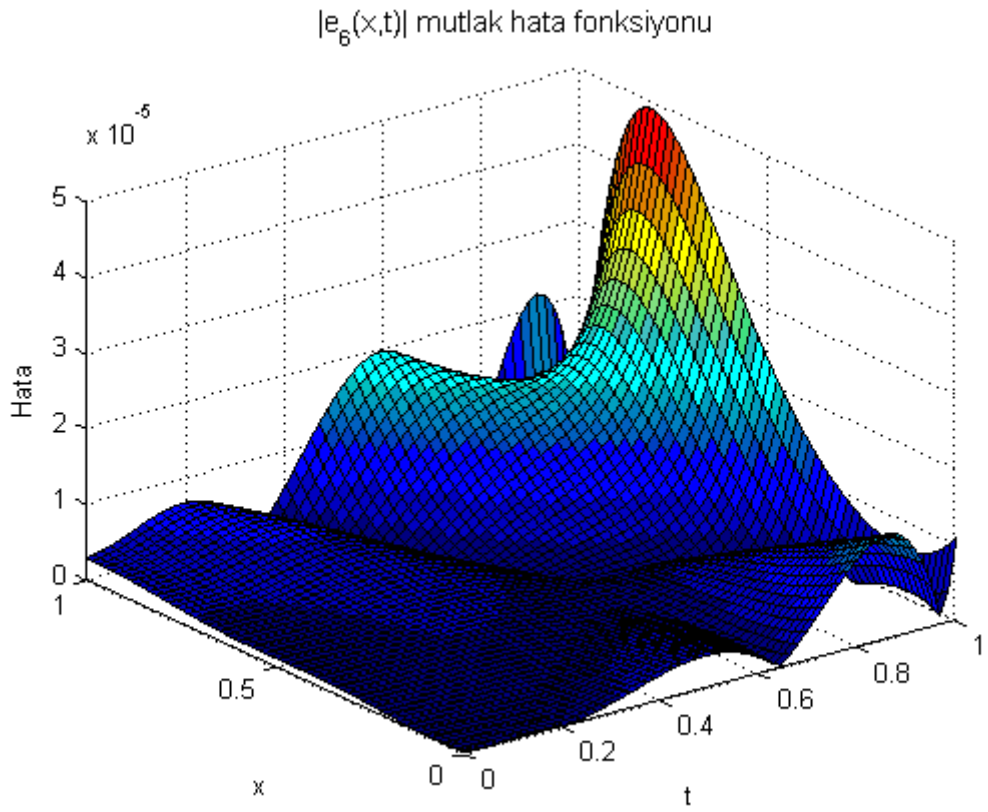
Şekil 4.20. Örnek 4.3.1' in $N = 3$ ve $M = 4$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

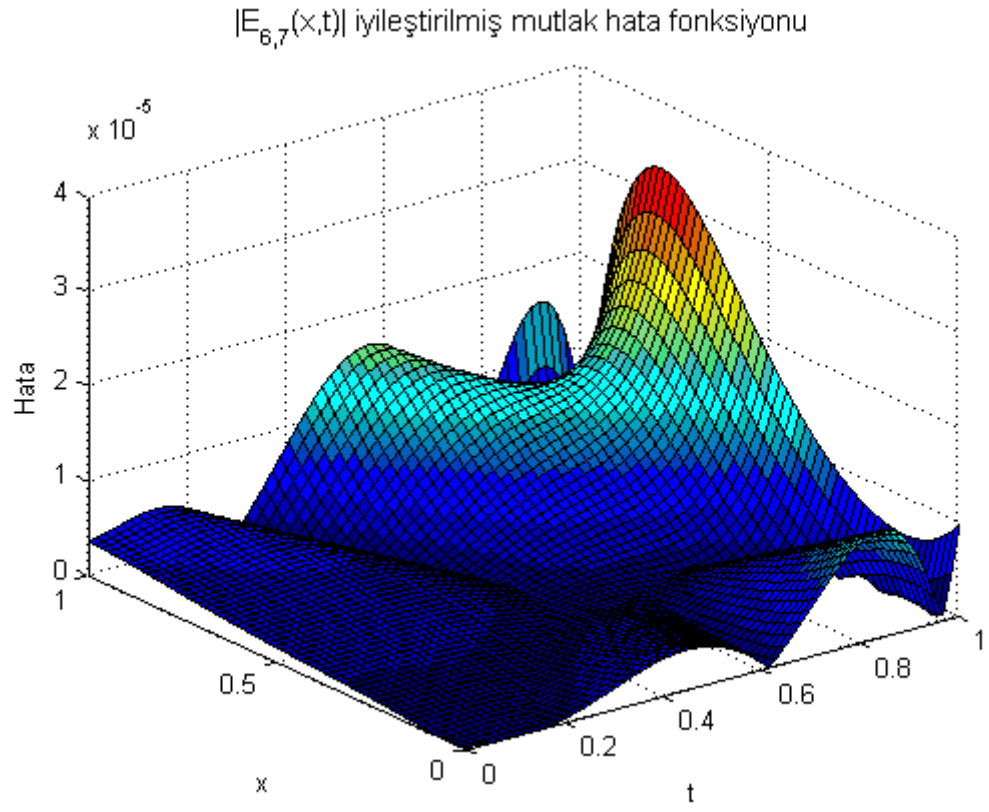
$|e_5(x,t)|$ mutlak hata fonksiyonu



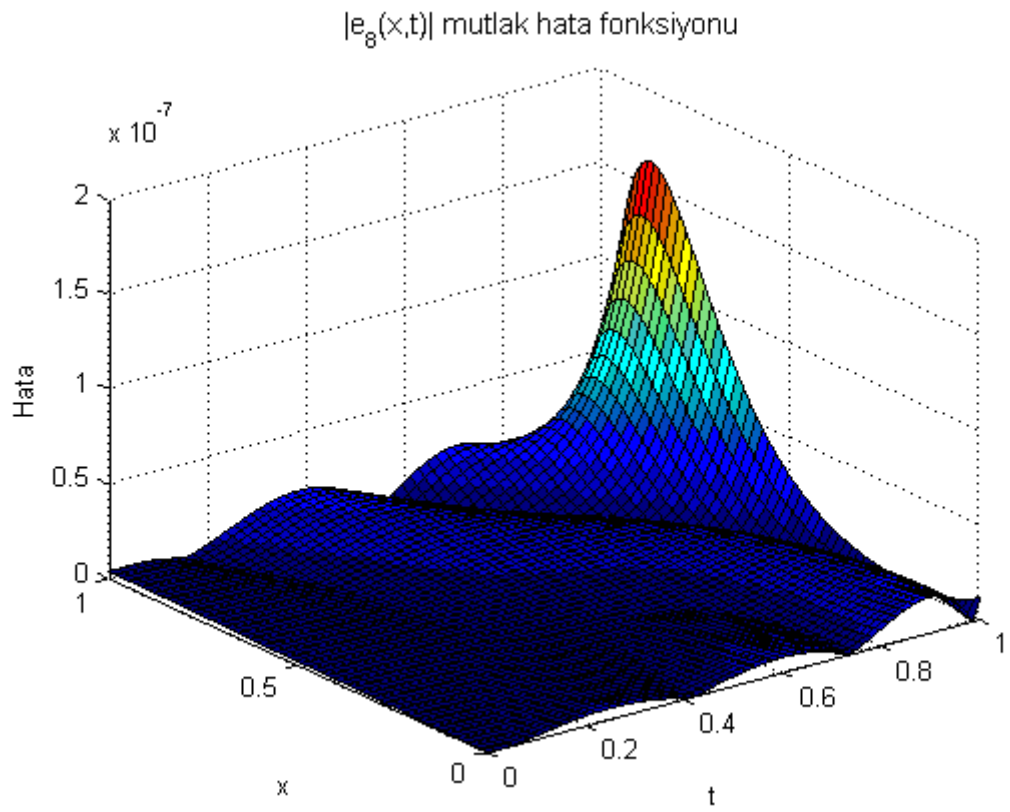


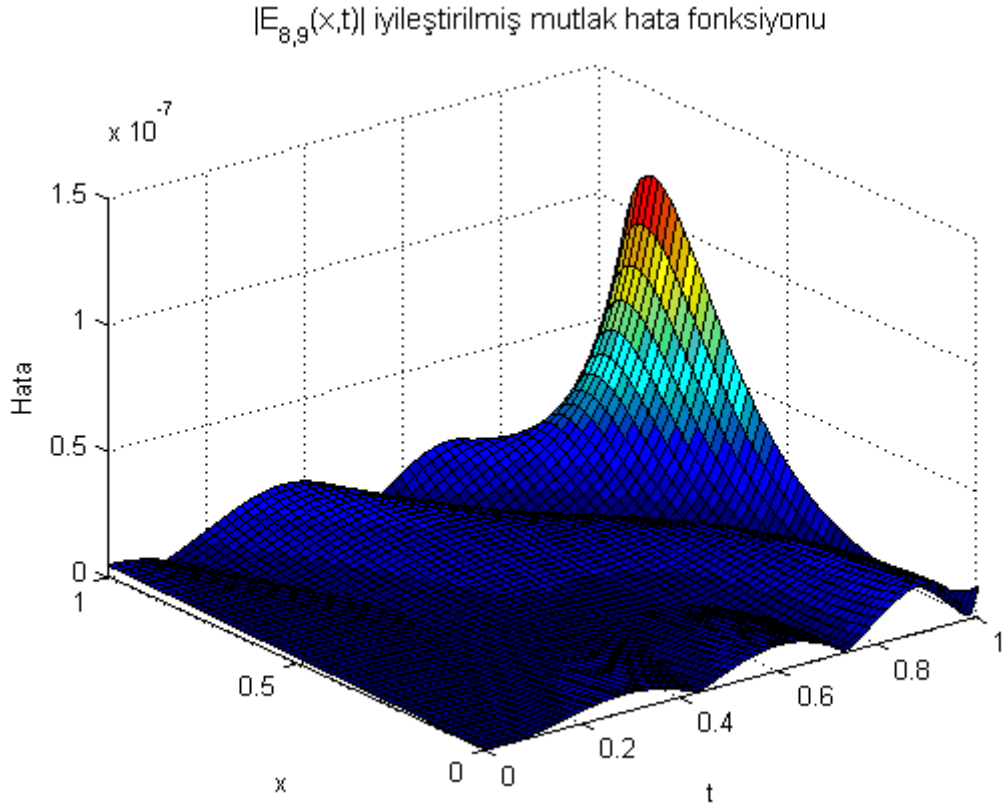
Şekil 4.21. Örnek 4.3.1' in $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması





Şekil 4.22. Örnek 4.3.1' in $N = 6$ ve $M = 7$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması





Şekil 4.23. Örnek 4.3.1' in $N = 8$ ve $M = 9$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

Örnek 4.3.2: Bu örneğimizde,

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial u}{\partial t} + u = \frac{\partial^2 u}{\partial x^2} + (2 - 2t + t^2)(x - x^2)e^{-t} + 2t^2e^{-t}$$

hiperbolik tip kısmi diferansiyel denklemini

$$\begin{cases} u(x,0) = 0 \\ \frac{\partial u(x,0)}{\partial t} = 0 \end{cases}$$

başlangıç koşulları ve

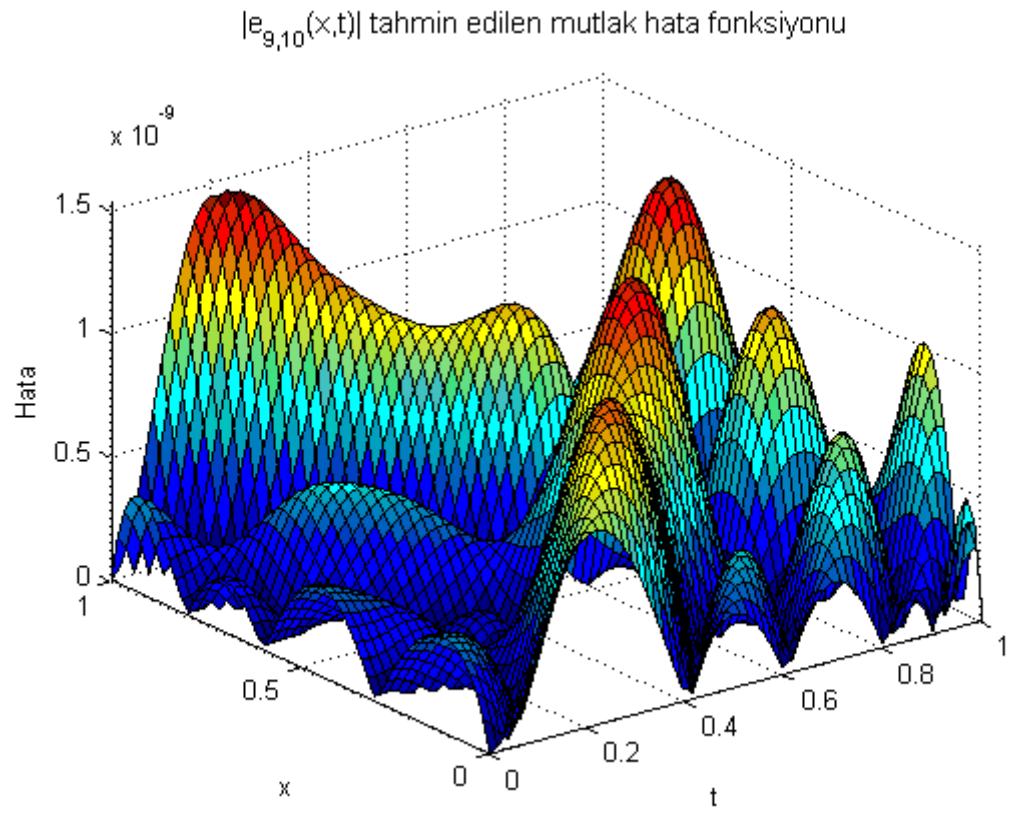
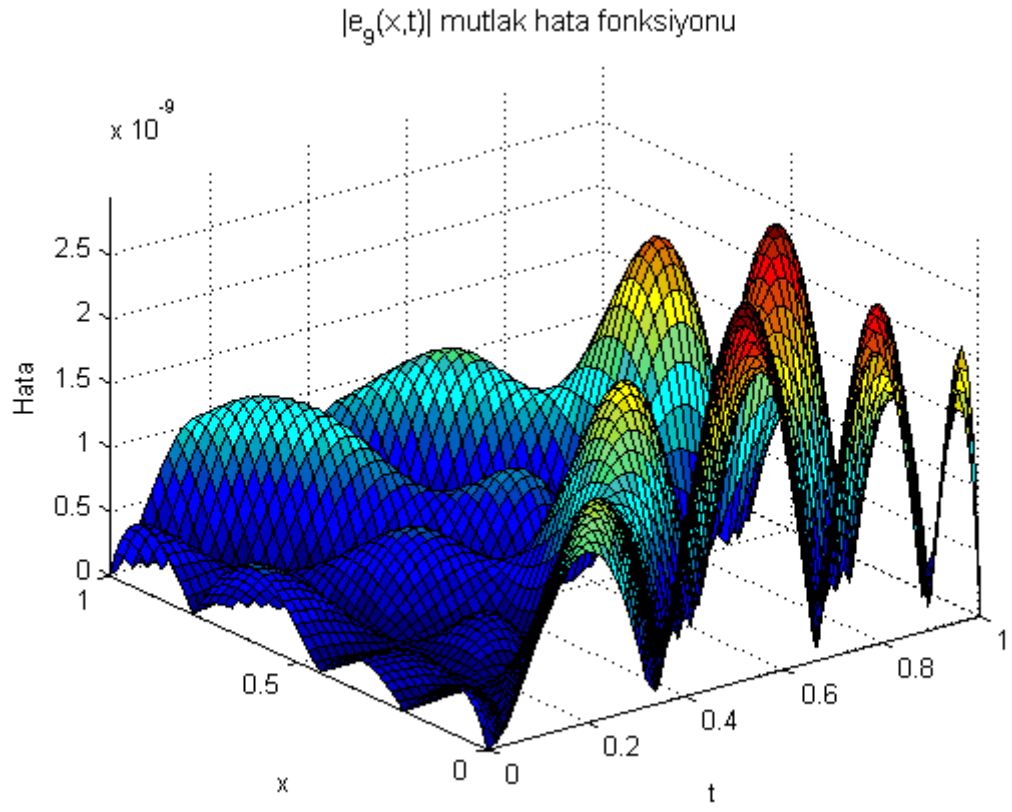
$$\begin{cases} u(0,t) = 0 \\ u(1,t) = 0 \end{cases}$$

sınır koşulları ile $0 \leq x \leq 1$, $0 \leq t \leq 1$ bölgesinde ele alalım (El-Azab ve M. El-Gamel, 2007). Problemin tam çözümü $u(x,t) = t^2(x - x^2)e^{-t}$ ' dir.

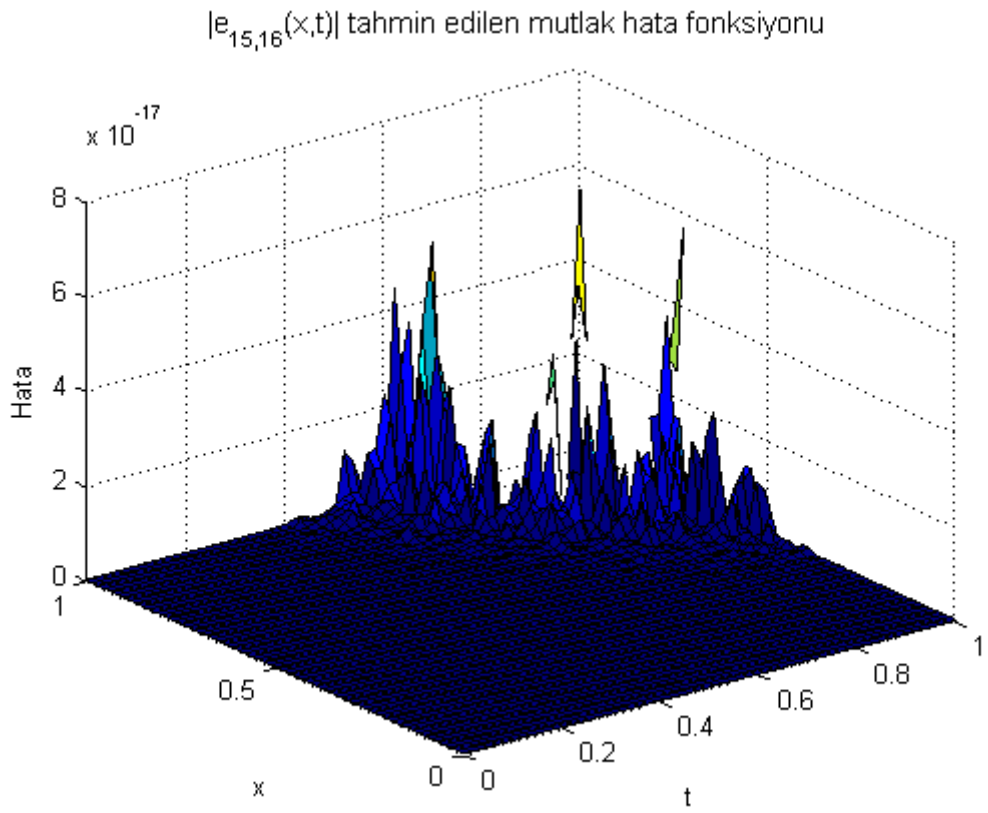
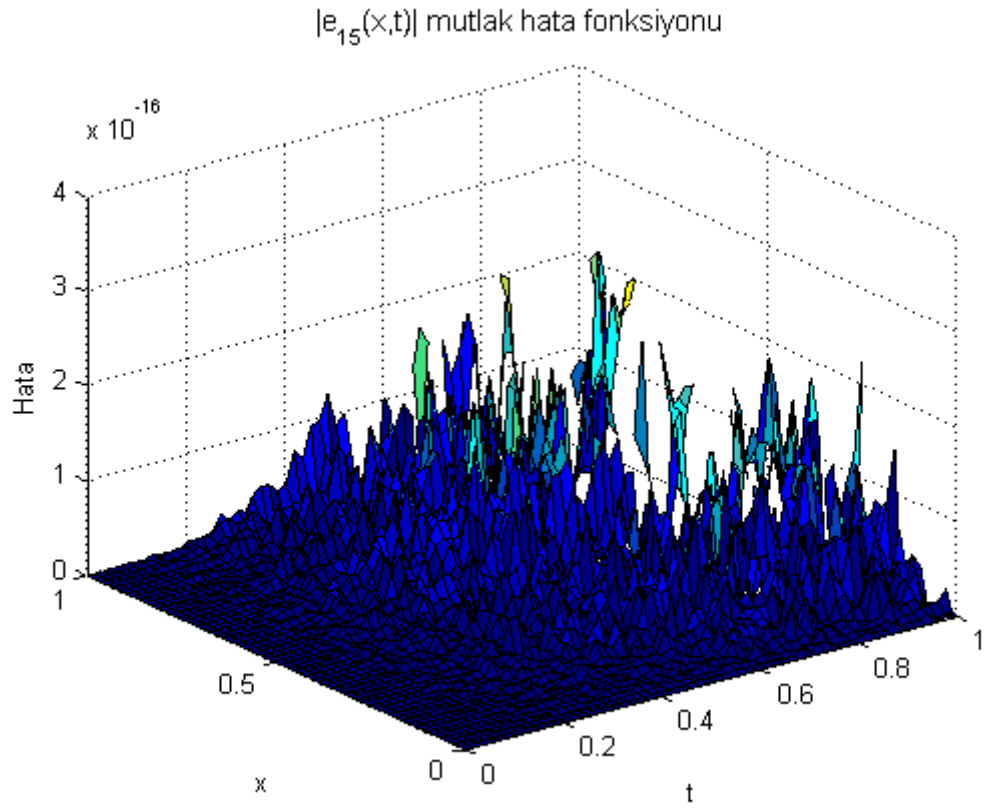
Sunulan yöntemi $N = 5, 7, 9$ için uygulayarak yaklaşık çözümleri hesapladık. Yaklaşık çözümler Maple' da yazılan bilgisayar programı ile virgülden sonra 30 hanede duyarlılıkla elde edildi. Çizelge 4.8' de, L_2 ve L_∞ hataları verilir. Çizelge 4.9' da, $t = 1$ zamanında farklı x değerleri için sunulan metot, Rothe-Wavelet metodu (RWM) (El-Azab ve M. El-Gamel, 2007) ve Chebyshev Tau metodu (CTM) (Saadatmandi ve Dehghan 2010)' nun mutlak hataları karşılaştırılır. Karşılaştırılan noktalarda, sunulan metodun RWM' den daha iyi sonuçlar verdiği gözlemlenir ve sunulan metot ve CTM' nin mutlak hatalarının birbirine çok yakın olduğu görülür. Şekil 4.24 ve Şekil 4.25' de $(N, M) = (9, 10)$ ve $(N, M) = (15, 16)$ için elde edilen $|e_N(x, t)|$ gerçek mutlak hata fonksiyonları ve $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonları karşılaştırılır. Şekil 4.26 $|e_9(x, t)|$ mutlak hata fonksiyonu ve $|E_{9,10}(x, t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırmasını verir. $[0, 1] \times [0, 1]$ bölgesinde, $|e_9(x, t)|$ mutlak hata fonksiyonunun maksimum değeri 2.8264×10^{-9} , $|e_{9,10}(x, t)|$ tahmin edilen mutlak hata fonksiyonunun maksimum değeri 1.4649×10^{-9} ve $|E_{9,10}(x, t)|$ iyileştirilmiş mutlak hata fonksiyonunun maksimum değeri 1.4864×10^{-9} olarak hesaplanır.

Çizelge 4.8. Örnek 4.3.2' nin L_2 ve L_∞ hataları

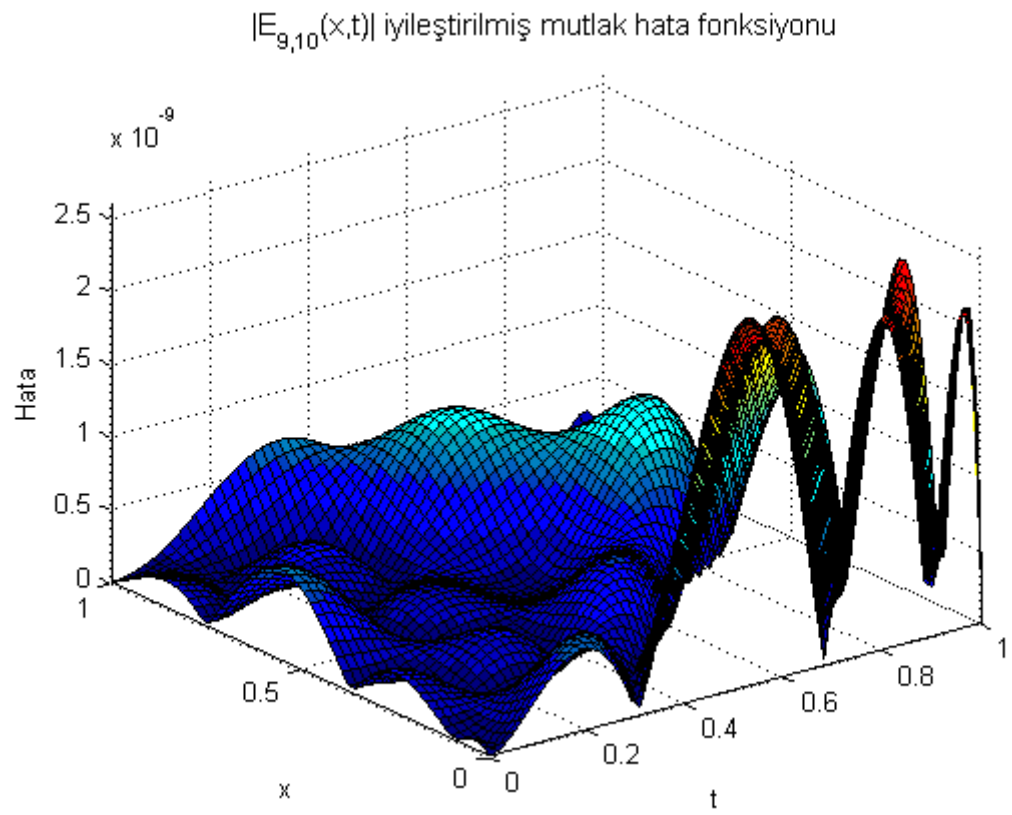
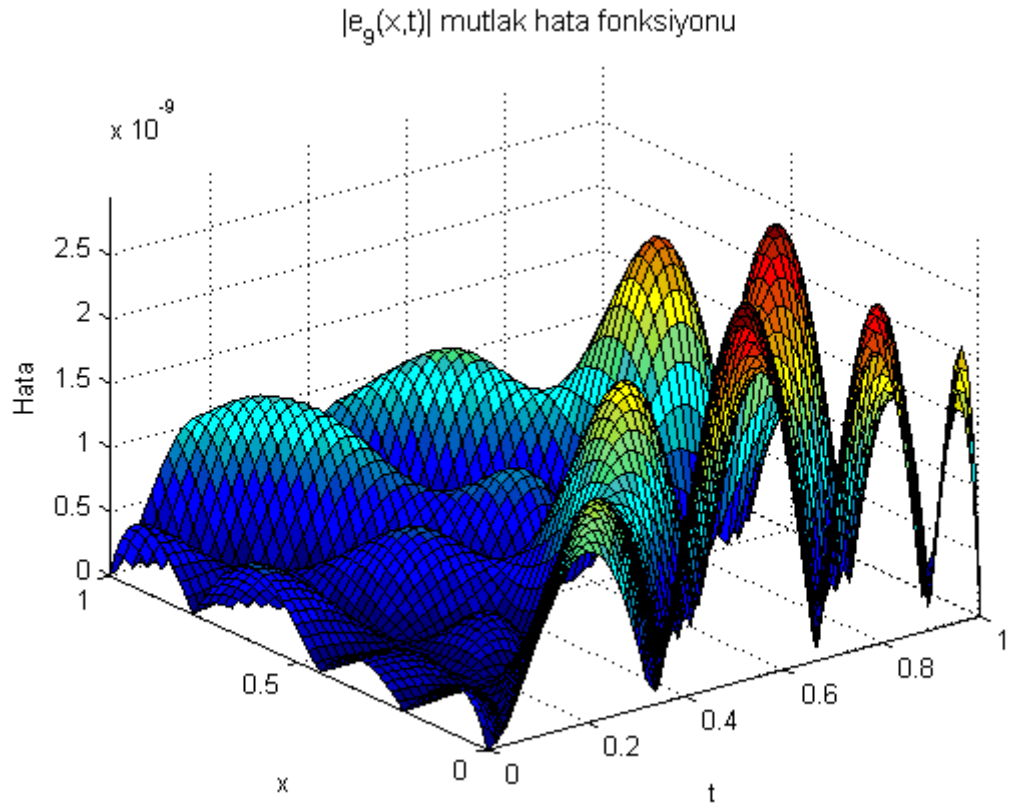
Sunulan metot							
N	5	7	9	N	5	7	9
L_2	1.5×10^{-4}	6.4×10^{-7}	1.99×10^{-9}	L_∞	3.9×10^{-4}	1.4×10^{-6}	2.8×10^{-9}



Şekil 4.24. Örnek 4.3.2' nin $N = 9$ ve $M = 10$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.25. Örnek 4.3.2' nin $N = 15$ ve $M = 16$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.26. Örnek 4.3.2' nin $N = 9$ ve $M = 10$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

Çizelge 4.9. Örnek 4.3.2' nin $u(x_i,1)$ için mutlak hatalarının karşılaştırılması

	RWM (El-Azab ve M. El-Gamel, 2007)	CTM (Saadatmandi ve Dehghan 2010)	Sunulan metot
x_i	$n = 6, m = 9$	$n = m = 7$	$N = 7$
0.125	1.2e-6	5.8e-8	1.25e-7
0.250	1.9e-6	1.0e-7	1.06e-8
0.375	2.4e-6	1.2e-7	3.72e-7
0.500	2.6e-6	1.3e-7	6.21e-7
0.625	2.4e-6	1.2e-7	3.71e-7
0.750	1.9e-6	1.0e-7	1.08e-8
0.875	1.2e-6	5.8e-8	1.25e-7

Örnek 4.3.3: Şimdi,

$$x^2 \frac{\partial^2 u}{\partial x^2} - t^2 \frac{\partial^2 u}{\partial t^2} = 0$$

değişken katsayılı kısmi diferansiyel denklemini (Keşan, 2003)

$$\begin{cases} u(x,0) = 1 \\ u(x,1) = e^t \end{cases}$$

başlangıç koşulları ve

$$\begin{cases} u(0,t) = 1 \\ u(1,t) = e^x \end{cases}$$

sınır koşulları ile $0 \leq x \leq 1, 0 \leq t \leq 1$ bölgesinde ele alalım.

$N = 3,5$ için yöntemi uygulayarak,

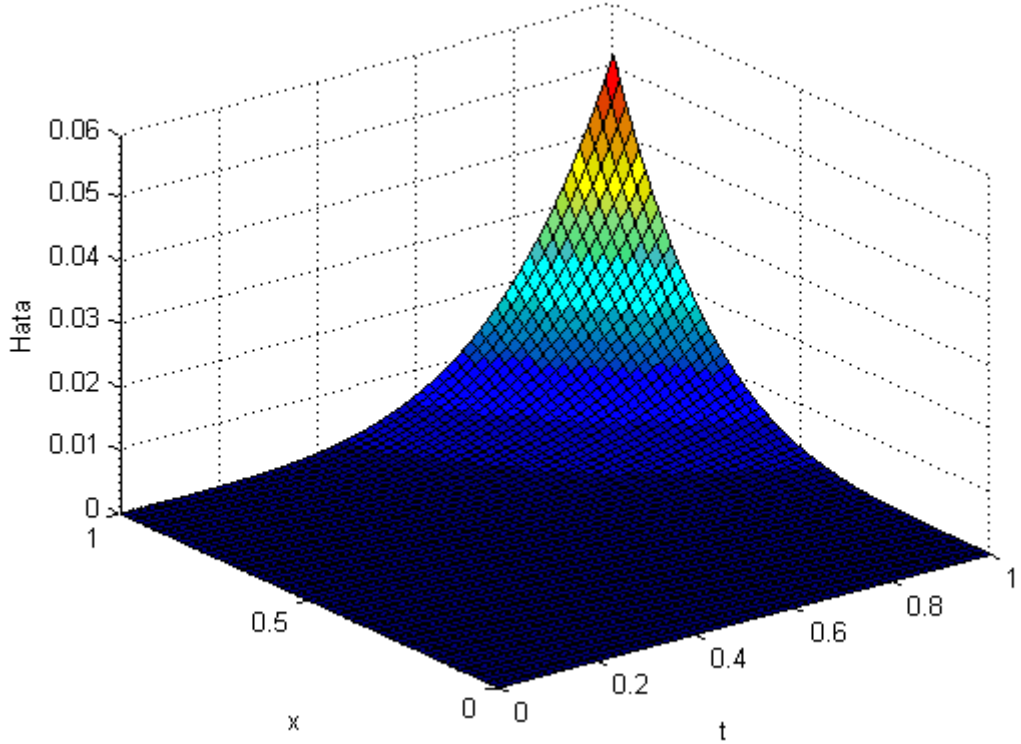
$$\begin{aligned} u_3(x,t) = & 1 + (0.5000000000e-30)x + (0.5000000000e-30)t + 1.013990470xt \\ & - (0.7224855260e-28)x^2t - (0.7257173950e-28)xt^2 \\ & + 0.4256649465x^2t^2 - (0.4409020358e-28)x^2t^3 - (0.6250000000e-31)x^3 \\ & - (0.6250000000e-31)t^3 - (0.4903488722e-28)x^3t^2 + 0.2786264125x^3t^3 \end{aligned}$$

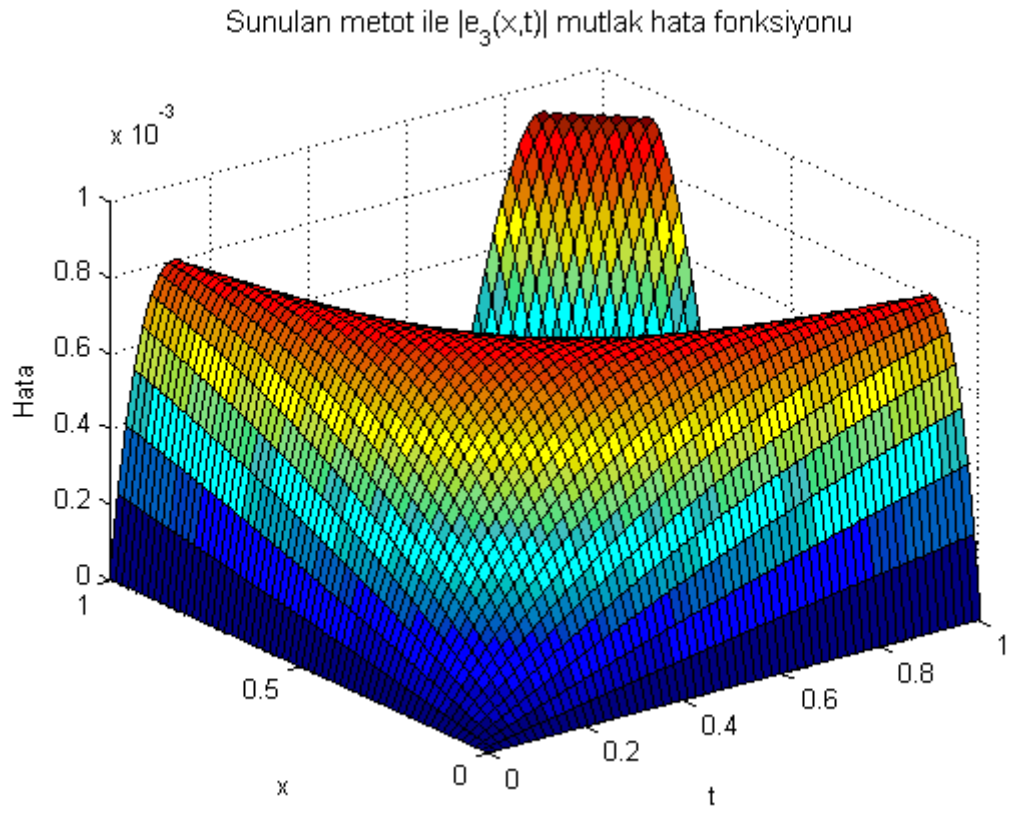
$$\begin{aligned} u_5(x,t) = & 1 - (0.1453848152e-28)x - (0.2129032258e-30)t + 1.000082545xt \\ & - (0.1627403796e-27)xt^2 + (0.5257407258e-27)x^2t + 0.4990687823x^2t^2 \\ & + (0.9953696235e-29)t^3 - (0.5949869445e-28)x^3 - (0.7e-11)x^4 \\ & + (0.9574508969e-26)x^2t^3 + (0.1103306738e-25)x^3t^2 + 0.1704097760x^3t^3 \end{aligned}$$

$$\begin{aligned}
& -(0.1e-9)xt^3 - (0.1e-9)x^3t - (0.7619187549e-29)x^5 - (0.1e-10)t^4 \\
& -(0.4e-11)x^4t^2 + (0.1584835216e-26)x^4t + (0.2528212882e-25)x^4t^3 \\
& +(0.3550191906e-26)x^5t^2 + (0.3e-11)x^5t + (0.1e-10)x^5t^3 - (0.3e-11)x^2t^4 \\
& +(0.3159129825e-26)x^2t^5 + (0.7418926280e-27)xt^4 \\
& +(0.2469515086e-25)x^3t^4 + (0.3e-11)xt^5 + (0.131e-10)x^3t^5 \\
& +(0.3486644681e-1)x^4t^4 + (0.8079960454e-26)x^4t^5 \\
& +(0.7965022741e-26)x^5t^4 + (0.3599193549e-29)t^5 + (0.1385427840e-1)x^5t^5
\end{aligned}$$

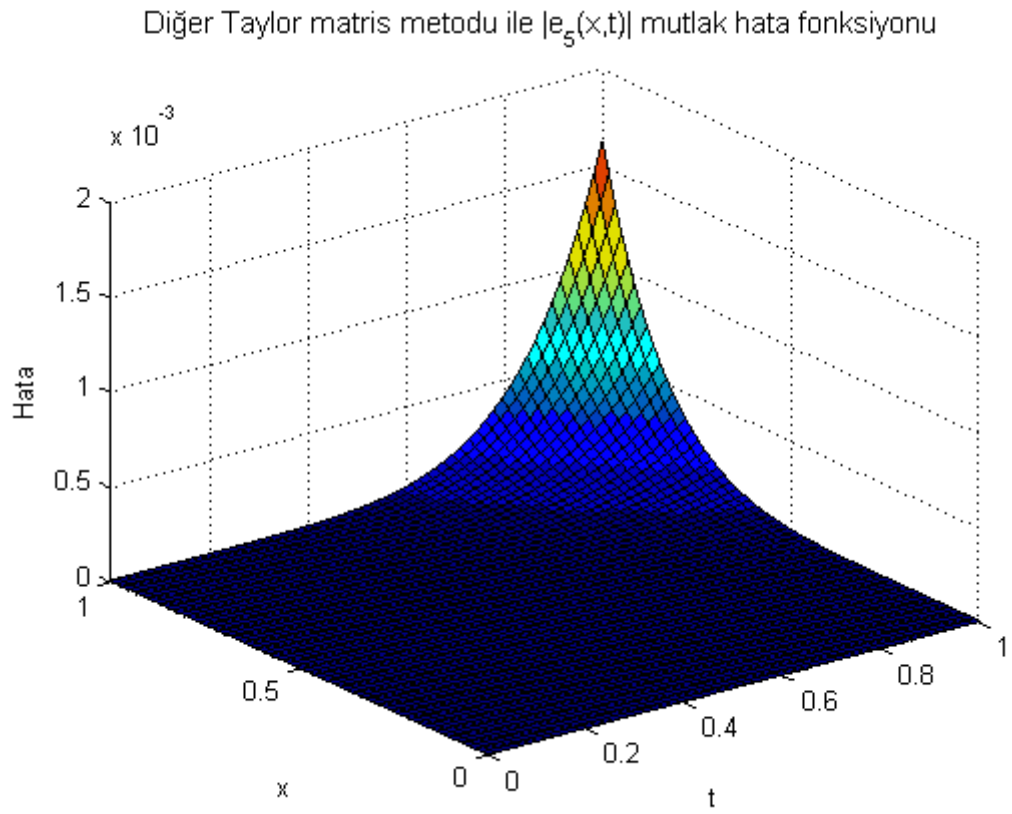
Bessel polinom çözümlerini elde ederiz. Bu problemin tam çözümü $u(x,t) = e^{xt}$ ' dir. Şekil 4.27' de $N=3$ için Taylor matris yöntemi (Keşan, 2003) ve sunulan yöntem ile elde edilen $|e_N(x,t)|$ mutlak hata fonksiyonları karşılaştırılır. Şekil 4.28 $N=5$ için diğer Taylor matris yöntemi (Bülbül, 2011) ve sunulan yöntem ile elde edilen $|e_N(x,t)|$ mutlak hata fonksiyonlarının karşılaştırmasını gösterir. Yaklaşık çözümler ve hata fonksiyonları Maple' da yazılan bilgisayar kodunda virgülden sonra 30 hane kullanılarak elde edildiğini not ediyoruz.

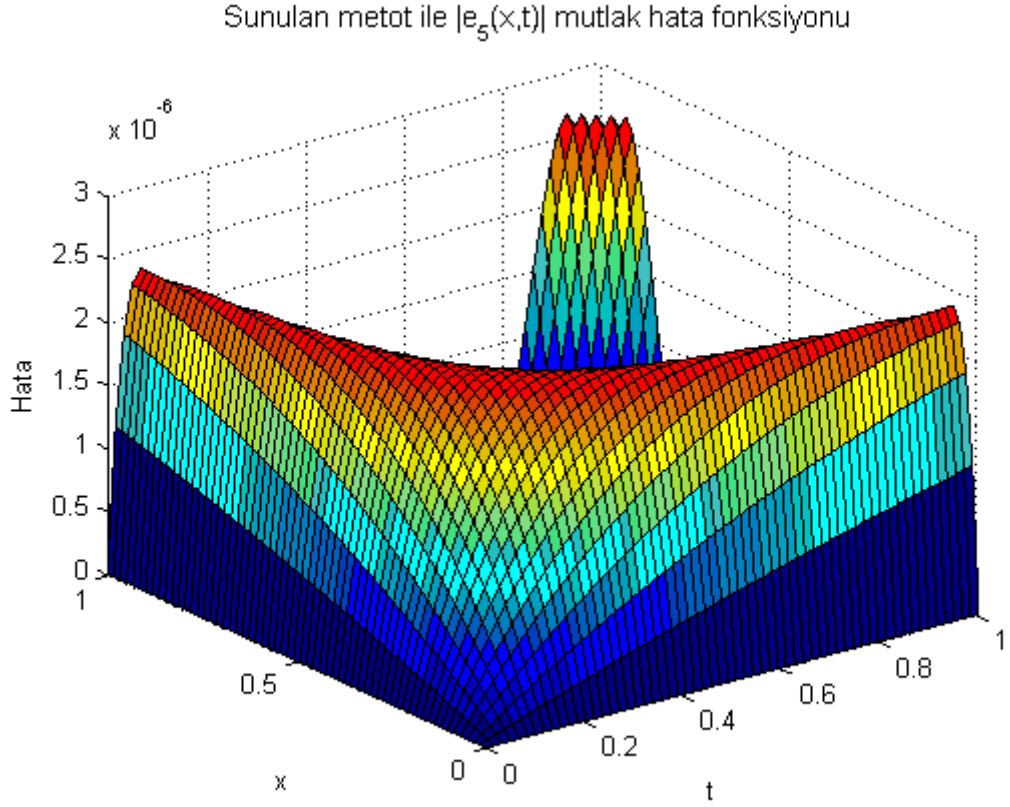
Taylor matris metodu ile $|e_3(x,t)|$ mutlak hata fonksiyonu





Şekil 4.27. Örnek 4.3.3' ün $N = 3$ için Taylor matris yöntemi ve sunulan yöntem ile elde edilen $|e_N(x,t)|$ mutlak hata fonksiyonlarının karşılaştırılması





Şekil 4.28. Örnek 4.3.3' ün $N = 5$ için diğer Taylor matris yöntemi ve sunulan yöntem ile elde edilen $|e_N(x,t)|$ mutlak hata fonksiyonlarının karşılaştırılması

4.4. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon Problemi için Örnekler

Örnek 4.4.1: İlk olarak, (Kadalbajoo ve Gupta, 2009)

$$\frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + (2x+1) \frac{\partial u}{\partial x} + x^2 u = \frac{e^{x+t}}{\varepsilon} (x^2 + 2x + 2 - \varepsilon)$$

singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon denklemini

$$u(x,0) = \frac{e^x}{\varepsilon}, \quad 0 \leq x \leq 1$$

başlangıç koşulu ve

$$u(0,t) = \frac{e^t}{\varepsilon}, \quad 0 \leq t \leq 1,$$

$$u(L,t) = \frac{e^{1+t}}{\varepsilon}, \quad 0 \leq t \leq 1$$

sınır koşulları ile $0 \leq x \leq 1$, $0 \leq t \leq 1$ bölgesinde ele alalım. Burada, ε : $0 < \varepsilon \ll 1$ olacak şekilde bir difüzyon katsayısı ya da singüler pertürbasyon parametresidir.

Problemin tam çözümü $u(x,t) = \frac{e^{x+t}}{\varepsilon}$ ' dir.

Kesim (3.4)' de sunulan prosedürü $N = 5, 6, 7, 8, 9$ için takip ederek Bessel polinom çözümlerini hesapladık. Yaklaşık çözümler ve hata fonksiyonları Matlab'de yazılan bilgisayar kodu ile hesaplandı. Çizelge 4.10 ve Çizelge 4.11' de N ve ε ' nun farklı değerleri için L_2 ve L_∞ hatalarını veriyoruz. Ayrıca (N, M) ve ε ' nun farklı değerleri için $|e_N(x,t)|$ mutlak hata fonksiyonları, $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonları ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonları Çizelge 4.12' de karşılaştırılır. Şekil 4.29-Şekil 4.32' de (N, M) ve ε ' nun çeşitli değerleri için $|e_N(x,t)|$ mutlak hata fonksiyonlarının ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması sunulur. $|e_N(x,t)|$ mutlak hata fonksiyonları ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonları Şekil 4.33-Şekil 4.36' da karşılaştırılır. N değeri artarken hataların azaldığı ve ε değeri küçüldükçe hataların arttığı Çizelge 4.10- Çizelge 4.12 ve Şekil 4.29-Şekil 4.36' dan gözlemlenir.

Çizelge 4.10. Örnek 4.4.1' in L_2 hataları

N	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
5	1.9982×10^{-4}	1.7535×10^{-3}	1.7367×10^{-2}	1.7352×10^{-1}
6	43851×10^{-6}	5.3522×10^{-5}	5.3718×10^{-4}	5.3727×10^{-3}
7	1.9434×10^{-7}	2.3204×10^{-6}	2.3288×10^{-5}	2.3291×10^{-4}
8	7.0223×10^{-9}	5.1130×10^{-8}	5.5541×10^{-7}	5.5985×10^{-6}
9	3.3883×10^{-9}	4.8773×10^{-8}	5.1873×10^{-7}	5.2150×10^{-6}

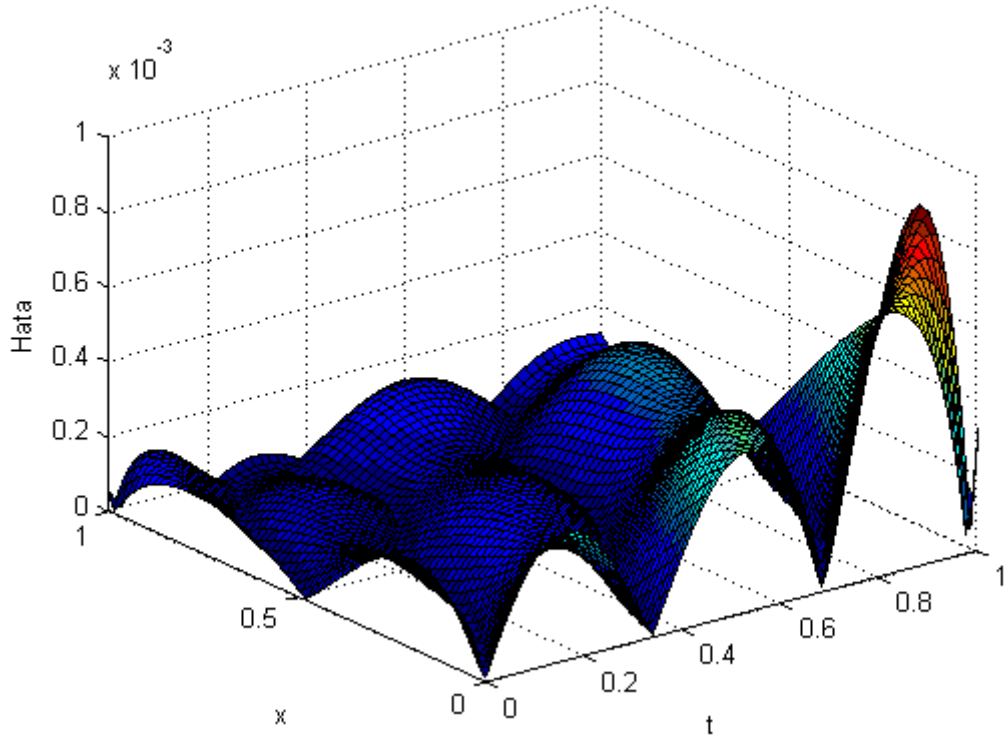
Çizelge 4.11. Örnek 4.4.1' in L_∞ hataları

N	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
5	9.6181×10^{-4}	6.0181×10^{-3}	6.3998×10^{-2}	6.5455×10^{-1}
6	1.80×10^{-5}	2.200×10^{-4}	2.150×10^{-3}	2.150×10^{-2}
7	1.5525×10^{-6}	1.1333×10^{-5}	1.1365×10^{-4}	1.150×10^{-3}
8	1.2692×10^{-8}	1.1429×10^{-7}	1.3333×10^{-6}	1.3429×10^{-5}
9	6.8182×10^{-9}	8.500×10^{-8}	9.250×10^{-7}	9.00×10^{-6}

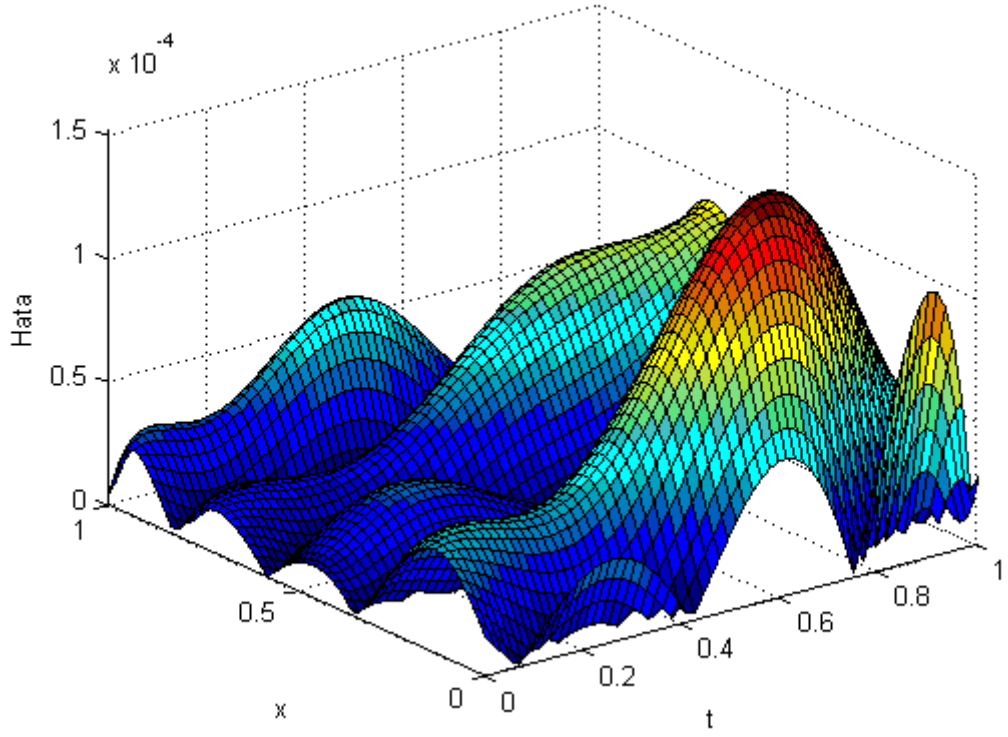
Çizelge 4.12. Örnek 4.4.1' in Bessel polinom çözümlerinin ve iyileştirilmiş Bessel polinom çözümlerinin maksimum hataları ve tahmin edilen maksimum hataların karşılaştırılması

(N, M)	$\varepsilon = 10^{-1}$		$\varepsilon = 10^{-3}$	
	(5,6)	(7,8)	(5,6)	(7,8)
$ e_N(x, t) $ için maksimum hata	9.6181×10^{-4}	1.5525×10^{-6}	6.3998×10^{-2}	1.1365×10^{-4}
$ e_{N,M}(x, t) $ için maksimum hata	1.4567×10^{-4}	1.9348×10^{-7}	1.7962×10^{-2}	3.2566×10^{-5}
$ E_{N,M}(x, t) $ için maksimum hata	8.5885×10^{-4}	1.2882×10^{-6}	4.9361×10^{-2}	8.4277×10^{-5}

$\varepsilon=10^{-1}$ için $|e_5(x, t)|$ mutlak hata fonksiyonu

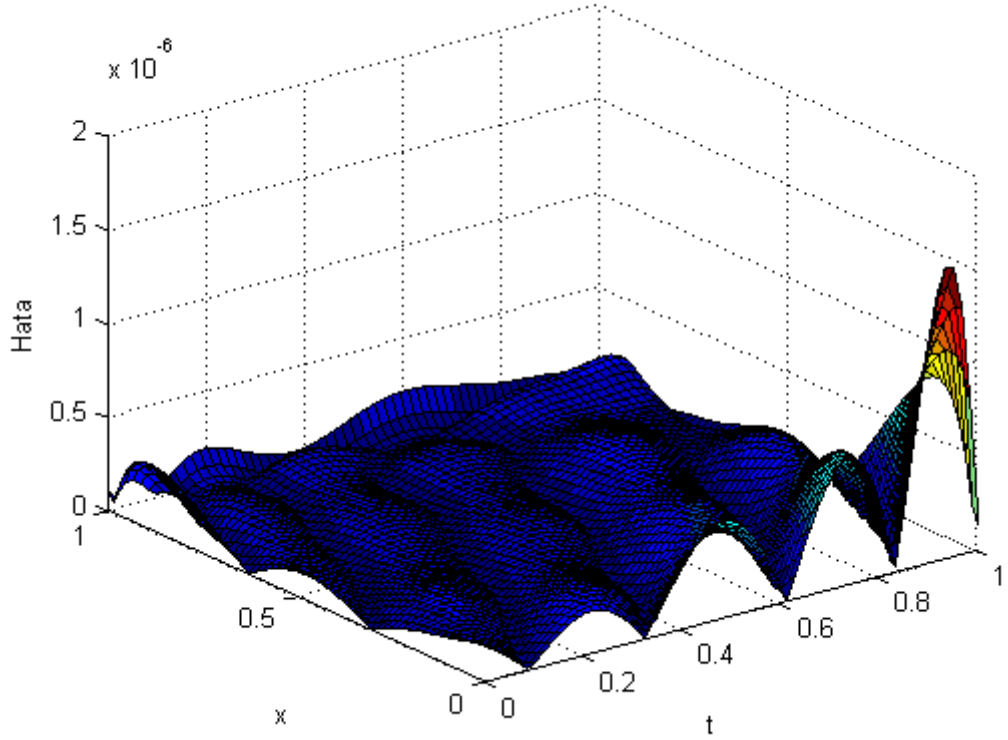


$\varepsilon=10^{-1}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

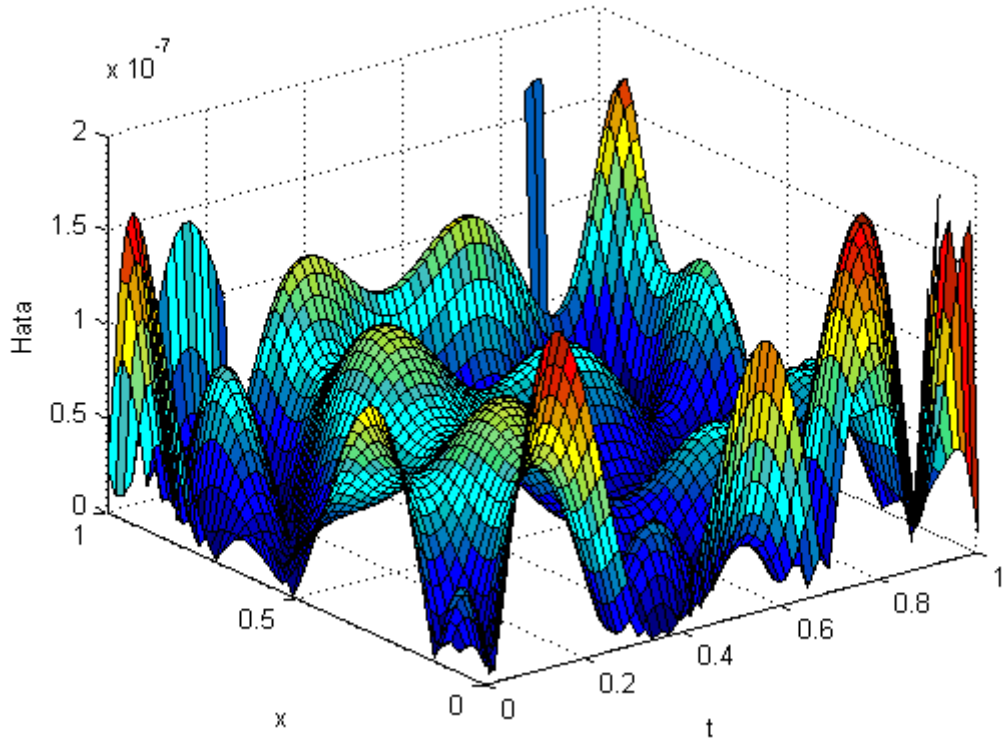


Şekil 4.29. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-1}$ için $|e_7(x,t)|$ mutlak hata fonksiyonu

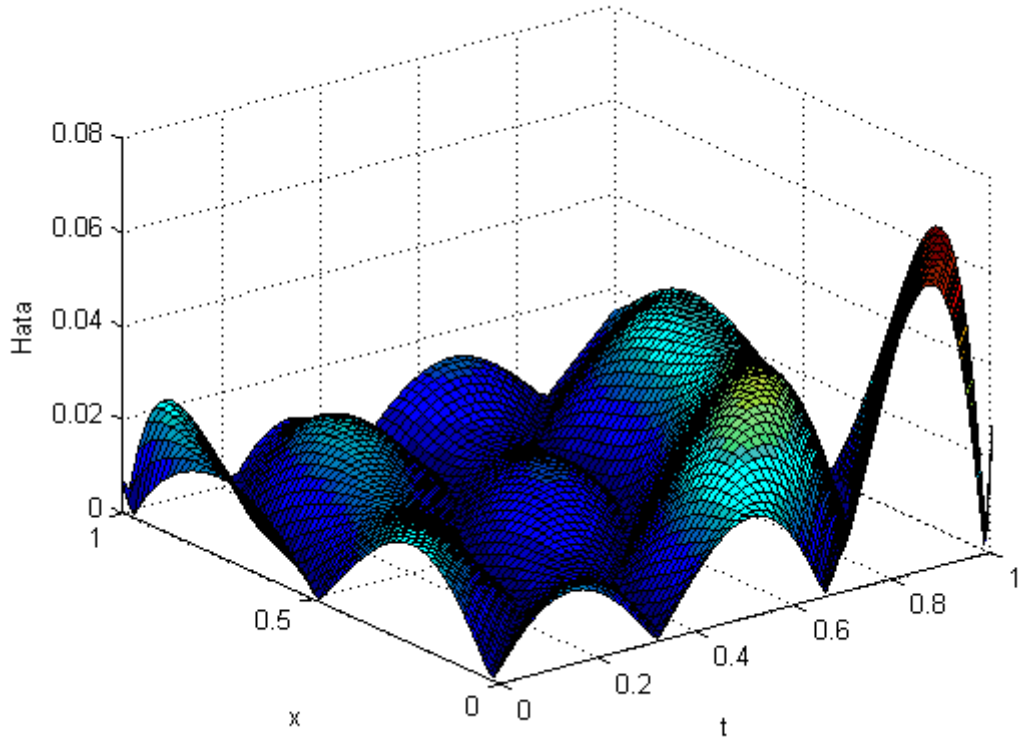


$\varepsilon=10^{-1}$ için $|e_{7,8}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

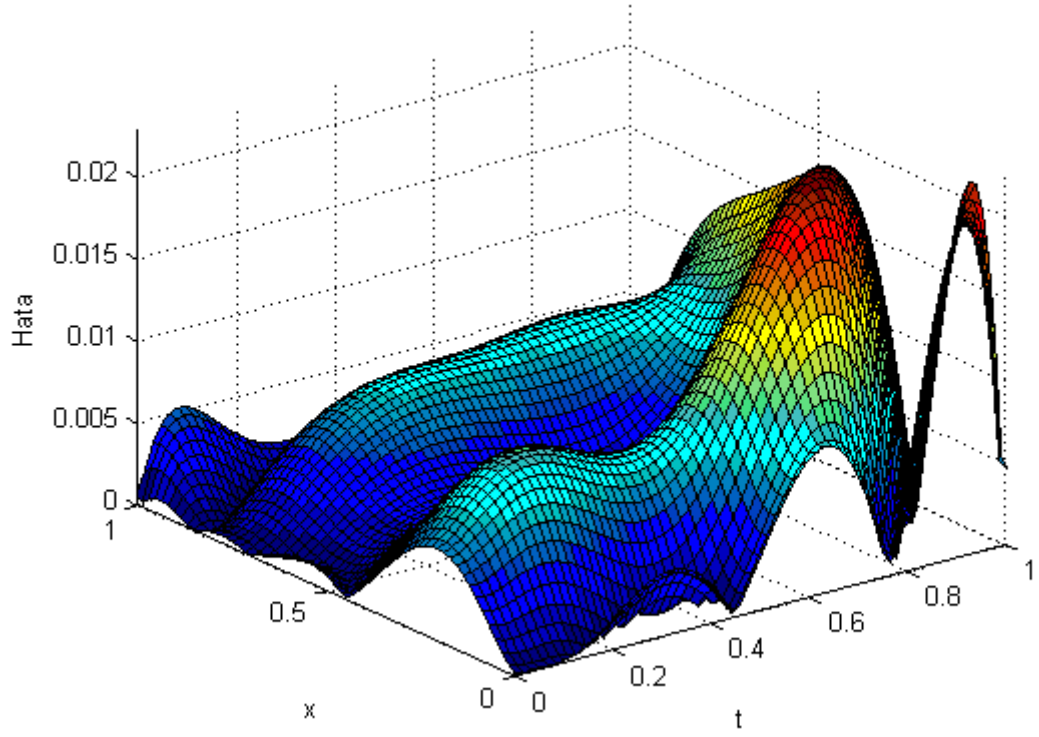


Şekil 4.30. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-3}$ için $|e_5(x,t)|$ mutlak hata fonksiyonu

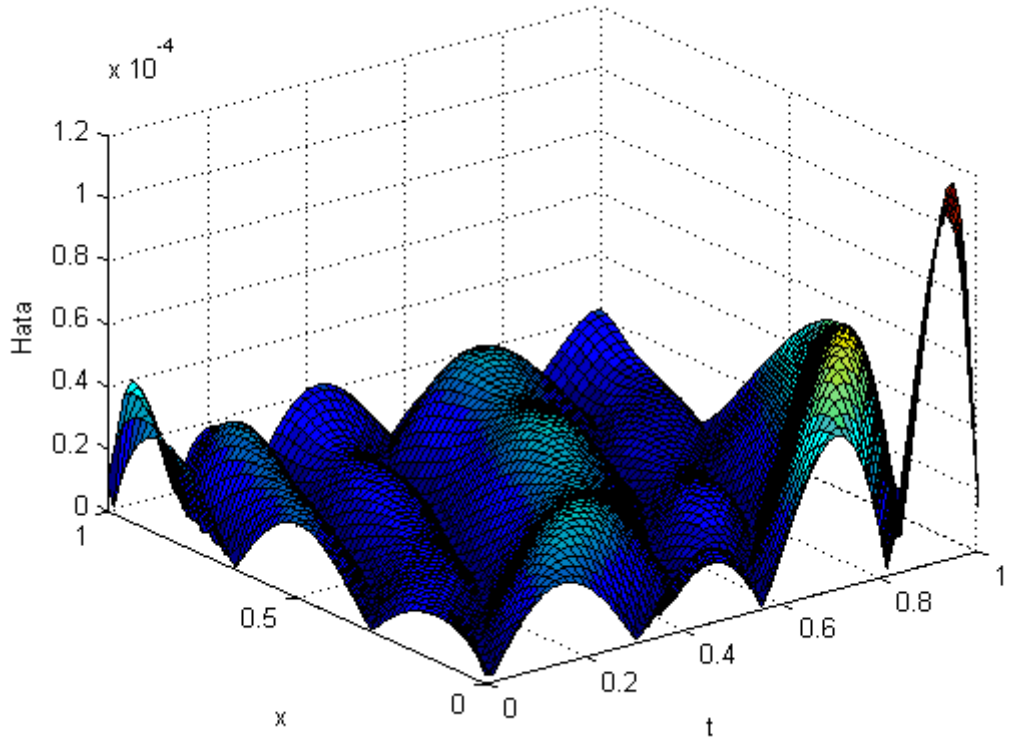


$\varepsilon=10^{-3}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

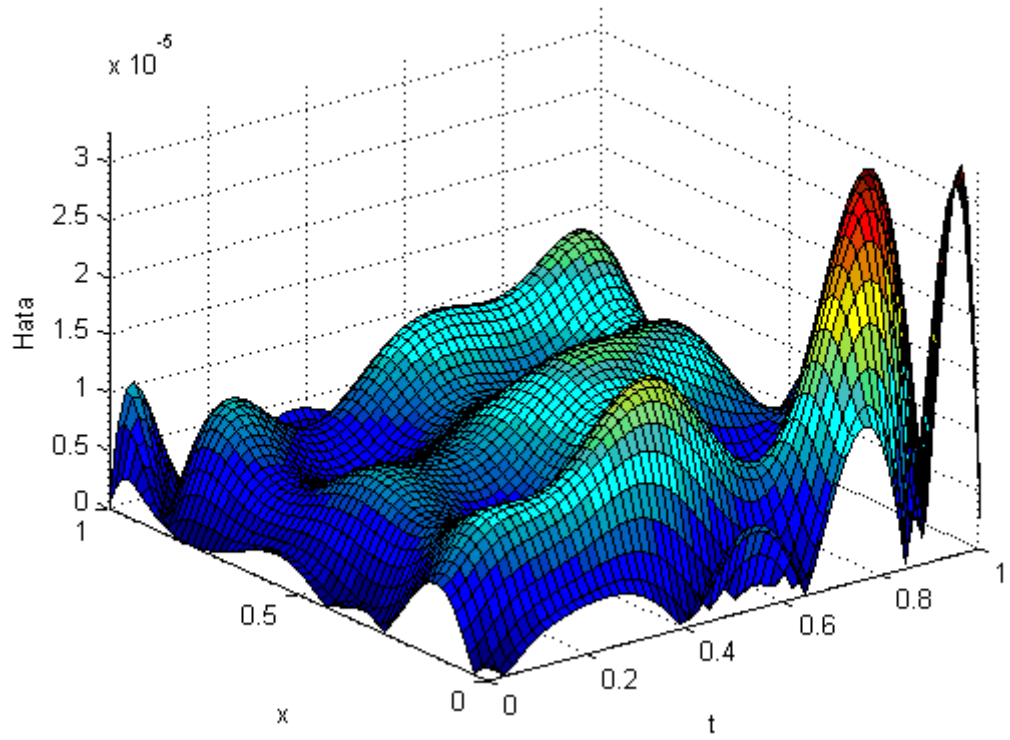


Şekil 4.31. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{NM}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-3}$ için $|e_7(x,t)|$ mutlak hata fonksiyonu

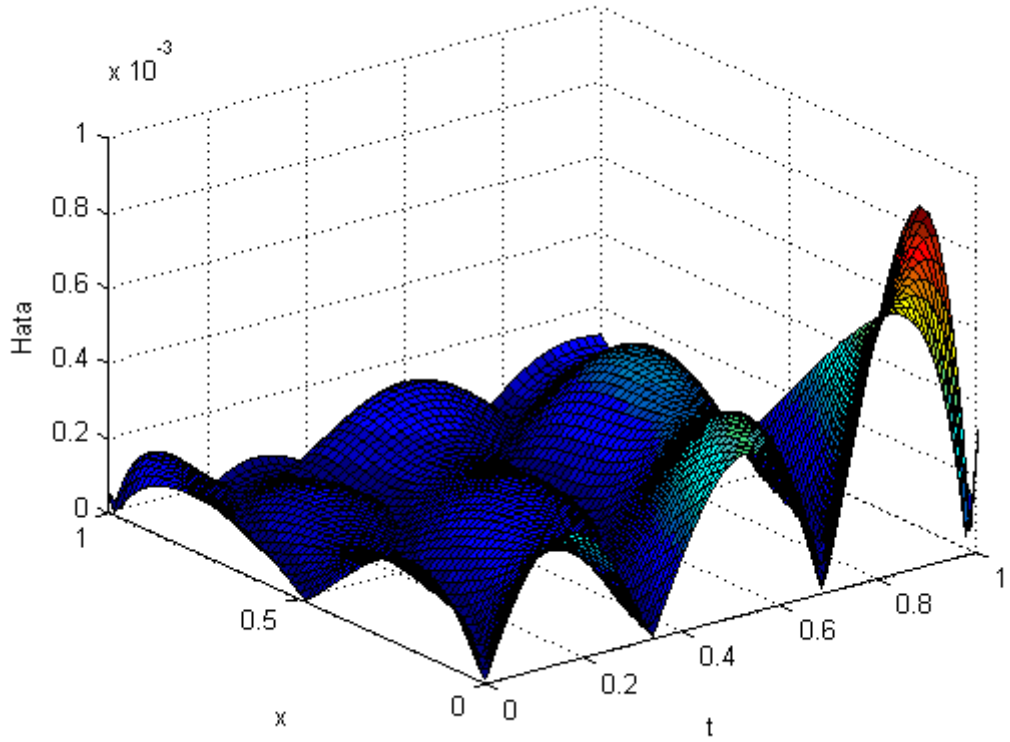


$\varepsilon=10^{-3}$ için $|e_{7,8}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

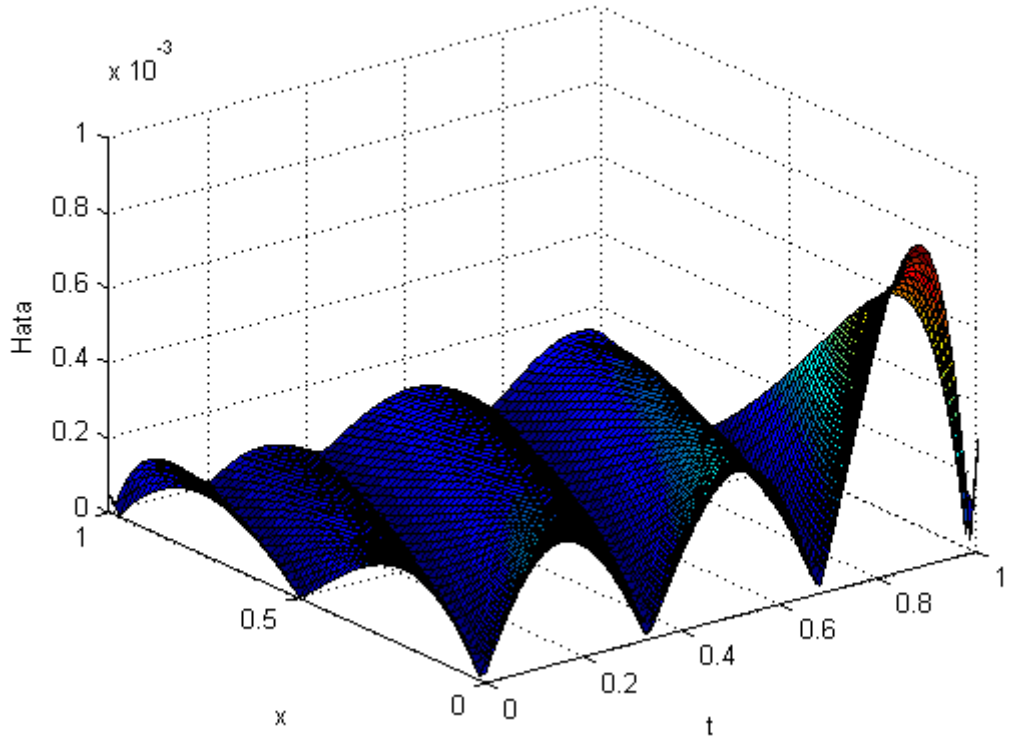


Şekil 4.32. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-1}$ için $|e_5(x,t)|$ mutlak hata fonksiyonu

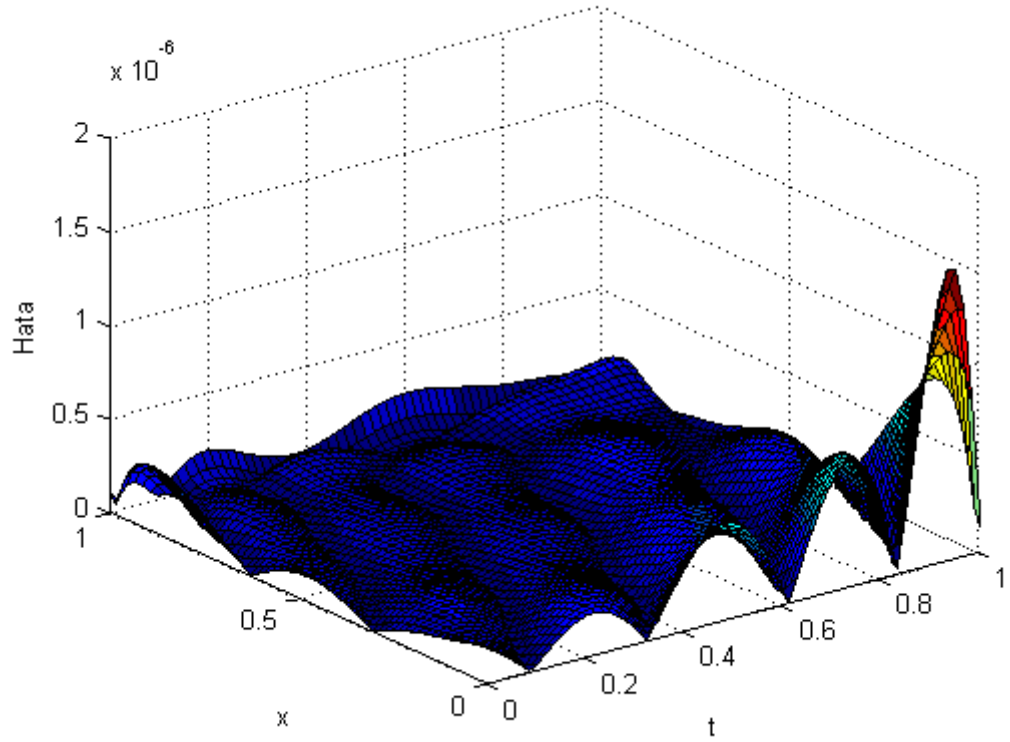


$\varepsilon=10^{-1}$ için $|E_{5,6}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonu

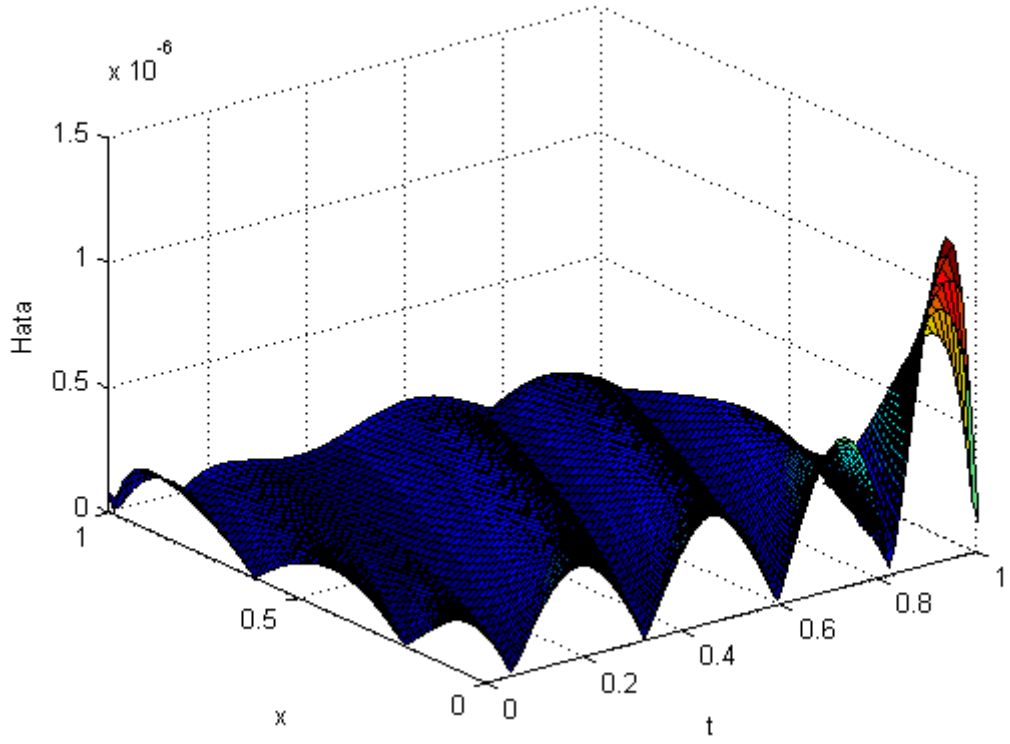


Şekil 4.33. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-1}$ için $|e_7(x,t)|$ mutlak hata fonksiyonu

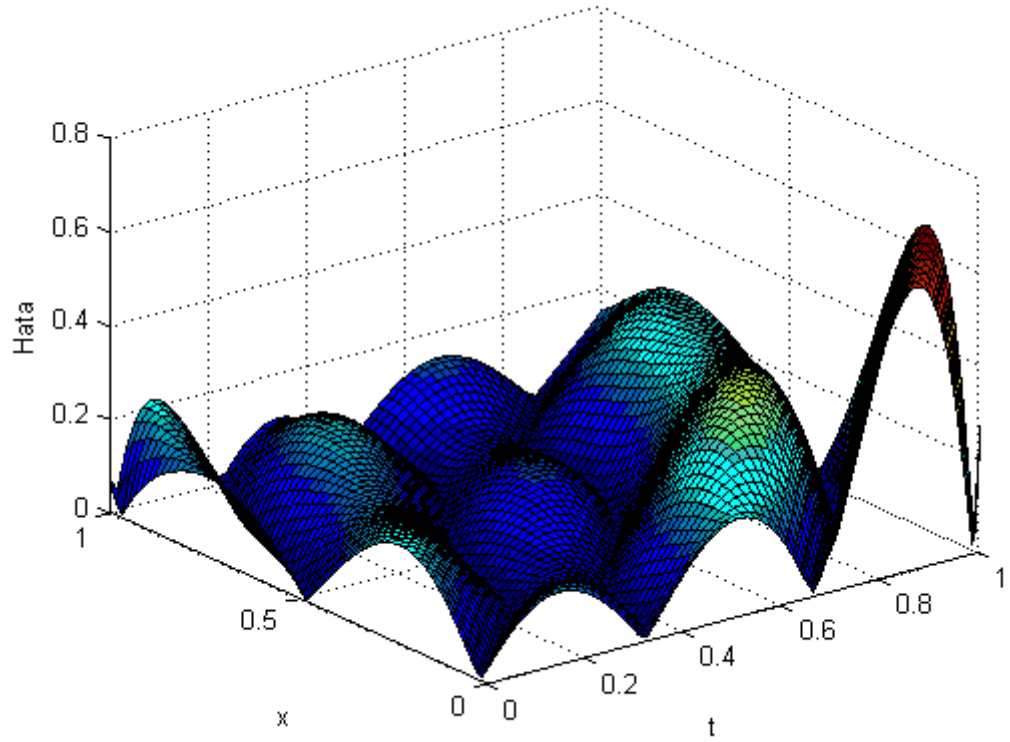


$\varepsilon=10^{-1}$ için $|E_{7,8}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonu

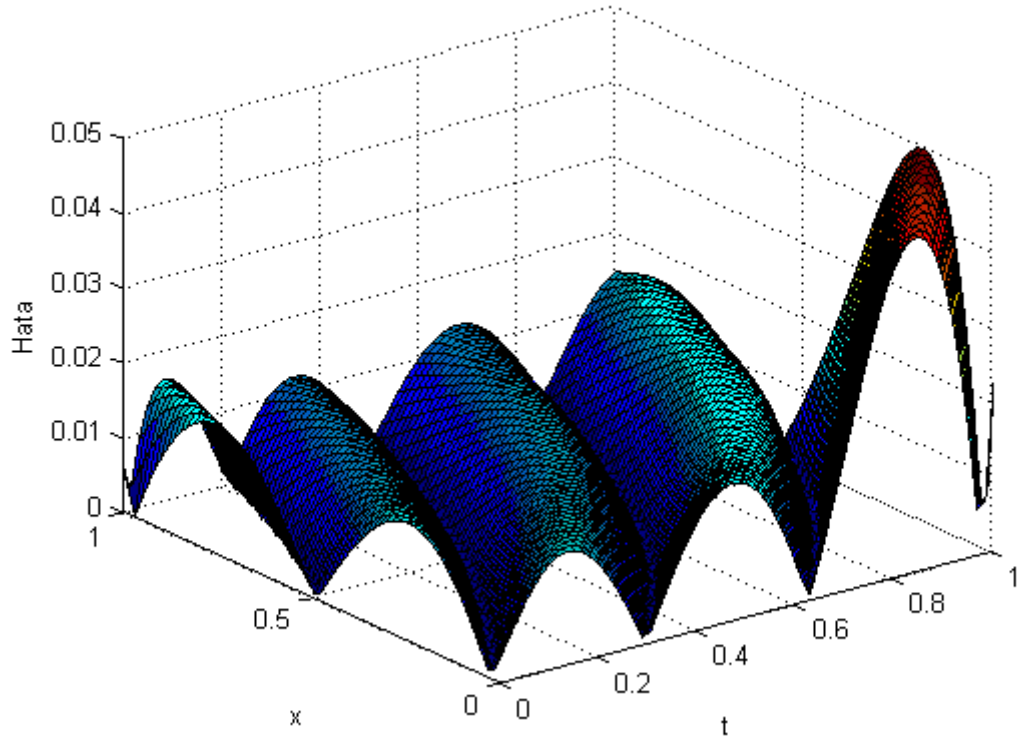


Şekil 4.34. Örnek 4.4.1' in $\varepsilon = 10^{-1}$, $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-3}$ için $|e_5(x,t)|$ mutlak hata fonksiyonu

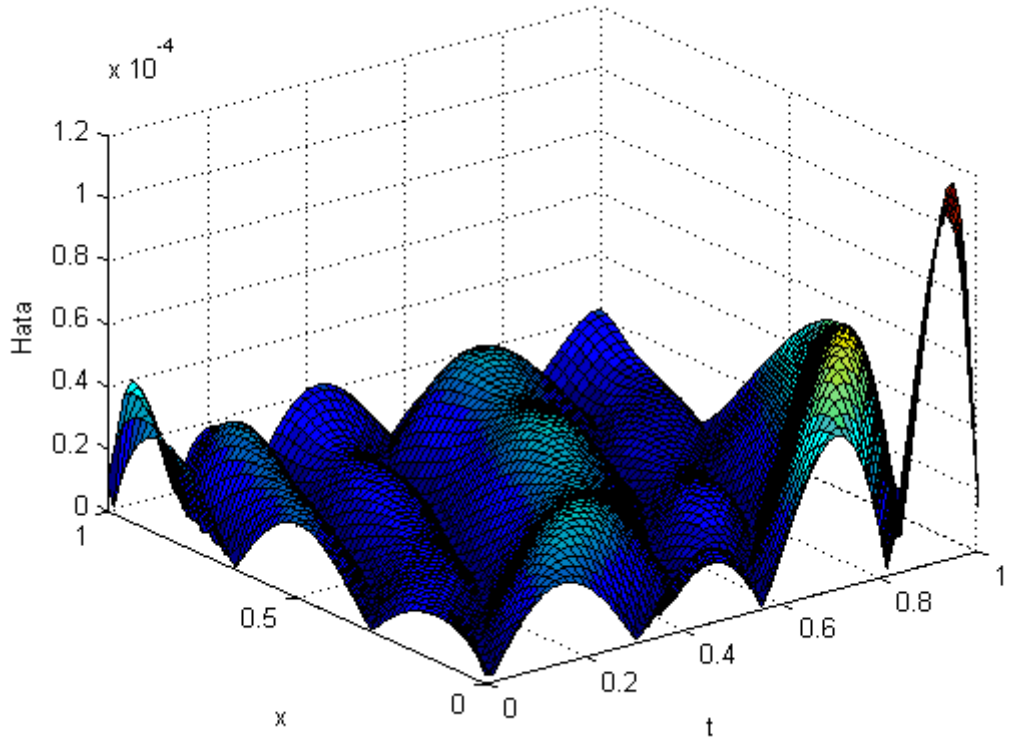


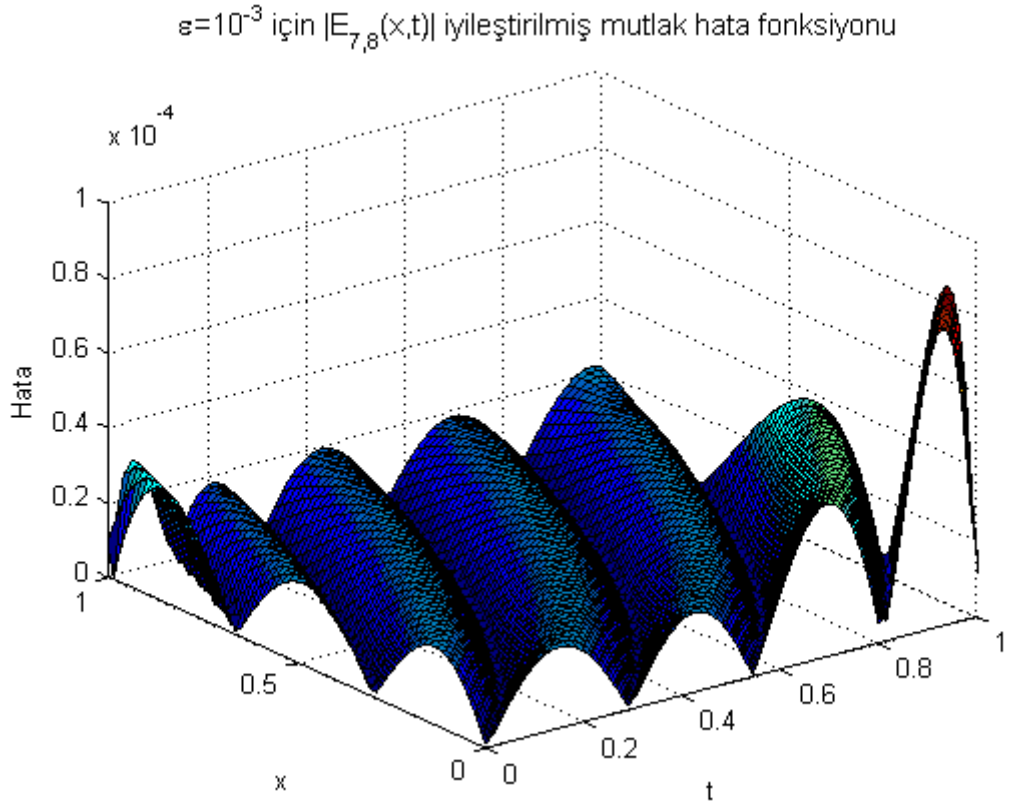
$\varepsilon=10^{-3}$ için $|E_{5,6}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonu



Şekil 4.35. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 5$ ve $M = 6$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

$\varepsilon=10^{-3}$ için $|e_7(x,t)|$ mutlak hata fonksiyonu





Şekil 4.36. Örnek 4.4.1' in $\varepsilon = 10^{-3}$, $N = 7$ ve $M = 8$ için $|e_N(x,t)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,t)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

Örnek 4.4.2: (Clavero, 2003) Bu örnekte,

$$\frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + (2 - x^2) \frac{\partial u}{\partial x} + xu = 10t^2 e^{-t} x(1 - x)$$

singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon denklemini

$$u(x, 0) = 0, \quad 0 \leq x \leq 1$$

başlangıç koşulu ve

$$u(0, t) = 0, \quad 0 \leq t \leq 1,$$

$$u(1, t) = 0, \quad 0 \leq t \leq 1$$

sınır koşulları ile $0 \leq x \leq 1$, $0 \leq t \leq 1$ bölgesinde ele alalım. Burada, $\varepsilon: 0 < \varepsilon \ll 1$ olacak şekilde bir difüzyon katsayısı ya da singüler pertürbasyon parametresidir.

Kesim (4.4)' de sunulan prosedür kullanılarak, N ve M ' nin çeşitli değerleri için yaklaşık çözümleri elde ettik. Yaklaşık çözümleri ve hata fonksiyonlarını Matlab' de yazılan bilgisayar kodu ile hesapladık. $\varepsilon = 2^{-1}$, $N = 5$ ve $M = 6$ için $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümünün grafikleri Şekil 4.37' de ve $\varepsilon = 2^{-14}$, $N = 5$ ve $M = 6$ için $u_N(x, t)$ Bessel polinom çözümü ve $u_{N,M}(x, t)$ iyileştirilmiş Bessel polinom çözümünün grafikleri Şekil 4.38' de verilir. Bu problemin tam çözümü mevcut değildir. Bu yüzden L_2 ve L_∞ hataları hesaplanamaz. Fakat önceki örneklerden hata fonksiyonu için yapılan tahminin oldukça etkili olduğunu biliyoruz. Böylece, yöntemin güvenilirliğini hata tahmini ile test edebiliriz. Şekil 4.39-Şekil 4.42' de $\varepsilon = 2^{-1}$ ve (N, M) ' nin çeşitli değerleri için $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırılması verilir. Şekil 4.43' de $\varepsilon = 2^{-14}$ ve (N, M) ' nin çeşitli değerleri için $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonları karşılaştırılır. Şekil 4.44-Şekil 4.47' de aynı (N, M) değeri için $\varepsilon = 2^{-1}$ ve $\varepsilon = 2^{-14}$ alınarak elde edilen $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonları karşılaştırılır.

(3.109)' da verilen

$$R_N(x, t) = L[u_N(x, t)] - G(x, t)$$

rezidüel fonksiyonu bu problem için

$$R_N(x, t) = \frac{\partial u_N}{\partial t} - \varepsilon \frac{\partial^2 u_N}{\partial x^2} + (2 - x^2) \frac{\partial u_N}{\partial x} + xu_N - 10t^2 e^{-t} x(1 - x)$$

olur. Ekstra bir ölçüm için, çözümlerin doğruluğu $|R_N(x, t)|$ hata fonksiyonu ile kontrol edilebilir. Çizelge 4.13' de $\varepsilon = 2^{-1}$ ve $N = 2, 4, 7, 9$ için $t = 1$ zamanındaki bazı $|R_N(x_i, 1)|$ mutlak rezidüel hataları veriyoruz.

Çözümlerin güvenilirliğinin ekstra bir ölçümü için N ve ε ' nun farklı değerleri ile sunulan yöntem, Euler implicit metodu (Clavero, 2003), B-spline sıralama metodu (BSM) (Kadalbajoo vd., 2008) ve parçalı analitik metodu (Ramos, 2005) için

$$e_{\varepsilon}^N = \max \{ |u_N(x, t) - u_{2N}(x, t)|, 0 \leq x \leq 1, 0 \leq t \leq 1 \}$$

ile elde edilen maksimum noktasal hatalar Çizelge 4.14’ de karşılaştırılır.

Sonuç olarak N ve M değerleri arttırıldığında hataların azaldığı ve ε değeri küçüldükçe hataların arttığı Çizelge 4.13-Çizelge 4.14 ve Şekil 4.39-Şekil 4.47’ den görülür ve ayrıca yapılan karşılaştırmalardan bizim yöntem ile diğer metotlara göre daha küçük N değerleri için hemen hemen aynı sonuçların elde edildiği görülür.

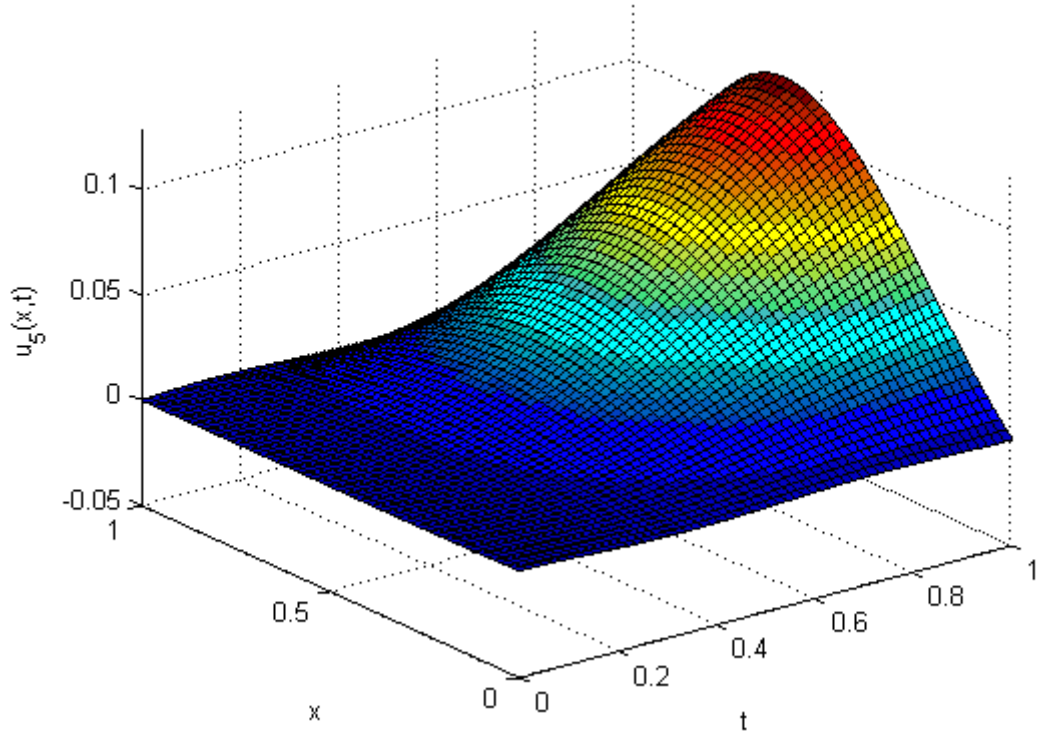
Çizelge 4.13. Örnek 4.4.2’nin $\varepsilon = 10^{-1}$ için $t = 1$ zamanındaki $|R_N(x_i, 1)|$ hataları

x_i	Sunulan yöntem			
	$N = 2$	$N = 4$	$N = 7$	$N = 9$
0	3.4414e-001	2.7339e-003	4.3217e-005	2.8937e-006
0.125	2.2897e-002	1.9274e-003	3.3953e-004	1.3234e-005
0.250	2.8882e-001	4.2649e-003	3.5866e-004	1.1146e-005
0.375	4.5398e-001	3.7479e-003	6.3568e-005	6.1311e-006
0.500	5.1872e-001	8.3327e-004	3.7500e-004	1.8466e-005
0.625	4.8337e-001	3.0808e-003	6.3611e-006	5.1207e-006
0.750	3.4827e-001	5.7004e-003	4.9111e-004	1.2397e-005
0.875	1.1379e-001	3.8822e-003	4.8803e-004	1.0829e-005
1	2.1976e-001	6.3210e-003	6.0717e-005	3.8149e-006

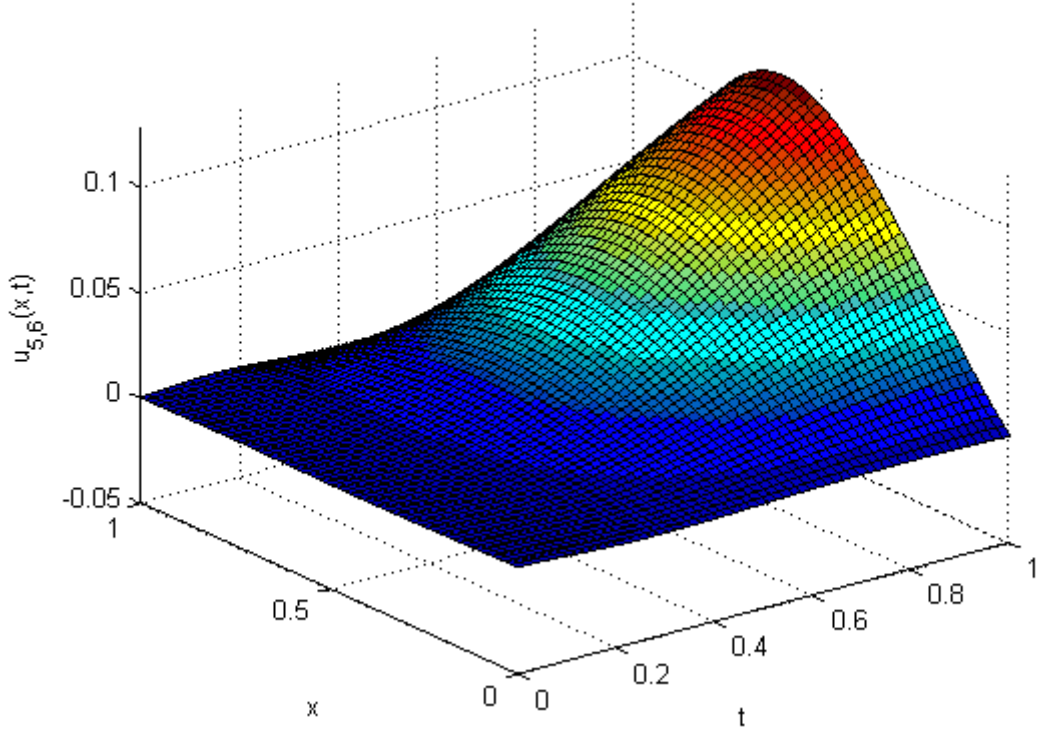
Çizelge 4.14. Bizim metodun ve başka metotların N ve ε ’nun çeşitli değerleri için e_{ε}^N maksimum noktasal hatalarının karşılaştırılması

$N \rightarrow$	3	4	16	32	16	32
ε	Sunulan yöntem		Düzgün sıralama noktaları ile BSM (Kadalbajoo vd., 2008)		Parçalı analitik metot (Ramos, 2005)	
$\downarrow$						
2^{-2}	1.7917e-02	1.0909e-03	2.0301e-2	1.1131e-2	2.6e-3	9.9218e-4
2^{-4}	2.4545e-02	1.1414e-02	2.8100e-2	1.8578e-2	1.15e-2	5.1e-3
2^{-6}	4.2727e-02	1.1875e-02	3.0483e-1	1.2750e-1	2.25e-2	1.67e-2
2^{-8}	5.1250e-02	2.9091e-02	8.3956e-1	4.6488e-1	1.52e-2	1.44e-2
2^{-10}	5.3750e-02	3.2500e-02	1.2697	9.594e-1	1.33e-2	7.9e-3
2^{-12}	5.5000e-02	3.3333e-02	1.4406	1.3296	1.40e-2	6.7e-3
2^{-14}	5.6250e-02	3.4545e-02	1.4904	1.4702	1.41e-2	6.9e-3
	Euler implicit metodu (Clavero vd., 2003)		Shishkin sıralama noktaları ile BSM (Kadalbajoo vd., 2008)			
2^{-2}	1.1249e-2	6.3202e-3	5.8459e-3	1.4220e-3		
2^{-4}	1.6783e-2	8.1043e-3	4.6929e-3	2.8234e-3		
2^{-6}	3.0900e-2	1.5221e-2	3.7566e-3	1.2640e-3		
2^{-8}	3.5742e-2	1.9347e-2	1.2536e-2	2.6440e-3		
2^{-10}	3.6717e-2	2.0475e-2	1.6606e-2	5.3053e-3		
2^{-12}	3.6931e-2	2.0732e-2	1.7491e-2	6.3301e-3		
2^{-14}	3.6982e-2	2.0794e-2	1.7860e-2	6.4345e-3		

$\varepsilon=2^{-1}$ için $|u_5(x,t)|$ Bessel polinom çözümü

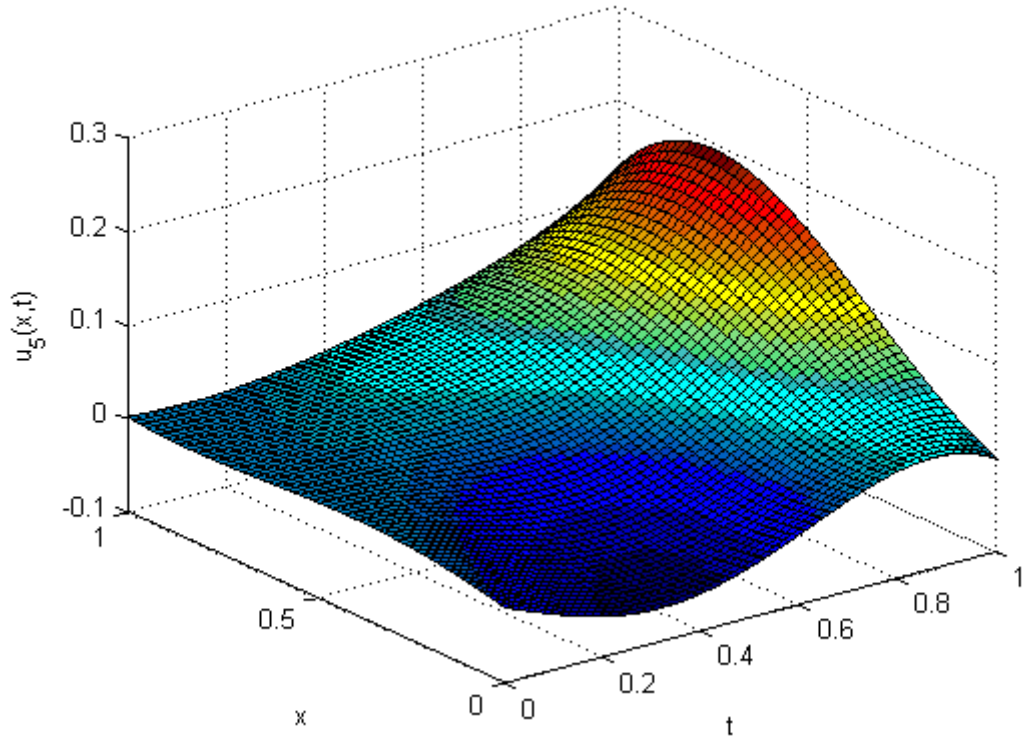


$\varepsilon=2^{-1}$ için $|u_{5,6}(x,t)|$ iyileştirilmiş Bessel polinom çözümü

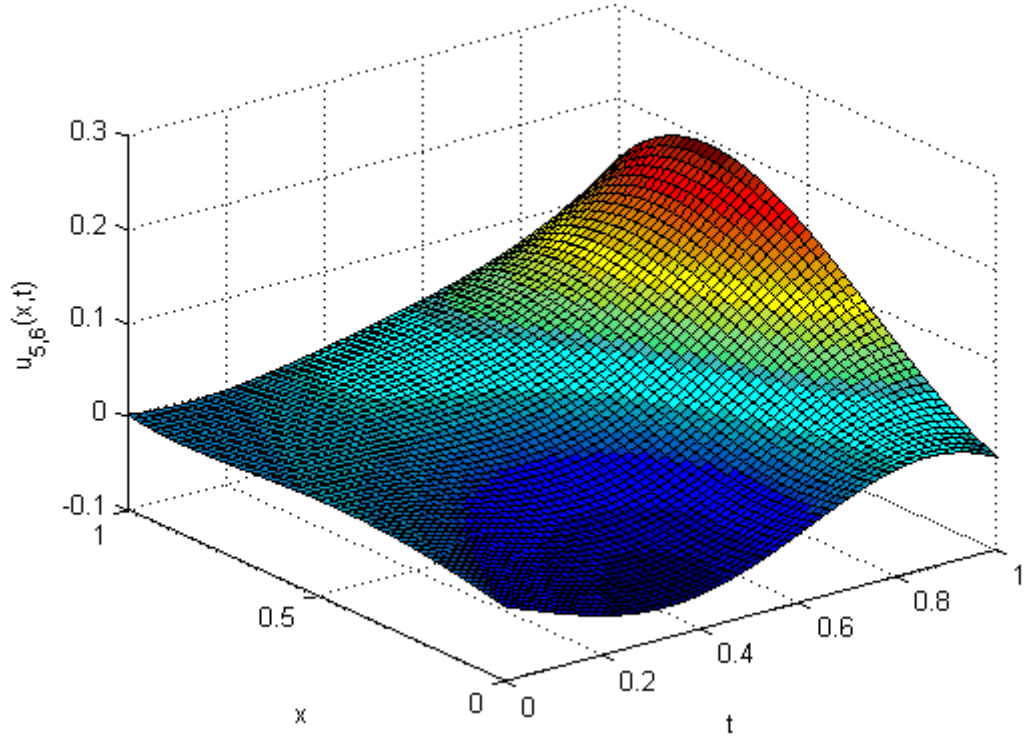


Şekil 4.37. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $N = 5$ ve $M = 6$ için $u_N(x,t)$ Bessel polinom çözümü ve $u_{N,M}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması

$\varepsilon=2^{-14}$ için $|u_5(x,t)|$ Bessel polinom çözümü

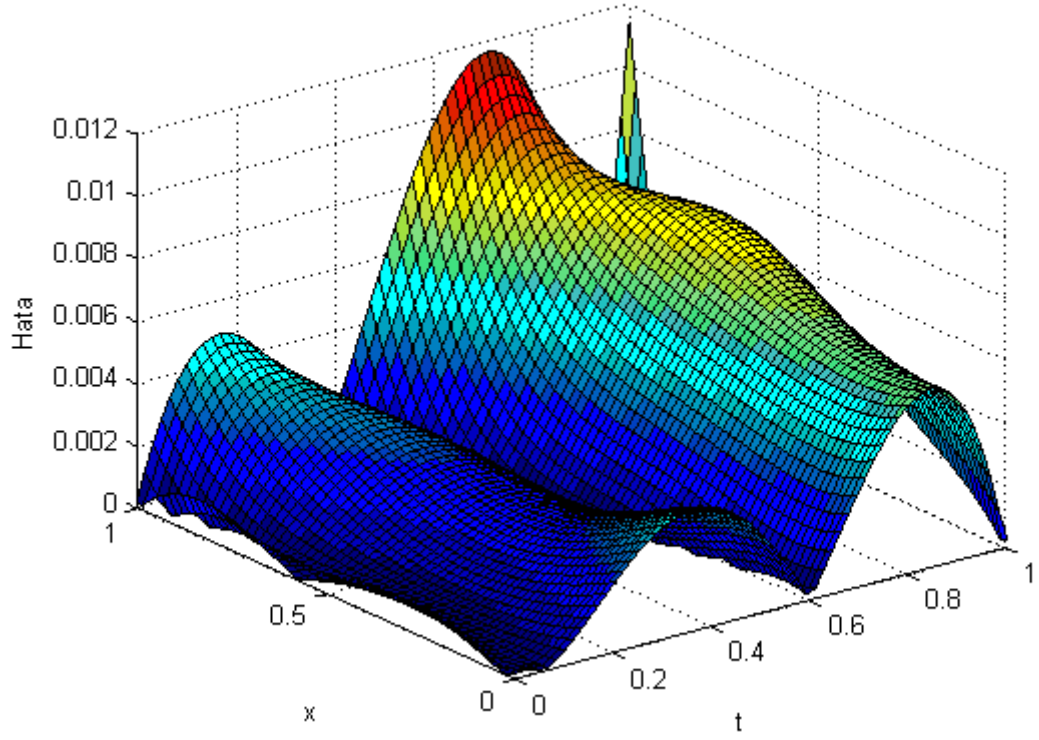


$\varepsilon=2^{-14}$ için $|u_{5,6}(x,t)|$ iyileştirilmiş Bessel polinom çözümü

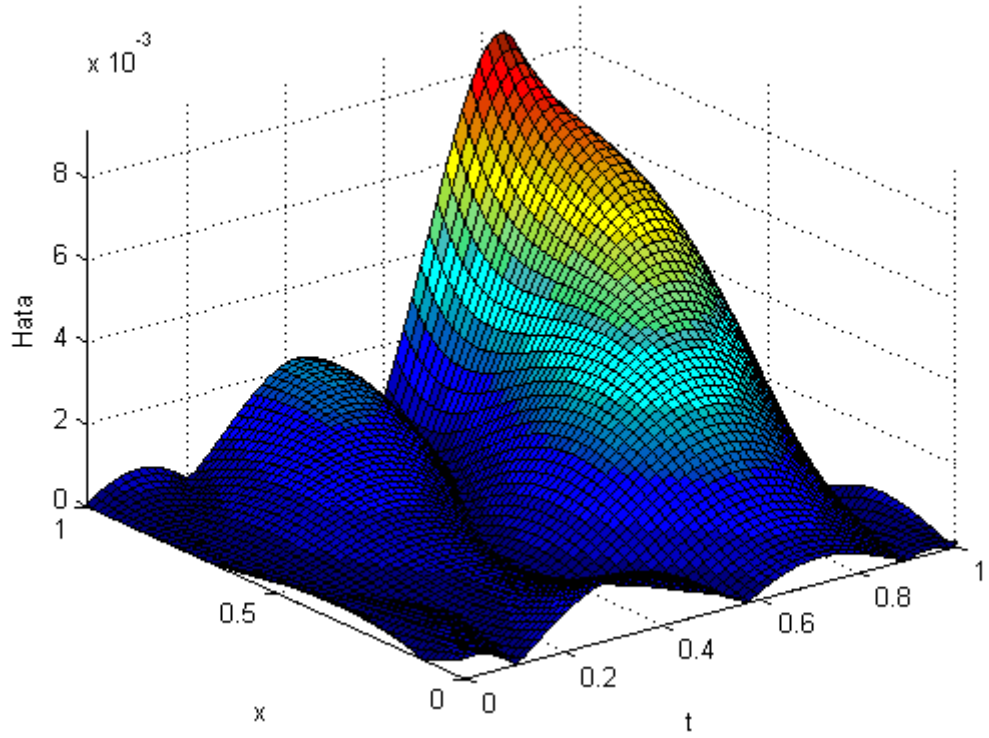


Şekil 4.38. Örnek 4.4.2' nin $\varepsilon = 2^{-14}$, $N = 5$ ve $M = 6$ için $u_N(x,t)$ Bessel polinom çözümü ve $u_{N,M}(x,t)$ iyileştirilmiş Bessel polinom çözümünün karşılaştırılması

$\varepsilon=2^{-1}$ için $|e_{3,4}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

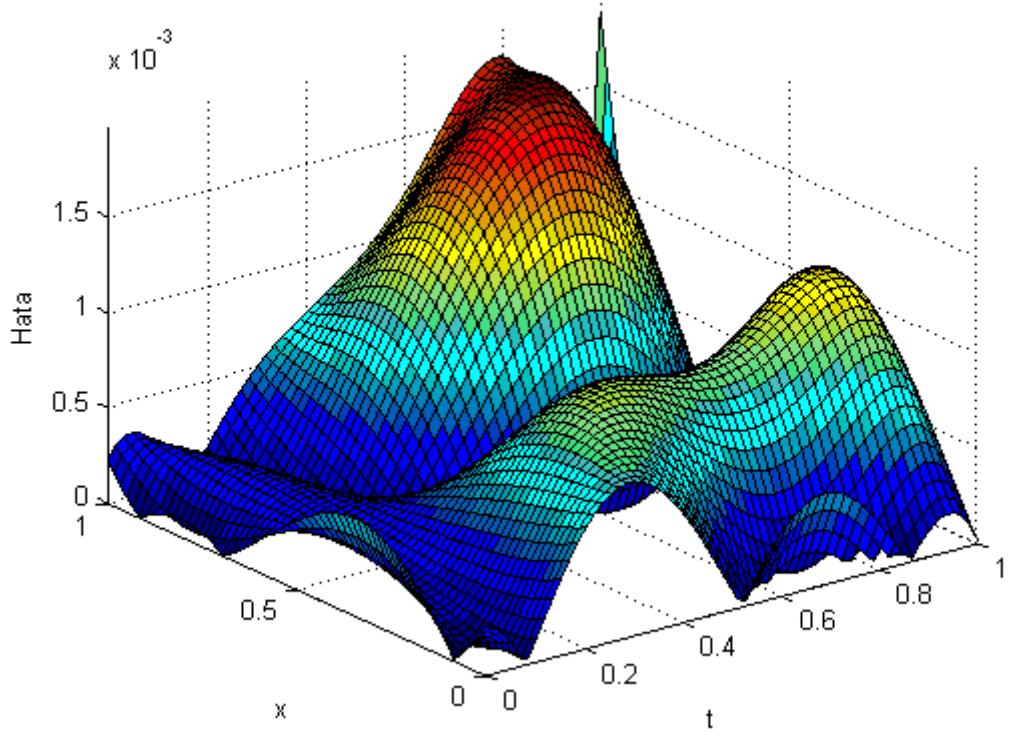


$\varepsilon=2^{-1}$ için $|e_{3,5}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

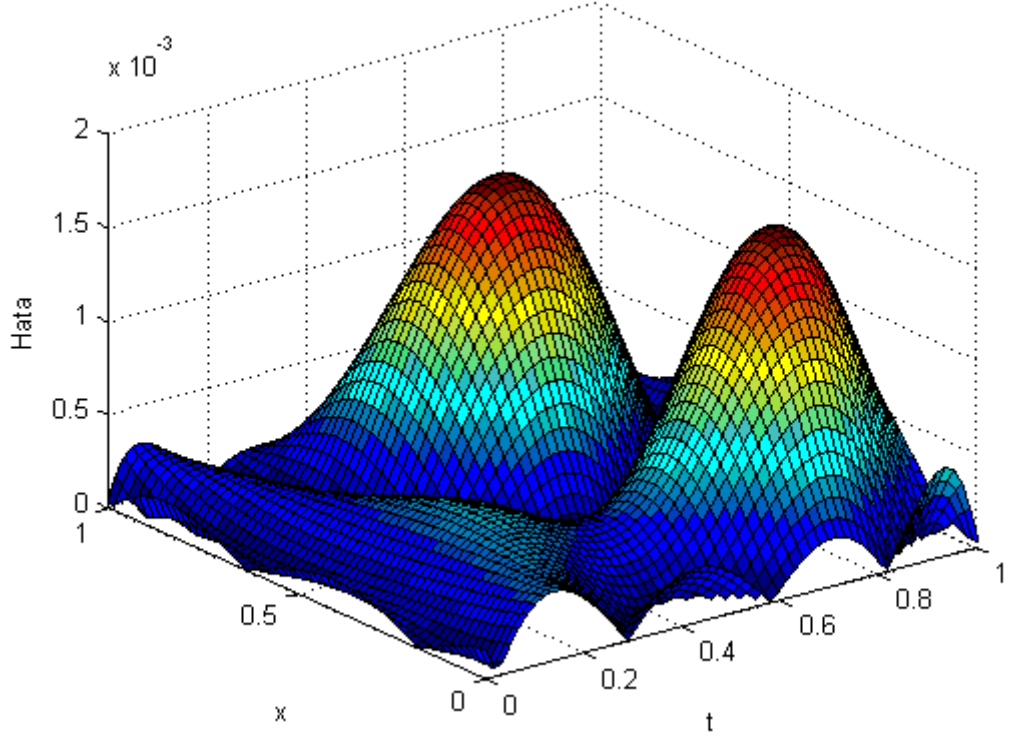


Şekil 4.39. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N,M) = (3,4)$ ve $(N,M) = (3,5)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{4,5}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

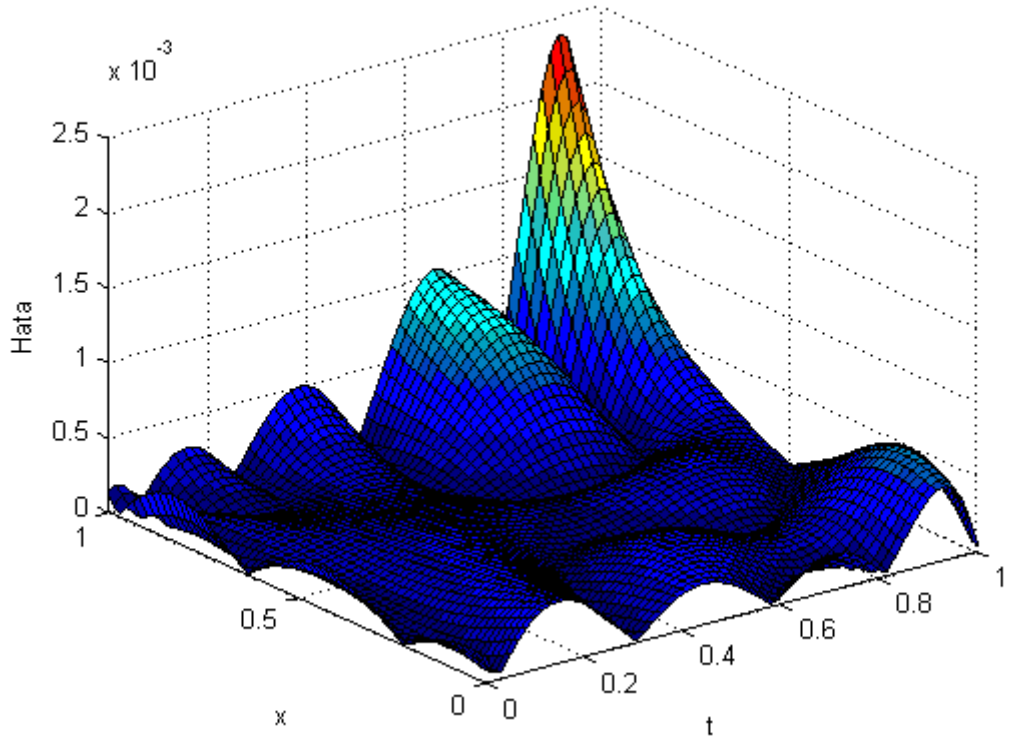


$\varepsilon=2^{-1}$ için $|e_{4,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

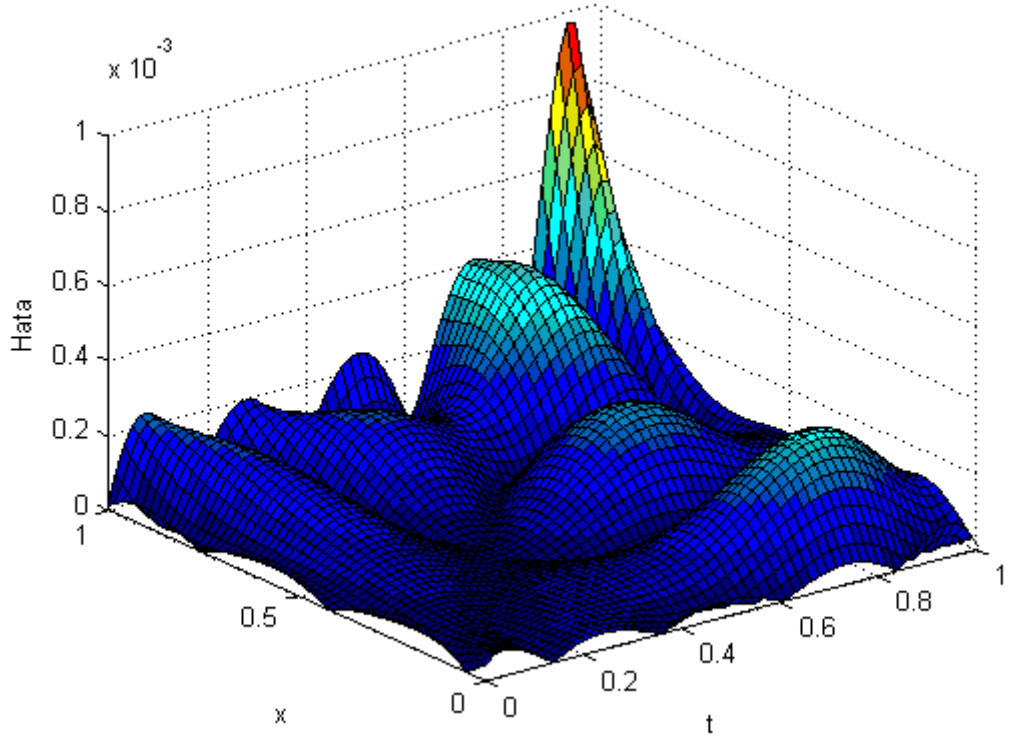


Şekil 4.40. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (4, 5)$ ve $(N, M) = (4, 6)$ için $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

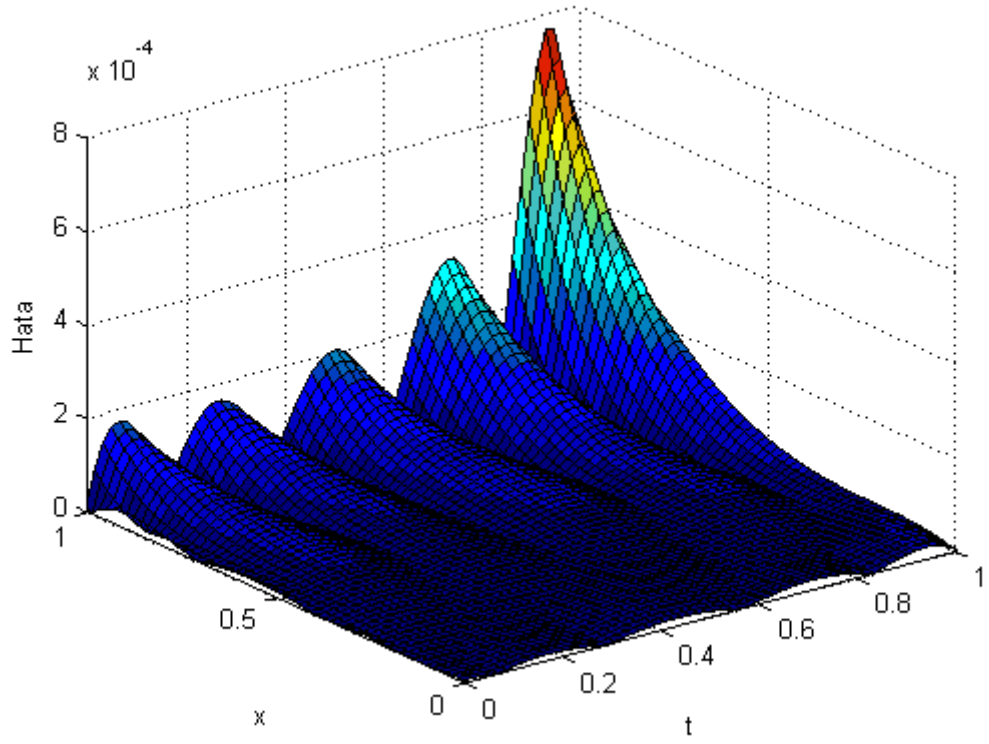


$\varepsilon=2^{-1}$ için $|e_{5,7}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

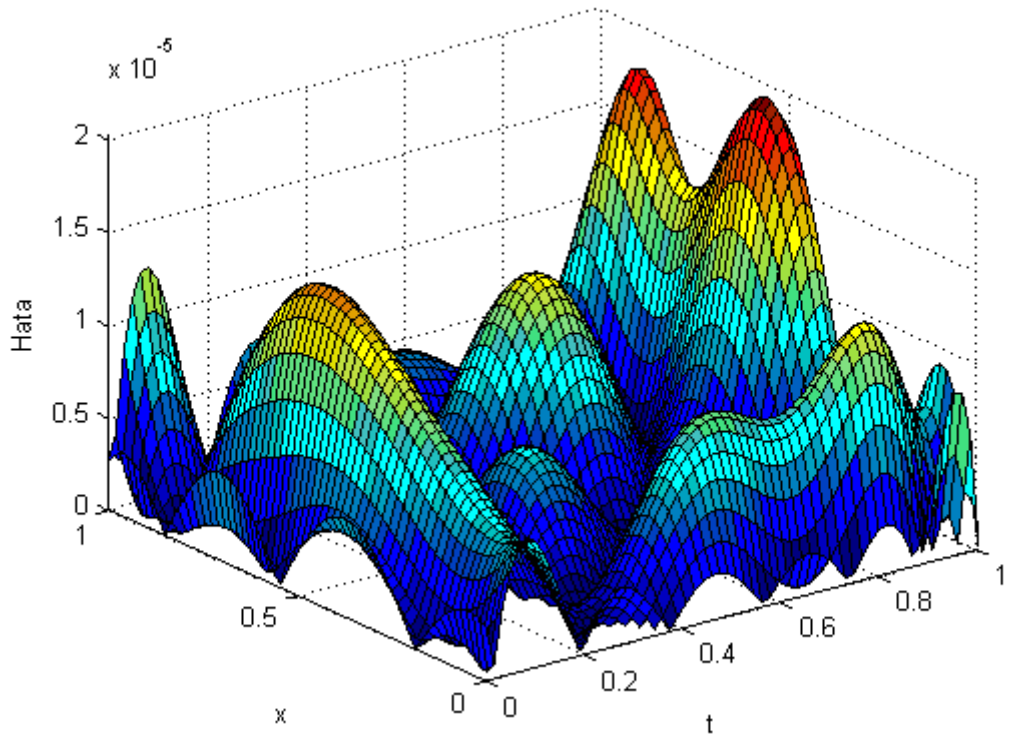


Şekil 4.41. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N, M) = (5, 6)$ ve $(N, M) = (5, 7)$ için $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{6,7}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

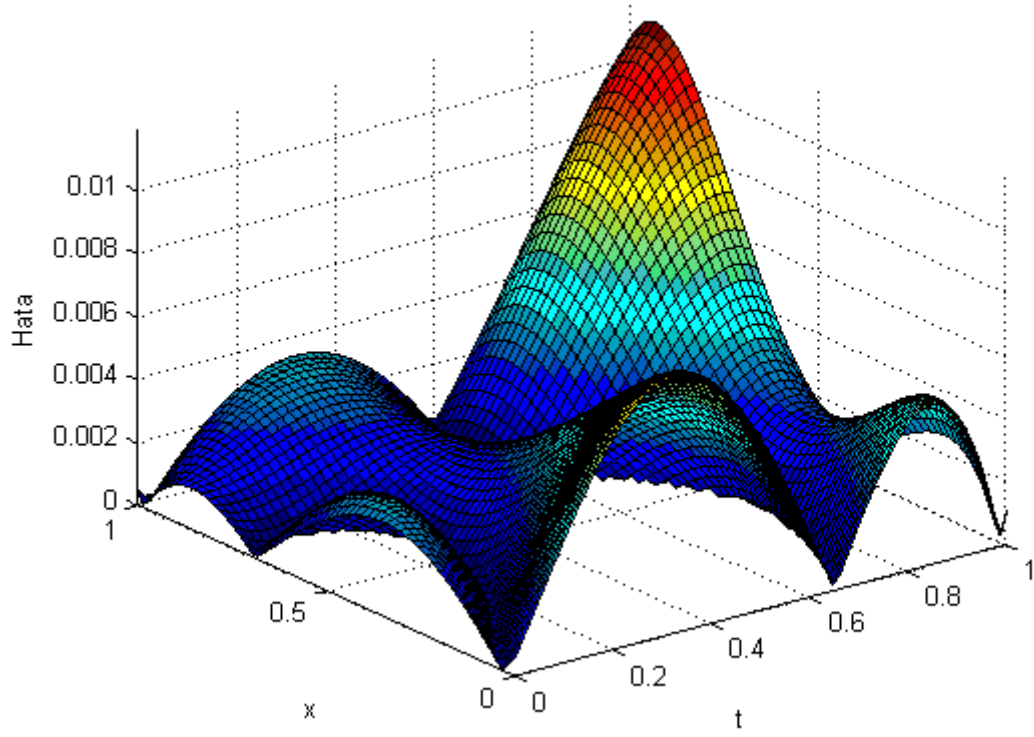


$\varepsilon=2^{-1}$ için $|e_{6,8}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

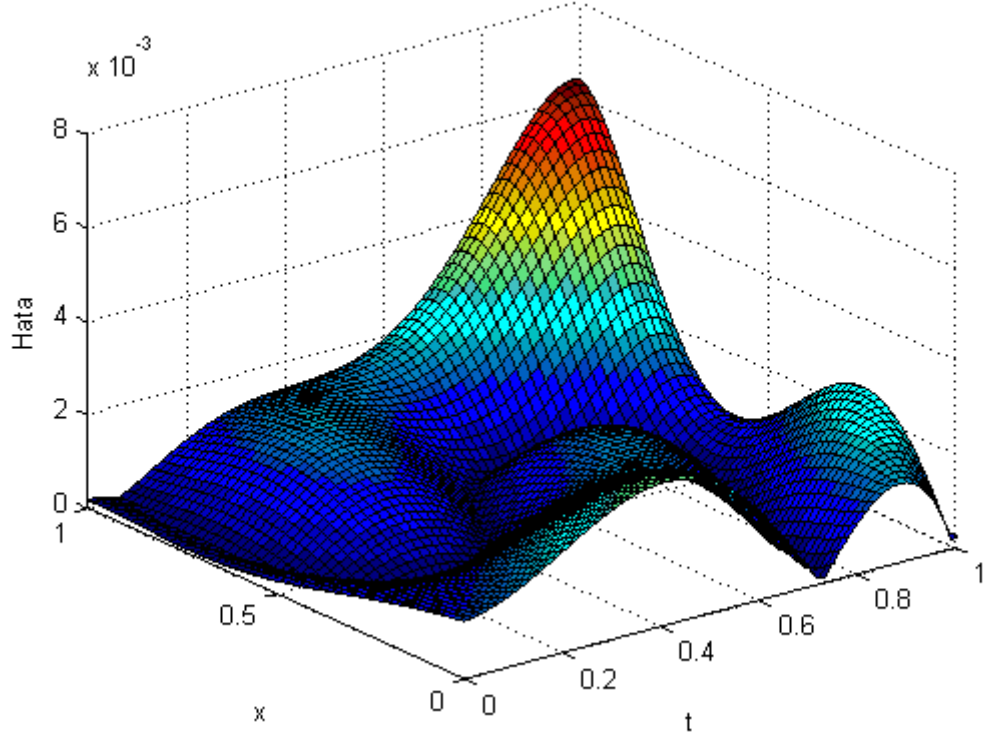


Şekil 4.42. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $(N,M) = (6,7)$ ve $(N,M) = (6,8)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

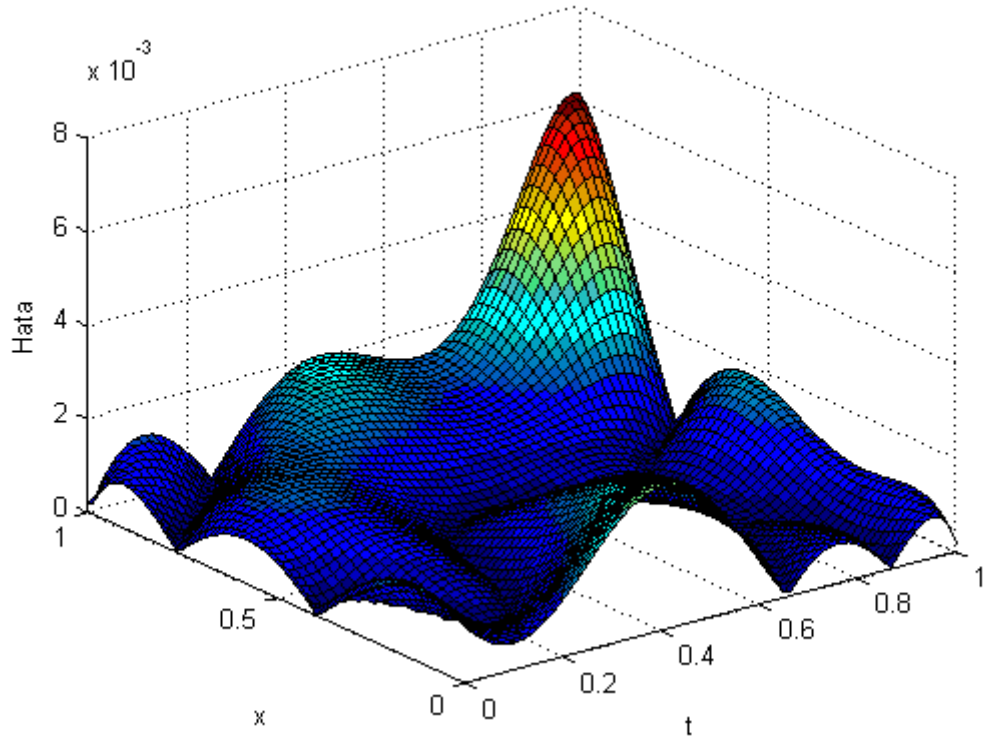
$\varepsilon=2^{-14}$ için $|e_{3,4}(x,t)|$ tahmin edilen mutlak hata fonksiyonu



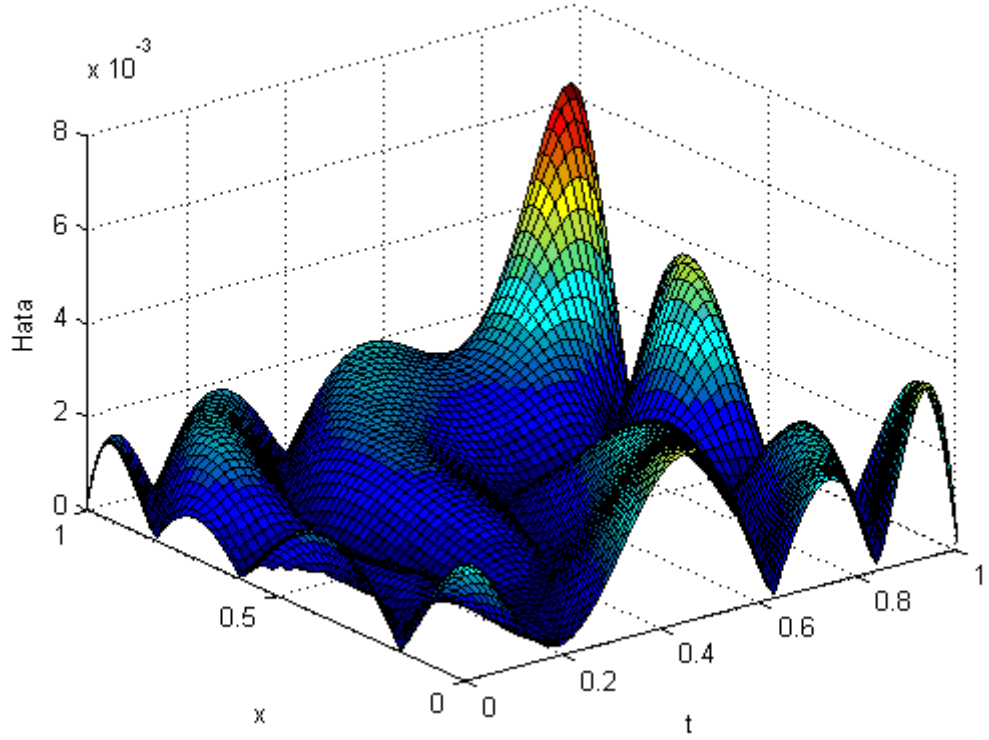
$\varepsilon=2^{-14}$ için $|e_{4,5}(x,t)|$ tahmin edilen mutlak hata fonksiyonu



$\varepsilon=2^{-14}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

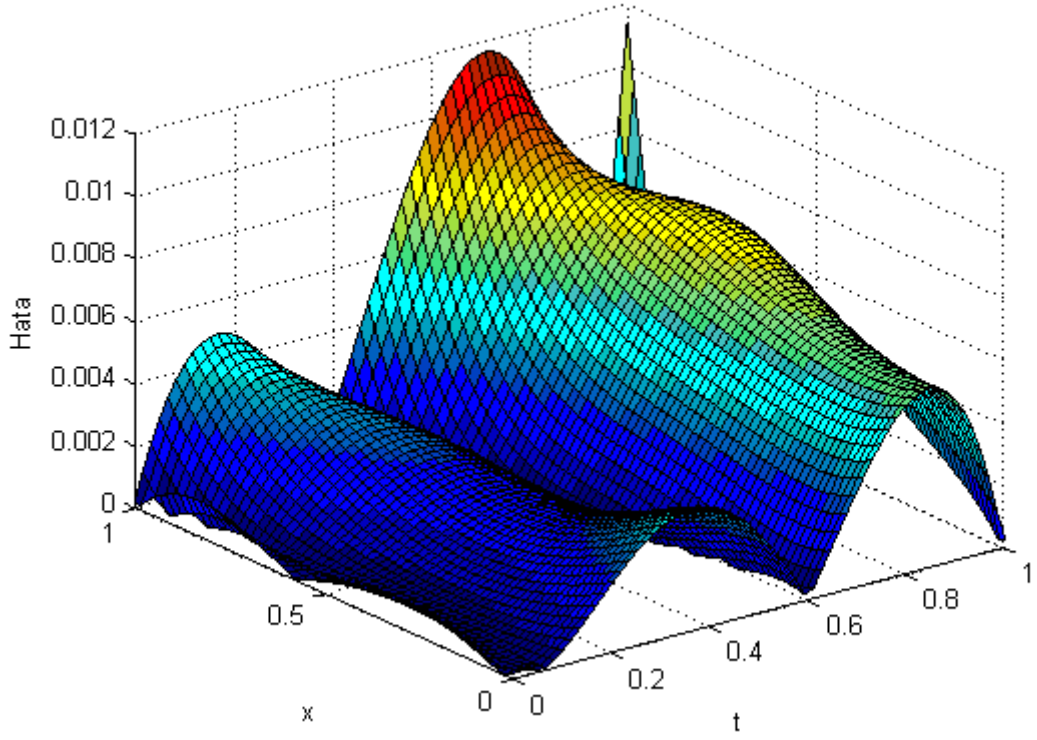


$\varepsilon=2^{-14}$ için $|e_{6,7}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

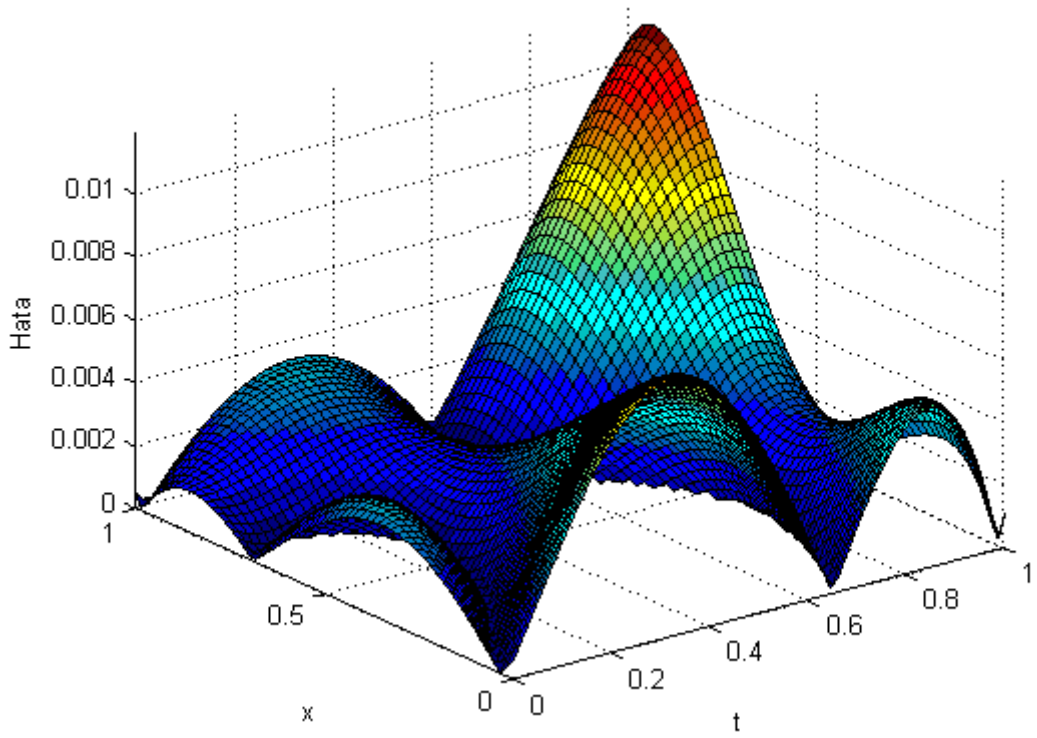


Şekil 4.43. Örnek 4.4.2' nin $\varepsilon = 2^{-14}$, $(N,M)=(3,4)$ ve $(N,M)=(4,5)$, $(N,M)=(4,6)$ ve $(N,M)=(6,7)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{3,4}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

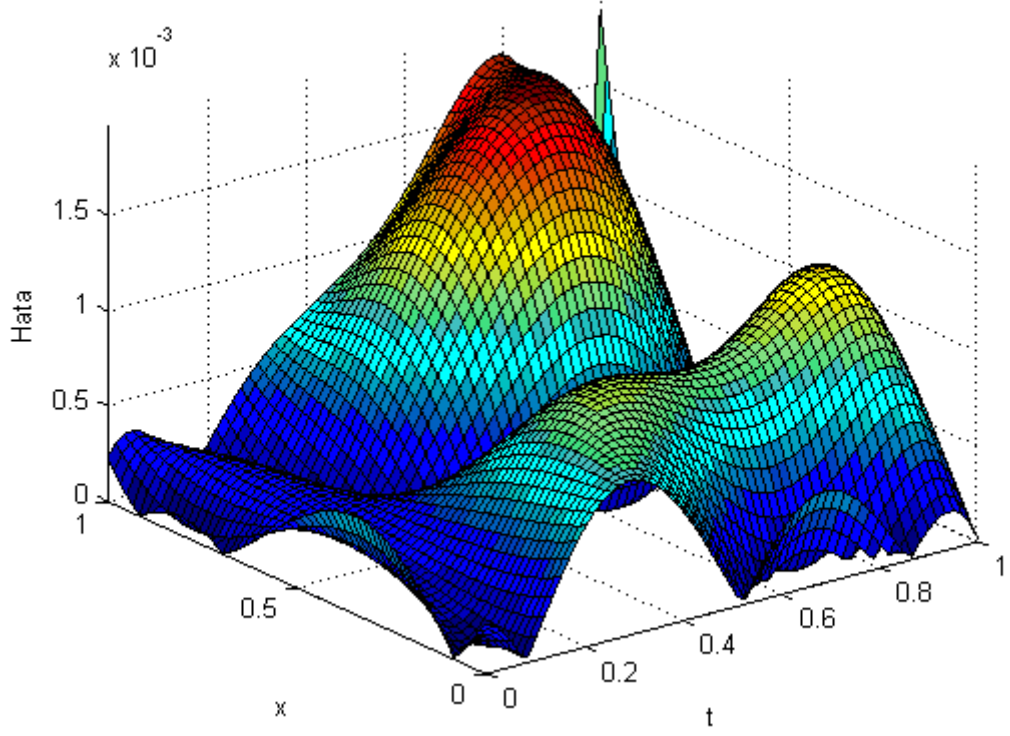


$\varepsilon=2^{-14}$ için $|e_{3,4}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

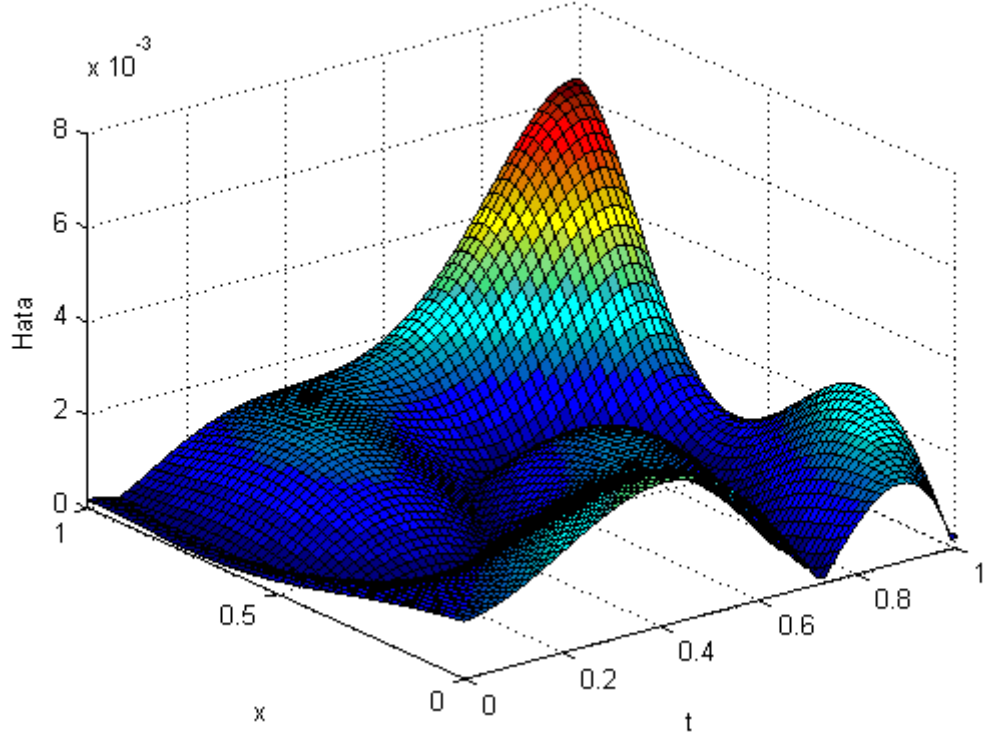


Şekil 4.44. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N,M) = (3,4)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{4,5}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

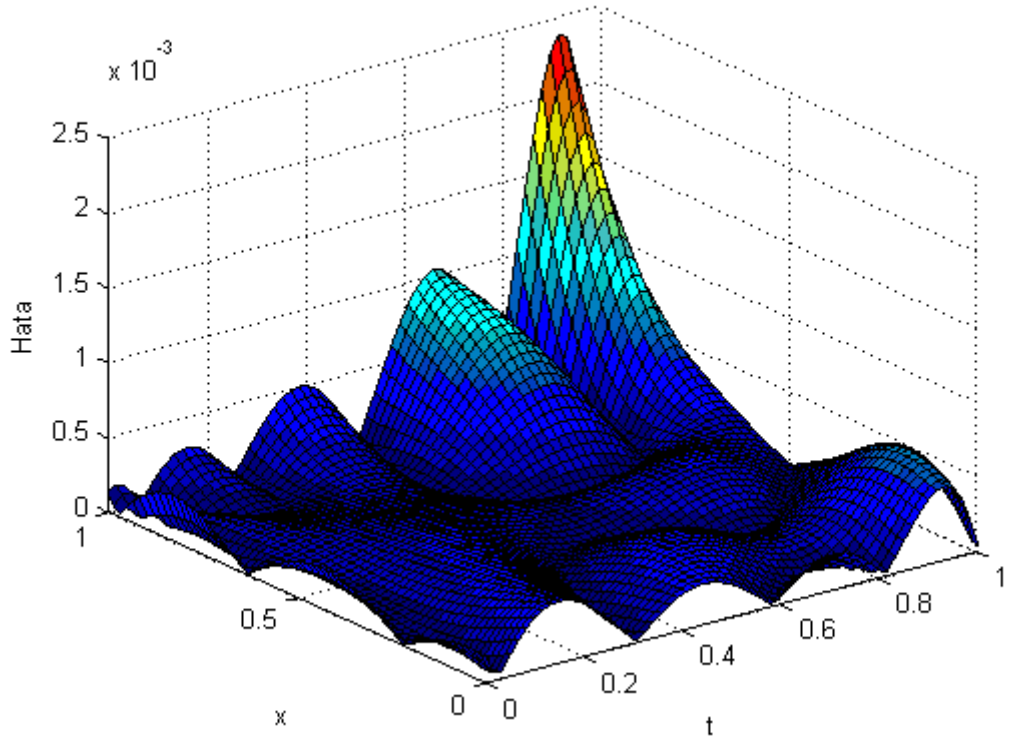


$\varepsilon=2^{-14}$ için $|e_{4,5}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

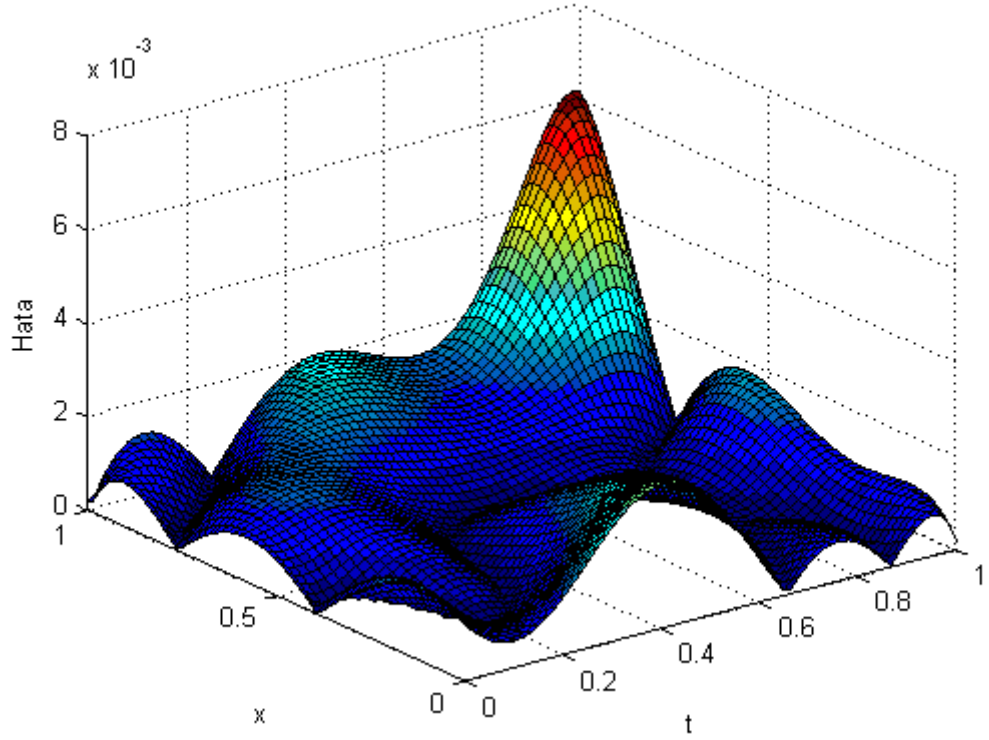


Şekil 4.45. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N,M) = (4,5)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

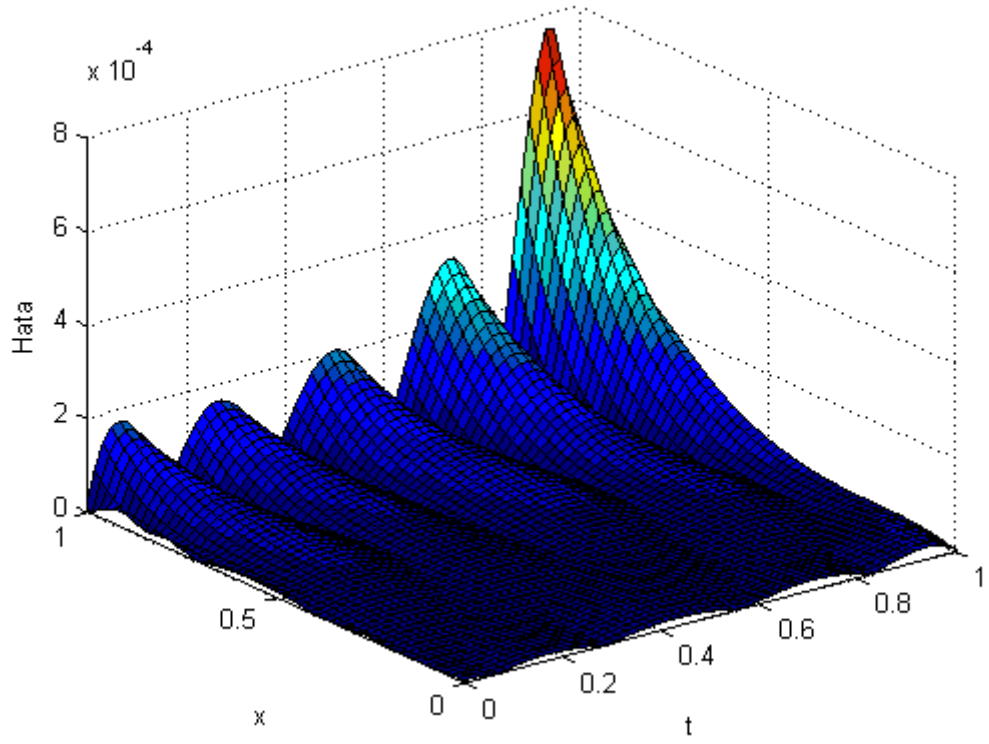


$\varepsilon=2^{-14}$ için $|e_{5,6}(x,t)|$ tahmin edilen mutlak hata fonksiyonu

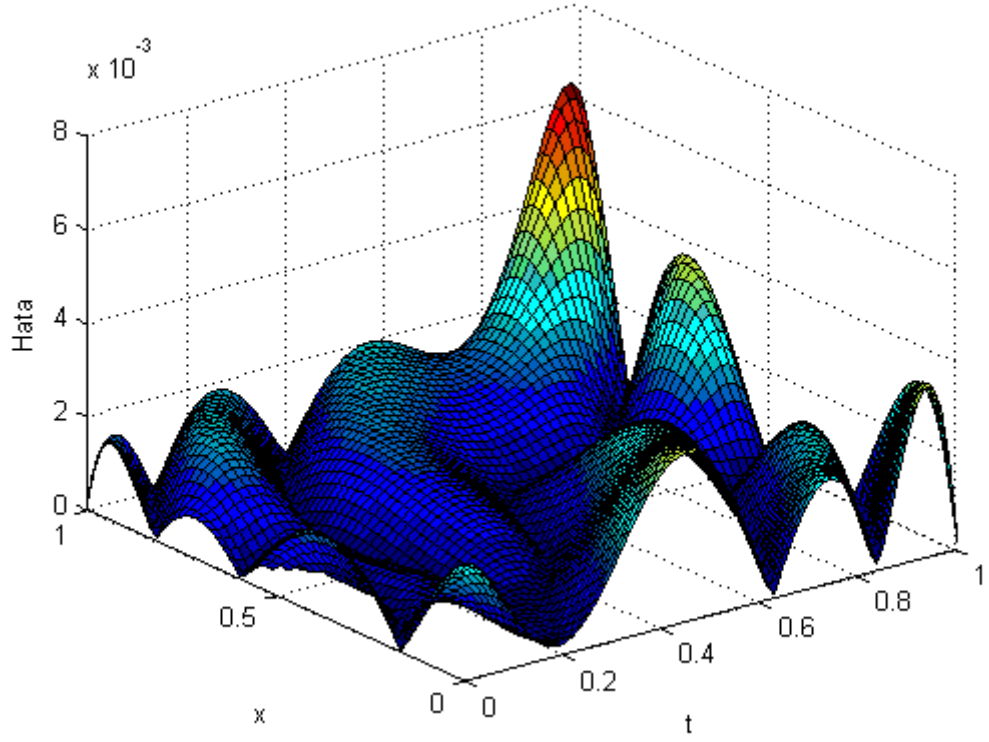


Şekil 4.46. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N, M) = (5, 6)$ için $|e_{N,M}(x, t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

$\varepsilon=2^{-1}$ için $|e_{6,7}(x,t)|$ tahmin edilen mutlak hata fonksiyonu



$\varepsilon=2^{-14}$ için $|e_{6,7}(x,t)|$ tahmin edilen mutlak hata fonksiyonu



Şekil 4.47. Örnek 4.4.2' nin $\varepsilon = 2^{-1}$, $\varepsilon = 2^{-14}$ ve $(N,M) = (6,7)$ için $|e_{N,M}(x,t)|$ tahmin edilen mutlak hata fonksiyonlarının karşılaştırması

4.5. İki Boyutlu İkinci Mertebeden Lineer Kısmi Diferansiyel Denklem Problemi için Örnekler

Örnek 4.5.1: İlk olarak,

$$u_{tt}(x, y, t) + xu_t(x, y, t) + (x + y)u_{xx}(x, y, t) - (x + y - t)u_x(x, y, t) - (2xt)u_{yy}(x, y, t) - 2u(x, y, t) = e^{x+y} \left[x - 2t + (x + y)t - (x + y - t)t - 2xt^2 \right]$$

iki boyutlu lineer değişken katsayılı kısmi diferansiyel denklemini

$$u(x, y, 0) = 0, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1,$$

$$u(x, y, 1) = e^{x+y}, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1,$$

$$u(x, 0, t) = e^x t, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1,$$

$$u(x, 1, t) = e^{x+1} t, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1,$$

$$u(0, y, t) = e^y t, \quad 0 \leq y \leq 1, \quad 0 \leq t \leq 1$$

ve

$$u(1, y, t) = e^{y+1} t, \quad 0 \leq y \leq 1, \quad 0 \leq t \leq 1,$$

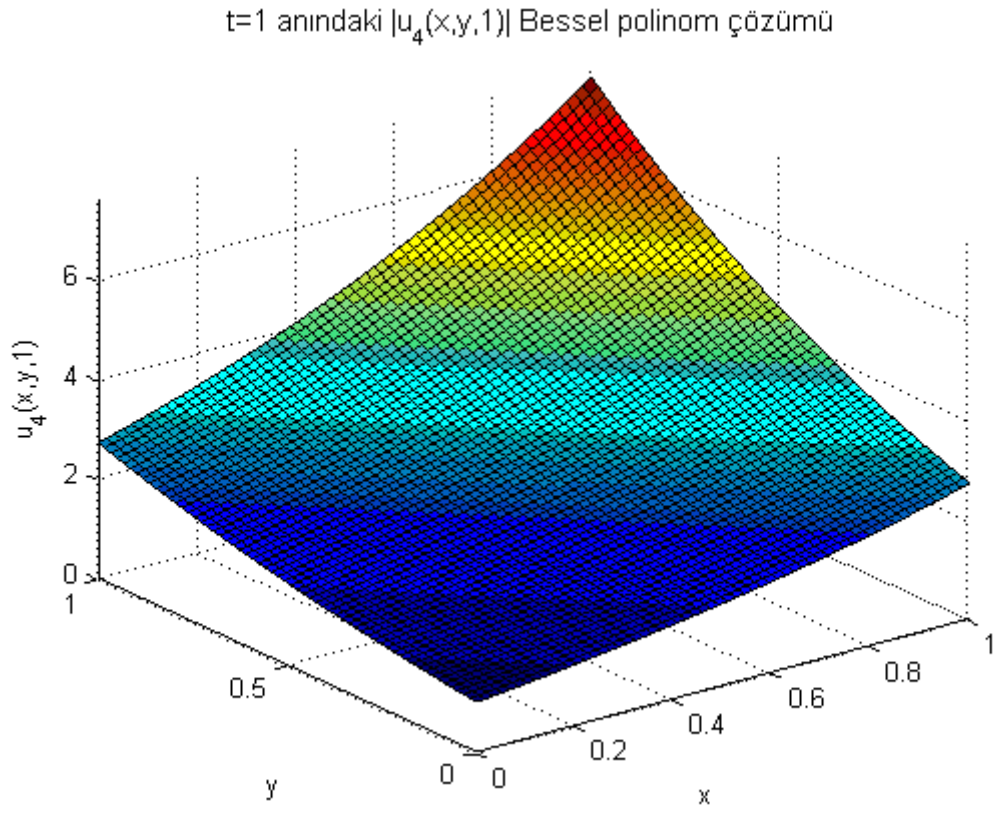
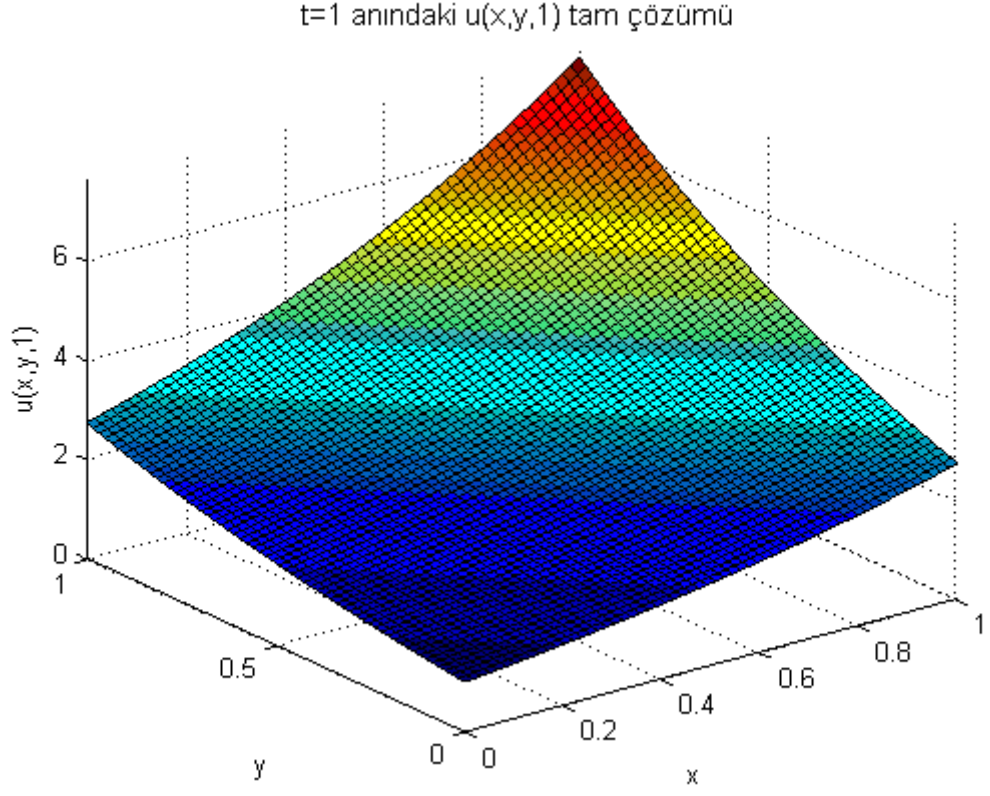
sınır koşulları altında $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 1]$ uzayında ele alalım. Problemin tam çözümü $u(x, y, t) = te^{x+y}$ ' dir.

$N = 4$ ve $M = 5$ için Kesim (3.5)' de sunulan işlemler sırasıyla yapılarak $u_4(x, y, t)$ Bessel polinom çözümü ve $u_{4,5}(x, y, t)$ iyileştirilmiş Bessel polinom çözümü elde edilir. Yaklaşık çözümler ve hata fonksiyonları Maple' da yazılan bilgisayar kodunda virgülden sonra 30 hane kullanılarak elde edildi. $t = 1$ zamanındaki $u(x, y, 1) = e^{x+y}$ tam çözümü, $u_4(x, y, 1)$ Bessel polinom çözümü ve $u_{4,5}(x, y, 1)$ iyileştirilmiş Bessel polinom çözümünün grafiklerini Şekil 4.48' de sunuyoruz. $t = 1$ Zamanındaki

$|e_4(x, y, 1)|$ mutlak hata fonksiyonu ve $|e_{4,5}(x, y, 1)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırması Şekil 4.49’ da verilir. $t = 1/2$ zamanındaki $|e_4(x, y, 1/2)|$ mutlak hata fonksiyonu ve $|e_{4,5}(x, y, 1/2)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırması Şekil 4.50’ de verilir. Şekil 4.51 $t = 1/2$ zamanındaki $|e_4(x, y, 1/2)|$ mutlak hata fonksiyonu ve $|E_{4,5}(x, y, 1/2)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırmasını gösterir. $t = 0$ zamanındaki $|e_7(x, y, 0)|$ mutlak hata fonksiyonu ve $|e_{7,8}(x, y, 0)|$ tahmin edilen mutlak hata fonksiyonunu Şekil 4.52’ de karşılaştırıyoruz. Çizelge 4.15’ de farklı zamanlardaki maksimum hatalar sunulur.

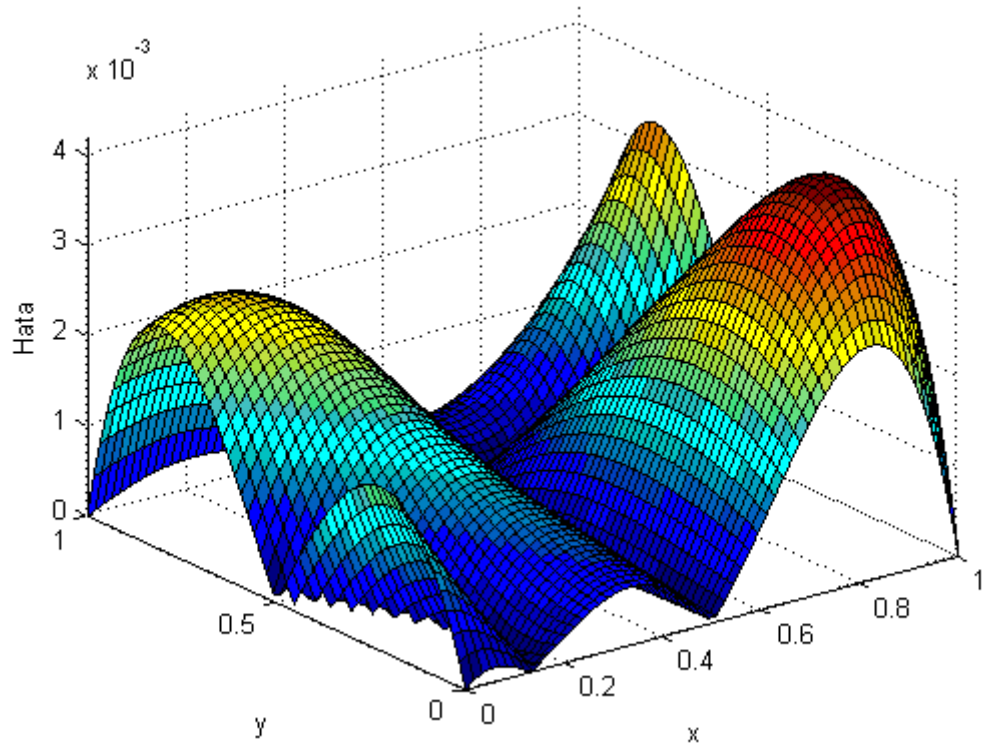
Çizelge 4.15. Örnek 4.5.1’ in farklı zamanındaki maksimum hataları

$t = 1/2$ zamanında			
(N, M)	$ e_N(x, t) $ için	$ e_{N,M}(x, t) $ için	$ E_{N,M}(x, t) $ için
$(4, 5)$	3.4×10^{-3}	4.31×10^{-4}	3.3×10^{-3}
$t = 0$ zamanında			
(N, M)	$ e_N(x, t) $ için	$ e_{N,M}(x, t) $ için	$ E_{N,M}(x, t) $ için
$(4, 5)$	2.9×10^{-3}	9.5×10^{-4}	2.9×10^{-3}
$t = 0$ zamanında			
(N, M)	$ e_N(x, t) $ için	$ e_{N,M}(x, t) $ için	$ E_{N,M}(x, t) $ için
$(7, 8)$	1.483×10^{-7}	4.173×10^{-8}	1.356×10^{-7}
$t = 1$ zamanında			
(N, M)	$ e_N(x, t) $ için	$ e_{N,M}(x, t) $ için	$ E_{N,M}(x, t) $ için
$(4, 5)$	4.0×10^{-3}	7.01×10^{-4}	4.0×10^{-3}

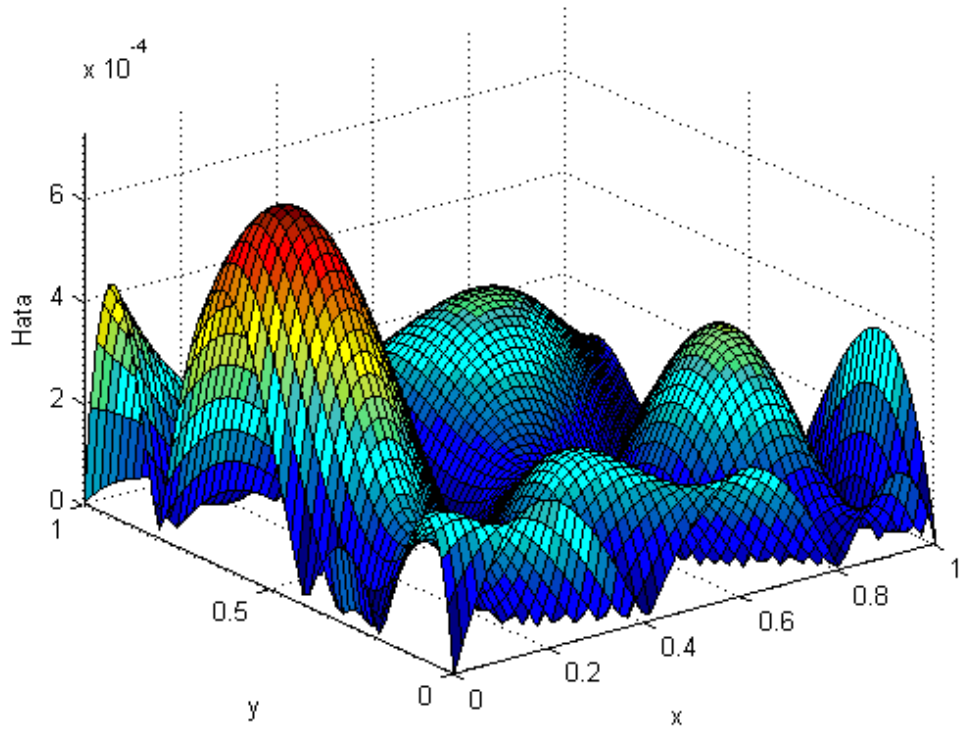


Şekil 4.48. Örnek 4.5.1' in $t = 1$ zamanındaki $u(x,y,1)$ tam çözümü ve $N = 4$ için $u_N(x,y,1)$ Bessel polinom çözümünün karşılaştırması

$t=1$ anındaki $|e_4(x,y,1)|$ mutlak hata fonksiyonu

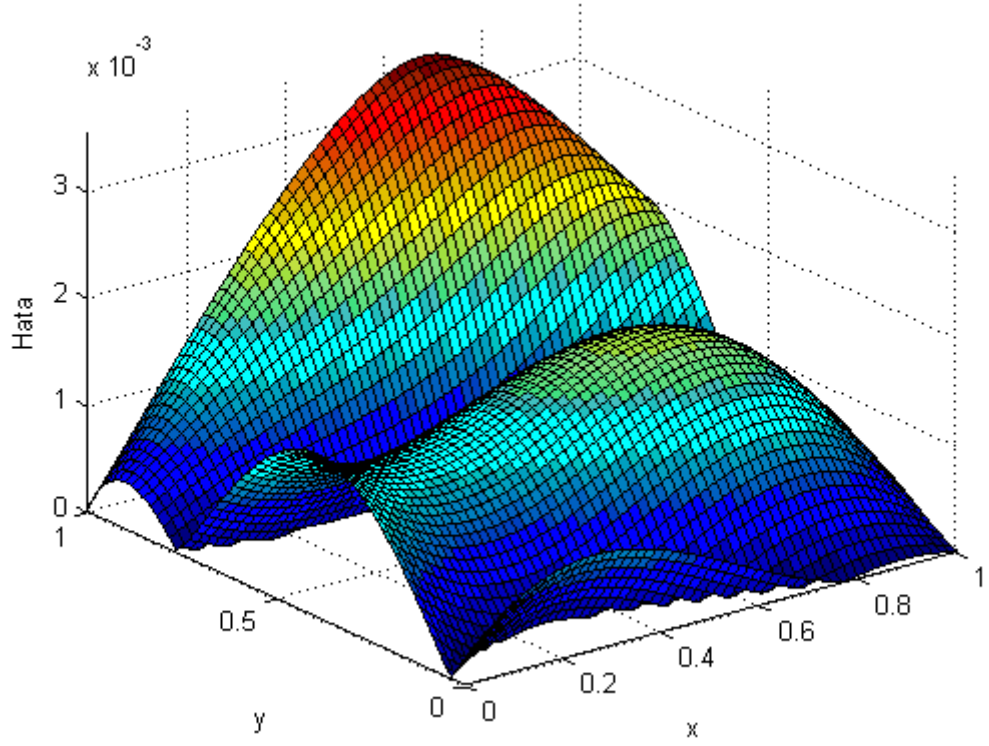


$t=1$ anındaki $|e_{4,5}(x,y,1)|$ tahmin edilen mutlak hata fonksiyonu

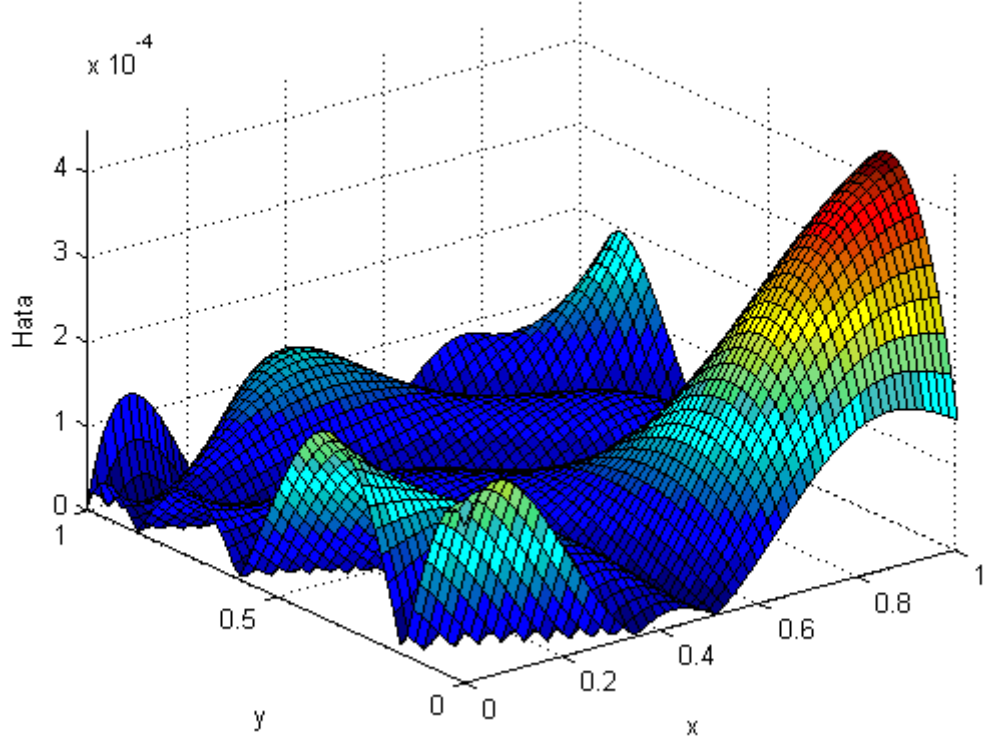


Şekil 4.49. Örnek 4.5.1' in $t=1$ zamanındaki $N=4$ ve $M=5$ için $|e_N(x,y,1)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,y,1)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$t=1/2$ anındaki $|e_4(x,y,1/2)|$ mutlak hata fonksiyonu

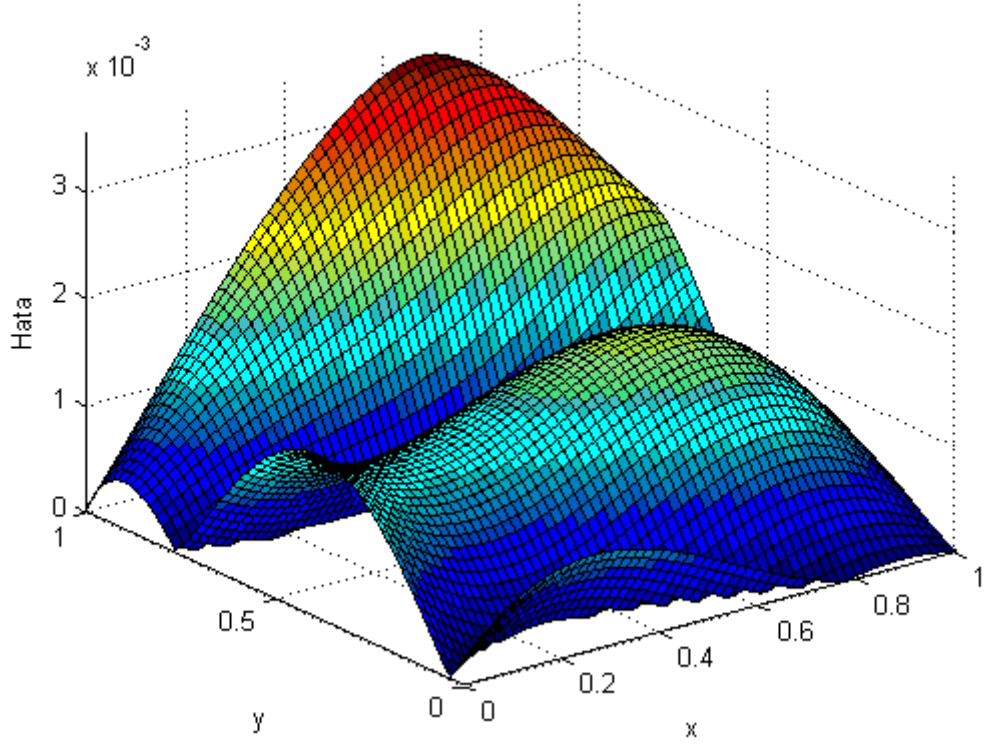


$t=1/2$ anındaki $|e_{4,5}(x,y,1/2)|$ tahmin edilen mutlak hata fonksiyonu

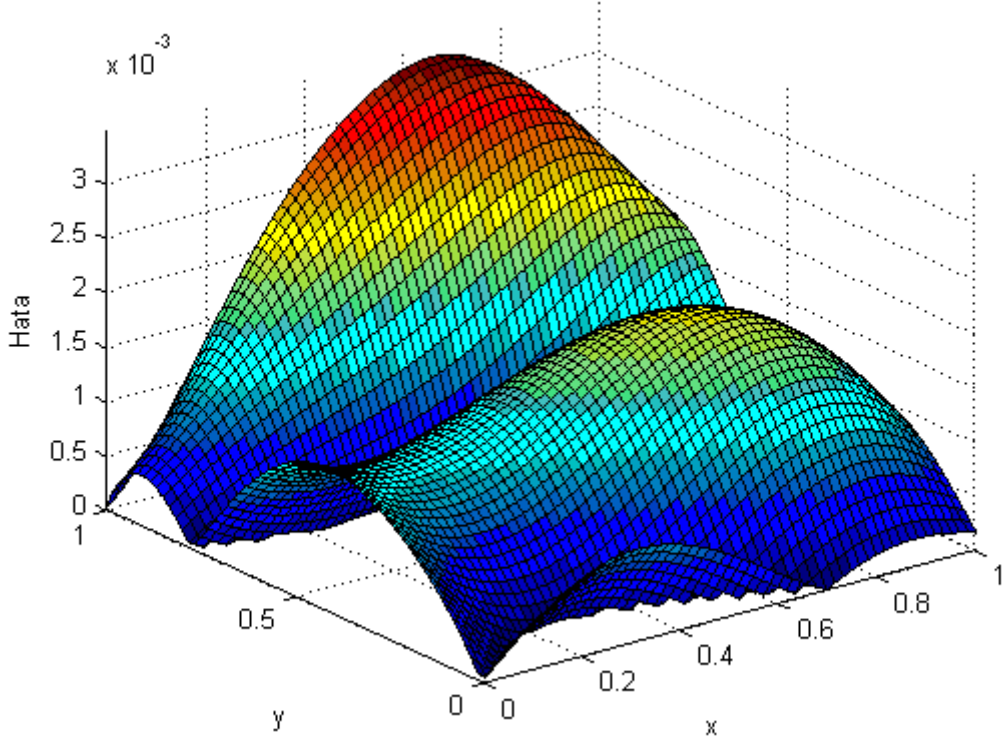


Şekil 4.50. Örnek 4.5.1' in $t = 1/2$ zamanındaki $N = 4$ ve $M = 5$ için $|e_N(x,y,1/2)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,y,1/2)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

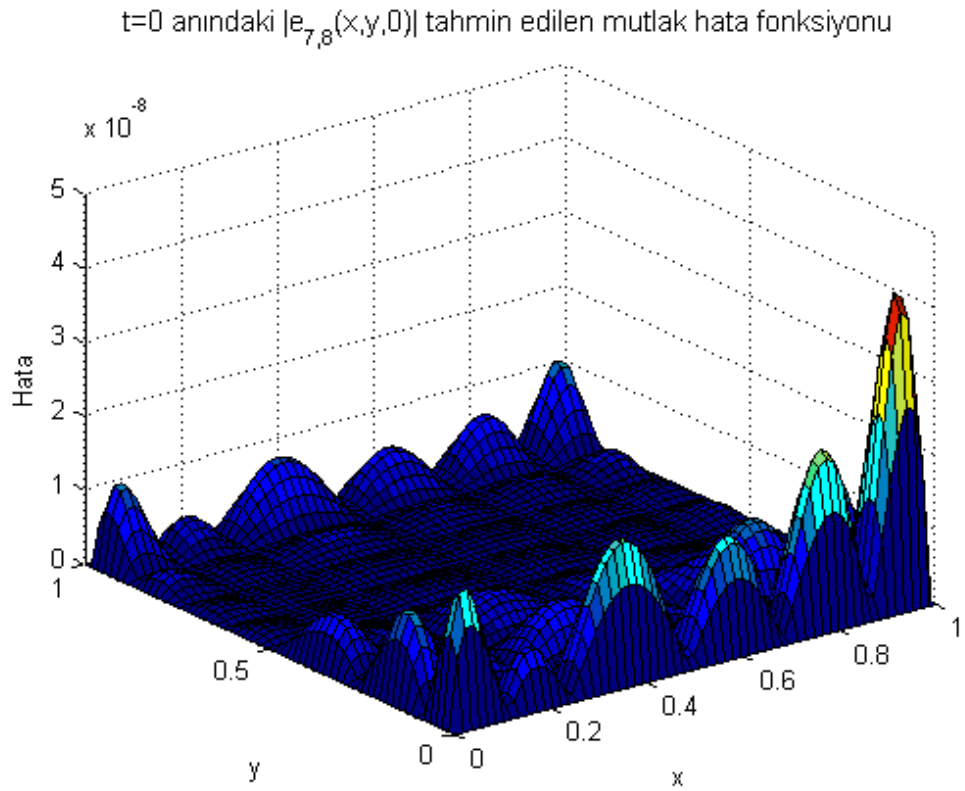
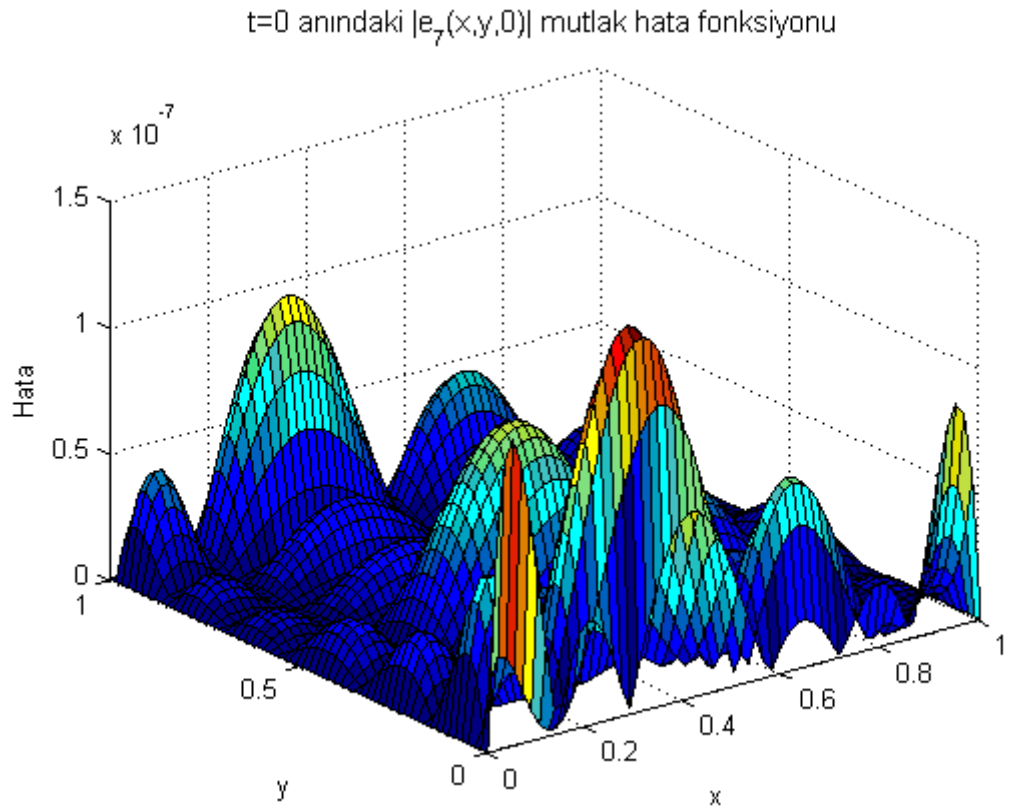
$t=1/2$ anındaki $|e_4(x,y,1/2)|$ mutlak hata fonksiyonu



$t=1/2$ anındaki $|E_{4,5}(x,y,1/2)|$ iyileştirilmiş mutlak hata fonksiyonu



Şekil 4.51. Örnek 4.5.1' in $t = 1/2$ zamanındaki $N = 4$ ve $M = 5$ için $|e_N(x,y,1/2)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x,y,1/2)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması



Şekil 4.52. Örnek 4.5.1' in $t = 0$ zamanındaki $N = 7$ ve $M = 8$ için $|e_N(x, y, 0)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x, y, 0)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

Örnek 4.5.2: Şimdi, (Bülbül, 2011)

$$u_{tt}(x, y, t) + 2u_t(x, y, t) + u_{xx}(x, y, t) + u_{yy}(x, y, t) + u(x, y, t) = 2e^{x+y-t}$$

iki boyutlu lineer sabit katsayılı hiperbolik kısmi diferansiyel denklemini

$$u(x, y, 0) = 0, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

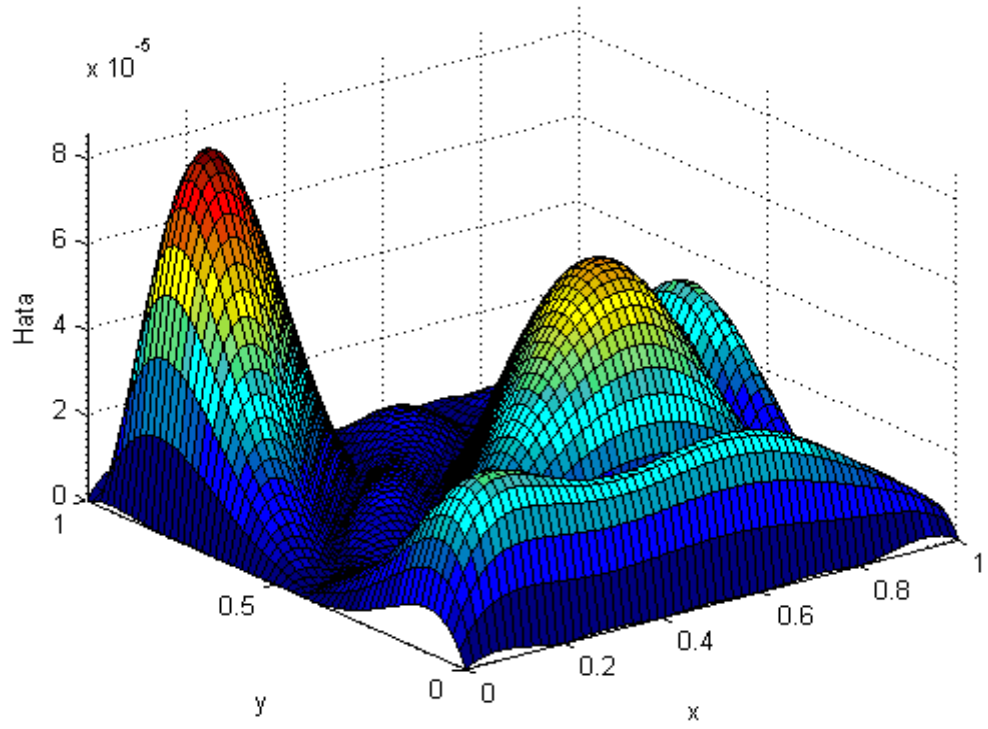
ve

$$u_t(x, y, 0) = -e^{x+y}, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

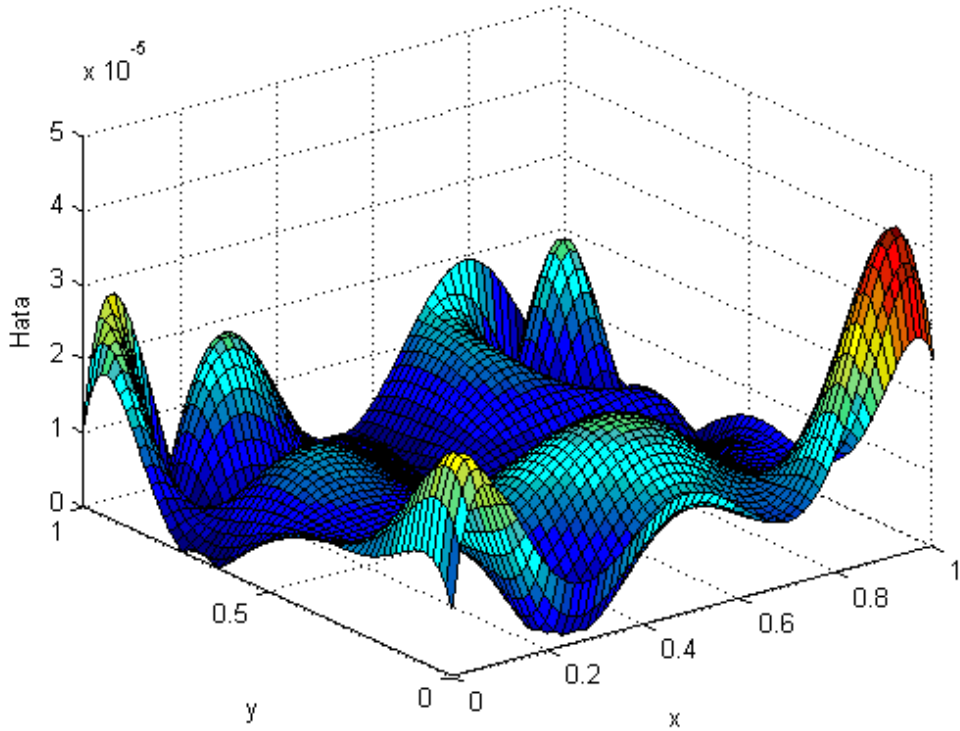
başlangıç koşulları altında $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 1]$ uzayında ele alalım. Verilen problemin tam çözümü $u(x, y, t) = e^{x+y-t}$ dir.

Bu problem $N = 5$ alınarak sunulan yöntem ile çözüldüğünde elde edilen yaklaşık çözümün ve $N = 5$ için Taylor matris yöntemi (Bülbül, 2011) ile elde edilen yaklaşık çözümün $t = 0 : 0.1 : 1$ zamanlarındaki L_2 ve L_∞ hataları Çizelge 4.16' da karşılaştırılır. Çizelge 4.16' dan sunulan yöntemin hatalarının Taylor yönteminin hatalarından daha küçük olduğunu gözlemliyoruz. Şekil 4.53' de $t = 0$ zamanındaki $|e_5(x, y, 0)|$ mutlak hata fonksiyonu ve $|e_{5,6}(x, y, 0)|$ tahmin edilen mutlak hata fonksiyonu karşılaştırılır. $t = 0$ zamanındaki $|e_5(x, y, 0)|$ mutlak hata fonksiyonu ve $|E_{5,6}(x, y, 0)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırması Şekil 4.54' de sunulur. Yaklaşık çözümler ve hata fonksiyonları Maple' da yazılan bilgisayar programında virgülden sonra 30 hanede duyarlılıkla hesaplandığını not ediyoruz.

$t=0$ anındaki $|e_5(x,y,0)|$ mutlak hata fonksiyonu

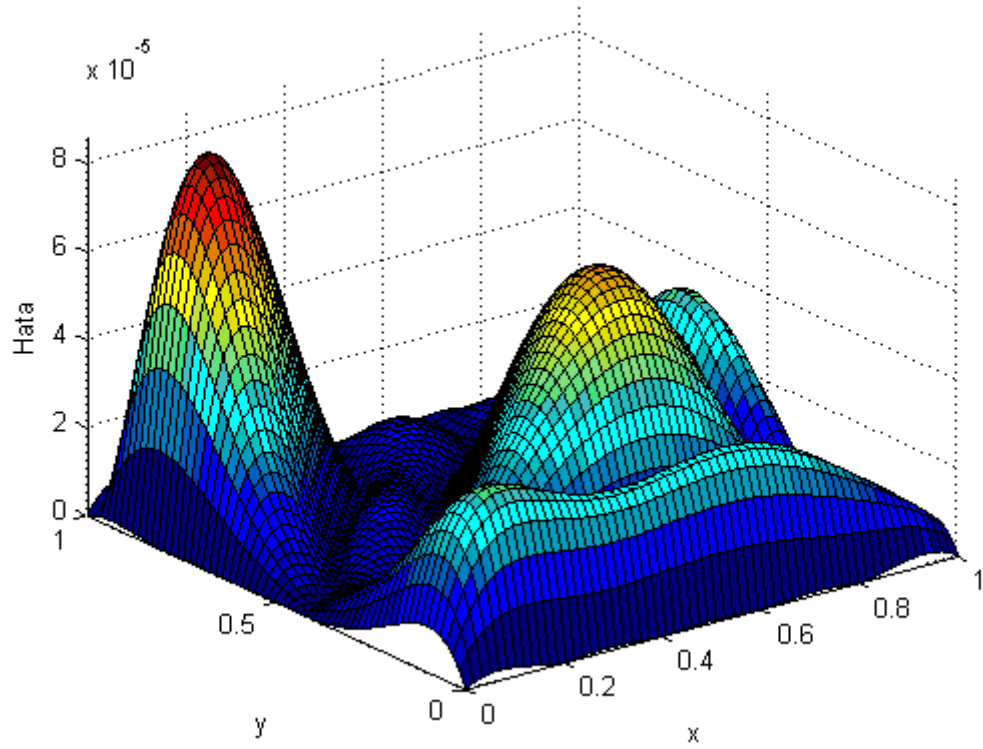


$t=0$ anındaki $|e_{5,6}(x,y,0)|$ tahmin edilen mutlak hata fonksiyonu

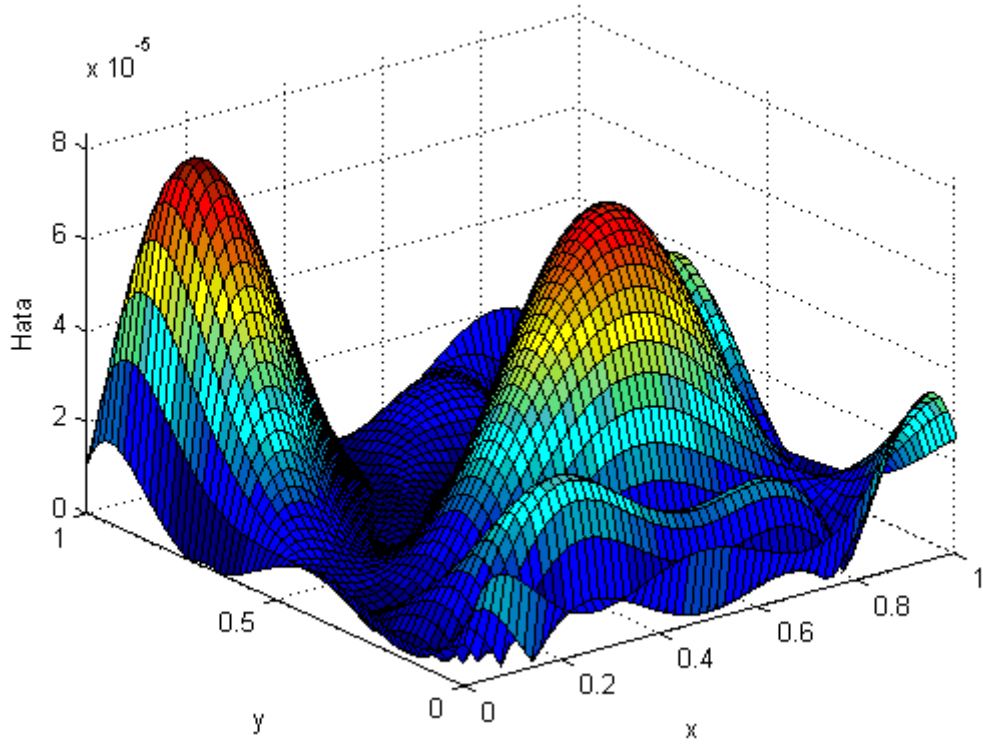


Şekil 4.53. Örnek 4.5.2' nin $t = 0$ zamanındaki $N = 5$ ve $M = 6$ için $|e_N(x,y,0)|$ mutlak hata fonksiyonu ve $|e_{N,M}(x,y,0)|$ tahmin edilen mutlak hata fonksiyonunun karşılaştırılması

$t=0$ anındaki $|e_5(x,y,0)|$ mutlak hata fonksiyonu



$t=0$ anındaki $|E_{5,6}(x,y,0)|$ iyileştirilmiş mutlak hata fonksiyonu



Şekil 4.54. Örnek 4.5.2' nin $t = 0$ zamanındaki $N = 5$ ve $M = 6$ için $|e_N(x, y, 0)|$ mutlak hata fonksiyonu ve $|E_{N,M}(x, y, 0)|$ iyileştirilmiş mutlak hata fonksiyonunun karşılaştırılması

Çizelge 4.16. Örnek 4.5.2'nin $N = 5$ için sunulan yöntem ve Taylor matris yönteminin L_2 ve L_∞ hatalarının karşılaştırılması

t	$N = 5$ için sunulan yöntem		$N = 5$ için Taylor matris yöntemi (Bülbül, 2011)	
	L_∞	L_2	L_∞	L_2
0	8.2962e-005	2.5893e-005	8.77e-03	1.93e-02
0.1	4.8601e-004	7.5001e-005	9.18e-03	2.40e-02
0.2	1.5352e-003	1.9694e-004	1.18e-02	7.62e-02
0.3	2.1919e-003	3.0611e-004	1.63e-02	2.83e-02
0.4	2.4776e-003	5.3960e-004	2.25e-02	4.63e-02
0.5	4.2375e-003	1.2839e-003	3.00e-02	4.44e-02
0.6	9.9380e-003	2.6961e-003	3.89e-02	8.88e-02
0.7	2.1660e-002	4.7969e-003	4.59e-02	9.46e-04
0.8	3.7513e-002	7.4997e-003	7.48e-02	1.47e-01
0.9	5.6473e-002	1.0566e-002	5.16e-02	4.44e-01
1	7.6356e-002	1.3583e-002	5.24e-02	6.12e-01

5.TARTIŞMA VE SONUÇLAR

Fizik ve mühendislik alanlarındaki birçok problemde kısmi diferansiyel denklemler ile karşılaşılır. Kısmi diferansiyel denklemlerin tam çözümlerinin elde edilmesinin zor olduğu durumlarda ya da tam çözümlerinin olmaması durumlarında nümerik metotlara ihtiyaç duyulur. Son yıllarda, adi diferansiyel, fark, integral ve integro-diferansiyel-fark denklemlerin nümerik çözümleri için Bessel sıralama metodu kullanıldı. Bu çalışmada,

- Neuman, Dirichlet ve integral koşullu dalga problemi
- Neuman, Dirichlet ve Robin koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi
- Başlangıç ve sınır koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi
- Singüler pertürbe olmuş bir boyutlu parabolik konveksiyon-difüzyon problemi
- İki boyutlu ikinci mertebeden lineer değişken katsayılı kısmi diferansiyel denklem problemi

için Bessel sıralama metodu uygulandı ve her bir problem için rezidüel fonksiyonu temel alan hata tahmini verildi. Ayrıca, her bir problemin Bessel polinom çözümleri bu hata fonksiyonu yardımıyla iyileştirildi. İlavenen, bu çalışmamızda yöntemin karmaşıklığı üzerinede çalıştık. Başlangıç ve sınır koşullu genel formda ikinci mertebe lineer kısmi diferansiyel denklem problemi için yöntemin karmaşıklığı hesaplandı.

Bu çalışmanın dördüncü bölümünde yöntemin uygulaması için ele alınan tüm problemler için örnekler sunuldu. Örneklerde, u tam çözümlerin, u_N yaklaşık çözümlerin, $u_{N,M}$ iyileştirilmiş yaklaşık çözümlerin, e_N gerçek mutlak hata fonksiyonlarının, $e_{N,M}$ tahmin edilen mutlak hata fonksiyonlarının ve $E_{N,M}$ iyileştirilmiş mutlak hata fonksiyonlarının farklı N değerleri için grafikleri çizildi. Şekillerden ve Çizelgelerden $e_{N,M}$ tahmin edilen mutlak hata fonksiyonlarının e_N

gerçek mutlak hata fonksiyonlarına oldukça yakın olduğu gözlemlenir ve yapılan iyileştirmeninde etkili olduğu görülür.

Hata fonksiyonu tahmini oldukça etkili olduğundan herhangi bir problemin tam çözümü bilinmediği zaman sonuçların güvenilirliğini $e_{N,M}$ tahmin edilen mutlak hata fonksiyonu ile ölçebiliriz. Örneğin, Örnek 4.4.2' nin tam çözümü mevcut değildir. Bu problemde, çözümlerin güvenilirliğini $e_{N,M}$ aracılığıyla test ettik. Ayrıca, çözümlerin doğruluğu ($R_{N,M}$ rezidüel fonksiyon) kullanılarak da ölçüm yapıldı. Çözümlerin güvenilirliğinin ekstra bir ölçümü için N ve ε ' nun farklı değerleri ile sunulan yöntem, Euler implicit metodu (Clavero, 2003), B-spline sıralama metodu (BSM) (Kadalbajoo vd., 2008) ve parçalı analitik metodu (Ramos, 2005) için

$$e_{\varepsilon}^N = \max \left\{ |u_N(x,t) - u_{2N}(x,t)|, 0 \leq x \leq 1, 0 \leq t \leq 1 \right\}$$

ile elde edilen maksimum noktasal hatalar karşılaştırıldı. Yapılan karşılaştırmada, bizim sonuçların oldukça etkili olduğu görülür. Dahası, N değeri artarken hataların azaldığı görülür.

Çoğu örnekte extra ölçüm için, L_2 ve L_{∞} hatalarının ikisi ile de yöntemin güvenilirliğini test ettik. L_2 ve L_{∞} hatalarında da N değeri artarken hataların azaldığı gözlemlenir.

Diğer örneklerde de başka metotlar ile karşılaştırmalar yapıldı. Yapılan karşılaştırmalardan genel olarak bizim metodun oldukça etkili olduğu gözlemlenir.

Yaklaşık çözümler Matlab, Maple ve Mathematica gibi bilgisayar programları kullanılarak kolayca hesaplanabilir. Bizim örneklerde, yöntemin uygulamaları Matlab ve Maple programlarında yazılan kodlar çalıştırılarak elde edildi. Bu kodlar sayesinde yaklaşık çözümler kısa sürede hesaplandı. Ayrıca, bu problemler için genellikle Maple' da yaklaşık çözümlerin daha kısa sürede elde edildiğini gözlemledik.

Örneklerde N kesme sınırı artarken yaklaşık çözümlerin tam çözüme daha da yaklaştığı saptandı; fakat kesme sınırı N ' nin çok büyük seçilmesi durumunda, bilgisayarda işlem yükü artacağından, bilgisayar programları daha yavaş sonuç verecektir. Ayrıca, N değeri artarken programdan kaynaklanan hesaplama hatalarında

büyür. Bu nedenle terim sayısını arttırmak çok kullanışlı değildir. Bu sebeple, N değeri yeteri kadar büyük seçildiğinde tam çözüme oldukça yakın çözümler elde edilebilir.

Bu yöntem başka kısmi diferansiyel denklem problemleri için geliştirilerek uygulanabilir.

KAYNAKLAR

- Akyüz-Daşcıoğlu, A. ve Sezer, M. (2005) Chebyshev polynomial solutions of systems of higher-order linear Fredholm–Volterra integro-differential equations, *J Frank Inst*, 342 (6): 688-701.
- Akyüz-Daşcıoğlu, A. (2009) Chebyshev polynomial approximation for high-order partial differential equations with complicated conditions, *Numer Meth Part Diff Equ*, 25 (3): 610-621.
- Anar, İ. E. (2005) *Kısmi Diferansiyel Denklemler*, Palme Yayıncılık, Ankara, 560s.
- Ateş, İ. ve Yıldırım, A. (2010) Comparison between variational iteration mMethod and homotopy perturbation method for linear and nonlinear partial differential equations with the nonhomogeneous initial conditions, *Numer Meth Part Diff Equ*, 26 (6): 1581-1593.
- Azari, H. ve Zhang, S. (2010) A numerical approximation to the solution of an inverse heat conduction problem, *Numer Meth Part Diff Equ*, 26 (1): 95-106.
- Bashier, E.B.M. ve Patidar, K.C. (2011) A fitted numerical method for a system of partial delay differential equations, *Comput Math Appl*, 61 (6): 1475-1492.
- Benito, J.J., Urena, F. ve Gavete L. (2007) Solving parabolic and hyperbolic by the generalized finite difference method, *Appl Math Comput*, 209 (2): 208-233.
- Bernatz, R. (2010) *Fourier Series and Numerical Methods for Partial Differential Equations*, John Wiley & Sons, Canada, 318s.
- Biazar, J. ve Ebrahimi, H. (2007) An Approximation to the Solution of Telegraph Equation by Adomian Decomposition Method, *Int Math Forum*, 2 (45): 2231-2236.
- Biazar, J. ve Ghazvini, H. (2008) Homotopy perturbation method for solving hyperbolic partial differential equations, *Comput Math Appl*, 56 (2): 453-458.
- Biazar, J. ve Aminikhah, H. (2009) Study of convergence of homotopy perturbation method for systems of partial differential equations, *Comput Math Appl*, 58 (11-12): 2221-2230.

- Biazar, J., Ebrahimi, H. ve Ayati, Z. (2009a) An approximation to the solution of Telegraph equation by variational iteration method, *Numer Meth Part Diff Equ*, 25 (4): 797-801.
- Biazar, J., Badpeima, F. ve Azimi, F. (2009b) Application of the homotopy perturbation method to Zakharov-Kuznetsov equations, *Comput Math Appl*, 58 (11-12): 2391-2394.
- Bildik, N. ve Konuralp, A. (2006) Two-dimensional differential transform method, Adomian's decomposition method, and variational iteration method for partial differential equations, *Int J Comput Math*, 83 (12): 973-987.
- Bouziani, A. (1999) On a class of parabolic equations with a nonlocal boundary condition, *Acad Roy Belg Bull Cl Sci*, 10: 61-77.
- Boyd S. ve Vandenberghe, L. (2004) *Convex optimization*, Cambridge University Press, New York, 716s.
- Bratsos, A. G. (2009) On the numerical solution of the Klein-Gordon equation, *Numer Meth Part Diff Equ*, 25 (4): 939-951.
- Brezis H. ve Browder F. (1998) Partial Differential Equations in the 20th Century, *Adv Math*, 135: 76-144.
- Bülbül, B., 2011. *İkinci mertebeden kısmi diferansiyel denklemlerin Taylor polinom çözümleri*, Doktora Tezi, Muğla Üniversitesi, Muğla, 88s.
- Bülbül, B. ve Sezer, M. (2011) Taylor polynomial solution of hyperbolic type partial differential equations with constant coefficients, *Int J Comput Math*, 88 (3): 533-544.
- Chung, T.J. (2010) *Computational Fluid Dynamics*, Cambridge University Press, New York, 1034s.
- Clavero, C., Jorge, J.C. ve Lisbona, F. (2003) A uniformly convergent scheme on a nonuniform mesh for convection-diffusion parabolic problems, *J Comput Appl Math*, 154: 415-429.
- Çağlıyan, Ç. ve Çelebi, O. (2010) *Kısmi diferansiyel denklemler*, Dora basım yayın dağıtım, Bursa, 264s.
- Çelik, İ. (2006) Collocation method and residual correction using Chebyshev series, *Appl Math Comput* 174: 910-920.

- Çilingir, İ. (2005) *Eliptik Türden Kısmi türevli diferansiyel denklemlerin sonlu farklar metodu ile çözümleri ve yakınsaklık analizi*, Yüksek lisans Tezi, Ondokuz Mayıs Üniversitesi, Samsun, 62s.
- Dağ, İ., Korkmaz, A. ve Saka, B. (2010) Cosine expansion-based differential quadrature algorithm for numerical solution of the RLW equation, *Numer Meth Part Diff Equ*, 26 (3): 544-560.
- Darvishi, M.T. ve Khani, F. (2009) A series solution of the foam drainage equation, *Comput Math Appl*, 58 (2): 360-368.
- Debnath, L. (2005) *Nonlinear partial differential equations for scientists and engineers*, Texas Pan-American University, Edinburg, 737s.
- Dehghan, M. (2005) On the solution of an initial-boundary value problem that combines Neumann and integral condition for the wave equation, *Numer Meth Part Diff Equ*, 21 (1): 24-40.
- Dehghan, M. (2007) The one-dimensional heat equation subject to a boundary integral specification, *Chaos Solitons Fract*, 32: 661-675.
- Dehghan, M. ve Shokri, A. (2009a) A meshless method for numerical solution of a linear hyperbolic equation with variable coefficients in two space dimensions, *Numer Meth Part Diff Equ*, 25 (2): 494-506.
- Dehghan, M. ve Shokri, A. (2009b) Numerical solution of the nonlinear Klein-Gordon equation using radial basis functions, *J Comput Appl Math*, 230: 400-410.
- Dereli, Y. ve Dağ, İ. (2012) Numerical solutions of the Kawahara type equations using radial basis functions, *Numer Meth Part Diff Equ*, 28 (2): 542-553.
- Dereli, Y. (2012) Solitary wave solutions of the MRLW equation using radial basis functions, *Numer Meth Part Diff Equ*, 28 (1): 235-247.
- Doha, E.H. ve Al-Kholi, F.M.R. (2001) An efficient double Legendre spectral method for parabolic and elliptic partial differential equations, *Int J Comput Math*, 78 (3): 413-432.
- El-Azab, M. S. ve El-Game, I. M. (2007) A numerical algorithm for the solution of telegraph equations, *Appl Math Comput*, 190 (1): 757-764.
- Ferziger, J.H. ve Peric, M. (2002) *Computational methods for fluid dynamics*, Siproinger-Verlag, Berlin, 423s.

- Greiner, W. (2000) *Relativistic Quantum Mechanics-Wave Equations*, Springer, Berlin.
- Gülsu, M. ve Sezer, M. (2006) Taylor collocation method for solution of systems of high-order linear Fredholm Volterra integro-differential equations, *Int J Comput Math*, 83 (4): 429-448.
- Hasanoğlu, A. (2010) *Kısmi türevli denklemler*, Literatür yayıncılık, İstanbul, 429s.
- Hemeda, A.A. (2008) Variational iteration method for solving wave equation, *Comput Math Appl*, 56 (8): 1948-1953.
- Herron, I.H. ve Foster, M.R. (2008) *Partial Differential Equations in Fluid Dynamics*, Cambridge University Press, New York, 282s.
- Higham, N. J. (2002) *Accuracy and stability of numerical algorithms*, SIAM, United States of America, 680s.
- Işık, O.R. (2012) *Lineer fonksiyonel denklemlerin Bernstein polinomları ve rasyonel fonksiyonlara dayalı sıralama yöntemi ile çözümleri üzerine*, Doktora Tezi, Muğla Üniversitesi, Muğla, 107s.
- İnc, M. (2010) He's Homotopy perturbation method for solving Korteweg-de Vries Burgers equation with initial condition, *Numer Meth Part Diff Equ*, 26 (5): 1224-1235.
- Kadalbajoo, M. K., Gupta, V. ve Awasthi, A. (2008) A uniformly convergent B-spline collocation method on a nonuniform mesh for singularly perturbed one-dimensional time-dependent linear convection-diffusion problem, *J Comput Appl Math*, 220: 271-289.
- Kadalbajoo, M. K. ve Gupta, V. (2009) Numerical solution of singularly perturbed convection-diffusion problem using parameter uniform B-spline collocation method, *J Math Anal Appl*, 355: 439-452.
- Karamete, A. ve Sezer, M. (2002) A Taylor collocation method for the solution of linear integro-differential equations, *Int J Comput Math*, 79 (9): 987-1000.
- Kelleci, A. ve Yıldırım, A. (2011) An efficient numerical method for solving coupled Burgers' equation by combining homotopy perturbation and pade techniques, *Numer Meth Part Diff Equ*, 27 (4): 982-995.
- Keşan C. (2003) Chebyshev polynomial solutions of second-order linear partial differential equations, *Appl Math Comput*, 134 (1): 109-124.

- Khalifa, A.K., Elbarbary, E.M.E. ve Abd Elrazek, M.A. (2003) Chebyshev expansion method for solving second and fourth-order elliptic equations, *Appl Math Comput*, 135 (2-3): 307-318.
- Khater, A.H. ve Temsah, R.S. (2008) Numerical solutions of the generalized Kuramoto–Sivashinsky equation by Chebyshev spectral collocation methods, *Comput Math Appl* 56 (6): 1465-1472.
- Kumar B.V.R. ve Mehra, M. (2006) A Three-step wavelet Galerkin method for parabolic and hyperbolic partial differential equations, *Int J Comput Math*, 83 (1): 143-157.
- Kurnaz, A., Oturanç, G. ve Kırıs, M.E. (2005) n-dimensional differential transformation method for solving PDEs, *Int J Comput Math*, 82 (3): 369-380.
- Kurt, N. ve Sezer, M. (2008) Polynomial solution of high-order linear Fredholm integro-differential equations with constant coefficients, *J Frank Inst*, 345 (8): 839-850.
- Kurulay, M. ve Bayram, M. (2009) A novel power series method for solving second order partial differential equations, *Eur J Pure Appl Math*, 2(2): 268-277.
- Lakestani, M. ve Dehghan, M. (2009) Numerical solution of Fokker-Planck equation using the cubic B-spline scaling functions, *Numer Meth Part Diff Equ*, 25 (2): 418-429.
- Lenferink, W. (2002) A second order scheme for a time-dependent, singularly perturbed convection-diffusion equation, *J Comput Appl Math*, 143: 49-68.
- Liu, X. ve Tai, K. (2006) Point interpolation collocation method for the solution of partial differential equations. *Engin Anal Bound Elem*, 30 (7): 598-609.
- Liu, N. ve Lin, E.B. (2010) Legendre wavelet method for numerical solutions of partial differential equations, *Numer Meth Part Diff Equ*, 26 (1): 81-94.
- Ma, H.C., Yu, Y.D. ve Ge, D.J. (2009) The auxiliary equation method for solving the Zakharov-Kuznetsov (ZK) equation, *Comput Math Appl*, 58 (11-12): 2523-2527.
- Moreira, D. ve Vilhena, M. (2010) *Air pollution and turbulence: modeling and applications*, Taylor Francis group, New York, 332s.
- Morton, K.W. ve Mayers, D.F. (2005) *Numerical Solution of Partial Differential Equations: An Introduction*, Cambridge University Press, New York, 278s.

- Murphy, C. P. ve Evans, D. J. (1981) Chebyshev series solution of the two dimensional heat equation, *Math Comput Simul*, 23(2): 157-162.
- Nas, Ş., Yalçınbaş ve S., Sezer, M. (2000) A Taylor polynomial approach for solving high-order linear Fredholm integro-differential equations, *Int J Math Educ Sci Technol*, 31 (2): 213-225.
- Norhashidah, HJ. M. A. ve Evans, D.J. (2004) Preconditioned rotated iterative methods in the solution of elliptic partial differential equations, *Int J Comput Math*, 81 (9): 1163-1174.
- Oliveira, F.A. (1980) Collocation and residual correction, *Numer Math*, 36: 27-31.
- Parand, K. ve Rad, J.A. (2012) Exp-function method for some nonlinear PDE's and a nonlinear ODE's, *J King Saud Univ Sci*, 24 (1): 1-10.
- Pinchover, Y. ve Rubinstein, J. (2005) *An introduction to partial differential equations*, Cambridge University Press, New York, 371s.
- Polyanin, A. D. (2002) *Handbook of linear partial differential equations for engineers and scientists*, ACRC Press Company, New York, 781s.
- Ponsoda, P., Defez, E., Rosello, M.D. ve Romero, J.V. (2008) A stable numerical method for solving variable coefficient advection–diffusion models, *Comput Math Appl*, 56 (3): 754-768.
- Pozrikidis, C. (2009) *Fluid Dynamics: Theory, Computation and Numerical Simulation*, Springer, New York, 773s.
- Ramos, J.I. (2005) A piecewise-analytical method for singularly perturbed parabolic problems, *Appl Math Comput*, 161: 501-512.
- Revelli, R. ve Ridolfi, L. (2008) Generalized collocation method for two-dimensional reaction-diffusion problems with homogeneous Neumann boundary conditions, *Comput Math Appl*, 56 (9): 2360-2370.
- Reyes, E., Rodriguez, F. ve Martin, J.A. (2008) Analytic-numerical solutions of diffusion mathematical models with delays, *Comput Math Appl*, 56(3): 743-753.
- Roussy, G. ve Percy, J. A. (1995) *Foundations and Industrial Applications of Microwaves and Radio Frequency Fields*, John Wiley, New York.

- Saadatmandi, A. ve Dehghan, M. (2010) Numerical solution of hyperbolic telegraph equation using the Chebyshev Tau method, *Numer Meth Part Diff Equ*, 26 (1): 239-252.
- Safari, M., Ganji, D.D. ve Moslemi, M. (2009) Application of He's variational iteration method and Adomian's decomposition method to the fractional KdV Burgers Kuramoto equation, *Comput Math Appl*, 58 (11-12): 2091-2097.
- Sezer, M. (1989) A Chebyshev polynomial approximation for Dirichlet problem, *J Fac Sci Ege Univ Ser A*, 12 (2): 69-77.
- Sezer, M. ve Gulsu, M. (2005) Polynomial solution of the most general linear Fredholm integrodifferential-difference equation by means of Taylor matrix method, *Int J Complex Var*, 50 (5): 367-382.
- Shahmorad, S. (2005) Numerical solution of general form linear Fredholm Volterra integro differential equations by the tau method with an error estimation, *Appl Math Comput* 167: 1418-1429.
- Shakeri, F. ve Dehghan, M. (2008) The method of lines for solution of the one-dimensional wave equation subject to an integral conservation condition, *Comput Math Appl* 56: 2175-2188.
- Stavroulakis, I.P. ve Tersian, S.A. (2004) *Partial Differential Equations: An Introduction With Mathematica and Maple*, Word Scientific Publishing Co. Pte.Ltd., London, 306s.
- Şahin, N., Yüzbaşı, Ş. ve Sezer, M. (2011a) A Bessel Polynomial approach for solving general linear Fredholm integro-differential-difference equations, *Int J Comput Math*, 88 (14): 3093-3111.
- Şahin, N., Yüzbaşı, Ş. ve Gülsu, M. (2011b) A collocation approach for solving systems of linear Volterra integral equations with variable coefficients, *Comput Math Appl*, 62 (2): 755-769.
- Trahan, J., Kaw, A. ve Martin, K. (2006) *Computational Time for Finding the Inverse of a Matrix: LU Decomposition vs. Naive Gaussian Elimination*, Ders notları, URL:http://nm.mathforcollege.com/simulations/mcd/04sle/mcd_sle_sim_inv ersecomptime.pdf.
- Watkins, D. S. (2002) *Fundamentals of matrix computations*. John Wiley & Sons, New York, 618s.
- Weinberger, H.F. (1965) *A first course in partial differential equations*, Blaisdell publishing company, New York, 456s.

- Wen, Y.X. ve Zho, X.W. (2009) Exact solutions for the generalized nonlinear heat conduction equations using the exp-function method, *Comput Math Appl*, 58 (11-12): 2464-2467.
- Yao, H. (2011) Reproducing kernel method for the solution of nonlinear hyperbolic telegraph equation with an Integral condition, *Numer Meth Part Diff Equ*, 27 (4): 867-886.
- Yıldırım, A. (2009) Application of He's homotopy perturbation method for solving the Cauchy reaction-diffusion problem, *Comput Math Appl*, 57 (4): 612-618.
- Yildirim, A. ve Berberler, M.E. (2010) Homotopy perturbation method for numerical solutions of KdV-Burgers' and Lax's seventh-order KdV equations, *Numer Meth Part Diff Equ*, 26 (5): 1040-1053.
- Yousefi, S. A. (2010) Legendre multiwavelet Galerkin method for solving the hyperbolic telegraph equation, *Numer Meth Part Diff Equ*, 26 (3): 535-543.
- Yousefi, S. A., Barikbin, Z. ve Dehghan, M. (2010) Bernstein Ritz-Galerkin method for solving an initial-boundary value problem that combines Neumann and integral condition for the wave equation, *Numer Meth Part Diff Equ*, 26 (5): 1236-1246.
- Yüksel, G., 2011. *İkinci merteye lineer kısmi diferansiyel denklemlerin Chebyshev polinom çözümleri*, Doktora Tezi, Muğla Üniversitesi, Muğla, 95s.
- Yüzbaşı, Ş. (2009) *Lineer diferansiyel, integral ve integro-diferansiyel denklemlerin Besssel polinom çözümleri*, Yüksek Lisans Tezi, Muğla Üniversitesi, 180s.
- Yüzbaşı, Ş., Şahin, N. ve Sezer, M. (2012) A Bessel collocation method for numerical solution of generalized pantograph equations, *Numer Meth Part Diff Equ*, 28 (4): 1105-1123.
- Zhu, S.D., Chu, Y.M. ve Qiu, S.L. (2009) The homotopy perturbation method for discontinued problems arising in nanotechnology, *Comput Math Appl*, 58 (11-12): 2398-2401.

Ek A. Dalga Problemi İçin Matlab Kodu

```
function DalgaOrnek1(N,M)
format rat
D=besselD(N)
syms x
for i=1:(N+1)
    X(1,i)=x^(i-1);
end
X
Jx=X*D'
syms y
Jy=subs(Jx,y)
B=zeros(N+1,N+1);
for i=1:N
    B(i+1,i)=i;
end
P=(D')^(-1)*B'*D'
PP=kron(eye(N+1),P)
Qy=kron(eye(N+1),Jy)
W=Jx*Qy*(PP)^2-Jx*P^2*Qy
s=1;
WM=zeros((N+1)^2,(N+1)^2);
for i=1:N+1
    for j=1:N+1
        WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
        WM(s,k)=WW(1,k);
    end
    s=s+1;
end
WM
g=-2*(x-y)*exp(-x-y)
s=1
G=zeros((N+1)^2,1);
for i=1:N+1
    for j=1:N+1
        G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
        GM(s,1)=G(1,1);
    end
    s=s+1;
```

```

    end
    end
GM
    a=0
    b=1
    c=0
    d=1
    ub=Jx*subs(Qy,c)
    s=1;
    Ub=zeros(1,(N+1)^2);
    for i=1:N+1
        Ub=[subs(ub,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        UB(s,k)=Ub(1,k);
    end
    s=s+1;
    end
    UB
    us=Jx*subs(Qy,d)
    s=1;
    Us=zeros(1,(N+1)^2);
    for i=1:N+1
        Us=[subs(us,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        US(s,k)=Us(1,k);
    end
    s=s+1;
    end
    US
    ut=Jx*subs(Qy,c)*PP
    s=1;
    U=zeros(1,(N+1)^2);
    for i=1:N+1
        U=[subs(ut,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        Ut(s,k)=U(1,k);
    end
    s=s+1;
    end
    Ut
    vb=subs(Jx,a)*Qy
    s=1;
    Vb=zeros(1,(N+1)^2);
    for i=1:N+1
        Vb=[subs(vb,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        VB(s,k)=Vb(1,k);
    end
    s=s+1;
    end

```

```

VB
for i=1:(N+1)
L(1,i)=b^(i)/i;
end
L
vn=L*D'*Qy
s=1;
Vn=zeros(1,(N+1)^2);
for i=1:N+1
Vn=[subs(vn,(i-1)/N)];
for k=1:((N+1)*(N+1))
VN(s,k)=Vn(1,k);
end
s=s+1;
end
VN
syms x
f=0
m=x*exp(-x)
h=0
k=-2*y*exp(-y-1)+y*exp(-y)
s=1;
F=zeros((N+1),1);
for i=1:N+1
F=[subs(f,(i-1)/N)];
FF(s,1)=F(1,1);
s=s+1;
end
s=1;
Mm=zeros((N+1),1);
for i=1:N+1
Mm=[subs(m,(i-1)/N)];
MM(s,1)=Mm(1,1);
s=s+1;
end
s=1;
H=zeros((N+1),1);
for i=1:N+1
H=[subs(h,(i-1)/N)];
HH(s,1)=H(1,1);
s=s+1;
end
s=1;
K=zeros((N+1),1);
for i=1:N+1
K=[subs(k,(i-1)/N)];
KK(s,1)=K(1,1);
s=s+1;
end
WW=[UB;Ut;VB;VN;WM]

```

```

GG=[FF;MM;HH;KK;GM]
Wg=[WW GG]
an=WW\GG
syms t
U=Jx*subs(Qy,y)*an
syms x y
Rn=diff(U,y,2)-diff(U,x,2)-g
format rat
D1=besselD(M)
syms x
for i=1:(M+1)
    X1(1,i)=x^(i-1);
end
X1
Jx1=X1*D1'
syms y
Jy1=subs(Jx1,y)
B1=zeros(M+1,M+1);
for i=1:M
    B1(i+1,i)=i;
end
B1;
P1=(D1')^(-1)*B1'*D1'
PP1=kron(eye(M+1),P1)
Qy1=kron(eye(M+1),Jy1)
alfa=1
beta=1
W1=Jx1*Qy1*(PP1)^2-Jx1*P1^2*Qy1
s=1;
WM1=zeros((M+1)^2,(M+1)^2);
for i=1:M+1
    for j=1:M+1
        WW1=[subs(W1,{x,y},{(i-1)/M,(j-1)/M})];
        WM1(s,k)=WW1(1,k);
    end
    s=s+1;
end
WM1
g1=-Rn
s=1
G1=zeros((M+1)^2,1);
for i=1:M+1
    for j=1:M+1
        G1=[subs(g1,{x,y},{(i-1)/M,(j-1)/M})];
        GM1(s,1)=G1(1,1);
    end
    s=s+1;
end
end

```

```

GM1
    a=0
    b=1
    c=0
    d=1
    ub1=Jx1*subs(Qy1,c)
    s=1;
    Ub1=zeros(1,(M+1)^2);
    for i=1:M+1
    Ub1=[subs(ub1,(i-1)/M)];
    for k=1:((M+1)*(M+1))
        UB1(s,k)=Ub1(1,k);
    end
    s=s+1;
    end
    us1=Jx1*subs(Qy1,d)
    s=1;
    Us1=zeros(1,(M+1)^2);
    for i=1:N+1
    Us1=[subs(us1,(i-1)/M)];
    for k=1:((M+1)*(M+1))
        US1(s,k)=Us1(1,k);
    end
    s=s+1;
    end
    ut1=Jx1*subs(Qy1,c)*PP1
    s=1;
    U1=zeros(1,(M+1)^2);
    for i=1:M+1
    U1=[subs(ut1,(i-1)/M)];
    for k=1:((M+1)*(M+1))
        Ut1(s,k)=U1(1,k);
    end
    s=s+1;
    end
    vb1=subs(Jx1,a)*Qy1
    s=1;
    Vb1=zeros(1,(M+1)^2);
    for i=1:M+1
    Vb1=[subs(vb1,(i-1)/M)];
    for k=1:((M+1)*(M+1))
        VB1(s,k)=Vb1(1,k);
    end
    s=s+1;
    end
    VB1
    for i=1:(M+1)
    LL(1,i)=b^(i)/i;
    end
LL

```



```

vn1=LL*D1'*Qy1
s=1;
Vn1=zeros(1,(M+1)^2);
for i=1:M+1
Vn1=[subs(vn1,(i-1)/M)];
for k=1:((M+1)*(M+1))
VN1(s,k)=Vn1(1,k);
end
s=s+1;
end
VN1
syms x
f=0
m=0
h=0
k=0
s=1;
F1=zeros((M+1),1);
for i=1:M+1
F1=[subs(f,(i-1)/M)];
FF1(s,1)=F1(1,1);
s=s+1;
end
s=1;
Mm1=zeros((M+1),1);
for i=1:M+1
Mm1=[subs(m,(i-1)/M)];
MM1(s,1)=Mm1(1,1);
s=s+1;
end
s=1;
H1=zeros((M+1),1);
for i=1:M+1
H1=[subs(h,(i-1)/M)];
HH1(s,1)=H1(1,1);
s=s+1;
end
s=1;
K1=zeros((M+1),1);
for i=1:M+1
K1=[subs(k,(i-1)/M)];
KK1(s,1)=K1(1,1);
s=s+1;
end
WW1=[UB1;Ut1;VB1;VN1;WM1]
GG1=[FF1;MM1;HH1;KK1;GM1]
an1=WW1\GG1
eNM=Jx1*subs(Qy1,y)*an1
syms x y
h1=diff(eNM,x,2)-diff(eNM,y,2)

```

```

UNM=vpa(U+eNM)
Ug=x*y*exp(-(x+y))
eN=U-Ug
ENM=U+eNM-Ug
for i=0:0.2:1
    for j=0:0.2:1
        cozumUN=vpa([subs(U,{x,y},{i,j})],16)
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        tamcozum=vpa([subs(Ug,{x,y},{i,j})],10)
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        cozumUNM=vpa([subs(UNM,{x,y},{i,j})],10)
    end
end
format short e
for i=0:0.2:1
    for j=0:0.2:1
        errorfonksbessel=abs([subs(h,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        errorfonksbessel=abs([subs(h1,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesseleN=abs([subs(U-Ug,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesseleNM=abs([subs(eNM,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesselENM=abs([subs(U+eNM-Ug,{x,y},{i,j})])
    end
end
eN
eNM=subs(eNM,y,t)
ENM

```

**Ek B. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci
Mertebe Lineer Kısmi Diferansiyel Denklemler İçin Matlab Kodu**

```

function NeumannDirichletRobinOrnek1(N,Kk)
format rat
D=besselD(N)
syms x
for i=1:(N+1)
    X(1,i)=x^(i-1);
end
X
Jx=X*D'
syms y
Jy=subs(Jx,y)
B=zeros(N+1,N+1);
for i=1:N
    B(i+1,i)=i;
end
B;
P=(D')^(-1)*B'*D'
PP=kron(eye(N+1),P)
Qy=kron(eye(N+1),Jy)
alfa=1
beta=1
W=Jx*P^2*Qy+2*x*y*Jx*P*Qy*PP-x*Jx*Qy*(PP)^2+y*Jx*Qy
s=1;
WM=zeros((N+1)^2,(N+1)^2);
for i=1:N+1
    for j=1:N+1
WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
for k=1:((N+1)*(N+1))
    WM(s,k)=WW(1,k);
end
s=s+1;
end
end
WM
g=4*exp(2*x+y)+4*x*y*exp(2*x+y)-x*exp(2*x+y)+y*exp(2*x+y)
s=1
G=zeros((N+1)^2,1);
for i=1:N+1
    for j=1:N+1
G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
GM(s,1)=G(1,1);
s=s+1;
end
end
GM
a=0

```

```

    b=1
    c=0
    d=1
    ub=Jx*subs(Qy,c)
    s=1;
    Ub=zeros(1,(N+1)^2);
    for i=1:N+1
    Ub=[subs(ub,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        UB(s,k)=Ub(1,k);
    end
    s=s+1;
    end
    UB
    us=2*Jx*subs(Qy,d)+Jx*subs(Qy,c)*PP
    s=1;
    Us=zeros(1,(N+1)^2);
    for i=1:N+1
    Us=[subs(us,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        US(s,k)=Us(1,k);
    end
    s=s+1;
    end
    US
    ut=Jx*subs(Qy,c)*PP
    s=1;
    U=zeros(1,(N+1)^2);
    for i=1:N+1
    U=[subs(ut,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        Ut(s,k)=U(1,k);
    end
    s=s+1;
    end
    Ut
    vb=subs(Jx,a)*Qy
    s=1;
    Vb=zeros(1,(N+1)^2);
    for i=1:N+1
    Vb=[subs(vb,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        VB(s,k)=Vb(1,k);
    end
    s=s+1;
    end
    VB
    vs=subs(Jx,b)*Qy
    s=1;
    Vs=zeros(1,(N+1)^2);

```

```

    for i=1:N+1
Vs=[subs(vs,(i-1)/N)];
for k=1:((N+1)*(N+1))
    VS(s,k)=Vs(1,k);
end
    s=s+1;
end
    VS
    vx=subs(Jx,a)*P*Qy
    s=1;
    V=zeros(1,(N+1)^2);
    for i=1:N+1
V=[subs(vx,(i-1)/N)];
for k=1:((N+1)*(N+1))
    Vx(s,k)=V(1,k);
end
    s=s+1;
end
    Vx
syms x
f=exp(2*x)
m=2*exp(2*x+1)+exp(2*x)
h=exp(y)
k=2*exp(y)
    s=1;
    F=zeros((N+1),1);
    for i=1:N+1
F=[subs(f,(i-1)/N)];
    FF(s,1)=F(1,1);
s=s+1;
end
    FF
    s=1;
    M=zeros((N+1),1);
    for i=1:N+1
M=[subs(m,(i-1)/N)];
    MM(s,1)=M(1,1);
s=s+1;
end
    MM
    s=1;
    H=zeros((N+1),1);
    for i=1:N+1
H=[subs(h,(i-1)/N)];
    HH(s,1)=H(1,1);
s=s+1;
end
    HH
    s=1;
    K=zeros((N+1),1);

```

```

    for i=1:N+1
K=[subs(k,(i-1)/N)];
    KK(s,1)=K(1,1);
s=s+1;
    end
    KK
    WW=[UB;US;VB;Vx;WM]
    GG=[FF;MM;HH;KK;GM]
    syms t
        an=WW\GG
        U=Jx*subs(Qy,y)*an
    syms x t
    Uu=U
    Rn=diff(U,x,2)+2*x*y*diff(diff(U,x,1),y,1)-x*diff(U,y,2)+y*U-g
    N=Kk
    format rat
D=besselD(N)
    syms x
    for i=1:(N+1)
        X(1,i)=x^(i-1);
    end
    X
    Jx=X*D'
    syms y
    Jy=subs(Jx,y)
    B=zeros(N+1,N+1);
    for i=1:N
        B(i+1,i)=i;
    end
    B;
    P=(D')^(-1)*B'*D'
    PP=kron(eye(N+1),P)
    Qy=kron(eye(N+1),Jy)
    alfa=1
    beta=1
    W=Jx*P^2*Qy+2*x*y*Jx*P*Qy*PP-x*Jx*Qy*(PP)^2+y*Jx*Qy
    s=1;
    WM=zeros((N+1)^2,(N+1)^2);
    for i=1:N+1
        for j=1:N+1
            WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
        for k=1:((N+1)*(N+1))
            WM(s,k)=WW(1,k);
        end
s=s+1;
        end
    end
    WM
    g=-Rn
    s=1

```

```

    G=zeros((N+1)^2,1);
    for i=1:N+1
        for j=1:N+1
            G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
            GM(s,1)=G(1,1);
            s=s+1;
        end
    end
    GM
    a=0
    b=1
    c=0
    d=1
    ub=Jx*subs(Qy,c)
    s=1;
    Ub=zeros(1,(N+1)^2);
    for i=1:N+1
        Ub=[subs(ub,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        UB(s,k)=Ub(1,k);
    end
    s=s+1;
    end
    UB
    us=2*Jx*subs(Qy,d)+Jx*subs(Qy,c)*PP
    s=1;
    Us=zeros(1,(N+1)^2);
    for i=1:N+1
        Us=[subs(us,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        US(s,k)=Us(1,k);
    end
    s=s+1;
    end
    US
    ut=Jx*subs(Qy,c)*PP
    s=1;
    U=zeros(1,(N+1)^2);
    for i=1:N+1
        U=[subs(ut,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        Ut(s,k)=U(1,k);
    end
    s=s+1;
    end
    Ut
    vb=subs(Jx,a)*Qy
    s=1;
    Vb=zeros(1,(N+1)^2);
    for i=1:N+1

```

```

Vb=[subs(vb,(i-1)/N)];
for k=1:((N+1)*(N+1))
    VB(s,k)=Vb(1,k);
end
s=s+1;
end
VB
vs=subs(Jx,b)*Qy
s=1;
Vs=zeros(1,(N+1)^2);
for i=1:N+1
Vs=[subs(vs,(i-1)/N)];
for k=1:((N+1)*(N+1))
    VS(s,k)=Vs(1,k);
end
s=s+1;
end
VS
vx=subs(Jx,a)*P*Qy
s=1;
V=zeros(1,(N+1)^2);
for i=1:N+1
V=[subs(vx,(i-1)/N)];
for k=1:((N+1)*(N+1))
    Vx(s,k)=V(1,k);
end
s=s+1;
end
Vx
syms x
f=0
m=0
h=0
k=0
s=1;
F=zeros((N+1),1);
for i=1:N+1
F=[subs(f,(i-1)/N)];
FF(s,1)=F(1,1);
s=s+1;
end
FF
s=1;
M=zeros((N+1),1);
for i=1:N+1
M=[subs(m,(i-1)/N)];
MM(s,1)=M(1,1);
s=s+1;
end
MM

```



```

s=1;
H=zeros((N+1),1);
for i=1:N+1
H=[subs(h,(i-1)/N)];
HH(s,1)=H(1,1);
s=s+1;
end
HH
s=1;
K=zeros((N+1),1);
for i=1:N+1
K=[subs(k,(i-1)/N)];
KK(s,1)=K(1,1);
s=s+1;
end
KK
WW=[UB;US;VB;Vx;WM]
GG=[FF;MM;HH;KK;GM]
syms t
an=WW\GG
eNM=Jx*subs(Qy,y)*an
syms x y
h1=diff(Uu,x,2)+2*x*y*diff(diff(Uu,y,1),x,1)-x*diff(Uu,y,2)+y*Uu-
(4*exp(2*x+y)+4*x*y*exp(2*x+y)-x*exp(2*x+y)+y*exp(2*x+y))
UNM=(Uu+eNM)
Ug=exp(2*x+y)
eN=Uu-Ug
ENM=Uu+eNM-Ug
for i=0:0.2:1
for j=0:0.2:1
cozumUN=vpa([subs(Uu,{x,y},{i,j})],16)
end
end
for i=0:0.2:1
for j=0:0.2:1
tamcozum=vpa([subs(Ug,{x,y},{i,j})],10)
end
end
for i=0:0.2:1
for j=0:0.2:1
cozumUNM=vpa([subs(UNM,{x,y},{i,j})],10)
end
end
format short e
for i=0:0.2:1
for j=0:0.2:1
errorfonksbessel=abs([subs(h,{x,y},{i,j})])
end
end
for i=0:0.2:1

```

```

    for j=0:0.2:1
        errorfonksbessel=abs([subs(h1,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesseleN=abs([subs(Uu-Ug,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesseleNM=abs([subs(eNM,{x,y},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBesselENM=abs([subs(Uu+eNM-Ug,{x,y},{i,j})])
    end
end
eN
eNM=subs(eNM,y,t)
ENM

```

**Ek C. Neuman, Dirichlet ve Robin Koşulları Altında Genel Formda İkinci
Mertebe Lineer Kısmi Diferansiyel Denklemler İçin Maple Kodu**

```

> # Neuman, Dirichlet ve Robin Koşullu örnek 2
> restart;
> Digits:=30;
> n:=6;
> N=n;
> with(linalg):
> a:=0;
> b:=2;
> c:=0;
> d:=3;
> for i from 0 to n do x[i]:=evalf((b-a)*i/n); od;
> for j from 0 to n do t[j]:=evalf((d-c)*j/n); od;
> Xx:=array(1..1,1..n+1):
> for i to 1 do for j to n+1 do Xx[i,j]:=x^(j-1) od; od;
> M:=array(1..n+1,1..n+1):
> for i to n+1 do for j to n+1 do M[i,j]:=0 od; od;
> for i to n+1 do M[i,i]:=1/(factorial(0)*factorial(i-1)*2^(i-1)); od;
> for i to n-1 do M[i,i+2]:=1/((-1)*factorial(1)*factorial(i)*2^(i+1)); od;
> for i to n-3 do M[i,i+4]:=1/(factorial(2)*factorial(i+1)*2^(i+3)); od;
> for i to n-5 do M[i,i+6]:=1/((-1)*factorial(3)*factorial(i+2)*2^(i+5)); od;
> for i to n-7 do M[i,i+8]:=1/(factorial(4)*factorial(i+3)*2^(i+7)); od;
> for i to n-9 do M[i,i+10]:=1/((-1)*factorial(5)*factorial(i+4)*2^(i+9)); od;
> for i to n-11 do M[i,i+12]:=1/(factorial(6)*factorial(i+5)*2^(i+11)); od;
> for i to n-13 do M[i,i+14]:=1/((-1)*factorial(7)*factorial(i+6)*2^(i+13)); od;
> for i to n-15 do M[i,i+16]:=1/(factorial(8)*factorial(i+7)*2^(i+15)); od;
> for i to n-17 do M[i,i+18]:=1/((-1)*(factorial(9)*factorial(i+8)*2^(i+17)); od;
> for i to n-19 do M[i,i+20]:=1/(factorial(10)*factorial(i+9)*2^(i+19)); od;
> for i to n-21 do M[i,i+22]:=1/((-1)*(factorial(11)*factorial(i+10)*2^(i+21)); od;
> for i to n-23 do M[i,i+24]:=1/(factorial(12)*factorial(i+11)*2^(i+23)); od;
> for i to n-25 do M[i,i+26]:=1/((-1)*(factorial(13)*factorial(i+12)*2^(i+25)); od;
> for i to n-27 do M[i,i+28]:=1/(factorial(14)*factorial(i+13)*2^(i+27)); od;
> for i to n-29 do M[i,i+30]:=1/((-1)*(factorial(15)*factorial(i+14)*2^(i+29)); od;
> DD:=transpose(M);
> Jx:=evalm(Xx&*DD);
> Xt:=array(1..1,1..n+1):
> for i to 1 do for j to n+1 do Xt[i,j]:=t^(j-1) od; od;
> Qt:=evalm(Xt&*DD);
> Q:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to n+1 do for j to (n+1)*(n+1) do Q[i,j]:=0 od; od;
> for i to 1 do for j to n+1 do Q[i,j]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+1,j+(n+1)]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+2,j+2*(n+1)]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+3,j+3*(n+1)]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+4,j+4*(n+1)]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+5,j+5*(n+1)]:=Qt[i,j]; od; od;
> for i to 1 do for j to n+1 do Q[i+6,j+6*(n+1)]:=Qt[i,j]; od; od;

```

```

> BB:=array(1..n+1,1..n+1):
> for i to n+1 do for j to n+1 do BB[i,j]:=0 od; od;
> for i to n do BB[i+1,i]:=i od;
> B:=array(1..n+1,1..n+1):
> B:=transpose(BB):
> P:=evalm(DD^(-1)&*B&*DD):
> PP:=array(1..(n+1)*(n+1),1..(n+1)*(n+1)):
> for i to (n+1)*(n+1) do for j to (n+1)*(n+1) do PP[i,j]:=0 od;od;
> for i to n+1 do for j to n+1 do PP[i,j]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+(n+1),j+(n+1)]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+2*(n+1),j+2*(n+1)]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+3*(n+1),j+3*(n+1)]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+4*(n+1),j+4*(n+1)]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+5*(n+1),j+5*(n+1)]:=P[i,j];od;od;
> for i to n+1 do for j to n+1 do PP[i+6*(n+1),j+6*(n+1)]:=P[i,j];od;od;
> W:=evalm(Jx&*P^2&*Q+2*Jx&*P&*Q&*PP+2*x*exp(t)*Jx&*Q&*(PP)^2-
x*Jx&*P&*Q-t*sin(x)*Jx&*Q):
> W1:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W1[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W1[i,j]:=eval(W[1,j],[x=x[0],t=t[i-1]])
od;od;
> W2:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W2[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W2[i,j]:=eval(W[1,j],[x=x[1],t=t[i-1]])
od;od;
> W3:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W3[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W3[i,j]:=eval(W[1,j],[x=x[2],t=t[i-1]])
od;od;
> W4:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W4[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W4[i,j]:=eval(W[1,j],[x=x[3],t=t[i-1]])
od;od;
> W5:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W5[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W5[i,j]:=eval(W[1,j],[x=x[4],t=t[i-1]])
od;od;
> W6:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W6[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W6[i,j]:=eval(W[1,j],[x=x[5],t=t[i-1]])
od;od;
> W7:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W7[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W7[i,j]:=eval(W[1,j],[x=x[6],t=t[i-1]])
od;od;
> W8:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W8[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W8[i,j]:=eval(W[1,j],[x=x[7],t=t[i-1]])
od;od;
> W9:=array(1..(n+1),1..(n+1)*(n+1)):

```

```

> for i to (n+1) do for j to (n+1)*(n+1) do W9[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W9[i,j]:=eval(W[1,j],[x=x[8],t=t[i-1]])
od;od:
> W10:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W10[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W10[i,j]:=eval(W[1,j],[x=x[9],t=t[i-1]])
od;od:
> W11:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W11[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W11[i,j]:=eval(W[1,j],[x=x[10],t=t[i-1]])
od;od:
> W12:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W12[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W12[i,j]:=eval(W[1,j],[x=x[11],t=t[i-1]])
od;od:
> W13:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W13[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W13[i,j]:=eval(W[1,j],[x=x[12],t=t[i-1]])
od;od:
> W14:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W14[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W14[i,j]:=eval(W[1,j],[x=x[13],t=t[i-1]])
od;od:
> W15:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W15[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W15[i,j]:=eval(W[1,j],[x=x[14],t=t[i-1]])
od;od:
> W16:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W16[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W16[i,j]:=eval(W[1,j],[x=x[15],t=t[i-1]])
od;od:
> W17:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W17[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W17[i,j]:=eval(W[1,j],[x=x[16],t=t[i-1]])
od;od:
> W18:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W18[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W18[i,j]:=eval(W[1,j],[x=x[17],t=t[i-1]])
od;od:
> W19:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W19[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W19[i,j]:=eval(W[1,j],[x=x[18],t=t[i-1]])
od;od:
> W20:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W20[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W20[i,j]:=eval(W[1,j],[x=x[19],t=t[i-1]])
od;od:
> W21:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W21[i,j]:=0 od;od;

```

```

> for i to (n+1) do for j to (n+1)*(n+1) do W21[i,j]:=eval(W[1,j],[x=x[20],t=t[i-1]])
od;od:
> W22:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W22[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W22[i,j]:=eval(W[1,j],[x=x[21],t=t[i-1]])
od;od:
> W23:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W23[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W23[i,j]:=eval(W[1,j],[x=x[22],t=t[i-1]])
od;od:
> W24:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W24[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W24[i,j]:=eval(W[1,j],[x=x[23],t=t[i-1]])
od;od:
> W25:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W25[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W25[i,j]:=eval(W[1,j],[x=x[24],t=t[i-1]])
od;od:
> W26:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do W26[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do W26[i,j]:=eval(W[1,j],[x=x[25],t=t[i-1]])
od;od:
> WW:=array(1..(n+1)*(n+1),1..(n+1)*(n+1)):
> for i to (n+1)*(n+1) do for j to (n+1)*(n+1) do WW[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i,j]:=W1[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+(n+1),j]:=W2[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+2*(n+1),j]:=W3[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+3*(n+1),j]:=W4[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+4*(n+1),j]:=W5[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+5*(n+1),j]:=W6[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WW[i+6*(n+1),j]:=W7[i,j] od;od;
> k1:=eval(evalm(Jx&*Q),t=0):
> K1:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do K1[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do K1[i,j]:=eval(k1[1,j],x=x[i-1]) od;od;
> k21:=eval(evalm(Jx&*Q),t=3):
> k22:=eval(evalm(Jx&*P&*Q),t=0):
> K2:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do K2[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do K2[i,j]:=eval(k21[1,j]-k22[1,j],x=x[i-1])
od;od:
> k3:=eval(evalm(Jx&*Q),x=2):
> K3:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do K3[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do K3[i,j]:=eval(k3[1,j],t=t[i-1]) od;od;
> k4:=eval(evalm((-1)*Jx&*Q&*PP),x=0):
> K4:=array(1..(n+1),1..(n+1)*(n+1)):
> for i to (n+1) do for j to (n+1)*(n+1) do K4[i,j]:=0 od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do K4[i,j]:=eval(k4[1,j],t=t[i-1]) od;od;
> f1:=x->cos(x):

```

```

> F1:=linalg[matrix](n+1,1,[seq(evalf(f1(x[i-1])),i=1..n+1))]:
> f2:=x->exp(3)*cos(x)+sin(x):
> F2:=linalg[matrix](n+1,1,[seq(evalf(f2(x[i-1])),i=1..n+1))]:
> h1:=t->cos(2)*exp(t):
> H1:=linalg[matrix](n+1,1,[seq(evalf(h1(t[i-1])),i=1..n+1))]:
> h2:=t->-exp(t):
> H2:=linalg[matrix](n+1,1,[seq(evalf(h2(t[i-1])),i=1..n+1))]:
> WWW:=array(1..(n+1)^2+4*(n+1),1..(n+1)*(n+1)):
> for i to (n+1)^2+4*(n+1) do for j to (n+1)*(n+1) do WWW[i,j]:=0 od;od;
> for i to (n+1)^2 do for j to (n+1)*(n+1) do WWW[i,j]:=WW[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WWW[i+(n+1)^2,j]:=K1[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WWW[i+(n+1)^2+(n+1),j]:=K2[i,j] od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WWW[i+(n+1)^2+2*(n+1),j]:=K3[i,j]
od;od;
> for i to (n+1) do for j to (n+1)*(n+1) do WWW[i+(n+1)^2+3*(n+1),j]:=K4[i,j]
od;od;
> g:=-cos(x)*exp(t)+x*sin(x)*exp(t)+2*x*exp(t)^2*cos(x)-2*sin(x)*exp(t)-
t*sin(x)*cos(x)*exp(t):
> G1:=array(1..(n+1),1..1):
> for i to (n+1) do G1[i,1]:=0 od:
> for i to (n+1) do G1[i,1]:=eval(g,[x=x[0],t=t[i-1]]) od:
> G2:=array(1..(n+1),1..1):
> for i to (n+1) do G2[i,1]:=0 od:
> for i to (n+1) do G2[i,1]:=eval(g,[x=x[1],t=t[i-1]]) od:
> G3:=array(1..(n+1),1..1):
> for i to (n+1) do G3[i,1]:=0 od:
> for i to (n+1) do G3[i,1]:=eval(g,[x=x[2],t=t[i-1]]) od:
> G4:=array(1..(n+1),1..1):
> for i to (n+1) do G4[i,1]:=0 od:
> for i to (n+1) do G4[i,1]:=eval(g,[x=x[3],t=t[i-1]]) od:
> G5:=array(1..(n+1),1..1):
> for i to (n+1) do G5[i,1]:=0 od:
> for i to (n+1) do G5[i,1]:=eval(g,[x=x[4],t=t[i-1]]) od:
> G6:=array(1..(n+1),1..1):
> for i to (n+1) do G6[i,1]:=0 od:
> for i to (n+1) do G6[i,1]:=eval(g,[x=x[5],t=t[i-1]]) od:
> G7:=array(1..(n+1),1..1):
> for i to (n+1) do G7[i,1]:=0 od:
> for i to (n+1) do G7[i,1]:=eval(g,[x=x[6],t=t[i-1]]) od:
> G8:=array(1..(n+1),1..1):
> for i to (n+1) do G8[i,1]:=0 od:
> for i to (n+1) do G8[i,1]:=eval(g,[x=x[7],t=t[i-1]]) od:
> G9:=array(1..(n+1),1..1):
> for i to (n+1) do G9[i,1]:=0 od:
> for i to (n+1) do G9[i,1]:=eval(g,[x=x[8],t=t[i-1]]) od:
> G10:=array(1..(n+1),1..1):
> for i to (n+1) do G10[i,1]:=0 od:
> for i to (n+1) do G10[i,1]:=eval(g,[x=x[9],t=t[i-1]]) od:
> G11:=array(1..(n+1),1..1):
> for i to (n+1) do G11[i,1]:=0 od:

```

```

> for i to (n+1) do G11[i,1]:=eval(g,[x=x[10],t=t[i-1]]) od:
> G12:=array(1..(n+1),1..1):
> for i to (n+1) do G12[i,1]:=0 od:
> for i to (n+1) do G12[i,1]:=eval(g,[x=x[11],t=t[i-1]]) od:
> G13:=array(1..(n+1),1..1):
> for i to (n+1) do G13[i,1]:=0 od:
> for i to (n+1) do G13[i,1]:=eval(g,[x=x[12],t=t[i-1]]) od:
> G14:=array(1..(n+1),1..1):
> for i to (n+1) do G14[i,1]:=0 od:
> for i to (n+1) do G14[i,1]:=eval(g,[x=x[13],t=t[i-1]]) od:
> G15:=array(1..(n+1),1..1):
> for i to (n+1) do G15[i,1]:=0 od:
> for i to (n+1) do G15[i,1]:=eval(g,[x=x[14],t=t[i-1]]) od:
> G16:=array(1..(n+1),1..1):
> for i to (n+1) do G16[i,1]:=0 od:
> for i to (n+1) do G16[i,1]:=eval(g,[x=x[15],t=t[i-1]]) od:
> G17:=array(1..(n+1),1..1):
> for i to (n+1) do G17[i,1]:=0 od:
> for i to (n+1) do G17[i,1]:=eval(g,[x=x[16],t=t[i-1]]) od:
> G18:=array(1..(n+1),1..1):
> for i to (n+1) do G18[i,1]:=0 od:
> for i to (n+1) do G18[i,1]:=eval(g,[x=x[17],t=t[i-1]]) od:
> G19:=array(1..(n+1),1..1):
> for i to (n+1) do G19[i,1]:=0 od:
> for i to (n+1) do G19[i,1]:=eval(g,[x=x[18],t=t[i-1]]) od:
> G20:=array(1..(n+1),1..1):
> for i to (n+1) do G20[i,1]:=0 od:
> for i to (n+1) do G20[i,1]:=eval(g,[x=x[19],t=t[i-1]]) od:
> G21:=array(1..(n+1),1..1):
> for i to (n+1) do G21[i,1]:=0 od:
> for i to (n+1) do G21[i,1]:=eval(g,[x=x[20],t=t[i-1]]) od:
> G22:=array(1..(n+1),1..1):
> for i to (n+1) do G22[i,1]:=0 od:
> for i to (n+1) do G22[i,1]:=eval(g,[x=x[21],t=t[i-1]]) od:
> G23:=array(1..(n+1),1..1):
> for i to (n+1) do G23[i,1]:=0 od:
> for i to (n+1) do G23[i,1]:=eval(g,[x=x[22],t=t[i-1]]) od:
> G24:=array(1..(n+1),1..1):
> for i to (n+1) do G24[i,1]:=0 od:
> for i to (n+1) do G24[i,1]:=eval(g,[x=x[23],t=t[i-1]]) od:
> G25:=array(1..(n+1),1..1):
> for i to (n+1) do G25[i,1]:=0 od:
> for i to (n+1) do G25[i,1]:=eval(g,[x=x[24],t=t[i-1]]) od:
> G26:=array(1..(n+1),1..1):
> for i to (n+1) do G26[i,1]:=0 od:
> for i to (n+1) do G26[i,1]:=eval(g,[x=x[25],t=t[i-1]]) od:
> G:=array(1..(n+1)*(n+1),1..1):
> for i to (n+1)*(n+1) do G[i,1]:=0 od:
> for i to (n+1) do G[i,1]:=G1[i,1] od:
> for i to (n+1) do G[i+(n+1),1]:=G2[i,1] od:

```



```

> for i to (n+1) do G[i+2*(n+1),1]:=G3[i,1] od:
> for i to (n+1) do G[i+3*(n+1),1]:=G4[i,1] od:
> for i to (n+1) do G[i+4*(n+1),1]:=G5[i,1] od:
> for i to (n+1) do G[i+5*(n+1),1]:=G6[i,1] od:
> for i to (n+1) do G[i+6*(n+1),1]:=G7[i,1] od:
> GG:=array(1..(n+1)^2+4*(n+1),1..1):
> for i to (n+1)^2+4*(n+1) do GG[i,1]:=0 od:
> for i to (n+1)^2 do GG[i,1]:=G[i,1] od:
> for i to (n+1) do GG[i+(n+1)^2,1]:=F1[i,1] od:
> for i to (n+1) do GG[i+(n+1)^2+(n+1),1]:=F2[i,1] od:
> for i to (n+1) do GG[i+(n+1)^2+2*(n+1),1]:=H1[i,1] od:
> for i to (n+1) do GG[i+(n+1)^2+3*(n+1),1]:=H2[i,1] od:
> Aa:=blockmatrix(1,2,[WWW,GG]):
> with(LinearAlgebra):
> A1:=gausselim(Aa):
> WWW1:=array(1..(n+1)^2,1..(n+1)^2):
> for i to (n+1)^2 do for j to (n+1)^2 do WWW1[i,j]:=A1[i,j];od;od:
> GG1:=array(1..(n+1)^2,1..1):
> for i to (n+1)^2 do GG1[i,1]:=A1[i,(n+1)^2+1];od:
> A:=evalf(evalm(inverse(WWW1)&*GG1)):
> yak:=evalf(evalm(Jx&*Q&*A)[1,1]):
> tam:=cos(x)*exp(t):
> aaaa:=eval(tam-yak,[x=x,t=t]):
> plot3d(abs(aaaa),x=0..2,t=0..3);

```

**Ek D. Başlangıç ve Sınır Koşulları Altında Genel Formda İkinci Mertebe Lineer
Kısmi Diferansiyel Denklemler için Matlab Kodu**

```

function BaslangicSindirOrnek1(N,Kk)
format rat
D=besselD(N)
syms x
for i=1:(N+1)
    X(1,i)=x^(i-1);
end
X
Jx=X*D'
syms y
Jy=subs(Jx,y)
B=zeros(N+1,N+1);
for i=1:N
    B(i+1,i)=i;
end
B;
P=(D')^(-1)*B'*D'

PP=kron(eye(N+1),P)
Qy=kron(eye(N+1),Jy)
syms x y
aa=1
bb=-2*x*y
cc=x^2*y^2
dd=x^2*y
ee=-2
ff=x
W=aa*Jx*P^2*Qy+bb*Jx*P*Qy*(PP)+cc*Jx*Qy*(PP)^2+dd*Jx*P*Qy+ee*Jx*Qy*
(PP)+ff*Jx*Qy
s=1;
WM=zeros((N+1)^2,(N+1)^2);
for i=1:N+1
    for j=1:N+1
        WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
    for k=1:((N+1)*(N+1))
        WM(s,k)=WW(1,k);
    end
    s=s+1;
    end
    end
    WM
g=-exp(x+y)+x^2*y^2*exp(x+y)-2*x*y*exp(x+y)+x^2*y*exp(x+y)+x*exp(x+y)
s=1
G=zeros((N+1)^2,1);
for i=1:N+1
    for j=1:N+1

```

```

G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
GM(s,1)=G(1,1);
s=s+1;
end
end
GM
a=0
b=1
c=0
d=1
ub=Jx*subs(Qy,c)
s=1;
Ub=zeros(1,(N+1)^2);
for i=1:N+1
Ub=[subs(ub,(i-1)/N)];
for k=1:((N+1)*(N+1))
UB(s,k)=Ub(1,k);
end
s=s+1;
end
UB
us=Jx*subs(Qy,d)
s=1;
Us=zeros(1,(N+1)^2);
for i=1:N+1
Us=[subs(us,(i-1)/N)];
for k=1:((N+1)*(N+1))
US(s,k)=Us(1,k);
end
s=s+1;
end
US
ut=Jx*subs(Qy,c)*PP
s=1;
U=zeros(1,(N+1)^2);
for i=1:N+1
U=[subs(ut,(i-1)/N)];
for k=1:((N+1)*(N+1))
Ut(s,k)=U(1,k);
end
s=s+1;
end
Ut
vb=subs(Jx,a)*Qy
s=1;
Vb=zeros(1,(N+1)^2);
for i=1:N+1
Vb=[subs(vb,(i-1)/N)];
for k=1:((N+1)*(N+1))
VB(s,k)=Vb(1,k);

```

```

end
    s=s+1;
end
    VB
    vs=subs(Jx,b)*Qy
    s=1;
    Vs=zeros(1,(N+1)^2);
    for i=1:N+1
Vs=[subs(vs,(i-1)/N)];
for k=1:((N+1)*(N+1))
    VS(s,k)=Vs(1,k);
end
    s=s+1;
end
    VS
syms x
f=exp(x)
m=exp(x)
h=exp(y)
k=exp(1)*exp(y)
    s=1;
    F=zeros((N+1),1);
    for i=1:N+1
F=[subs(f,(i-1)/N)];
    FF(s,1)=F(1,1);
s=s+1;
end
    FF
    s=1;
    M=zeros((N+1),1);
    for i=1:N+1
M=[subs(m,(i-1)/N)];
    MM(s,1)=M(1,1);
s=s+1;
end
    MM
    s=1;
    H=zeros((N+1),1);
    for i=1:N+1
H=[subs(h,(i-1)/N)];
    HH(s,1)=H(1,1);
s=s+1;
end
    HH
    s=1;
    K=zeros((N+1),1);
    for i=1:N+1
K=[subs(k,(i-1)/N)];
    KK(s,1)=K(1,1);
s=s+1;

```

```

end
KK
WW=[UB;Ut;VB;VS;WM]
GG=[FF;MM;HH;KK;GM]
syms t
an=WW\GG
U=Jx*subs(Qy,t)*an
uU=U
syms x t
Rn=subs(diff(U,x,2)-
2*x*t*diff(diff(U,x,1),t,1)+x^2*t^2*diff(U,t,2)+x^2*t*diff(U,x,1)-
2*diff(U,t,1),t,y)+x*subs(U,t,y)-g
N=Kk
format rat
D=besselD(N)
syms x
for i=1:(N+1)
    X(1,i)=x^(i-1);
end
X
Jx=X*D'
syms y
Jy=subs(Jx,y)
B=zeros(N+1,N+1);
for i=1:N
    B(i+1,i)=i;
end
B;
P=(D')^(-1)*B'*D'
PP=kron(eye(N+1),P)
Qy=kron(eye(N+1),Jy)
syms x y
aa=1
bb=-2*x*y
cc=x^2*y^2
dd=x^2*y
ee=-2
ff=x
W=aa*Jx*P^2*Qy+bb*Jx*P*Qy*(PP)+cc*Jx*Qy*(PP)^2+dd*Jx*P*Qy+ee*Jx*Qy*
(PP)+ff*Jx*Qy
s=1;
WM=zeros((N+1)^2,(N+1)^2);
for i=1:N+1
    for j=1:N+1
        WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
        for k=1:((N+1)*(N+1))
            WM(s,k)=WW(1,k);
        end
        s=s+1;
    end
end

```

```

end
    WM
    g=-Rn
    s=1
    G=zeros((N+1)^2,1);
    for i=1:N+1
        for j=1:N+1
G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
        GM(s,1)=G(1,1);
s=s+1;
        end
    end
GM
    a=0
    b=1
    c=0
    d=1
    ub=Jx*subs(Qy,c)
    s=1;
    Ub=zeros(1,(N+1)^2);
    for i=1:N+1
Ub=[subs(ub,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        UB(s,k)=Ub(1,k);
    end
    s=s+1;
    end
    UB
    us=Jx*subs(Qy,d)
    s=1;
    Us=zeros(1,(N+1)^2);
    for i=1:N+1
Us=[subs(us,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        US(s,k)=Us(1,k);
    end
    s=s+1;
    end
    US
    ut=Jx*subs(Qy,c)*PP
    s=1;
    U=zeros(1,(N+1)^2);
    for i=1:N+1
U=[subs(ut,(i-1)/N)];
    for k=1:((N+1)*(N+1))
        Ut(s,k)=U(1,k);
    end
    s=s+1;
    end
    Ut

```

```

        vb=subs(Jx,a)*Qy
        s=1;
        Vb=zeros(1,(N+1)^2);
        for i=1:N+1
            Vb=[subs(vb,(i-1)/N)];
        for k=1:((N+1)*(N+1))
            VB(s,k)=Vb(1,k);
        end
        s=s+1;
        end
        VB
        vs=subs(Jx,b)*Qy
        s=1;
        Vs=zeros(1,(N+1)^2);
        for i=1:N+1
            Vs=[subs(vs,(i-1)/N)];
        for k=1:((N+1)*(N+1))
            VS(s,k)=Vs(1,k);
        end
        s=s+1;
        end
        VS
syms x
f=0
m=0
h=0
k=0
        s=1;
        F=zeros((N+1),1);
        for i=1:N+1
            F=[subs(f,(i-1)/N)];
            FF(s,1)=F(1,1);
        s=s+1;
        end
        FF
        s=1;
        M=zeros((N+1),1);
        for i=1:N+1
            M=[subs(m,(i-1)/N)];
            MM(s,1)=M(1,1);
        s=s+1;
        end
        MM
        s=1;
        H=zeros((N+1),1);
        for i=1:N+1
            H=[subs(h,(i-1)/N)];
            HH(s,1)=H(1,1);
        s=s+1;
        end

```

```

HH
s=1;
K=zeros((N+1),1);
for i=1:N+1
K=[subs(k,(i-1)/N)];
KK(s,1)=K(1,1);
s=s+1;
end
KK
WW=[UB;Ut;VB;VS;WM]
GG=[FF;MM;HH;KK;GM]
syms t
an=WW\GG
eNM=Jx*subs(Qy,t)*an
syms x t
ENM=uU+eNM
for i=0:0.2:1
for j=0:0.2:1
cozum=vpa([subs(uU,{x,t},{i,j})],16)
end
end
Ug=exp(x+t)
ENM=uU+eNM-Ug
for i=0:0.2:1
for j=0:0.2:1
tamcozum=vpa([subs(Ug,{x,t},{i,j})],10)
end
end
format short e
for i=0:0.2:1
for j=0:0.2:1
erroreN=abs([subs(uU-Ug,{x,t},{i,j})])
end
end
for i=0:0.2:1
for j=0:0.2:1
erroreNM=abs([subs(eNM,{x,t},{i,j})])
end
end
for i=0:0.2:1
for j=0:0.2:1
errorfonksENM=abs([subs(ENM,{x,t},{i,j})])
end
end
syms x t
eN=uU-Ug
eNM
ENM

```


**Ek E. Singüler Pertürbe Olmuş Bir Boyutlu Parabolik Konveksiyon-Difüzyon
Problemi İçin Matlab Kodu**

```
function SingulerParabolikEpsilonOrnek7(N)
format rat
D=besselD(N)
syms x
for i=1:(N+1)
    X(1,i)=x^(i-1);
end
X
Jx=X*D'
syms y
Jy=subs(Jx,y)
B=zeros(N+1,N+1);
for i=1:N
    B(i+1,i)=i;
end
B;
P=(D')^(-1)*B'*D'
PP=kron(eye(N+1),P)
Qy=kron(eye(N+1),Jy)
axy=2-x^2
bxy=x
format long
kc=vpa(2^-1,30)
W= Jx*Qy*(PP)-kc*Jx*P^2*Qy+axy*Jx*P*Qy+bxy*Jx*Qy
s=1;
WM=zeros((N+1)^2,(N+1)^2);
for i=1:N+1
    for j=1:N+1
        WW=[subs(W,{x,y},{(i-1)/N,(j-1)/N})];
    for k=1:((N+1)*(N+1))
        WM(s,k)=WW(1,k);
    end
    s=s+1;
    end
    end
    WM
    g=10*y^2*exp(-y)*x*(1-x)
    s=1
    G=zeros((N+1)^2,1);
    for i=1:N+1
        for j=1:N+1
            G=[subs(g,{x,y},{(i-1)/N,(j-1)/N})];
            GM(s,1)=G(1,1);
            s=s+1;
        end
    end
end
```

```

GM
a=0
b=1
c=0
d=1
ub=Jx*subs(Qy,c)
s=1;
Ub=zeros(1,(N+1)^2);
for i=1:N+1
Ub=[subs(ub,(i-1)/N)];
for k=1:((N+1)*(N+1))
    UB(s,k)=Ub(1,k);
end
s=s+1;
end
UB
us=Jx*subs(Qy,d)
s=1;
Us=zeros(1,(N+1)^2);
for i=1:N+1
Us=[subs(us,(i-1)/N)];
for k=1:((N+1)*(N+1))
    US(s,k)=Us(1,k);
end
s=s+1;
end
US
ut=Jx*subs(Qy,c)*PP
s=1;
U=zeros(1,(N+1)^2);
for i=1:N+1
U=[subs(ut,(i-1)/N)];
for k=1:((N+1)*(N+1))
    Ut(s,k)=U(1,k);
end
s=s+1;
end
Ut
vb=subs(Jx,a)*Qy
s=1;
Vb=zeros(1,(N+1)^2);
for i=1:N+1
Vb=[subs(vb,(i-1)/N)];
for k=1:((N+1)*(N+1))
    VB(s,k)=Vb(1,k);
end
s=s+1;
end
VB
vs=subs(Jx,b)*Qy

```

```

        s=1;
        Vs=zeros(1,(N+1)^2);
        for i=1:N+1
            Vs=[subs(vs,(i-1)/N)];
            for k=1:((N+1)*(N+1))
                VS(s,k)=Vs(1,k);
            end
            s=s+1;
        end
        VS
syms x y
f=0
m=0
h=0
k=0
        s=1;
        F=zeros((N+1),1);
        for i=1:N+1
            F=[subs(f,(i-1)/N)];
            FF(s,1)=F(1,1);
            s=s+1;
        end
        FF
        s=1;
        M=zeros((N+1),1);
        for i=1:N+1
            M=[subs(m,(i-1)/N)];
            MM(s,1)=M(1,1);
            s=s+1;
        end
        MM
        s=1;
        H=zeros((N+1),1);
        for i=1:N+1
            H=[subs(h,(i-1)/N)];
            HH(s,1)=H(1,1);
            s=s+1;
        end
        HH
        s=1;
        K=zeros((N+1),1);
        for i=1:N+1
            K=[subs(k,(i-1)/N)];
            KK(s,1)=K(1,1);
            s=s+1;
        end
        KK
        WW=[UB;VB;VS;WM]
        GG=[FF;HH;KK;GM]
syms t

```

```

an=WW\GG
U=Jx*subs(Qy,t)*an
syms x t
h=diff(U,t,1)-kc*diff(U,x,2)+subs(axy,y,t)*diff(U,x,1)+subs(bxy,y,t)*U-subs(g,y,t)
for i=0:0.2:1
    for j=0:0.2:1
        cozum=vpa([subs(UU,{x,t},{i,j})],16)
    end
end
Ug=exp(x+t)/kc
for i=0:0.2:1
    for j=0:0.2:1
        tamcozum=vpa([subs(Ug,{x,t},{i,j})],10)
    end
end
format short e
for i=0:0.2:1
    for j=0:0.2:1
        errorfonksbessel=abs([subs(h,{x,t},{i,j})])
    end
end
for i=0:0.2:1
    for j=0:0.2:1
        absoluteerrorBessel=abs([subs(U-Ug,{x,t},{i,j})])
    end
end
hata=U-Ug

```

ÖZGEÇMİŞ

Karabük' ün Yenice İlçesi Yortanpazarı Beldesinde 20.09.1984 tarihinde doğdu. 1991 yılında başladığı Yortanpazarı ilköğretim okulundan 1999 yılında mezun oldu. 1999 yılında başladığı Yortanpazarı Lisesi' ni 2002 yılında birincilik ile bitirdi. 2002 yılında Muğla Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümünü kazandı. 2005 yılında Matematik bölümündeki eğitim-öğrenimini birincilikle 3 yılda tamamladı. 2005 yılında Muğla Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı' nda tezli yüksek lisansa ingilizce şartlı olarak başladı. 2006 yılında ingilizce sınavında (Üniversiteler Arası Dil Sınavında) başarılı olarak Yrd. Doç. Dr. Niyazi ŞAHİN danışmanlığında tezli yüksek lisansın ders aşamasına başladı. 2007 yılında Muğla Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı' nda Araştırma Görevlisi olarak çalışmaya başladı. Ocak 2009' da yüksek lisanstan mezun oldu. Şubat 2009' da Yrd. Doç. Dr. Niyazi ŞAHİN danışmanlığında Doktora öğrenimine başladı. Halen Muğla Sıtkı Koçma Üniversitesinde araştırma görevlisi olarak çalışmaya devam etmektedir.

Şuayip Yüzbaşı