



**İKİ BOYUTLU HİPERBOLİK BİR PROBLEMDE
LİNEER OLMAYAN OPTİMİZASYON İLE TERS
KATSAYI BELİRLEME**

Faika Derya ŞENDUR

**Danışman: Prof. Dr. Murat SUBAŞI
Doktora Tezi
Matematik Ana Bilim Dalı
2024**

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANA BİLİM DALI

**İKİ BOYUTLU HİPERBOLİK BİR PROBLEMDE LİNEER OLMAYAN
OPTİMİZASYON İLE TERS KATSAYI BELİRLEME**

(Inverse Coefficient Identification By Nonlinear Optimization in A Two Dimensional
Hyperbolic Problem)

DOKTORA TEZİ

Faika Derya ŞENDUR

Danışman: Prof. Dr. Murat SUBAŞI

Erzurum
Ağustos, 2024

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü

TEZ KABUL VE ONAY TUTANAĞI

İKİ BOYUTLU HİPERBOLİK BİR PROBLEMDE LİNEER OLMAYAN
OPTİMİZASYON İLE TERS KATSAYI BELİRLEME

Prof. Dr. Murat SUBAŞI danışmanlığında, Faika Derya ŞENDUR tarafından hazırlanan bu çalışma, 22/08/2024 tarihinde aşağıdaki jüri tarafından Matematik Anabilim Dalı, Uygulamalı Matematik Bilim Dalı'nda doktora tezi olarak **oybirliği** ile kabul edilmiştir.

Jüri Başkanı:	Prof. Dr. Murat ÖZDEMİR <i>Atatürk Üniversitesi</i>	Aslı ıslak imzalıdır
Danışman:	Prof. Dr. Murat SUBAŞI <i>Atatürk Üniversitesi</i>	Aslı ıslak imzalıdır
Jüri Üyesi:	Prof. Dr. Ali GÜROL <i>Atatürk Üniversitesi</i>	Aslı ıslak imzalıdır
Jüri Üyesi:	Prof. Dr. Şuayip YÜZBAŞI <i>Bartın Üniversitesi</i>	Aslı ıslak imzalıdır
Jüri Üyesi:	Doç. Dr. Hülya DURUR <i>Ardahan Üniversitesi</i>	Aslı ıslak imzalıdır

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Saltuk Buğrahan CEYHUN

Enstitü Müdürü

Aslı ıslak imzalıdır

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Doktora tezi olarak Prof. Dr. Murat SUBAŞI danışmanlığında sunulan “İki Boyutlu Hiperbolik Bir Problemden Lineer Olmayan Optimizasyon ile Ters Katsayı Belirleme” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	9	30
Kuramsal Temeller	26	30
Materyal ve Metot	5	35
Araştırma Bulguları	6	20
Tartışma ve Sonuç	0	20
Tezin Geneli	12	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuk sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Faika Derya ŞENDUR	Prof. Dr. Murat SUBAŞI
28.8.2024	28.8.2024
İmza: Aslı ıslak imzalıdır	İmza: Aslı ıslak imzalıdır

* Tez ile ilgili YÖKTEZ'de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun .../.../... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

Enstitü Yönetim Kurulunun .../.../... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.



Babam Aziz ŐENDUR'a ithafen...

TEŐEKKÜR

Doktora tezi olarak sunduđum bu alıřmada, alıřma konusunun tayin edilmesinde ve alıřma sürecinin her ařamasında bilgi, tecrube ve kıymetli zamanını esirgemeyerek sevgi, sabır ve destekle bana rehberlik eden Atatürk Üniversitesi Matematik Bölümü öğretim üyelerinden deđerli danışman hocam Prof. Dr. Murat SUBAŐI' ya en içten teşekkürlerimi sunmayı bir bor bilirim.

Ayrıca, tez izleme komitesinde ve tez savunma jürisinde yer alarak deđerli görüşleri ile alıřmalarımıza destek olan saygıdeđer hocalarıma da ok teşekkür ediyorum.

Akademik eğitimim boyunca attığım her adımda bana maddi ve manevi destek olan, inanan, bu günlere gelebilmem için hiçbir fedakarlıktan kaçınmayan ve sevgisini her zaman hissettiğim canım babam Aziz ŐENDUR' a gönülden teşekkür ediyorum. Ne yazık ki, doktora eğitimimin başlarında, 2019 yılında, onu kaybetmenin derin üzüntüsünü yaşadım. Onun sevgisini ve desteđini her zaman hissettim ve kendisine olan minnettarlığımı ifade etmek istiyorum.

Tez alıřmam süresince, bana destek olan deđerli aileme ve her zaman yanımda olan, yardım ve teşvikleriyle beni cesaretlendiren kıymetli arkadaşlarıma gönülden teşekkür ediyorum.

Faika Derya ŐENDUR

ÖZET

DOKTORA TEZİ

İKİ BOYUTLU HİPERBOLİK BİR PROBLEMDE LİNEER OLMAYAN OPTİMİZASYON İLE TERS KATSAYI BELİRLEME

Faika Derya ŞENDUR

Danışman: Prof. Dr. Murat SUBAŞI

Amaç: Bu çalışmanın amacı, iki boyutlu hiperbolik bir başlangıç-sınır değer problemde, çözüme ait bir final anı gözlemi kullanılarak problemde yer alan bir katsayı fonksiyonunun belirlenmesini incelemektir. Bu doğrultuda, önce ele alınan hiperbolik başlangıç-sınır değer probleminin direkt çözümü için etkili bir metodun kullanılması, ardından bir optimizasyon tekniğinin güncellenerek bu tür problemlerde katsayı fonksiyonunu belirlemesi sağlanacaktır. Bu güncellenmiş tekniğin, matematiksel modellemeye yeni bir yaklaşım getirmesi hedeflenmektedir. Ayrıca, hazırlayacağımız simülasyonlar ile bu tür problemlerde çözümün etkili bir şekilde elde edilmesi amaçlanmaktadır.

Yöntem: Bu tez çalışması, iki boyutlu hiperbolik bir başlangıç-sınır değer probleminde, çözümün bir final anı gözleminden yola çıkarak ters katsayı fonksiyonunun belirlenmesiyle ilgilidir. Direkt problemin çözümü, zayıf çözüm yaklaşımı ve sonlu elemanlar yöntemi ile elde edilmiştir. Ters problem kısmında, katsayı fonksiyonunun belirlenmesi için, lineer olmayan en küçük kareler optimizasyon yöntemlerinden Levenberg–Marquardt yöntemi güncellenerek probleme uyarlanmıştır. Bu aşamada, klasik Levenberg–Marquardt yöntemine aday parametre fonksiyonları adapte edilmiş ve yerleştirilen ağırlık katsayıları sayesinde optimizasyon süreci başarıyla tamamlanmıştır. Ayrıca, MATLAB programı kullanılarak hem direkt problemin hem de ters problemin çözümünü veren simülasyonlar hazırlanmıştır. İncelenen tekniğin ve hazırlanan simülasyonların başarısı, üretilen nümerik problemler üzerinden test edilmiştir.

Bulgular: İncelenen problemde katsayı fonksiyonu çözümün bir final anı gözleminden belirleneceği için nümerik yöntemlerin kullanılması kaçınılmazdır. Direkt problemin çözümünde literatürde mevcut olan sonlu elemanlar yöntemi kullanılmıştır. Ters katsayı belirleme kısmında ise lineer olmayan en küçük kareler optimizasyon yöntemlerinden Levenberg–Marquardt yönteminin seçilen problem için güncellenmiş hali oldukça başarılı sonuçlar vermiştir. Çalışmada sunulan test problemleri üzerinden elde edilen veriler ve analiz sonuçları görsel ve sayısal olarak gösterilmiştir. Bu örnekler, araştırma bulgularının somut ve anlaşılır bir şekilde sunulmasına katkı sağlamıştır.

Sonuç: İki boyutlu bir hiperbolik başlangıç-sınır değer probleminde ters katsayı belirleme problemi için Levenberg-Marquardt optimalleştirme tekniğinin uyarlanmış hali ile katsayı fonksiyonu başarılı olarak elde edilebilmiştir. Çalışmanın esas katkısı bu çözümün şimdiye kadar benzer problemlerde uygulanmamış bir optimizasyon tekniği sayesinde belirlenmiş olmasıdır. Bu katkı, oluşturulan yazılım ve simülasyonlarla desteklenerek, araştırmanın pratik uygulamalar üzerindeki etkisini güçlendirmektedir. Ayrıca, yeterli sayıda nümerik test yaparak teorik sonuçları doğrulamak, araştırmanın güvenilirliğini artırmakta ve bulguların daha geniş bir kitleye ulaşmasını sağlamaktadır. Belirlenen bu yöntemin, hiperbolik denklemler için ters problemlerin çözümünde önemli bir araç olabileceği ortaya konmuştur.

Anahtar Kelimeler: Ters Problemler, Hiperbolik Denklemler, Sonlu Elemanlar Yöntemi, Sayısal Optimizasyon.

Ağustos 2024, 93 sayfa

ABSTRACT

DOCTORAL DISSERTATION

INVERSE COEFFICIENT IDENTIFICATION BY NONLINEAR OPTIMIZATION IN A TWO DIMENSIONAL HYPERBOLIC PROBLEM

Faika Derya ŞENDUR

Supervisor: Prof. Dr. Murat SUBAŞI

Purpose: The aim of this study is to examine the determination of a coefficient function in a two-dimensional hyperbolic initial-boundary value problem using a final-time observation of the solution. In this direction, first an effective method will be used for the direct solution of the hyperbolic initial-boundary value problem, then an optimization technique will be updated to determine the coefficient function in such problems. It is aimed that this updated technique will bring a new approach to mathematical modelling. In addition, the aim is to obtain a solution effectively for such problems with the simulations we will prepare.

Method: This thesis study is related to the determination of the inverse coefficient function in a two-dimensional hyperbolic initial-boundary value problem, using a final moment observation of the solution. The solution of the direct problem is obtained by the weak solution approach and the finite element method. In the inverse problem part, the Levenberg–Marquardt method, one of the nonlinear least squares optimization methods, is updated and adapted to the problem in order to determine the coefficient function. At this stage, the candidate parameter functions are adapted to the classical Levenberg–Marquardt method and the optimization process is successfully completed thanks to the weight coefficients placed. In addition, simulations that give the solution of both the direct problem and the inverse problem are prepared using the MATLAB program. The success of the examined technique and the prepared simulations is tested on the generated numerical problems.

Findings: In the problem under consideration, since the coefficient function will be determined from a final time observation of the solution, it is inevitable to use numerical methods. The finite element method available in the literature was used in the solution of the direct problem. In the inverse coefficient determination part, the updated version of the Levenberg–Marquardt method, one of the nonlinear least squares optimization methods, for the selected problem gave quite successful results. The data obtained from the test problems presented in the study and the analysis results are shown visually and numerically. These examples contributed to the presentation of the research findings in a concrete and understandable way.

Results: The coefficient function has been successfully obtained by adapting the Levenberg–Marquardt optimization technique for the inverse coefficient determination problem in a two-dimensional hyperbolic initial-boundary value problem. The main contribution of the study is that this solution has been achieved by an optimization technique that has not been applied to similar problems so far. This contribution strengthens the impact of the research on practical applications by supporting the software and simulations developed. In addition, verifying the theoretical results by performing a sufficient number of numerical tests increases the reliability of the research and ensures that the findings reach a wider audience. It has been demonstrated that this determined method can be an important tool in solving inverse problems for hyperbolic equations.

Keywords: Inverse Problems, Hyperbolic Equations, Finite Element Method, Numerical Optimization.

August 2024, 93 pages

İÇİNDEKİLER

TEZ KABUL VE ONAY TUTANAĞI	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iv
ÖZET	v
ABSTRACT	vi
İÇİNDEKİLER.....	vii
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
SİMGELER DİZİNİ	xi
GİRİŞ.....	1
KURAMSAL TEMELLER.....	7
Literatürdeki Çalışmalar.....	12
MATERYAL VE METOT	17
Sonlu Elemanlar Yönteminin Genel Yapısı.....	17
Kısmı diferansiyel denkleminin zayıf formülasyonu.....	17
Sonlu elemanlar.....	19
Uzayda ayırıklaştırma	20
İyi Tanımlanmış (well-posed)- Kötü Tanımlanmış (ill-posed) Problem	23
Bazı Optimizasyon Yöntemlerinin Temel Tanımları ve Karşılaştırmalar	27
Steepest Descent metodu (Gradyen metodu)	28
Newton metodu	29
Gauss-Newton metodu	30
Levenberg-Marquardt metodu	31
Örnek: Rosenbrock fonksiyonunun minimizasyonu.....	32
ARAŞTIRMA BULGULARI	37
Direkt Problem için Sonlu Elemanlar Yöntemi ve Zayıf Çözüm	38
Zayıf çözüm	38
Sonlu elemanlar yöntemi.....	39
Ters Katsayı Belirleme Problemi	46
Ters katsayı belirleme probleminin çözümü	47
Nümerik Örnekler	51
Örnek 1:.....	51
Örnek 2:.....	55

TARTIŞMA VE SONUÇ.....	60
KAYNAKLAR.....	61
EKLER	64
EK-1. Direkt Problemin Çözümü için Yazılmış Matlab Arayüzünün Kodu	64
EK-2. Örnek 1 ile Verilen Seçenekler için Gürültüsüz Gözlemlere Ait Matlab Arayüz Ekranları.....	67
EK-3. Örnek 1 ile Verilen P3 Seçeneği için Bazı Gürültülü Gözlemlere Ait Matlab Arayüz Ekranları	69
EK-4. Örnek 1 ile Verilen Ters Problemde Gürültülü Gözlemlere Karşılık Gelen Çözümü Bulma için Yazılmış Matlab Arayüzünün Kodu	72
EK-5. Örnek 2 ile Verilen Seçenekler için Gürültüsüz Gözlemlere Ait Matlab Arayüz Ekranları.....	75
EK-6. Örnek 2 ile Verilen P2 Seçeneği için Bazı Gürültülü Gözlemlere Ait Matlab Arayüz Ekranları	77
ÖZGEÇMİŞ.....	80

TABLolar DİZİNİ

Tablo 1. Örnek 1 için Bazı h ve Karşılık Gelen Hata Norm Değerleri	52
Tablo 2. Örnek 1 için Bazı Aday Parametre Fonksiyonları.....	53
Tablo 3. Örnek 1 için Bazı Aday Parametre Fonksiyonları için Elde Edilen Sonuçlar.....	53
Tablo 4. Örnek 1 için P3 Probleminde İterasyon Sonuçları	54
Tablo 5. Örnek 1 için P3 Problemindeki Gürültülere Karşılık Gelen Çözümlerin Bağlı Değişimi	54
Tablo 6. Örnek 2 için Bazı h ve Karşılık Gelen Hata Norm Değerleri	57
Tablo 7. Örnek 2 için Bazı Aday Parametre Fonksiyonları.....	57
Tablo 8. Örnek 2 için Bazı Aday Parametre Fonksiyonları için Elde Edilen Sonuçlar.....	58
Tablo 9. Örnek 2 için P2 Probleminde İterasyon Sonuçları	58
Tablo 10. Örnek 2 için P2 Problemindeki Gürültülere Karşılık Gelen Çözümlerin Bağlı Değişimi	59

ŞEKİLLER DİZİNİ

Şekil 1. Operatör denklemlerde çözümün varlığının şematik yapısı.....	24
Şekil 2. Operatör denklemlerde çözümün tekliğinin şematik yapısı.....	24
Şekil 3. Operatör denklemlerde çözümün kararlılığının şematik yapısı.....	25
Şekil 4. Operatör denklemlerde düzgünleştirme tekniğinin şematik yapısı.....	27
Şekil 5. Rosenbrock fonksiyonunun minimizasyonunda Steepest Descent metodu.....	33
Şekil 6. Rosenbrock fonksiyonunun minimizasyonunda Newton metodu.....	34
Şekil 7. Rosenbrock fonksiyonunun minimizasyonunda Gauss-Newton metodu.....	34
Şekil 8. Rosenbrock fonksiyonunun minimizasyonunda Levenberg-Marquardt metodu.....	35
Şekil 9. Rosenbrock fonksiyonunun minimizasyonunda optimizasyon metotlarının karşılaştırılması.....	36
Şekil 10. Direkt problemin şematik yapısı.....	37
Şekil 11. Direkt problemin çözümü için hazırlanan simülasyonu gösteren MATLAB arayüz ekranı.....	45
Şekil 12. Ters problemin şematik yapısı.....	47
Şekil 13. Örnek 1 için tanım kümesi ve $h = 0,1$ için sonlu eleman ağı.....	51
Şekil 14. Örnek 1 için $T = 1$ anında gerçek dalga fonksiyonu.....	52
Şekil 15. Örnek 1 için $T = 1$ anında hesaplanan dalga fonksiyonu.....	52
Şekil 16. Örnek 1 için $h = 0,1$ değerine karşılık hata değerleri.....	52
Şekil 17. Örnek 2 için tanım kümesi ve $h = 0,15$ için sonlu eleman ağı.....	56
Şekil 18. Örnek 2 için $T = 2$ anında gerçek dalga fonksiyonu.....	56
Şekil 19. Örnek 2 için $T = 2$ anında hesaplanan dalga fonksiyonu.....	56
Şekil 20. Örnek 2 için $h = 0,15$ değerine karşılık hata değerleri.....	57

SİMGELER DİZİNİ

λ	Damping(sönümlenme) parametresi
∇	Gradient operatörü
Φ	Operatör uzayından çözüm uzayına tanımlanan dönüşüm
Γ	Çözüm uzayından gözlem uzayına tanımlanan dönüşüm
δ_{ij}	Kronoker delta fonksiyonu
$(x, y) \in \Omega \subset \mathbb{R}^2$	\mathbb{R}^2 uzayında verilen bölge
$L_2(0, T]$	$(0, T]$ aralığında ölçülebilir ve karesi integrallenebilir fonksiyonlar uzayı
$L_2(\Omega)$	Ω bölgesinde ölçülebilir ve karesi integrallenebilir fonksiyonlar uzayı
$L_\infty(\Omega)$	Ω bölgesinde esaslı sınıra sahip fonksiyonlar uzayı
$H^1(\Omega)$	Kendisi ve birinci mertebeden genelleştirilmiş türevleri $L_2(\Omega)$ uzayına ait olan fonksiyonların oluşturduğu Hilbert uzayı
$H_0^1(\Omega)$	Kendisi ve birinci mertebeden genelleştirilmiş türevleri $L_2(\Omega)$ uzayına ait olan ve Ω bölgesinin sınırında sıfır değerine sahip fonksiyonların oluşturduğu Hilbert uzayı
$H_g^1(\Omega)$	Kendisi ve birinci mertebeden genelleştirilmiş türevleri $L_2(\Omega)$ uzayına ait olan ve Ω bölgesinin sınırında g değerine sahip fonksiyonların oluşturduğu Hilbert uzayı

GİRİŞ

Dalga denklemleri, dalgaların yayılımını modellemek için kullanılan hiperbolik kısmi diferansiyel denklemlerdir. Hiperbolik denklemler, iki veya daha fazla değişkenli fonksiyonlar ile oluşturulmuş kısmi diferansiyel denklemlerin özel bir sınıfıdır. Bu denklemler, birinci ve ikinci mertebeden türevlerin bir arada yer aldığı denklemler olup genellikle dalgaların ve titreşimlerin matematiksel modellerini sağlarlar.

Hiperbolik denklemler bir ortam içerisindeki titreşimlerin ve özellikle dalgaların nasıl yayıldığını tanımlarlar. Bu nedenle “dalga denklemleri” olarak adlandırılırlar. Dalga denklemlerinin en basitlerinden biri bir-boyutlu haldeki salınım yapan yay problemine ait olanıdır. İki-boyutlu halde bir davulcunun titreştirdiği davul zarı örnek olarak düşünülebilir. Üç-boyutlu halin hayal edilmesi biraz daha zor olsa da örnek olarak şeffaf bir jelatin içerisindeki bir sıvıda süspansiyon halinde yer alan taneciklerin jelatin zarfa yaptığı çarpma hareketi olarak alınabilir (Yükselen 2009).

Hiperbolik denklemlerin tarihçesi, dalga denklemlerinin incelenmeye başlandığı 18. yüzyıla dayanır. Jean le Rond D’Alembert (1747) yayınladığı çalışmasında, titreşen bir telin davranışını incelemek için ilk dalga denklemini modellemiştir. Bu çalışma, hiperbolik denklemler alanındaki ilk temel adımlardan biridir (Graff 2012).

18. yüzyılın sonlarına doğru, D’Alembert ve Leonhard Euler, dalga denklemlerinin matematiksel çözümlemesi üzerinde çalışmışlardır. 1766 yılında Leonard Euler, kavisli çubukların davranışına dayanarak bir zilin titreşimlerini analiz etmeye çalışmıştır. James Bernoulli de (1789) bu sorunun analizini yapmaya çalışmıştır (Graff 2012).

19. yüzyılda, Joseph-Louis Lagrange ve Pierre-Simon Laplace gibi matematikçiler, hiperbolik denklemlerin analitik çözümünü derinleştirmişlerdir.

20. yüzyılda, hiperbolik denklemler teorisi, matematiksel analizin önemli bir parçası haline geldi. John von Neumann ve Kurt Friedrichs gibi matematikçiler, hiperbolik denklemlerin teorik temellerini incelemişlerdir ve hiperbolik sistemlerin istikrarını ve çözümlerini analiz etmişlerdir.

Soyut ve uygulamalı matematikte dalga denklemleriyle ilgili hiperbolik problemler birçok bilim insanının dikkatini çekmiştir. İzlenebilir davranışından dolayı dalganın iç dinamikleri hakkında biraz bilgi edinme isteği bu ilginin nedeni olmuştur. Geniş uygulama alanları nedeniyle, birçok bilim insanı, çeşitli sistemlerin davranışlarını modellemek ve ilgili sayısal algoritmalar geliştirmek için hiperbolik problemlere ve ilgili evrimsel denklemlere büyük önem vermektedir.

Günümüzde, hiperbolik denklemler, bilgisayar simülasyonları ve sayısal analiz teknikleri sayesinde birçok alanda uygulanmaktadır. Bu denklemler, fizik, mühendislik ve diğer disiplinlerde ses dalgaları, elektromanyetik dalgalar, su dalgaları ve sismik dalgalar gibi olguları incelemek için kullanılır. Hiperbolik denklemler, mühendislik, fizik, bilgisayar grafikleri ve görüntü işleme gibi alanlarda geniş bir uygulama yelpazesine sahiptir. Bu nedenle, hiperbolik denklemler ve onların uygulamaları, matematik ve bilimde sürekli gelişen bir araştırma alanıdır.

Ters problem; bir matematiksel sistem hakkında bilinen bir sonuç veya yapılan gözlemden yola çıkarak, sistemin özelliklerini veya parametrelerini tahmin etmeye çalışan bir problemdir.

Ters problemlerin kökenleri oldukça eski olmasına rağmen, modern anlamda sistemli bir şekilde ele alınmaları 19. yüzyılın sonlarına dayanmakta olup, 20. yüzyılın başlarında ve ortalarında yapılan teorik çalışmalarla şekillenmiştir. Matematiksel analiz, integral denklemler teorisi ve diferansiyel denklemler teorisi gibi alanların gelişmesiyle birlikte, ters problemlere olan ilgi artmıştır. Ters problemlerin tarihçesi, Newton ve Fourier gibi erken dönem matematikçilerden başlayarak, Gauss, Abel, Hadamard, Tikhonov, Kolmogorov ve Wiener gibi önemli isimlerin katkılarıyla zenginleşmiş, ters problemlerin anlaşılması ve çözülmesinde önemli adımlar atılmış ve modern dönemde Tarantola gibi matematikçilerle gelişmeye devam etmiştir.

Jacques Hadamard, ters problemlerin doğasını daha da anlamak için önemli katkılarda bulunmuştur. Hadamard (1908, 1932) yayınladığı çalışmalarında, lineer hiperbolik denklemlerin özellikleri, çözümleri ve bu denklemlerin “iyi tanımlanmış” olması için gereken koşulları detaylı bir şekilde incelemiştir. Ayrıca, çözümün varlığı, tekliği ve çözümdeki küçük değişikliklerin sonuca etkisinin kontrol edilebilir olması gibi konuları detaylı ve açık bir şekilde ele almıştır.

Andrey Tikhonov, ters problemlerin çözümünde kullanılan düzgünleştirme (regularization) yöntemlerini geliştirmiştir. Tikhonov (1963) yayınladığı çalışmasında, ters problemlerin kararsız doğasını ele almış ve düzgünleştirme teknikleriyle bu problemlerin çözümlerinin daha kararlı hale getirilmesini sağlamıştır. Tikhonov düzgünleştirme, günümüzde hala ters problemler için yaygın olarak kullanılan bir yöntemdir.

Ters problemlerin genellikle başlangıç verilerine göre sürekli olma şartını sağlayan nümerik çözümleri elde edilemez. Yapılan gözlemler gürültü içerebileceğinden bazı durumlarda ters problemin çözümünün varlığı bile mümkün değildir. Bu durumlar bilindiğinden ters problemlerde kararlı optimal çözümler bazı özel tekniklerle aranır. Kullanılan metotlar, bir elemanın tersini kararlı bir şekilde (başlangıç verilerine göre sürekli olma şartı ile) elde etmeye dayanır.

Ters problemlerin çözümü, genellikle çok zordur fakat sistemlerin daha iyi anlaşılmasına ve daha verimli bir şekilde çalıştırılmasına yardımcı olur. Bu nedenle bu problemlerin çözümleri önemlidir. Örneğin, bir makine tasarımı sırasında, makine özelliklerinin belirlenmesi için ters problemlerin çözümü kullanılabilir. Bu sayede, makine daha verimli bir şekilde çalıştırılabilir ve daha az maliyetli hale gelebilir. Bu nedenle, hiperbolik denklemler için ters problemlerin çözümü, gerçek hayattaki problemlere daha iyi yaklaşmak ve doğru sonuçlar elde etmek için kritik bir adımdır.

Ters problemler, birçok alanda kullanılır. Örneğin, fizik, mühendislik, bilim, ekonomi ve sağlık gibi alanlarda ters problemlerle karşılaşılır. Hiperbolik denklemlerle ilgili ters problemler, mühendislik uygulamalarındaki önemli rolü nedeniyle büyük ilgi görmektedir. Bu problemlerin çözümü, son yıllarda oldukça önemli bir araştırma alanı haline gelmiştir.

Denklemlerde bilinmeyen katsayılar içeren pek çok fiziksel modeller ve bu katsayıların belirlenmesinden oluşan ters problemlerin çözümü son yıllarda oldukça popüler bir araştırma alanı haline gelmiştir. Hiperbolik denklemler için ters problemler; titreşimli sicim, sismoloji, jeofizik, vs. gibi fiziksel uygulamalar için önemli bir yer tutar (Tekin 2018).

Günümüzde ters problemler, birçok uygulama alanında karşımıza çıkmaktadır. Tıbbi görüntüleme, manyetik rezonans görüntüleme (MRI), bilgisayarlı tomografi (CT), ultrasonografi, PET taramaları gibi tıbbi görüntüleme tekniklerinde, yeraltı görüntüleme ve petrol arama gibi jeofizik uygulamalarda, malzeme karakterizasyonu, sismoloji, akustik, elektriksel ölçümler ve daha birçok alanda ters problemler karşımıza çıkar.

Gelişen teknoloji ve matematiksel yöntemler, ters problemlerin çözümünde yeni ve ileri uygulamaların ortaya çıkmasına olanak sağlamaktadır. Özellikle yapay zeka, makine öğrenimi, derin öğrenme gibi alanlardaki ilerlemeler, ters problemlerin daha karmaşık ve büyük ölçekli veri setleriyle başa çıkmasını sağlamaktadır.

Hazırladığımız tez çalışmasında aşağıdaki hiperbolik başlangıç sınır değer problemi ele alınacaktır.

$x \in \Omega \subset \mathbb{R}^2$ bölgesindeki konum değişkenlerini ve $t \in (0, T]$ aralığındaki zaman değişkenini göz önünde bulundurarak, $Q_T := (x, t) \in \Omega \times (0, T]$ bölgesinde aşağıdaki hiperbolik problemi yazalım;

$$\frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c(x)\nabla u) + a(x)u = f(x, t), \quad (x, t) \in Q_T \quad (1.1)$$

$$u(x, 0) = u_0(x), \quad u_t(x, 0) = v_0(x), \quad x \in \Omega \quad (1.2)$$

$$u(x, t) = g(t), \quad x \in \partial\Omega, \quad t \in (0, T]. \quad (1.3)$$

Burada,

$$c(x) \in L_\infty(\Omega) \text{ ve } c(x) > 0, \quad a(x) \in L_\infty(\Omega), \quad f(x, t) \in L_2(Q_T), \quad (1.4)$$

$$u_0(x) \in H^1(\Omega), \quad v_0(x) \in L_2(\Omega), \quad g(t) \in L_2(0, T]$$

şeklindedir.

(1.1)-(1.3) denklemleriyle verilen problemin birçok fiziksel uygulaması vardır. Bu uygulamaların muhtemelen en önemlilerinden biri, Ω bölgesinde bulunan homojen veya homojen olmayan malzemedeki yapılmış bir zarın titreşimini ifade eden dalga denklemdir. Örneğin (1.1)-(1.3) probleminin $u(x, t)$ çözümü, zarın x noktasında ve t zamanda denge konumundan yer değiştirmesidir. Aslında dalga denklemi, Newton'un zar üzerindeki hareket denkleminin bir yorumu olarak kabul edilebilir; küçük bir zar parçasının mu_{tt} momentumundaki değişim, etki eden kuvvetlere eşittir.

Bu kuvvetler şu şekilde açıklanabilir;

$\nabla \cdot (c(x)\nabla u)$; zarın geriliminden kaynaklanan net kuvvettir: Gerilim bileşeni, zar parçası üzerinde net enine bir kuvvet oluşturur ve $c(x)$ fonksiyonuna, x noktasındaki gerilimin fonksiyonu denir.

$a(x)u$; geri getirme kuvvetidir ve zarın yer değiştirmesine karşı yöneltilmiş bir kuvvet olarak etki eder. Eğer u yer değiştirme pozitif olursa (Ω alanının üzerinde), o zaman kuvvet negatiftir.

$f(x, t)$ dış kuvvet olarak adlandırılır; Bu kuvvet, zar boyunca x ve t değişkenlerinin herhangi bir değerinde etkili olabilir. Örneğin, titreşimli davul yüzeyini tarif ederken, davulun yüzeyine çarpan ses dalgaları tarafından extra bir kuvvet uygulanabilir (Farlow 1993).

Bu tez çalışmasında (1.1)-(1.3) hiperbolik başlangıç sınır değer problemi ele alınmıştır. İlk olarak bu hiperbolik kısmi diferansiyel denkleminin çözüm uzayı belirlenmiştir. Sonra ele alınan problem için zayıf çözüm tanımlaması yapılarak bu çözüm için direkt probleme Sonlu Elemanlar Yöntemi uygulanmıştır. Daha sonra ortaya çıkan adi diferansiyel denklem sisteminin nümerik çözümü elde edilmiştir. MATLAB programlama dili kullanılarak elde edilen nümerik çözüm ile direkt problemin çözümü test edilmiştir ve direkt problemin çözümünü verecek simülasyonlar oluşturulmuştur. Ardından bu tezin amacı olan ve ele alınan hiperbolik problemin de yer aldığı bir ters problem oluşturulmuş ve bu problemin çözümü araştırılmıştır. Bu aşamada ters problem için düzgünleştirme tekniğini içeren bir optimalleştirme metodunun belirlenmesi hedeflenmiştir. Bu ters problem, gürültülü bir $w \in Y$ gözlemi verildiğinde,

$$F(c) = \|(\Gamma \circ \Phi \circ A)(c) - w\|^2$$

fonksiyoneli minimum yapacak $c(x)$ elemanını belirleme olarak özetlenebilir. Bu fonksiyoneli minimum yapacak bir optimalleştirme metodunun belirlenmesi üzerine incelemeler yapılmıştır. Son olarak da, bulunan ters katsayı belirleme probleminin optimal çözümünün sürekliliğinin, üretilen veriler üzerinde test edilmesi amaçlanmaktadır. Bu sebeple, istenen bağıl hata ile rastgele gürültü üretilmiştir ve bazı gürültülü gözlemler ile bunlara karşılık gelen optimal çözümler araştırılmıştır. MATLAB programlama dili kullanılarak ters katsayı fonksiyonunu hesaplayacak kodlar yazılmaya çalışılmıştır. Bu kapsamda grafiksel kullanıcı ara yüzleri oluşturularak belirli bir gürültü yüzdesine sahip gözlem fonksiyonu girildiğinde aranan ters katsayı fonksiyonunu belirleyip grafiği çizdirilebilmektedir.

Bu çalışmanın önemi şu şekilde ifade edilebilir:

- ✓ Çalışma, hiperbolik başlangıç sınır değer problemlerinin çözümü için belirlenen optimalleştirme tekniğini araştırmakta ve uygulamaktadır. Bu yeni matematiksel yaklaşım, hiperbolik denklemler için ters problemlerin çözümünde kullanılabilecek etkili bir yöntem olabilir.
- ✓ Hiperbolik denklemler ve ters problemler, mühendislik, fizik, jeofizik gibi birçok uygulama alanında önemli bir rol oynamaktadır. Bu çalışma, bu alanlarda karşılaşılan problemlere yeni bir bakış açısı getirerek uygulamalı matematikte ilerlemeyi hedeflemektedir.

- ✓ Çalışma, optimalleştirme tekniklerinin hiperbolik denklemler için nasıl uygulanabileceğini göstermektedir. Bu sayede, benzer problemlerin çözümünde kullanılacak yeni optimizasyon yöntemleri geliştirilmesi ve bu alanda ilerleme sağlanması amaçlanmaktadır.
- ✓ Belirlenen matematiksel yöntemin uygulanabilirliğini göstermek amacıyla yazılımlar geliştirilmesi planlanmaktadır. Bu yazılımlar, benzer problemlerin çözümünde araştırmacılara ve mühendislere yardımcı olabilir.

Sonuç olarak, bu çalışmanın önemi, hiperbolik başlangıç sınır değer problemlerinin çözümü için yeni bir matematiksel yaklaşım geliştirmesi, uygulamalı matematikte ilerleme sağlanması, optimizasyon tekniklerinin geliştirilmesine katkıda bulunması ve yazılım geliştirme sürecine odaklanmasıdır.

Tezin giriş bölümünde, dalga denklemleri, hiperbolik denklemler ve ters problemler hakkında tarihçesiyle birlikte genel bilgiler verilmiştir. Kuramsal temeller bölümünde, ileride kullanılacak temel tanımlar verilmiştir ve konu ile ilgili literatür incelemesi yapılmıştır. Materyal ve metot bölümünde, sonlu elemanlar yönteminin genel yapısı, ill-posed problemlerin tanımı ve optimizasyon teknikleri detaylı olarak açıklanmıştır. Araştırma bulguları bölümünde, direkt problem için sonlu elemanlar yöntemi ve zayıf çözümü ile ters problemin Levenberg-Marquardt metodu ile çözümü incelenmiştir. Elde edilen veriler ve analiz sonuçları görsel ve sayısal olarak gösterilmiştir. Sonuçlar yazılım ve simülasyonlarla da desteklenmiştir. Tartışma ve sonuç bölümünde ise, bu çalışmadan elde edilen sonuçlar ve literatüre katkısı açıklanmıştır.

KURAMSAL TEMELLER

Bu bölümde, ileride kullanılacak bazı temel kavramların tanımları ve hiperbolik problemler için optimizasyon ile ilgili literatürdeki bazı çalışmalar verilmiştir (Çağlıyan ve Çelebi 2013; Bayraktar 2017; Pişkin 2018; Şahin ve Yıldırım 2018).

Tanım 1 (Operatör):

Vektör uzayında tanımlı dönüşümlere operatör denir.

Tanım 2 (Fonksiyonel):

V, F cismi üzerinde bir vektör uzayı olmak üzere $f: V \rightarrow F$ operatörü fonksiyonel olarak adlandırılır. Eğer f lineer ise, o zaman f' ye lineer fonksiyonel adı verilir.

Tanım 3 (Adi- Kısmi Diferansiyel Denklem):

Diferansiyel denklem, tek bir bağımsız değişkene göre türev içerirse buna adi diferansiyel denklem denir. Bu denklemler genel olarak

$$f(x, y, y', y'', \dots, y^{(n)}) = 0$$

biçiminde veya eğer $y^{(n)}$ yalnız bırakılabiliyorsa

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)})$$

biçiminde yazılır.

Diferansiyel denklem, birden fazla bağımsız değişkene göre türev içerirse buna da kısmi diferansiyel denklem denir. Bu denklemler ise genel olarak

$$f\left(x, y, z, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \dots\right) = 0$$

veya

$$f(x, y, z, z_x, z_y, \dots) = 0$$

biçiminde yazılır.

Tanım 4 (Dalga Denklemi):

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

türündeki denklem bir boyutlu homojen dalga denklemi,

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

türündeki denklem ise bir boyutlu homojen olmayan dalga denklemi olarak adlandırılır. Burada x değişkeni konumu ve t değişkeni zamanı ifade etmektedir. c pozitif bir sabit sayıdır. F ise dalgaya etki ettiği bilinen bir dış kuvveti temsil eder.

Tanım 5 (Hiperbolik, Parabolik ve Eliptik Denklemler):

İkinci mertebeden hemen – hemen lineer

$$A(x, y)z_{xx} + 2B(x, y)z_{xy} + C(x, y)z_{yy} + F(x, y, z, z_x, z_y) = 0 \quad (2.1)$$

denklemini ele alalım. Burada A, B ve C ; xy – düzleminin bir Ω bölgesinde x ve y ' nin iki kez sürekli türetilebilir fonksiyonlarıdır ve aynı anda üçü de sıfır olmayan fonksiyonlar olarak kabul edilir.

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

olacak şekilde tanımlanan bu fonksiyona, (2.1) denkleminin diskriminantı denir.

Eğer bir Ω nın bir (x_0, y_0) noktasında

$\Delta(x_0, y_0) > 0$ ise, (2.1) denklemine bu noktada hiperboliktir,

$\Delta(x_0, y_0) = 0$ ise, (2.1) denklemine bu noktada paraboliktir,

$\Delta(x_0, y_0) < 0$ ise, (2.1) denklemine bu noktada eliptiktir,

denir. Bir Ω bölgesindeki her noktada hiperbolik, parabolik veya eliptik bir denklem varsa, Ω içinde bu denklem hiperbolik, parabolik veya eliptik olarak adlandırılır.

Genellikle bir denklem, katsayılarının tanımlandığı bir bölgede hiperbolik, parabolik veya eliptik olabilir. Özel olarak A, B ve C katsayıları (yani ikinci mertebeden türevlerin katsayıları) sabit olan bir denklem tüm düzlemde aynı türdendir.

Tanım 6 (Başlangıç-Sınır Değer Problemi):

Uygulamalı bilimlerde genellikle bir diferansiyel denklemin genel çözümü yerine onun bazı ek koşulları sağlayan çözümlerinin bulunması istenir. Eğer ek koşullar bağımlı değişken ve türevleri için tek bir noktada verilmişse, bu problem başlangıç-değer problemi olarak adlandırılır. Eğer koşullar en az iki farklı noktada verilmişse ise, bu problem sınır-değer problemi olarak bilinir.

Tanım 7 (Momentum):

Sabit m kütlesine sahip bir parçacık için Newton'un ikinci hareket yasası,

$$\vec{F}_{net} = m \frac{d\vec{v}_{net}}{dt} = \frac{d}{dt}(m\vec{v}_{net})$$

şeklinde yazılabilir. Bu ifade, parçacık üzerine etki eden net kuvvetin, parçacığın net hızı ile kütlesinin çarpımının zamanla değişim hızına eşit olduğunu söyler. Burada kütle ile hızın çarpımından oluşan terime ($m\vec{v}$) momentum denir. Aslında, momentum cismin hareket miktarıdır ve $\vec{p} = m\vec{v}$ ifadesi ile verilir. Momentum vektörel bir niceliktir; m kütlesi skaler bir nicelik olduğundan momentum hız ile aynı yönlüdür.

Tanım 8 (İmpuls (İtme) Kuvveti-Momentum Değişimi):

Belli bir zaman aralığında uygulanan bir kuvvet için momentum değişimini hesaplamak için,

$$\int_{t_i}^{t_s} \vec{F} dt = \int_{t_i}^{t_s} \left(\frac{d\vec{p}}{dt} \right) dt = \vec{p}_s - \vec{p}_i = \Delta\vec{p}$$

bağıntısı kullanılabilir. Bu eşitlikteki $\int_{t_i}^{t_s} \vec{F} dt$ niceliğine $\Delta t = t_s - t_i$ zaman aralığında parçacığa etki eden \vec{F} kuvvetinin impulsu, kuvvete ise impulsif kuvvet denir. Bu aynı zamanda sabit net bir kuvvetin anlık itmesi olarak da isimlendirilir. Bir parçacık üzerine etki eden \vec{F} kuvvetinin impulsu, bu kuvvetin neden olduğu parçacığın momentumundaki değişime eşittir. İmpuls ya da momentumdaki değişme,

$$\Delta\vec{p} = \vec{J} = \int \vec{F} dt = \vec{F}_{ort}\Delta t$$

olarak yazılır. İmpuls vektörünün (\vec{J} 'nin) yönü momentum değişiminin ($\Delta\vec{p}$ 'nin) yönü ile aynıdır. Ortalama kuvvet, çarpma süresi olan Δt zaman aralığında etkiyen kuvvetin ortalama değeridir.

Tanım 9 (Direkt Problem):

Bir matematiksel sistem verildiğinde sistemde yer alan fonksiyonlarından herhangi birinin parametre fonksiyonu olarak belirlenip, bu parametreye bağlı olarak sistemin çözümüne ait bir özelliğin belirlenmesi problemdir.

Tanım 10 (Ters Problem):

Matematiksel bir sistemin çözümüne ait bir özellikten (gözlemden) faydalanarak bu sistemde yer alan bir parametre fonksiyonunun belirlenmesi problemdir.

Örneğin; son yıllarda yaşanan salgınlar dikkate alındığında salgınlarla ilgili direkt problem, enfekte olan birey sayısının, belirli bir matematiksel model (örneğin, SIR modeli) kullanılarak zaman içindeki değişiminin öngörülmesidir. Ters problem ise, bu süreçte elde edilen verilerden hareketle hastalığın bulaşma oranı gibi model parametrelerinin ya da başlangıçtaki enfekte birey sayısının belirlenmesidir.

Tanım 11 (Hadamard Şartları):

Bir problemin

- i. Varlık: En az bir çözümü varsa
- ii. Teklik: Çözümü tek ise ve
- iii. Kararlılık: Çözüm giriş verilerine sürekli bağımlı ise yani başlangıç verilerinde olan küçük değişiklik çözümde de küçük değişiklik meydana getiriyor ise

probleme iyi tanımlanmış (well-posed) problem denir. Eğer bu şartlardan bir tanesi dâhi sağlanmıyorsa kötü tanımlanmış (ill-posed) problem denir. Bu şartlara Hadamard şartları denir.

Tanım 12 (Cauchy Dizisi ve Tam Metrik Uzay):

$X = (X, d)$ bir metrik uzay ve (x_n) bu uzayda bir dizi olsun. Verilmiş herhangi bir $\varepsilon > 0$ ve $m, n \geq n_0$ için

$$d(x_m, x_n) < \varepsilon$$

olacak şekilde bir $n_0 = n_0(\varepsilon)$ sayısı varsa (x_n) dizisine Cauchy dizisi (veya esas dizi) denir. X üzerindeki her (x_n) Cauchy dizisi yakınsaksa (X, d) metrik uzayı tam metrik uzay veya kısaca tam olarak adlandırılır.

Tanım 13 (İç Çarpım Uzayı):

X, F cismi üzerinde bir vektör uzayı olsun. $\langle \cdot, \cdot \rangle: X \times X \rightarrow F$ fonksiyonu,

- i. $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$,
- ii. $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$, ($\alpha \in F$)
- iii. $\langle x, y \rangle = \overline{\langle y, x \rangle}$,
- iv. $\langle x, x \rangle \geq 0$ ve $\langle x, x \rangle = 0 \Leftrightarrow x = 0$

şartlarını sağlıyorsa bu fonksiyona iç çarpım veya iç çarpım fonksiyonu ve X 'e de iç çarpım uzayı denir. İç çarpım uzayı genellikle $(X, \langle \cdot, \cdot \rangle)$ olarak gösterilir, burada $\langle \cdot, \cdot \rangle$ iç çarpım işaretini temsil eder.

Tanım 14 (Normlu Uzay):

N bir vektör uzayıdır ve $\| \cdot \|: N \rightarrow \mathbb{R}$ fonksiyonu, x ' in normunu $\|x\|$ ile gösterir.

Bu fonksiyon için

- i. $\|x\| = 0 \Leftrightarrow x = 0$,
- ii. $\|\alpha x\| = |\alpha|\|x\|$, ($\alpha \in F$) ve
- iii. $\|x + y\| \leq \|x\| + \|y\|$ (üçgen eşitsizliği),

şartlarını sağlıyorsa $\|\cdot\|$ ye N de veya N üzerinde norm ve N vektör uzayına da normlu uzay denir. Genellikle $(N, \|\cdot\|)$ ile gösterilir. $\|x\|$ sayısına ise x vektörünün normu (uzunluğu) denir.

Tanım 15 (Norm Metriği):

$(N, \|\cdot\|)$ normlu uzayında $d(x, y) = \|x - y\|$ olarak tanımlanan d metriğine, normun oluşturduğu metrik veya kısaca norm metriği denir.

Tanım 16 (Banach Uzayı):

$(N, \|\cdot\|)$ normlu uzayı, $d(x, y) = \|x - y\|$ olarak ifade edilen norm metriğine göre tam ise o zaman Banach uzayı olarak adlandırılır.

Tanım 17 (Hilbert Uzayı):

X bir iç çarpım uzayı ve $\|\cdot\|$, iç çarpım normu olmak üzere

$$d(x, y) = \|x - y\| = \sqrt{\langle x - y, x - y \rangle}$$

şeklinde tanımlanırsa (X, d) ' ye bir metrik uzayı denir. Eğer X iç çarpım uzayı, iç çarpım normu ile tanımlanan d metriğine göre tam ise, X 'e Hilbert uzayı adı verilir.

Bu tanıma göre Hilbert uzaylarına, özel Banach uzayları da denilebilir.

Tanım 18 ($L_2(\Omega)$ Uzayı):

$L_2(\Omega)$,

$$\iint |f|^2 d\Omega < \infty$$

olmak üzere tüm $f: \Omega \rightarrow \mathbb{R}$ fonksiyonlarına ölçülebilir fonksiyonlar uzayı denir. Bu uzayda sırasıyla iç çarpım ile norm aşağıdaki şekilde ifade edilir:

$$\langle f, g \rangle_{L_2(\Omega)} = \iint_{\Omega} f(x, t)g(x, t) dx dt,$$

$$\|f\|_{L_2(\Omega)} = \sqrt{\langle f, f \rangle_{L_2(\Omega)}}.$$

Yukarıda belirtilen iç çarpım ile $L_2(\Omega)$ uzayı bir Hilbert uzayıdır.

Tanım 19 ($H^1(\Omega)$ Uzayı):

$H^1(\Omega)$, Ω üzerinde tanımlı olan ve kendisi ile birinci mertebeden genelleştirilmiş kısmi türevleri $L_2(\Omega)$ uzayına ait olan fonksiyonların oluşturduğu bir Hilbert uzayıdır. Bu uzay üzerinde, sırasıyla iç çarpım ile norm

$$\langle f, g \rangle_{H^1(\Omega)} = \iint_{\Omega} \left(f(x, t)g(x, t) + \frac{\partial f(x, t)}{\partial x} \cdot \frac{\partial g(x, t)}{\partial x} + \frac{\partial f(x, t)}{\partial t} \cdot \frac{\partial g(x, t)}{\partial t} \right) dxdt,$$

$$\|f\|_{H^1(\Omega)} = \sqrt{\langle f, f \rangle_{H^1(\Omega)}}$$

olacak şekilde tanımlanır.

Teorem (Birinci Green Özdeşliği):

$u \in C^2(\Omega) \cap C^1(\bar{\Omega})$ olsun. (Yani; u fonksiyonu Ω bölgesinde iki defa türevli sürekli ve Ω bölgesinin kapanışında ($\bar{\Omega} = \Omega \cup \partial\Omega$) türevli sürekli olsun). Bu durumda

$$\int_{\Omega} v \Delta u dx = \int_{\partial\Omega} v \frac{\partial u}{\partial n} ds - \int_{\Omega} \nabla v \nabla u dx$$

dir. Burada n dışa doğru yönlendirilmiş birim vektör ve $\frac{\partial u}{\partial n} = n \cdot \nabla u$ dır.

Literatürdeki Çalışmalar

Hiperbolik denklemler için optimal kontrol problemleri literatürde, çeşitli yazarlar tarafından farklı açılardan çalışılmıştır. Bu çalışmalardan bazıları şu şekildedir:

Yamamoto (1999) çalışmasında,

$$u''(x, t) = \Delta u(x, t) - q(x)u(x, t), \quad x \in \Omega, 0 < t < T$$

$$u(x, 0) = a(x), \quad u'(x, 0) = b(x), \quad x \in \Omega$$

$$u(x, t) = \zeta(x, t), \quad x \in \partial\Omega, \quad 0 < t < T$$

ve

$$y''(x, t) = \Delta y(x, t) - q(x)y(x, t) + f(x)R(x, t), \quad x \in \Omega, 0 < t < T$$

$$y(x, 0) = y'(x, 0) = 0, \quad x \in \Omega$$

$$y(x, t) = 0, \quad x \in \partial\Omega, \quad 0 < t < T$$

lineer olmayan-lineer çok boyutlu hiperbolik ters problemler için ilkinde $q(x)$, ikincisinde $f(x)$ belirleyerek $u = u(q)$ ve $y = y(f)$ zayıf çözümlerinin tekliğini ve kararlılığını incelemiştir.

Maciag (2011) çalışmasında,

$$\frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad \tilde{\Omega} = \Omega \times (0, t)$$

başlangıç ve sınır koşullarıyla tamamlanan iki boyutlu dalga denklemi için dalga polinomlarını (dalga denklemi için Trefftz fonksiyonları) kullanarak direkt ve ters problemlerin yaklaşık çözümlerini araştırmıştır.

Subaşı and Saraç (2012) çalışmasında,

$$u_{tt} - a^2 u_{xx} = F(x, t), \quad (x, t) \in (0, l) \times (0, T)$$

$$u(x, 0) = \varphi(x), \quad u_t(x, 0) = \psi(x)$$

$$u(0, t) = 0, \quad u(l, t) = 0$$

hiperbolik denklemi için

$$J_a = \int_0^l |u_t(x, T; \varphi) - y_1(x)|^2 dx + \int_0^l |u_x(x, T; \varphi) - y_2(x)|^2 dx + \alpha \int_0^l \varphi_x^2 dx$$

fonksiyoneli kullanarak $\varphi(x) \in H^1(0, l)$ kontrol fonksiyonunu bulma problemini incelemiştir.

Goncharkii and Romanov (2012) çalışmalarında,

$$c^{-2}(r)u_{tt}(r, t) - \Delta u(r, t) = \delta(r - q)f(t)$$

$$u(r, t = 0) = u_t(r, t = 0) = 0$$

ters katsayı problemlerini çözmek amacıyla dalga denklemi için Green fonksiyonların yardımıyla elde edilen integral gösterimlere dayanan yaklaşım ile hata fonksiyonunun gradyanının bir kısmı diferansiyel denklem için ek problemin çözümü açısından doğrudan hesaplanmasına dayanan yaklaşımı karşılaştırmıştır.

Tagiyev (2012) çalışmasında,

$$\frac{\partial^2 u}{\partial t^2} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(k_i(x) \frac{\partial u}{\partial x_i} \right) + q(x, t)u = f(x, t), \quad (x, t) \in Q_t$$

$$u|_{t=0} = \varphi(x), \quad \frac{\partial u}{\partial t}|_{t=0} = \varphi_1(x), \quad u|_{S_T} = 0$$

lineer hiperbolik denklemi için

$$J(v) = \alpha_0 \int_{Q_T} |u(x, t, v) - z_0(x, t)|^2 dx dt + \alpha_1 \int_{\Omega} |u(x, T, v) - z_1(x)|^2 dx$$

fonksiyoneli kullanarak $k(x) = (k_1(x), k_2(x), \dots, k_n(x))$ olmak üzere $v = (k(x), q(x, t))$ katsayılarını kontrol etmiştir.

Serovajsky (2013) çalışmasında,

$$\begin{aligned} y'' - \Delta y + |y'|^p y' &= v, \quad (x, t) \in Q = \Omega \times (0, T) \\ y &= 0, \quad (x, t) \in \Sigma = S \times (0, T) \\ y(x, 0) = 0, y'(x, 0) &= 0, \quad x \in \Omega \end{aligned}$$

hiperbolik denklemi için

$$I(v) = \int_Q F(x, t; y[v](x, t), v(x, t)) dQ$$

fonksiyoneli kullanarak $v(x, t)$ fonksiyonunu kontrol etmiştir.

Ivanov and Potapov (2015) çalışmasında,

$$\begin{aligned} y_{tt} &= y_{xx} - \theta(x)y, \quad t > 0, \quad 0 < x < l \\ -\beta_0 y_x + \sigma_0 y|_{x=0} &= u_0(t), \quad \beta_1 y_x + \sigma_1 y|_{x=1} = u_1(t), \quad t > 0 \\ y|_{t=0} &= 0, \quad y_t|_{t=0} = 0, \quad 0 < x < l \end{aligned}$$

$u = (u_0(t), u_1(t))$ sınır kontrollerine karşılık gelen dalga denklemi için

$$\|y|_{t=T} - f^0(\cdot)\|^2 + \|y_t|_{t=T} - f^1(\cdot)\|^2$$

fonksiyoneli minimum yaparak belirli her sabit hedef $f = (f^0(x), f^1(x))$ için bir çift

$u^* = (u^*_0(t), u^*_1(t))$ sınır kontrolleri araştırmıştır.

Deiveegan *et al.* (2017) çalışmalarında,

$$\begin{aligned} u_{tt}(x, t) &= \Delta u(x, t) + \sigma(t)f(x), \quad x \in \Omega, 0 < t < T \\ u(x, 0) = u_t(x, 0) &= 0, \quad x \in \Omega \\ u(x, t) &= 0, \quad x \in \partial\Omega, 0 < t < T \end{aligned}$$

problemi için

$$J_\beta(f) = \frac{1}{2} \int_0^T \int_{\partial\Omega} \left| \frac{\partial u}{\partial n}(x, t, f) - g(x, t) \right|^2 dx dt + \frac{\beta}{2} \int_\Omega |f|^2 dx,$$

$$A = \{f \in L^2(\Omega): |f| \leq a\}, \quad J_\beta: A \subseteq L^2(\Omega) \rightarrow \mathcal{R}^+$$

fonksiyoneli minimum yapacak $f \in L^2(\Omega)$ kaynak terimini bulma problemini, Gradyan methodu ve iki parametrelili model fonksiyon methodunu kullanarak incelemiştir.

Tekin (2018) çalışmasında,

$$\begin{aligned} u_{tt} &= u_{xx} + a(t)u(x, t) + f(x, t), & (x, t) \in D_T \\ u(x, 0) &= \varphi(x), \quad u_t(x, 0) = \psi(x), & 0 \leq x \leq 1 \\ u(0, t) &= 0, \quad u_x(0, t) = u_x(1, t), & 0 \leq t \leq T \\ u(1, t) &= h(t), & 0 \leq t \leq T \end{aligned}$$

problemi için Fourier özfonksiyon açılımı yöntemini kullanarak $(a(t), u(x, t))$ çözüm çiftini bulma problemini incelemiştir.

Subaşı and Araz (2019) çalışmasında,

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} \left(k(x) \frac{\partial u}{\partial x} \right) + v(x)u &= f(x, t), & (x, t) \in \Omega = (0, l) \times (0, T) \\ u(x, 0) &= \varphi(x), \quad u_t(x, 0) = \psi(x), & x \in (0, l) \\ u(0, t) &= u(l, t) = 0, & t \in (0, T) \end{aligned}$$

hiperbolik problemi için

$$J(v) = \beta_0 \int_0^l |u(x, T; v) - y_1(x)|^2 dx + \beta_1 \int_0^l |u_t(x, T; v) - y_2(x)|^2 dx$$

fonksiyoneli kullanarak $v(x)$ kontrol fonksiyonunu incelemiştir.

Safarova (2020) çalışmasında,

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \Delta u + |u|u + vu &= f(x, t), & (x, t) \in Q \\ u &= 0, & (x, t) \in S, \quad u|_{t=0} = u_0(x), \quad \frac{\partial u}{\partial t}|_{t=0} = u_1(x), & x \in \Omega \\ u(x, T) &= \varphi(x), & x \in \Omega \end{aligned}$$

dalga denklemi için

$$J_\alpha(v) = J_0(v) + \frac{\alpha}{2} \int_\Omega |v(x)|^2 dx$$

fonksiyoneli minimalleştirecek $v \in V$ katsayısını kontrol etme problemini incelemiştir.

Akhundov and Habibova (2022) çalışmalarında,

$$u_{tt} - u_{xx} = f(t)g(x), \quad (x, t) \in D = (0, 1) \times (0, T]$$

$$u(x, 0) = \varphi(x), \quad u_t(x, 0) = \psi(x), \quad x \in [0, 1]$$

$$u(0, t) = u(1, t) = 0, \quad t \in [0, T]$$

$$\int_0^1 u(x, t) dx = h(t), \quad t \in [0, T]$$

olup $g(x)$, $\varphi(x)$, $\psi(x)$ ve $h(t)$ bazı verilen fonksiyonlardır. Ters problemin incelendiği bu çalışmada, hiperbolik denklemin sağ tarafındaki bilinmeyen katsayının belirlenmesi üzerine odaklanılmış ve çözümün benzersizliği, kararlılığı ve varlığıyla ilgili teoremler üzerinden analiz yapılmıştır.

MATERYAL VE METOT

Hazırlanan bu tez çalışması, (1.1)-(1.3) problemi kullanılarak bir direkt problemin ve buna bağlı bir ters problemin oluşturulmasını, sonrasında bu ters problemin optimizasyon teknikleri ile çözümünün bulunmasını içermektedir.

Sonlu Elemanlar Yönteminin Genel Yapısı

Sonlu Elemanlar yöntemi; karmaşık bir geometriyi, geometri üzerinde bir ağ oluşturarak alt alanların bir toplamı olarak tanımlar. Alt alanlar bir ağ oluşturur ve her bir köşe noktasına bir düğüm adı verilir. Bir sonraki adım, daha basit denklemler kullanılarak her bir alt etki alanında orijinal probleme yaklaşımdır. Ayrıca Sonlu Elemanlar yöntemi, ilgili hata fonksiyonunu en aza indirerek bir çözüme yaklaşımdır. Li (2019) tarafından yapılan çalışmadan çıkarılan sonuçlar aşağıdaki gibi ifade edilebilir.

İki boyutlu örnek bir kısmi diferansiyel denklem

$$m \frac{\partial^2 u}{\partial t^2} + d \frac{\partial u}{\partial t} - \nabla \cdot (c \nabla u) + au = f \quad (3.1)$$

şeklindedir. Sonlu elemanlar yöntemini kolayca açıklamak için, kısmi diferansiyel denklemi zaman türevli terimleri olmadan ele alalım. Bu yüzden (3.1) denklemdeki zaman türevli terimleri çıkartalım. Böylece

$$-\nabla \cdot (c \nabla u) + au = f \quad (3.2)$$

elde edilir. Zaman türevleri olmadığında, yukarıdaki denklemin çözümü zamana bağlı değildir. Bu nedenle yukarıdaki kısmi diferansiyel denkleme kararlı-durum problemi denir.

Çözüm bölgesinin her noktasında (3.1) veya (3.2) eşitliğini sağlayan u çözüm fonksiyonunu bulma yaklaşımına problem için güçlü formülasyon denir. Süreklilik şartlarının ağırlığından dolayı çoğu zaman güçlü formülasyon tercih edilmez. Bunun yerine daha az süreklilik şartı gerektiren zayıf formülasyonlar oluşturulur.

Kısmi diferansiyel denkleminin zayıf formülasyonu

Sonlu elemanlar yöntemi, (3.2) denklemin güçlü formülasyonunu çözmez, bunun yerine, kısmi diferansiyel denkleminin zayıf bir formülasyonunu çözer. Burada u çözümünün konum değişkenlerine göre ikinci türevleri yerine birinci türevleri kullanılır. Bu işlem, Green

özdeşliği kullanılarak u fonksiyonunun konum türevlerinden birinin v test fonksiyonları üzerine taşınmasıyla yapılır.

Zayıf formülasyon aşağıdaki şekilde elde edilir. Güçlü formülasyon, v test fonksiyonu ile çarpılır ve $(x, y) \in \Omega \subset \mathbb{R}^2$ olmak üzere Ω üzerinden integral alınır. Böylece

$$\int_{\Omega} -\nabla \cdot (c\nabla u)v dx + \int_{\Omega} auv dx = \int_{\Omega} f v dx \quad (3.3)$$

eşitliği elde edilir. dx notasyonu, Ω bölgesindeki hacim integrasyonudur. $dx = dx dy$ olmak üzere üç boyutta ise $dx = dx dy dz$ şeklindedir.

Green özdeşliğinden;

$$\int_{\Omega} -\nabla \cdot (c\nabla u)v dx = \int_{\Omega} (c\nabla u) \cdot \nabla v dx - \int_{\partial\Omega} (c\nabla u) \cdot n v ds \quad (3.4)$$

olmak üzere ds , bölgenin sınırı $\partial\Omega$ üzerindeki yüzey integrasyonunu gösterir. Bu, Ω alanı üzerinde dx ile gösterilen hacim integrasyonunun tersinedir.

(3.4) denkleminde eşitliğin sağ tarafı, (3.3) denkleminin ilk terimi yerine yazılırsa

$$\int_{\Omega} (c\nabla u) \cdot \nabla v dx - \int_{\partial\Omega} (c\nabla u) \cdot n v ds + \int_{\Omega} auv dx = \int_{\Omega} f v dx \quad (3.5)$$

olur. u çözümü,

$$(c\nabla u) \cdot n + qu = g, \quad (x, y) \in \partial\Omega$$

şeklindeki sınır şartları denklemini sağladığından dolayı

$$\int_{\Omega} (c\nabla u) \cdot \nabla v dx - \int_{\partial\Omega} (g - qu)v ds + \int_{\Omega} auv dx = \int_{\Omega} f v dx \quad (3.6)$$

denklemini elde edilir.

(3.6) denklemindeki u ' yu içeren tüm terimler sol tarafa alınırsa

$$\int_{\Omega} (c\nabla u) \cdot \nabla v dx + \int_{\partial\Omega} quv ds + \int_{\Omega} auv dx = \int_{\Omega} f v dx + \int_{\partial\Omega} gv ds \quad (3.7)$$

elde edilir. Böylece bu eşitliği sağlayan u fonksiyonunun bulunması problemi, süreklilik şartlarını azalttığından incelen problem için bir zayıf formülasyon olur.

Sonlu elemanlar

Şimdi u ve v için fonksiyonel uzayları seçelim. Fonksiyonel uzaylar, Ω ' nun sonlu elemanlar adı verilen küçük geometrik parçaların birleşimi halinde ayrıklaştırılmasıyla yakından ilişkilidir. Pratikte kullanılan en yaygın sonlu elemanlar, \mathbb{R}^2 'de üçgenler ve \mathbb{R}^3 'de dört yüzlülerdir.

Ω alanına,

$$\Omega \approx \mathcal{T}^h \equiv \bigcup_{i=1}^{N_T} T_i$$

olmak üzere N_T tane elemanın birleşimi ile yaklaşılabacaktır. Burada T_i , i . elemandır (her T_i , iki boyutta üçgen ve üç boyutta dört yüzlüdür). Elemanların birleşimine \mathcal{T}^h denir ve Ω alanı için sonlu elemanlar ağıdır. h sayısı elemanların boyutunu gösterir. Dolayısıyla, Ω için h ve N_T 'nin farklı olduğu farklı boyutlarda birkaç ağın olması mümkündür. Elemanların boyutu ne kadar küçük olursa, u 'nun yaklaşık çözümü o kadar doğru olur.

\mathcal{T}^h sonlu elemanlar ağındaki düğümler veya noktalar, elemanlardaki tüm köşelerin birleşimidir. İki boyutta her elemanda 3 köşe vardır. Üç boyutta ise her elemanda 4 köşe vardır. Ancak, elemanlar birbirine temas ettiğinden düğüm sayısının $3 \times N_T$ veya $4 \times N_T$ 'den çok daha az olduğu açık olmalıdır, dolayısıyla aynı düğüm birden fazla elemana ait olabilir. v_i^1, v_i^2, v_i^3 (iki boyutta) veya $v_i^1, v_i^2, v_i^3, v_i^4$ (üç boyutta) T_i elemanının köşeleri olsun. O zaman düğümlerin kümesi,

$$\{P_1, P_2, \dots, P_{N_p}\} = \bigcup \{v_i^1, \dots, v_i^k\}, \quad i = 1, \dots, N_T, \quad k = 3 \text{ veya } 4$$

şeklindedir. N_p sayısı \mathcal{T}^h daki toplam düğüm sayısıdır.

\mathbb{P}_1 elemanları

u ve v için kullanılacak en basit fonksiyon uzayı, 1. dereceden global parçalı sürekli polinomların uzayı olan \mathbb{P}_1 uzayıdır. Bu uzay,

$$\varphi_j(x, y), \quad j = 1, \dots, N_p$$

olacak şekilde bir dizi baz fonksiyona sahiptir. Burada, N_p düğüm sayısıdır.

$\varphi_j(x, y)$ baz fonksiyonu, x, y 'deki lineer fonksiyondur ve aşağıdaki özelliklere sahiptir:

$$\varphi_j(x, y) = \begin{cases} 0 & T_i' \text{ de } P_j \notin \{v_i^1, \dots, v_i^k\} \text{ ise,} \\ a_j^i x + b_j^i y + c_j^i x^2 + d_j^i xy + e_j^i y^2 + k_j^i & T_i' \text{ de } P_j \in \{v_i^1, \dots, v_i^k\} \text{ ise.} \end{cases} \quad (3.8)$$

$\varphi_j(x, y)$ baz fonksiyonunu T_i elemanı üzerinde tanımlanan $a_j^i, b_j^i, c_j^i, d_j^i, e_j^i, k_j^i$ polinom katsayılarını elde etmek için, T_i 'nin köşelerine aşağıdaki kısıtlama uygulanır:

$$\varphi_j(x, y) = \begin{cases} 1, & (x, y) = P_j \text{ ise,} \\ 0, & (x, y) \neq P_j \text{ ve } (x, y) \in \{v_i^1, \dots, v_i^k\} \text{ ise.} \end{cases} \quad (3.9)$$

Kısacası, $\varphi_j(x, y)$ baz fonksiyonunu elde etmek için, P_j 'nin bir tepe noktası olduğu tüm T_i üçgenler ve (3.9) denklemindeki kısıtlamalar uygulanarak, φ_j 'yi T_i elemanı üzerinde tanımlayan $a_j^i, b_j^i, c_j^i, d_j^i, e_j^i, k_j^i$ katsayılar bulunmalıdır. (3.8) denkleminde görüldüğü gibi, φ_j yalnızca P_j 'nin bir tepe noktası olduğu elemanlarda sıfırdan farklı olduğu için etkisi küçüktür. Elemanların büyük çoğunluğunda, φ_j özdeş olarak sıfırdır.

Çalışılacak fonksiyon uzayı,

$$\mathcal{U} = \left\{ f(x, y) = \sum_{j=1}^{N_p} f_j \varphi_j(x, y), \quad f_j \in \mathbb{R} \right\}$$

olmak üzere φ_j baz fonksiyonları tarafından gerilen uzaydır.

Yukarıda açıklanan φ_j 'nin seçimi için,

$$f(P_j) = f_j$$

olmak üzere f_j katsayısının sadece P_j düğümündeki f değeri olduğu görülmektedir. Bu, seçilen sonlu elemanlar fonksiyon uzayının seçiminin çok kullanışlı ve pratik bir özelliğidir.

Uzayda ayrıklaştırma

\mathcal{U} fonksiyon uzayını tanımladıktan sonra, bu fonksiyon uzayına ait kısmi diferansiyel denklem için yaklaşık bir çözüm arandığını varsayalım. Yani, yaklaşık çözüm aşağıdaki şekilde olacaktır;

$$u^h(x, y) = \sum_{j=1}^{N_p} U_j \varphi_j(x, y), \quad U_j \in \mathbb{R}. \quad (3.10)$$

h , baz fonksiyonların tanımlandığı \mathcal{T}^h sonlu elemanlar ağını hatırlatır. Yaklaşık çözümü bulmak için U_j katsayılarının bulunması yeterlidir. Ve U_j, P_j düğümündeki fonksiyonun değeri

ile çakıştığından, aynı zamanda sonlu elemanlar ağı düğümlerindeki yaklaşık çözümün değerleri de elde edilir.

Şimdi, (3.7) denklemini ele alalım ve u^h yaklaşık çözümünü yerine yazalım.

$$\int_{\Omega} (c\nabla u^h) \cdot \nabla v dx + \int_{\partial\Omega} q u^h v ds + \int_{\Omega} a u^h v dx = \int_{\Omega} f v dx + \int_{\partial\Omega} g v ds. \quad (3.11)$$

Buradan

$$\begin{aligned} \sum_{j=1}^{N_p} U_j \int_{\Omega} (c\nabla \varphi_j) \cdot \nabla v dx + \sum_{j=1}^{N_p} U_j \int_{\partial\Omega} q \varphi_j v ds \\ + \sum_{j=1}^{N_p} U_j \int_{\Omega} a \varphi_j v dx = \int_{\Omega} f v dx + \int_{\partial\Omega} g v ds \end{aligned} \quad (3.12)$$

elde edilir. Yukarıdaki, yaklaşık çözümün sağlaması gereken bir kısıtlamadır. N_p tane bilinmeyen katsayı bulunduğundan, N_p tane kısıtlama denklemlerine ihtiyaç vardır. Bu denklemler, v ' nin φ_i , $i = 1, \dots, N_p$ baz fonksiyonlarının her biri olarak seçilmesiyle elde edilir. N_p tane kısıtlama denklemleri aşağıdaki gibidir:

$$\begin{aligned} \sum_{j=1}^{N_p} U_j \int_{\Omega} (c\nabla \varphi_j) \cdot \nabla \varphi_i dx + \sum_{j=1}^{N_p} U_j \int_{\partial\Omega} q \varphi_j \varphi_i ds \\ + \sum_{j=1}^{N_p} U_j \int_{\Omega} a \varphi_j \varphi_i dx = \int_{\Omega} f \varphi_i dx + \int_{\partial\Omega} g \varphi_i ds \end{aligned}, \quad i = 1, \dots, N_p. \quad (3.13)$$

Dolayısıyla, N_p tane bilinmeyenli ve N_p tane denklemden oluşan sistemin bir tek $U = [U_1, \dots, U_{N_p}]^T$ çözümü vardır.

Şimdi aşağıdaki sonlu eleman matrislerini ve vektörlerini tanımlayarak (3.13) denklemini matris formunda yazalım;

$$K_{ij} \equiv \int_{\Omega} (c\nabla \varphi_j) \cdot \nabla \varphi_i dx, \quad i = 1, \dots, N_p, \quad j = 1, \dots, N_p, \quad (3.14)$$

$$Q_{ij} \equiv \int_{\partial\Omega} q \varphi_j \varphi_i ds, \quad i = 1, \dots, N_p, \quad j = 1, \dots, N_p, \quad (3.15)$$

$$A_{ij} \equiv \int_{\Omega} a \varphi_j \varphi_i dx, \quad i = 1, \dots, N_p, \quad j = 1, \dots, N_p, \quad (3.16)$$

$$F_i \equiv \int_{\Omega} f \varphi_i dx, \quad i = 1, \dots, N_p, \quad (3.17)$$

$$G_i \equiv \int_{\partial\Omega} g \varphi_i ds \quad i = 1, \dots, N_p. \quad (3.18)$$

Burada K matrisi sertlik matrisi olarak adlandırılır. A ve Q matrislerdir. F ve G , N_p uzunluğundaki sütun vektörleridir.

O halde, (3.13) denkleminin matris formu,

$$KU + AU + QU = F + G, \quad U = \begin{bmatrix} U_1 \\ \vdots \\ U_{N_p} \end{bmatrix} \quad (3.19)$$

şeklindedir.

u^h fonksiyonu, (3.2) denkleminin yaklaşık bir çözümüdür.

Şimdi zaman türevli terimlere sahip (3.11) denkleme geri dönelim. u^h yaklaşık çözümünde küçük bir değişiklik yaparak U_j ' nin sabit sayılar olması yerine, U_j ' yi zamanın fonksiyonları haline getirelim:

$$u^h(x, y, t) = \sum_{j=1}^{N_p} U_j(t) \varphi_j(x, y). \quad (3.20)$$

U_j fonksiyonlarının, $m \neq 0$ ve $d = 0$ olduğunda iki sürekli zaman türevine (dalga denklemi) ve $m = 0$ ve $d \neq 0$ olduğunda ise bir sürekli zaman türevine sahip olduğunu varsayalım. Ayrıca $\varphi_j(x, y)$ zamana bağlı değildir. Dolayısıyla, dalga denklemi ($m \neq 0$ ve $d = 0$) için,

$$M_{ij} \equiv \int_{\Omega} m \varphi_j \varphi_i dx, \quad i = 1, \dots, N_p, \quad j = 1, \dots, N_p \quad (3.21)$$

olmak üzere

$$M \frac{\partial^2 U}{\partial t^2} + KU + AU + QU = F + G, \quad U = \begin{bmatrix} U_1 \\ \vdots \\ U_{N_p} \end{bmatrix} \quad (3.22)$$

zamana bağılı matris denklem sistemi elde edilir.

O halde elde edilen (3.22) denklem sistemi ikinci dereceden adi diferansiyel denklem sistemidir. Bu sistemin başlangıç şartları

$$U_j(0) = u_0(P_j), \quad \frac{dU_j}{dt}(0) = v_0(P_j), \quad j = 1, \dots, N_p$$

şeklindedir. Böylece

$$\begin{aligned} M \frac{\partial^2 U}{\partial t^2} + KU + AU + QU &= F + G \\ u_0 &= U(0), \quad v_0 = \frac{dU}{dt}(0) \end{aligned} \quad (3.23)$$

problemi elde edilir. Bu problem, $(0, T]$ zaman aralığında herhangi bir nümerik yöntem ile çözülebilir.

İyi Tanımlanmış (well-posed)- Kötü Tanımlanmış (ill-posed) Problem

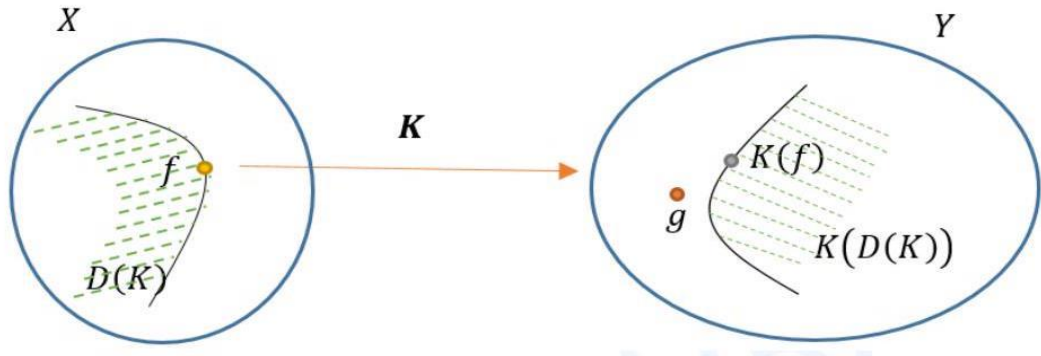
Aşağıdaki üç şart sağlanıyorsa verilen probleme *iyi tanımlanmış (well-posed) problem* denir.

- 1 Verilen $g \in Y$ için $Kf = g$ eşitliğini sağlayan en az bir $f \in X$ elemanını bulma (çözümün varlığı)
- 2 Verilen $g \in Y$ için $Kf = g$ eşitliğini sağlayan bir tek $f \in X$ elemanını bulma (çözümün tekliği)
- 3 $Kf = g$ eşitliğinde g fonksiyonundaki küçük değişikliklere karşılık f fonksiyonunda da küçük değişiklikler elde etme (çözümün kararlılığı)

Bu şartlardan en az biri sağlanmıyorsa probleme *kötü tanımlanmış (ill-posed) problem* denir.

Çözümün varlığı: Verilen $g \in Y$ için $Kf = g$ eşitliğini sağlayan en az bir $f \in X$ elemanını bulma.

K operatörünün tanım kümesi $D(K) \subseteq X$ olsun. K operatörü bu kümedeki elemanları $K(D(K))$ kümesine dönüştürecektir. Verilen $g \in Y$ elemanı aynı zamanda $K(D(K))$ kümesine ait değilse $Kf = g$ eşitliğini sağlayan bir $f \in X$ elemanı mevcut olmayacaktır.



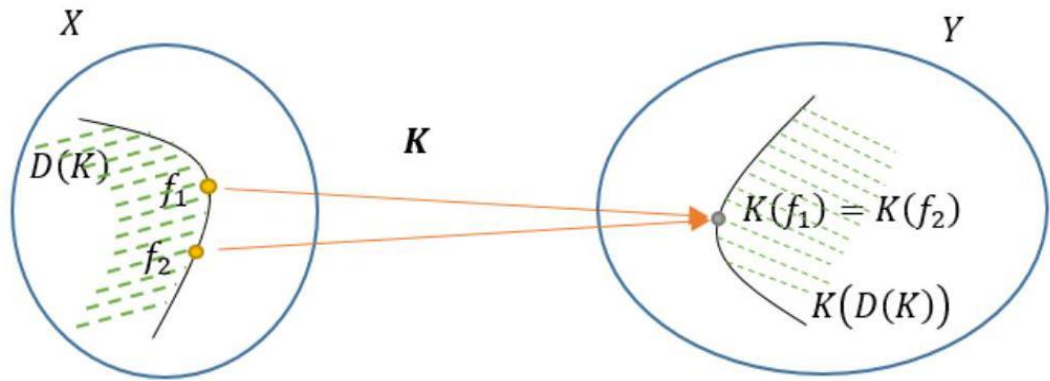
Şekil 1. Operatör denklemlerde çözümün varlığının şematik yapısı.

Örneğin; $\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ problemi için $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \in K(D(K))$ olup bu problemin bir $\begin{pmatrix} x \\ y \end{pmatrix}$ çözümü vardır ve $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ dir.

$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ problemi için $\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \notin K(D(K))$ olup bu problemin bir $\begin{pmatrix} x \\ y \end{pmatrix}$ çözümü yoktur. Optimal çözümü; $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1,6667 \\ -0,5 \end{pmatrix}$ dir.

Çözümün tekliği: Verilen $g \in Y$ için $Kf = g$ eşitliğini sağlayan bir tek $f \in X$ elemanını bulma.

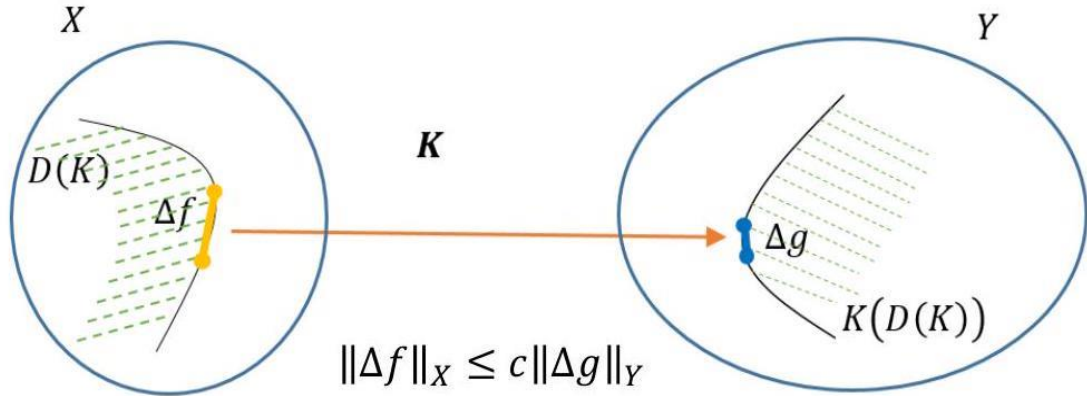
$f_1 \neq f_2$ elemanları için $K(f_1) = K(f_2) = g$ oluyorsa çözüm tek değildir.



Şekil 2. Operatör denklemlerde çözümün tekliğinin şematik yapısı.

Örneğin; $\begin{pmatrix} 1 & 2 \\ -2 & -4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 5 \\ -10 \end{pmatrix}$ problemi için $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ ve $\begin{pmatrix} 5 \\ 0 \end{pmatrix}$ elemanları birer çözümdür. Dahası bu sistemin sonsuz çözümü vardır.

Çözümün kararlılığı: $Kf = g$ eşitliğinde g fonksiyonundaki küçük değişikliklere karşılık f fonksiyonunda da küçük değişiklikler elde etme.



Şekil 3. Operatör denklemlerde çözümün kararlılığının şematik yapısı.

Örneğin;

$$\begin{aligned} x + 2y + 3z &= 1 \\ 2x + 4y + 6,001z &= 1 \end{aligned}$$

sistemini ele alalım.

Bu sistem $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6,001 \end{bmatrix}$ ve $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ için

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6,001 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

şeklinde $A\vec{x} = b$ matris formunda yazılabilir. Bu sistemin çözümü $x = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 600,2 \\ 1200,4 \\ -1000 \end{pmatrix}$

şeklinindedir. Şimdi b vektöründe bazı küçük değişiklikler yapalım ve bu değişikliklerin çözüme olan etkisini inceleyelim.

$$b_1 = \begin{pmatrix} 1 \\ 1,01 \end{pmatrix} \text{ ile verilen sağ taraf vektörü için çözüm } x_1 = \begin{pmatrix} 594,2 \\ 1188,4 \\ -990 \end{pmatrix} \text{ olup}$$

$$\|b - b_1\| = 0,01 \text{ iken } \|x - x_1\| = 16,73 \text{ olacaktır.}$$

Bu sonucu başka bir nokta ile kontrol edelim.

$$b_2 = \begin{pmatrix} 0,99 \\ 1 \end{pmatrix} \text{ ile verilen sağ taraf vektörü için çözüm } x_2 = \begin{pmatrix} 588,2 \\ 1176,4 \\ -980 \end{pmatrix} \text{ olup}$$

$$\|b - b_2\| = 0,01 \text{ iken } \|x - x_2\| = 33,47 \text{ olacaktır.}$$

Yani sistem, sağ taraf vektöründeki küçük değişikliklere karşılık tutarlı bir çözüm sergilememektedir.

Dolayısıyla, özellikle ters (invers) problemler çalışılırken problemin kötü tanımlanmış (ill-posed) olması çok sık karşılaşılan bir durumdur. Ters problemler; çözümün varlığı, tekliği ve kararlılığı olan Hadamard gereksinimlerine göre kötü tanımlanmıştır. Düzgünleştirme teknikleriyle kötü tanımlamanın üstesinden gelmek için birçok çalışma vardır (Engl et al. 1994; Yamamoto 1996; Cheng and Yamamoto 2000; Feng et al. 2003; Kabanikhin et al. 2005).

Uygulamalı matematikte kullanılan veriler genellikle ölçümlerden alınır veya tam verilerden ziyade nümerik hesaplamalarla elde edilmiş yaklaşık veriler kullanılır. Bu yüzden kullanılan verilerde bir miktar hata (gürültü) olması kaçınılmazdır.

Dolayısıyla, uygulamalı matematik ile uğraşırken her zaman verilerde gürültü olabileceği kabul edilip bazı regülerleştirme teknikleri kullanılarak optimal çözüm aranmıştır. Eğer gürültü ya da hata yoksa optimal çözüm denilen şey zaten tam çözüm olacaktır. Gürültü veya hata varsa da optimal çözüm ile bu gürültüye karşılık en uygun çözüm ortaya çıkmış olacaktır. O halde verilmiş $g \in Y$ elemanına karşılık $Kf = g$ eşitliğini sağlayan $f \in X$ elemanını bulma şeklinde bir invers problemi çalışıyorsak ve bu problem sadece teorik anlam ifade etmiyorsa özellikle bazı uygulamalara karşılık geliyorsa o verilmiş $g \in Y$ elemanında bir miktar gürültü bulunabileceği ya da g' nin yaklaşık bir ifade olabileceğini kabul etmemiz gerekir.

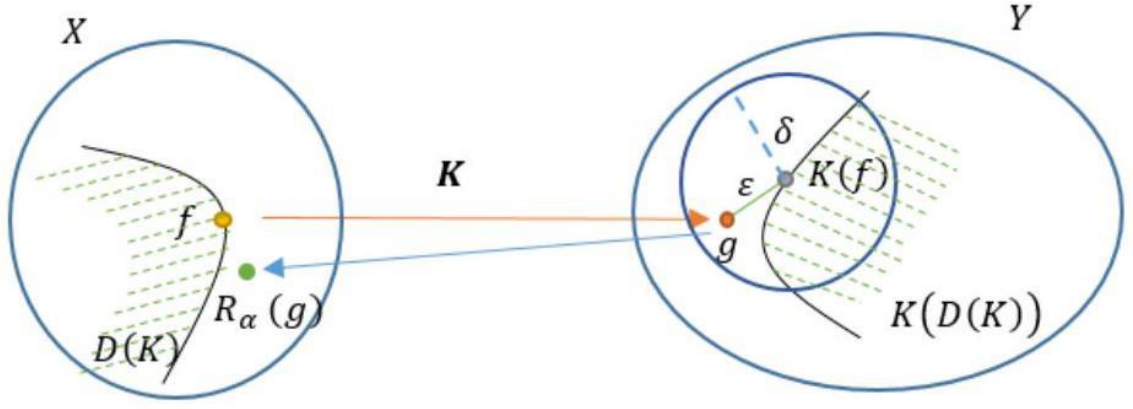
Bu durumu $g = K(f) + \varepsilon$ şeklinde gösteririz. Burada ε , üst sınırı bilinen ya da kontrol edilebilen gürültüyü ya da hatayı temsil eder. Gürültüdeki sınır $\|\varepsilon\|_Y \leq \delta$ ile gösterilir.

Çözümlerin tutarsızlığını önlemek için düzgünleştirme (regülerleştirme) teknikleri denilen teknikler kullanılır. Bu tekniklerden en önemlisi Tikhonov düzgünleştirme tekniğidir. Tikhonov düzgünleştirme, gürültüleri azaltmaya yönelik önemli bir yöntemdir.

O halde ters (invers) problem

Verilmiş $g = K(f) + \varepsilon$ elemanına karşılık gelen f elemanını bulma şeklinde düşünülmelidir.

Dolayısıyla ill-posed problemlerde α parametresine bağlı bir $R_\alpha(g)$ düzgünleştirme tekniği (Tikhonov Düzgünleştirme) kullanılır.



Şekil 4. Operatör denklemlerde düzgünleştirme tekniğinin şematik yapısı.

Burada $\delta \rightarrow 0$ için yani gürültü sıfıra giderken $R_\alpha(g) \rightarrow f$ şeklindedir.

Bazı Optimizasyon Yöntemlerinin Temel Tanımları ve Karşılaştırmalar

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m \rightarrow \mathbb{R}$$

$$f(x) = \frac{1}{2} \sum_{i=1}^m r_i(x)^2, \quad r_i(x) = y_i - M(x, t_i)$$

fonksiyonu için oluşturulmuş

$$\min_x f(x)$$

problemini ele alalım.

$f(x)$ fonksiyonunun $\nabla f(x)$ gradyeni

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

vektörüdür.

Hessian matrisi de

$$\nabla^2 f(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, 2, \dots, n$$

simetrik matrisidir.

$r(x) = [r_1(x), r_2(x), \dots, r_m(x)]^T$ vektörel fonksiyonunun Jakobyeni

$$[J(x)]_{m \times n} = \frac{\partial r_i(x)}{\partial x_j}, \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

matrisidir.

Bir x_* vektörünün $\min_x f(x)$ probleminin lokal minimumu olması için

- Gerek şart: x_* vektörü için $\nabla f(x_*) = 0$ olmasıdır.
- Yeter şart: $\nabla^2 f(x_*)$ matrisinin pozitif tanımlı olmasıdır.

Buna göre $f: \mathbb{R}^n \rightarrow \mathbb{R}$ dönüşümü için

$$f'(x) = \nabla f(x) = J(x)^T r(x)$$

$$f''(x) = \nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x) = J(x)^T J(x) + S(x)$$

yazılabilir.

Şimdi $\min_x f(x)$ problemi için bazı optimizasyon tekniklerini inceleyelim.

Steepest Descent metodu (Gradyen metodu)

f fonksiyonunun x noktasında bir h vektörü yönündeki değişimini inceleyelim. Taylor açılımı kullanılarak $\alpha > 0$ için

$$f(x + \alpha h) = f(x) + \alpha h^T f'(x) + O(\alpha^2)$$

yazılabilir.

Buradan yeteri kadar küçük α değeri için

$$f(x + \alpha h) \cong f(x) + \alpha h^T f'(x)$$

$$f(x + \alpha h) - f(x) \cong \alpha h^T f'(x)$$

elde edilir.

Eğer $h^T f'(x) < 0$ ise h vektörü f fonksiyonunun x noktasında azalma yönü olur.

f fonksiyonunun x noktasında h vektörü yönündeki bağıl değişimini inceleyelim.

$$\begin{aligned} \lim_{\alpha \rightarrow 0} \frac{f(x) - f(x + \alpha h)}{\alpha \|h\|} &= \lim_{\alpha \rightarrow 0} \frac{-\alpha h^T f'(x)}{\alpha \|h\|} = -\frac{1}{\|h\|} h^T f'(x) = -\frac{1}{\|h\|} \|h^T\| \|f'(x)\| \cos\theta \\ &= -\|f'(x)\| \cos\theta \end{aligned}$$

Burada, θ , $f'(x)$ ve h vektörleri arasındaki açıdır.

Bu bağıl değişimin maksimum olması için $\theta = \pi$ olması gerekir. Yani $f'(x)$ ve h vektörleri zıt yönlü olmalıdır.

Bir fonksiyonun maksimum veya minimum değerlerini bulmak için bağıl değişimin maksimum olduğu noktaları incelemek gerekir. Çünkü bağıl değişimin maksimum olduğu noktalar, bir fonksiyonun kritik noktalarıdır. Bu noktalarda fonksiyonun eğimi en hızlı değişir, fonksiyonun yönü değişir ve genellikle fonksiyonun en büyük (maksimum) veya en küçük (minimum) değerlerine ulaşılır.

O halde Steepest Descent Metodu (Gradyan Metodu) için h_{sd} adım uzunluğunu

$$h_{sd} = -f'(x) = -\nabla f(x)$$

$$h_{sd} = -J(x)^T r(x)$$

eşitliğinden elde ederiz.

Bu metod lokal minimuma lineer olarak yakınsar. Yani yakınsama genellikle yavaştır.

Newton metodu

f fonksiyonunu x noktasında iki kez sürekli türevlenebilen bir fonksiyon olsun. f' fonksiyonunu x noktasındaki Taylor açılımı

$$f'(x+h) = f'(x) + f''(x)h + O(\|h\|^2)$$

şeklindedir.

x_* vektörü lokal minimum ise bu noktada $f'(x_*) = 0$ olacağını biliyoruz. $x+h$ noktasının x_* noktasına yeterince yakın olduğunu kabul edelim. Bu yüzden Newton metodu iterasyonun son aşamalarında yani $x+h$ noktası x_* noktasına yakın olduğunda iyi sonuçlar verir.

Bu şart altında $f'(x+h) \cong f'(x_*) = 0$ ve h adımı da yeterince küçük olduğundan küçük $\|h\|$ sayısı için

$$0 \cong f'(x) + f''(x)h$$

$$f''(x)h \cong -f'(x)$$

yazılabilir.

$$f'(x) = \nabla f(x) = J(x)^T r(x)$$

$$f''(x) = \nabla^2 f(x) = J(x)^T J(x) + S(x)$$

eşitlikleri kullanılarak

$$f''(x)h \cong -f'(x)$$

için

$$[J(x)^T J(x) + S(x)]h \cong -J(x)^T r(x)$$

ifadesinden Newton metodu için h_n adım uzunluğunu

$$[J(x)^T J(x) + S(x)]h_n = -J(x)^T r(x)$$

eşitliğinden elde ederiz.

Burada $f''(x) = J(x)^T J(x) + S(x)$ matrisi pozitif tanımlı ise h_n vektörü azalma yönünü gösterecektir.

Herhangi bir adımdaki x noktası, x_* lokal minimum noktasının yakınında $f''(x)$ matrisi pozitif tanımlı olacak şekilde bir bölgede ise yakınsama karesel olacaktır. Diğer yandan bu bölgede $f''(x)$ matrisi negatif tanımlı ise yakınsama minimum nokta yerine maksimum nokta yönünde olabilir. O halde Steepest Descent metodunun garanti yakınsaması ile Newton metodunun hızını birleştirmek daha uygun olacaktır.

Bu durumda şöyle bir hibrit algoritma kullanılabilir:

x noktasında

$f''(x)$ pozitif tanımlı ise; $h = h_n$

Değilse; $h = h_{sd}$

$$x = x + h.$$

Gauss-Newton metodu

Newton metodundaki

$$[J(x)^T J(x) + S(x)]h_n = -J(x)^T r(x)$$

eşitliğinin

$$[J(x)^T J(x)]h_{gn} = -J(x)^T r(x)$$

şeklinde güncellenmesiyle oluşur.

Bazı durumlarda $S(x)$ teriminin elde edilmesi zorluklar içerdiğinden bu terimi çıkarmak bazı açılardan kolaylık sağlayabilir. $S(x_*) = 0$ ise yine karesel yakınsama sağlayacaktır. Diğer yandan $S(x_*)$ çok büyük ise bu terimi çıkarmak metodun yakınsamasını bozabilir.

Buradan azalma vektörü

$$h_{gn} = -[J(x)^T J(x)]^{-1} J(x)^T r(x)$$

$$h_{gn} = -J(x)^+ r(x)$$

olarak ifade edilebilir.

Levenberg-Marquardt metodu

Gauss-Newton metodunda h_{gn} vektörü teorik olarak azalma yönünü göstermekle birlikte kötü tanımlanmış (ill-posed) problemlerde veya $J(x)$ Jakobyeni için rank yetersizliği durumlarında bu metot yakınsamayabilir. Bu yüzden Gauss-Newton metodunda lineer adımlar için Tikhonov düzgünleştirme (regülerleştirme) yöntemi ile kararlılık sağlanarak Levenberg-Marquardt metodu oluşturulur.

Dolayısıyla bu metodun adım uzunluğu

$$[J(x)^T J(x) + \lambda I] h_{lm} = -J(x)^T r(x)$$

eşitliği ile elde edilir. Burada, λ damping (sönümleme) parametresidir ve I birim matristir.

$J(x)^T J(x)$ matrisi $J(x)^T J(x) + \lambda I$ ile değiştirilerek kötü tanımlanmış veya pozitif yarı tanımlı matris, pozitif tanımlı matrise dönüşür ve bu da h_{lm} doğrultusunun azalma yönü olmasını garanti eder.

Levenberg-Marquardt metodu, ters problem için düzgünleştirme tekniğini içeren bir optimalleştirme metodu ve en küçük kareler yöntemiyle optimize edilen bir parametre tahmin algoritmasıdır. Özellikle doğrusal olmayan modellerde ve gürültülü verilerle çalışırken etkili bir şekilde kullanılır. Levenberg-Marquardt metodunun uygulanmasında, başlangıç noktasının ve lambda değerinin seçimi kritiktir. Başlangıç noktası, çözümün yakınsama hızını ve doğruluğunu etkiler. Lambda'nın yüksek olması, çözümü istikrarlı hale getirirken, düşük olması Gauss-Newton'un hızlı yakınsama özelliğini korur. Metot, bir modelin parametrelerini uygun bir şekilde ayarlamak için iteratif adımlar kullanır. Bu adımlar, önceden belirlenen bir başlangıç çözümünden başlar ve her bir adımda en küçük kareler hatasını azaltmaya çalışarak çözüme doğru ilerler. Adaptif parametre olan lambda, iterasyon süresince dinamik olarak ayarlanır. Adımlar, genellikle en küçük kareler probleminin hatasını azaltan bir yönde ve hızla ilerleyen

bir yönde bir bileşim kullanılarak belirlenir. Algoritma, belli bir durdurma kriterine ulaşına veya maksimum iterasyon sayısına ulaşına kadar tekrarlar.

Levenberg-Marquardt metodu, Gauss-Newton metoduna benzer ancak bazı önemli değişiklikler içerir. Gauss-Newton metodunu, her adımda hesaplanan matrisin tersini alarak ve bu ters matrisi hesaplarken bir λ parametresi kullanarak geliştirir. Bu λ parametresi, adımların büyüklüğünü kontrol eder ve Gauss-Newton metoduna göre daha istikrarlı bir yakınsama sağlar. Büyük adımlar hızlı bir şekilde yakınsamak için kullanılırken, küçük adımlar yerel minimumlara yakınsamak için kullanılır.

Aynı zamanda, λ parametresi büyük olduğunda Gauss-Newton metoduna daha çok benzerken, λ parametresi küçük olduğunda Steepest Descent metoduna daha çok benzer. Levenberg-Marquardt metodu, hızlı bir şekilde yakınsamak için Gauss-Newton metodunun gücünü kullanırken, λ parametresini ayarlayarak Steepest Descent metodu ile de ilişkilidir. Levenberg-Marquardt metodu, Steepest Descent metodunun yavaş yakınsama sorununu azaltarak daha hızlı ve kararlı bir yakınsama sağlar.

Örnek: Rosenbrock fonksiyonunun minimizasyonu

Rosenbrock fonksiyonu;

$$f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

şeklinde olup iki boyutlu lineer olmayan bir optimizasyon problemidir. $f(x)$ fonksiyonu, orijin etrafındaki eğriliğinden dolayı Muz Fonksiyonu olarak da bilinir. Fonksiyonun minimuma doğru giden yolu, dar ve derin vadiler şeklinde olması metodların hızlı bir şekilde global minimuma ulaşmasını engelleyebilir. Zorlu bir test fonksiyonu olduğundan optimizasyon metodlarının performansını değerlendirmek için sıkça kullanılır.

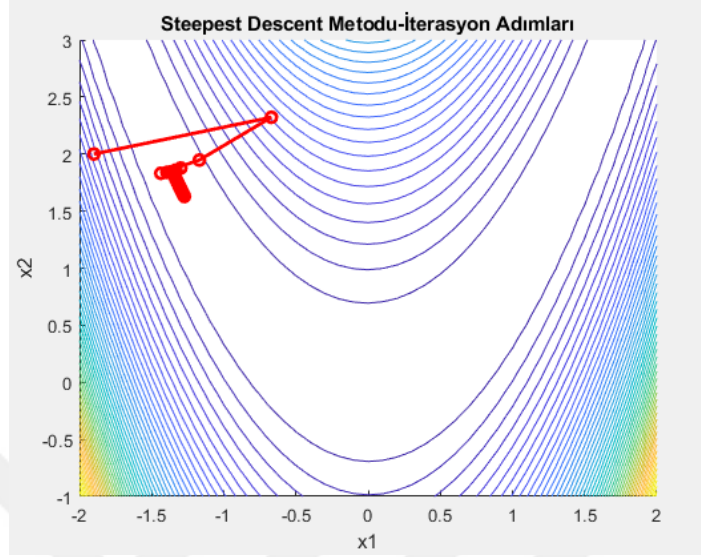
Bu fonksiyonun, global minimumu $x = [1,1]$ noktasında bulunur ve bu noktadaki minimum değeri $f(1,1) = 0$ olur.

Howard H. Rosenbrock tarafından (1960) yapılan makalede, Rosenbrock fonksiyonu tanıtılmıştır ve bu fonksiyonun optimizasyonunda kullanılan bazı yöntemler tartışılmıştır.

Şimdi bu fonksiyonunu, başlangıç noktası $x_0 = [-1,9,2]$ olmak üzere yukarıda tanıttığımız dört farklı optimizasyon metodu ile MATLAB'da çözelim ve iterasyon adımlarını grafikte gösterelim.

Steepest Descent metodu (Gradyen metodu) ile optimizasyon

Rosenbrock fonksiyonunu optimize etmek için Steepest Descent Metodu iterasyon adımlarını gösteren grafik aşağıdaki gibidir.

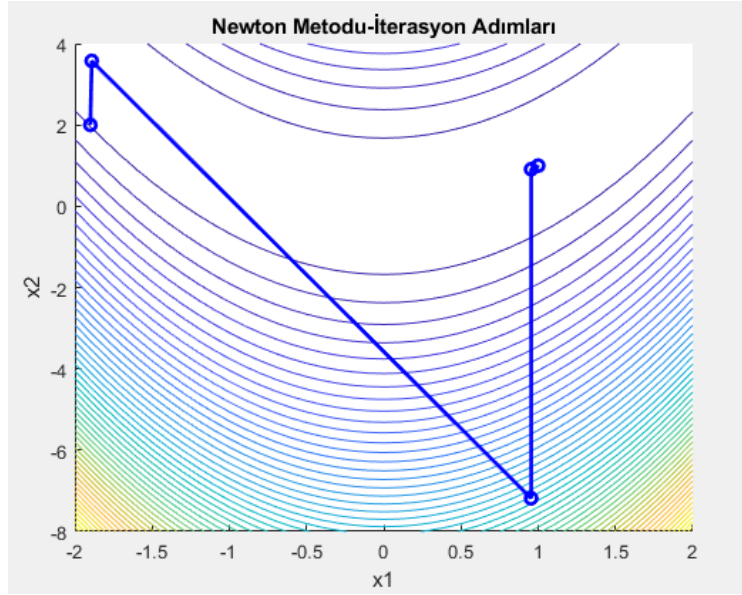


Şekil 5. Rosenbrock fonksiyonunun minimizasyonunda Steepest Descent metodu.

Steepest Descent metodu en basit ve en genel optimizasyon algoritmalarından biridir. Her iterasyonda sadece ilk türev hesaplandığı için hesaplama maliyeti düşüktür. Fakat Rosenbrock fonksiyonu gibi dar vadiler içeren fonksiyonlarda, Steepest Descent metodu yavaş yakınsama gösterir. Fonksiyonun eğimlerini takip ederken zigzag yapabilir ve bu da iterasyon sayısını artırır. Ayrıca yerel minimumlara takılabilir.

Newton metodu ile optimizasyon

Rosenbrock fonksiyonunu optimize etmek için Newton Metodu iterasyon adımlarını gösteren grafik aşağıdaki gibidir.

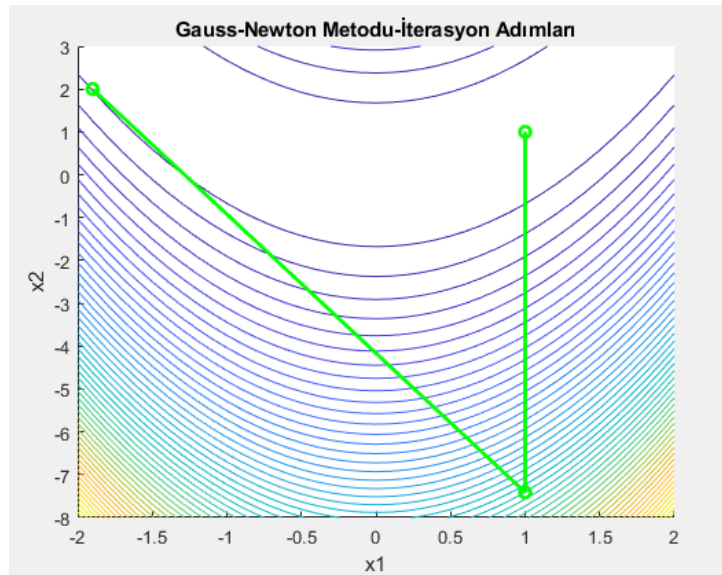


Şekil 6. Rosenbrock fonksiyonunun minimizasyonunda Newton metodu.

Newton metodu Rosenbrock fonksiyonu gibi ikinci türevlerin kullanıldığı fonksiyonlarda, Newton yöntemi genellikle hızlı ve kararlı bir şekilde yakınsar. Ancak Hessian matrisinin hesaplanması ve tersinin alınması maliyetlidir. Hessian matrisinin pozitif tanımlı olmadığı durumlarda ise yöntem başarısız olabilir veya yavaşlayabilir.

Gauss-Newton metodu ile optimizasyon

Rosenbrock fonksiyonunu optimize etmek için Gauss-Newton Metodu iterasyon adımlarını gösteren grafik aşağıdaki gibidir.

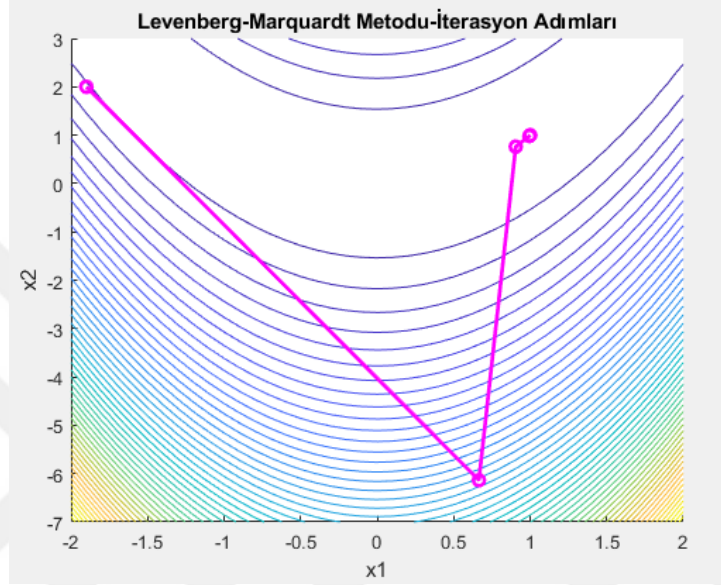


Şekil 7. Rosenbrock fonksiyonunun minimizasyonunda Gauss-Newton metodu.

Gauss-Newton yöntemi nonlineer en küçük kareler problemlerinde etkilidir. Newton yöntemine göre Hessian matrisini yaklaşık olarak kullanır ve bu nedenle hesaplama maliyeti düşüktür. Ancak sadece belirli türdeki problemlere uygundur ve yerel minimumlarda takılabilir.

Levenberg-Marquardt metodu ile optimizasyon

Rosenbrock fonksiyonunu optimize etmek için Levenberg-Marquardt Metodu iterasyon adımlarını gösteren grafik aşağıdaki gibidir.



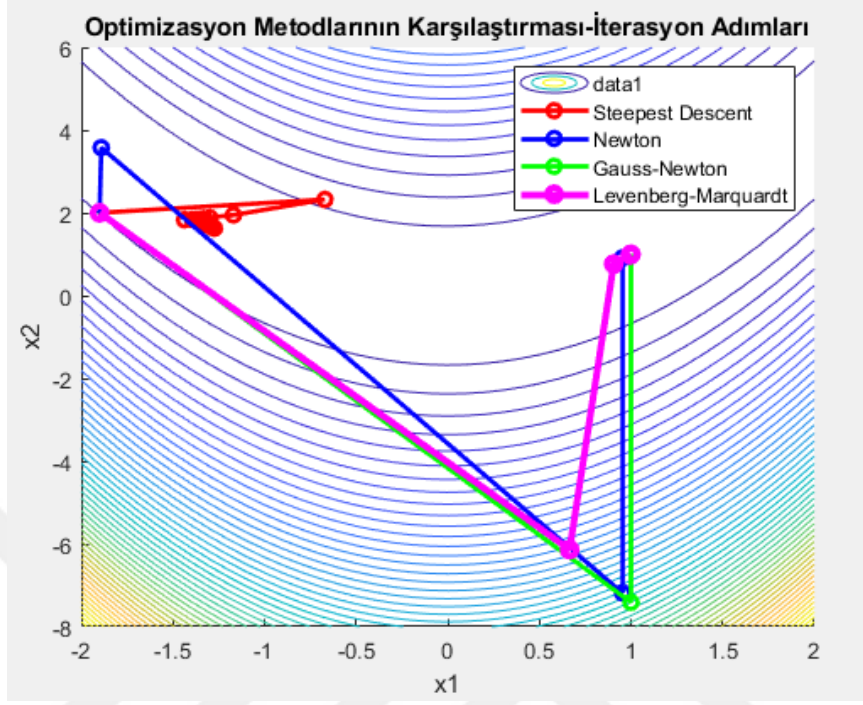
Şekil 8. Rosenbrock fonksiyonunun minimizasyonunda Levenberg-Marquardt metodu.

Levenberg-Marquardt metodu, Gauss-Newton ve Steepest Descent metodlarının bir kombinasyonudur ve genellikle daha iyi sonuç verir.

Levenberg-Marquardt yöntemi ise stabilite ve adaptif yapısı sayesinde yani Hem Gauss-Newton hem de Steepest Descent yöntemleri arasında geçiş yaparak, iterasyonlarda hızlı ve kararlı bir yakınsama sağlar. Hessian matrisine bir damping(sönümlleme) parametresi ekleyerek fonksiyonun eğiminin hızlı değiştiği bölgelerde adım boyutlarını küçülterek algoritmanın kararlı bir şekilde ilerlemesini sağlar. Bu parametre, algoritmanın yerel minimumlara takılma riskini azaltır ve global minimuma ulaşma olasılığını artırır. Daha geniş bir problem sınıfında daha güvenilir ve etkili sonuçlar sağlar. Bu nedenle genellikle daha iyi bir sonuç verir ve tercih edilir. Levenberg-Marquardt yöntemi, özellikle dar ve eğimli vadiler gibi zor optimizasyon problemlerinde daha etkili olabilir.

Optimizasyon metodlarının karşılaştırması

Rosenbrock fonksiyonunu optimize etmek için bu dört metodun iterasyon adımlarını karşılaştırarak gösteren grafik aşağıdaki gibidir.



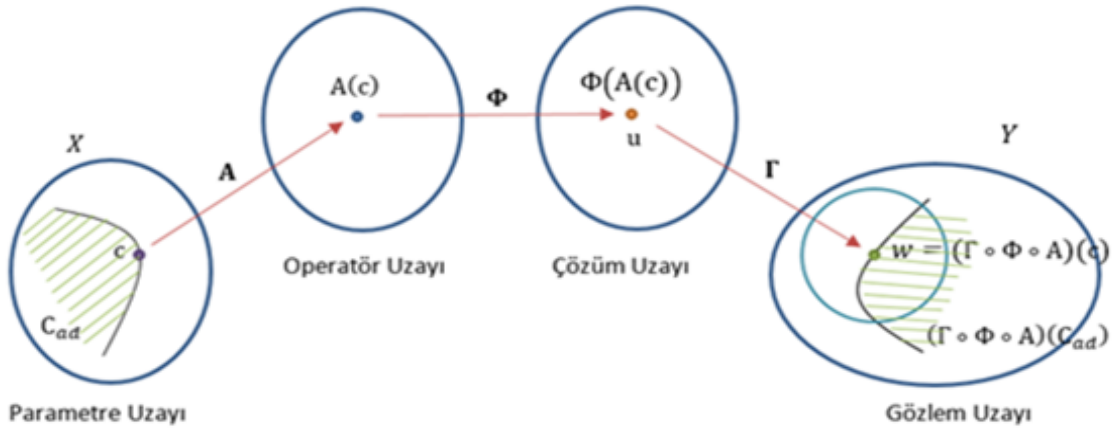
Şekil 9. Rosenbrock fonksiyonunun minimizasyonunda optimizasyon metodlarının karşılaştırılması.

Steepest Descent metodu, genellikle çok sayıda küçük adım atar ve yakınsama yavaştır. Gauss-Newton metodu, Hessian matrisinin yaklaşık formunu kullanarak daha büyük adımlar atabilir. Bu nedenle, iyi bir başlangıç noktasına sahipse hızlıca çözüm noktasına yakınsar. Ancak, başlangıç noktası çözümden uzak veya problem kötü koşullu olduğunda, Newton metodu ve Gauss-Newton metodu kararlılık sorunları yaşayabilir ve yanlış yöne gidebilir. Levenberg-Marquardt metodu, Steepest Descent metodu ve Gauss-Newton metodlarının bir birleşimi olup başlangıç noktası çözümden uzak veya problem kötü koşullu olsa bile yakınsama garantisi daha yüksektir ve zor problemlerde daha güvenilirdir. Grafik sonuçlarına bakıldığında, zor problem olarak bilinen Rosenbrock fonksiyonu için Levenberg-Marquardt metodunun iterasyon adımlarının daha kararlı ve hızlı olduğu görülmektedir. Sonuç olarak, Levenberg-Marquardt metodunun diğerlerine göre daha kararlı ve güvenilir bir yakınsama sağladığını yani daha iyi performans gösterdiğini grafik sonuçlarına bakarak söyleyebiliriz.

ARAŞTIRMA BULGULARI

İki boyutlu bir hiperbolik başlangıç-sınır değer probleminde ters katsayı belirleme problemi için Levenberg-Marquardt optimalleştirme tekniğinin uyarlanmış hali ile katsayı fonksiyonu başarılı olarak elde edilebilmiştir. Nümerik örnekler kullanılarak, elde edilen veriler ve analiz sonuçları görsel ve sayısal olarak gösterilmiştir. Bu örnekler, araştırma bulgularının somut ve anlaşılır bir şekilde sunulmasına katkı sağlamıştır.

Direkt problemin yapısı ve matematiksel ifadesi şu şekildedir.



Şekil 10. Direkt problemin şematik yapısı.

A dönüşümü, $C_{ad} \subset \{c \in L_{\infty}(\Omega) : c(x) > 0\}$ parametre uzayından operatör uzayına

$$A(c(x)) = -\nabla \cdot (c(x)\nabla u) + a(x)u$$

şeklinde tanımlanmış lineer olmayan dönüşüm olsun.

Φ dönüşümü operatör uzayından çözüm uzayına

$$\Phi(A(c(x))) = u$$

şeklinde tanımlanmış dönüşüm olsun.

Γ dönüşümü çözüm uzayından gözlem uzayına,

$$\Gamma(\Phi(A(c(x)))) = w$$

şeklinde u çözümünün final anındaki izi olarak tanımlanmış dönüşüm olsun.

Böylece direkt problem C_{ad} parametre uzayından $(\Gamma \circ \Phi \circ A)(C_{ad})$ gözlem uzayına

$$(\Gamma \circ \Phi \circ A)(c) = w$$

şeklinde tanımlanır.

(1.1)-(1.3) problemi için direkt problemin çözümü, zayıf çözüm yaklaşımı ve sonlu elemanlar yöntemi ile aşağıdaki şekilde elde edilmiştir.

Direkt Problem için Sonlu Elemanlar Yöntemi ve Zayıf Çözüm

Zayıf çözüm

(1.1)-(1.3) başlangıç-sınır değer probleminin (1.4) şartları altında $u_t \in C([0, T], L_2(\Omega))$ ve $u_{tt} \in C([0, T], H^{-1}(\Omega))$ zayıf türevleri ile $u \in C([0, T], H^1(\Omega))$ zayıf bir çözümü vardır (Evans 2002). Ayrıca bu çözüm

$$\|u\|_{L_\infty\{0,T;H^1(\Omega)\}} + \|u_t\|_{L_\infty(0,T;L_2(\Omega))} \leq c(\|\sigma\|_{L_2(0,T)}\|f\|_{L_2(\Omega)} + \|u_0\|_{H^1(\Omega)} + \|v\|_{L_2(\Omega)} + \|g\|_{L_2(0,T)})$$

eşitsizliğini sağlar.

Katsayı fonksiyonları ve diğer fonksiyonların süreklilik şartlarını sağlamadığı durumlarda integral eşitliklerini sağlayan zayıf çözüm bulunmaya çalışılır. Buna göre (1.1)-(1.3) başlangıç sınır değer problemi için zayıf çözüm şu şekilde tanımlanır:

(1.1) denklemini bölgenin sınırında sıfır olan ve bölge üzerinde integrallenebilen bir v test fonksiyonu ile çarpalım ve elde edilen denklemin Ω bölgesi üzerinden integralini alalım.

Her $t \in (0, T]$ için aşağıdaki integral eşitliği elde edilir:

$$\int_{\Omega} \ddot{u}v dx - \int_{\Omega} \nabla \cdot (c\nabla u)v dx + \int_{\Omega} auv dx = \int_{\Omega} f v dx. \quad (4.1)$$

Bu eşitlikte Green özdeşliği kullanılırsa

$$- \int_{\Omega} \nabla \cdot (c\nabla u)v dx = \int_{\Omega} (c\nabla u) \cdot \nabla v dx - \int_{\partial\Omega} (c\nabla u) \cdot \vec{n} v ds$$

olup

$$\int_{\Omega} \ddot{u}v dx + \int_{\Omega} (c\nabla u) \cdot \nabla v dx - \int_{\partial\Omega} (c\nabla u) \cdot \vec{n} v ds + \int_{\Omega} auv dx = \int_{\Omega} f v dx \quad (4.2)$$

elde edilir.

$$H_g^1 = \{v: \|v\|_{L_2(\Omega)} + \|\nabla v\|_{L_2(\Omega)} < \infty, v|_{\partial\Omega} = g\}$$

ve

$$H_0^1 = \{v: \|v\|_{L_2(\Omega)} + \|\nabla v\|_{L_2(\Omega)} < \infty, v|_{\partial\Omega} = 0\}$$

şeklinde tanımlansın.

O halde her $v \in H_0^1$ ve her $t \in (0, T]$ için (4.2) eşitliği

$$\int_{\Omega} \ddot{u}v dx + \int_{\Omega} ((c\nabla u) \cdot \nabla v + auv) dx = \int_{\Omega} f v dx \quad (4.3)$$

olarak yazılır. $\forall v \in H_0^1$ için (4.3) eşitliğini sağlayan $u \in H_g^1$ fonksiyonuna (1.1)-(1.3) probleminin zayıf çözümü denir.

Sonlu elemanlar yöntemi

Sonlu elemanlar yöntemini (4.3) direkt problemine uygulayalım. Ω bölgesi, N_T elemanlarının birleşimi olarak tanımlanır.

$$\Omega \approx \tau^h \equiv \bigcup_{i=1}^{N_T} \tau_i$$

Burada τ_i , i . elemandır. Elemanların birleşimine τ^h denir ve bu Ω bölgesi için sonlu eleman ağıdır. h sayısı elemanların boyutunu gösterir.

Yaklaşık çözüm,

$$u^h(x, y, t) \cong \sum_{j=1}^{N_p} u_j(t) \varphi_j(x, y)$$

formda aranmaktadır. Burada N_p bütün düğüm noktalarının sayısıdır.

$\varphi_j(x, y)$ baz fonksiyonu

$$\varphi_j(x, y) = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2$$

ikinci dereceden Lagrange şekil fonksiyonlarıdır.

φ_j baz fonksiyonunu elde etmek için, düğüm noktalarının köşe noktası olduğu tüm τ_i üçgenleri bulunur ve φ_j baz fonksiyonunu τ_i elemanı üzerinde tanımlayan altı a_j katsayıları da

$$\delta_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

şartını sağlayacak şekilde

$$\varphi_j(x_k, y_k) = \delta_{jk}, \quad j, k = 1, 2, \dots, N_p \quad (4.4)$$

ile belirlenir (Larson and Bengzon 2010). φ_j yalnızca düğüm noktalarının köşe noktası olduğu elemanlarda sıfırdan farklıdır. Elemanların büyük çoğunluğunda φ_j sıfırdır.

$$V_0^h = \{v^h: \|v^h\| + \|\nabla v^h\| < \infty, v^h|_{\partial\Omega} = 0\}$$

ve

$$V_g^h = \{v^h: \|v^h\| + \|\nabla v^h\| < \infty, v^h|_{\partial\Omega} = g\}$$

sonlu elemanlar uzayları olmak üzere $\forall v^h \in V_0^h$ için

$$\int_{\Omega} \ddot{u}^h v^h dx + \int_{\Omega} ((c\nabla u^h) \cdot \nabla v^h + a u^h v^h) dx = \int_{\Omega} f v^h dx \quad (4.5)$$

eşitliğinin $u^h \in V_g^h$ çözümüne problemin sonlu elemanlar çözümü denir.

$u_0^h \in V_0^h$ ve $u_g^h \in V_g^h$ olmak üzere u^h fonksiyonunu

$$u^h = u_0^h + u_g^h \quad (4.6)$$

şeklinde tanımlayalım. Böylece sınırda $u = g$ şartı u_g^h tarafından sağlanır. (4.6) fonksiyonu (4.5) eşitliğinde yerine yazılırsa

$$\begin{aligned} \int_{\Omega} (u_0^h + \ddot{u}_g^h) v^h dx + \int_{\Omega} ((c\nabla(u_0^h + u_g^h)) \cdot \nabla v^h + a(u_0^h + u_g^h) v^h) dx &= \int_{\Omega} f v^h dx \\ \int_{\Omega} \ddot{u}_0^h v^h dx + \int_{\Omega} (c\nabla u_0^h \nabla v^h + a u_0^h v^h) dx & \\ = \int_{\Omega} (f v^h - \ddot{u}_g^h v^h - c\nabla u_g^h \nabla v^h - a u_g^h v^h) dx & \end{aligned} \quad (4.7)$$

bulunur. u_g^h , sınır şartını sağlayacak şekilde seçilmiş bir fonksiyondur. Ω bölgesinin iç düğüm noktalarında sıfır, sınır düğüm noktalarında g fonksiyonuyla aynı değerleri alacak şekilde seçilir.

Böylece sonlu elemanlar ile çözülen problem $\forall v^h \in V_0^h$ için iç düğüm noktalarında (4.7) eşitliğini sağlayan $u_0^h \in V_0^h$ fonksiyonunun bulunması problemine dönüşür.

$$\alpha(u_0^h, v^h) = \int_{\Omega} (c\nabla u_0^h \nabla v^h + a u_0^h v^h) dx, \quad \forall v^h$$

$$\ell(v^h) = \int_{\Omega} (f v^h - \ddot{u}_g^h v^h - c \nabla u_g^h \nabla v^h - a u_g^h v^h) dx, \quad \forall v^h$$

diyelim.

$$\frac{d^2}{dt^2} (u_0^h, v^h)_{L^2} + \alpha (u_0^h, v^h)_{L^2} = \ell(v^h), \quad \forall v^h \in V_0^h, t \in (0, T] \quad (4.8)$$

eşitliği elde edilir.

Başlangıç şartları

$$u^h(x, 0) = u_0^h(x, 0) + u_g^h(x, 0)$$

$$\dot{u}^h(x, 0) = \dot{u}_0^h(x, 0) + \dot{u}_g^h(x, 0)$$

şeklinde olup $u_g^h(x, 0) = \dot{u}_g^h(x, 0) = 0$ olacak şekilde u_g^h seçilirse, u^h fonksiyonu ile u_0^h fonksiyonu aynı başlangıç şartlarına sahip olur.

Konumsal ayrıklaştırma

N_p bütün düğüm noktalarının sayısı ve N_i iç düğüm noktalarının sayısı olsun. $\varphi_j, j = 1, 2, \dots, N_i$ baz fonksiyonları kullanılarak $i = 1, 2, \dots, N_i$ için $v^h = \varphi_i$ olarak (4.8) eşitliği

$$\frac{d^2}{dt^2} (u_0^h, \varphi_i) + \alpha (u_0^h, \varphi_i) = \ell(\varphi_i), \quad i = 1, 2, \dots, N_i \quad (4.9)$$

sistemine dönüşür.

Şimdi u_0^h fonksiyonu için

$$u_0^h \cong \sum_{j=1}^{N_i} u_j(t) \varphi_j(x)$$

yaklaşımını oluşturalım.

Böylece

$$\sum_{j=1}^{N_i} u_j(t) \int_{\Omega} \varphi_j \varphi_i dx + \sum_{j=1}^{N_i} u_j(t) \alpha (\varphi_j, \varphi_i) = \ell(\varphi_i), \quad i = 1, 2, \dots, N_i$$

denklem sistemi ortaya çıkar.

$$M = M_{ij} = \int_{\Omega} \varphi_j \varphi_i dx, \quad i, j = 1, 2, \dots, N_i$$

$$A = A_{ij} = \alpha (\varphi_j, \varphi_i) = \int_{\Omega} (c \nabla \varphi_j \nabla \varphi_i + a \varphi_j \varphi_i) dx$$

ve

$$F = F_i = \int_{\Omega} (f \varphi_i - \ddot{u}_g^h \varphi_i - c \nabla u_g^h \nabla \varphi_i - a u_g^h \varphi_i) dx$$

olmak üzere $N_i \times N_i$ tipinde

$$M \frac{d^2 U}{dt^2} + AU = F(t), \quad t \in (0, T]$$

adi diferansiyel denklem sistemi ortaya çıkar. Burada U vektörü

$$U = \begin{bmatrix} U_1(t) \\ \vdots \\ U_j(t) \\ \vdots \\ U_{N_i}(t) \end{bmatrix} = \begin{bmatrix} u(P_1, t) \\ \vdots \\ u(P_j, t) \\ \vdots \\ u(P_{N_i}, t) \end{bmatrix}$$

şeklinde $j = 1, 2, \dots, N_i$ düğüm noktalarının t anındaki değerleridir.

$$U(0) = U_0(P_j) = \begin{bmatrix} u_0(P_1) \\ \vdots \\ u_0(P_{N_i}) \end{bmatrix}$$

$$\frac{d}{dt} U(0) = V_0(P_j) = \begin{bmatrix} v_0(P_1) \\ \vdots \\ v_0(P_{N_i}) \end{bmatrix}$$

başlangıç şartlarını temsil eden vektörlerdir.

Böylece ikinci mertebeden

$$M \frac{d^2 U}{dt^2} + AU = F(t)$$

$$U(0) = U_0$$

$$\frac{d}{dt} U(0) = V_0$$

adi diferansiyel denklem sistemi için başlangıç değer problemi ortaya çıkar.

Zaman ayrıklaştırma

Zaman içinde ayrıklaştırma için; $(0, T]$ zaman aralığını

$$0 = t_0 < t_1 < \dots < t_L = T$$

olmak üzere

$$k_l = t_l - t_{l-1}, \quad l = 1, 2, \dots, L$$

adımına sahip L tane aralığa bölelim.

$\dot{U} = Y$ diyelim. Böylece 2. mertebeden olan ilk denklem sistemi

$$M\dot{U} = MY$$

$$M\dot{Y} + AY = F(t)$$

birinci mertebeden adi diferansiyel denklem sistemine dönüşür.

Şimdi Crank-Nicolson yöntemini bu iki sistemin her birine uygularsak,

$$M \frac{U_l - U_{l-1}}{k_l} = M \frac{Y_l + Y_{l-1}}{2}$$
$$M \frac{Y_l - Y_{l-1}}{k_l} + A \frac{U_l + U_{l-1}}{2} = \frac{F_l + F_{l-1}}{2}$$

denklemlerinden

$$\begin{bmatrix} M & -\frac{k_l}{2}M \\ \frac{k_l}{2}A & M \end{bmatrix} \begin{bmatrix} U_l \\ Y_l \end{bmatrix} = \begin{bmatrix} M & \frac{k_l}{2}M \\ -\frac{k_l}{2}A & M \end{bmatrix} \begin{bmatrix} U_{l-1} \\ Y_{l-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ \frac{k_l}{2}(F_l + F_{l-1}) \end{bmatrix} \quad (4.10)$$

sistemi yazılır. Bu sistemin U_0 ve Y_0 başlangıç elemanları problemin U_0 ve V_0 başlangıç şartlarıdır. $(0, T]$ zaman aralığında (4.10) sistemi çözülür.

Crank-Nicolson, ileri Euler ve geriye doğru Euler yönteminin her birinin doğruluk, kararlılık ve hesaplama maliyeti ile ilgili kendine has özellikleri vardır. Basitçe söylemek gerekirse, ileri Euler çok hızlı, geriye doğru Euler sayısal olarak kararlı ve Crank-Nicolson en doğru olanıdır. Yani; Crank-Nicolson yöntemi, Euler yöntemine (ileri-geri) göre daha doğru sonuçlar verir. Ayrıca enerji koruma özelliğine sahiptir. Bu nedenle dalga denklemi için uygun bir yöntemdir.

Dalga denklemini çözmek için Crank-Nicolson yöntemini aşağıdaki algoritma ile özetleyebiliriz:

➤ **Algoritma: Dalga Denklemi için Crank-Nicolson Yöntemi**

1. Ω bölgesinin bir N_T üçgenlemesi oluşturun, parçalı sürekli lineer fonksiyonlara karşılık gelen V_0^h uzayını ve $\{\varphi_i\}_{i=1}^{N_i}$ baz fonksiyonunu tanımlayın.
2. $0 = t_0 < t_1 < \dots < t_L = T$, $I = (0, T]$ aralığında, L zaman adımları $k_l = t_l - t_{l-1}$ ile bir zaman ızgarası oluşturun.
3. U_0 ve Y_0 seçin.
4. **for** $l = 1, 2, \dots, L$ **do**
5. $N_i \times N_i$ kütle ve sertlik matrisleri M ve A ile $N_i \times 1$ yük vektörü $F(t)$ yi

$$M_{ij} = \int_0^1 \varphi_j \varphi_i dx, \quad A_{ij} = \int_0^1 (c \nabla \varphi_j \nabla \varphi_i + a \varphi_j \varphi_i) dx,$$

$$(F_l)_i = \int_0^1 (f(t_l) \varphi_i - \ddot{u}_g^h \varphi_i - c \nabla u_g^h \nabla \varphi_i - a u_g^h \varphi_i) dx$$

girişleriyle hesaplayın.

$$6. \quad \begin{bmatrix} M & \frac{-k_l}{2} M \\ \frac{k_l}{2} A & M \end{bmatrix} \begin{bmatrix} U_l \\ Y_l \end{bmatrix} = \begin{bmatrix} M & \frac{k_l}{2} M \\ \frac{-k_l}{2} A & M \end{bmatrix} \begin{bmatrix} U_{l-1} \\ Y_{l-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ \frac{k_l}{2} (F_l + F_{l-1}) \end{bmatrix}$$

lineer deklemler sistemini çözün.

7. end for

Şimdi aşağıdaki örneği ele alalım;

Örnek:

$x \in \Omega \subset \mathbb{R}^2$ bölgesindeki konum değişkenleri ve $t \in (0, 1]$ aralığındaki zaman değişkeni dikkate alındığında, $Q_T := (x, t) \in \Omega \times (0, 1]$ bölgesindeki hiperbolik denklemi yazalım:

$$\frac{\partial^2 u}{\partial t^2} - \nabla \cdot ((2 + \cos \pi x \cos \pi y) \nabla u) + (-4\pi^2 - 1 - 4\pi^2 \cos \pi x \cos \pi y) u = 0, \quad (x, t) \in Q_T$$

$$u(x, 0) = u_0(x) = \sin \pi x \sin \pi y, \quad u_t(x, 0) = v_0(x) = -\sin \pi x \sin \pi y, \quad x \in \Omega$$

$$u(x, t) = 0, \quad x \in \partial\Omega, \quad t \in (0, 1]$$

şeklindedir. Burada,

$$\begin{aligned} m &= 1 \\ d &= 0 \\ f &= 0 \end{aligned}$$

$$c(x) = 2 + \cos \pi x \cos \pi y$$

$$a(x) = -4\pi^2 - 1 - 4\pi^2 \cos \pi x \cos \pi y$$

$$u(x, 0) = \sin \pi x \sin \pi y$$

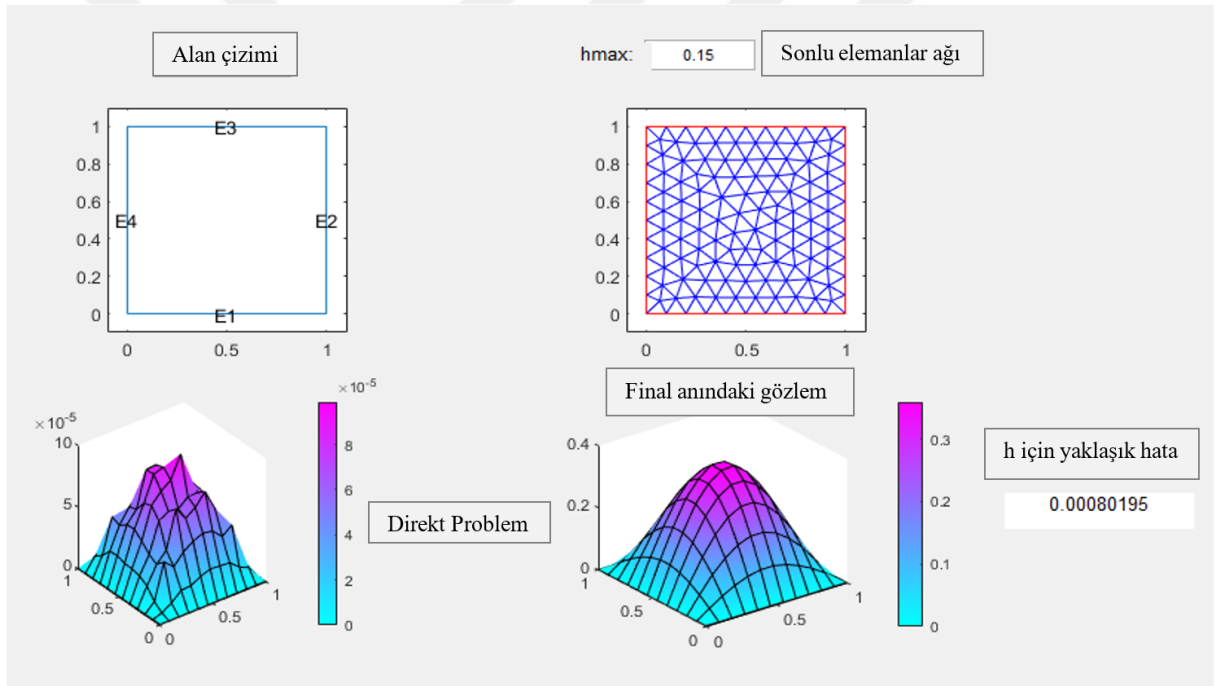
$$u_t(x, 0) = -\sin \pi x \sin \pi y$$

şeklindedir. Bu denklemin gerçek çözümünün

$$u_{ger} = e^{-1} \sin \pi x \sin \pi y$$

olduğu biliniyor. O halde şimdi çözüm tekniğimizi kullanarak direkt problemin nümerik çözümünü MATLAB programı ile bulalım. Daha sonra bulduğumuz çözümün gerçek çözüme ne kadar yaklaştığını test edelim.

Aşağıdaki MATLAB arayüz ekranında, hazırlanan simülasyon ile direkt problemin çözümü incelenebilir.



Şekil 11. Direkt problemin çözümü için hazırlanan simülasyonu gösteren MATLAB arayüz ekranı.

Matlab arayüz ekranı incelendiğinde, çözüm tekniğimiz kullanılarak $h_{max} = 0,15$ için nümerik çözümün gerçek çözüme minimum bir hata ile yaklaştığı görülmektedir. Sonuç olarak, iki boyutlu bir hiperbolik denklem için sonlu elemanlar yöntemi gerçek çözüme oldukça yakın değerler üretmiştir.

Bu arayüz ekranının Matlab kodu EK-1 ile verilmiştir.

Sonraki aşamada düzgünleştirilmiş şekliyle ortaya konulan ters problem için bir optimalleştirme tekniği belirlenmiştir. Bu teknikle gürültü $w \in Y$ verildiğinde,

$$F(c) = \|(\Gamma \circ \Phi \circ A)(c) - w\|^2$$

fonksiyoneli minimum yapacak $c(x)$ elemanı araştırılmıştır.

Ters Katsayı Belirleme Problemi

Deneylerden veya ölçümlerden alınan w gözlemi genellikle bir miktar hata ya da gürültü içerir. Bu yüzden ters problem şu şekilde tanımlanmalıdır.

$$\|\varepsilon\| \leq \delta \text{ ile } w = (\Gamma \circ \Phi \circ A)(c) + \varepsilon \text{ verildiğinde } c(x) \text{ parametresini belirleme} \quad (4.11)$$

Ters problemlerin genellikle başlangıç verilerine göre sürekli olma şartını sağlayan nümerik çözümleri elde edilemez. Yapılan gözlemler gürültü içerebileceğinden $w \notin (\Gamma \circ \Phi \circ A)(C_{ad})$ durumunda ters problemin çözümünün varlığı bile mümkün değildir. Bu durumlar bilindiğinden ters problemlerde kararlı optimal çözümler bazı özel tekniklerle aranır.

Kullanılan metotlar, $w \in (\Gamma \circ \Phi \circ A)(C_{ad})$ ise w elemanının tersini, $w \notin (\Gamma \circ \Phi \circ A)(C_{ad})$ ise w gözleminin $(\Gamma \circ \Phi \circ A)(C_{ad})$ kümesine izdüşümü olan $g = P(w)$ elemanı için bu elemanın tersini kararlı bir şekilde(başlangıç verilerine göre sürekli olma şartı ile) elde etmeye dayanır. Bu metotlar düzgünleştirme (regülerleştirme) stratejilerine dayanır.

Genel olarak her $c \in X$ için

$$\lim_{\alpha \rightarrow 0} R_\alpha (\Gamma \circ \Phi \circ A)(c) = c$$

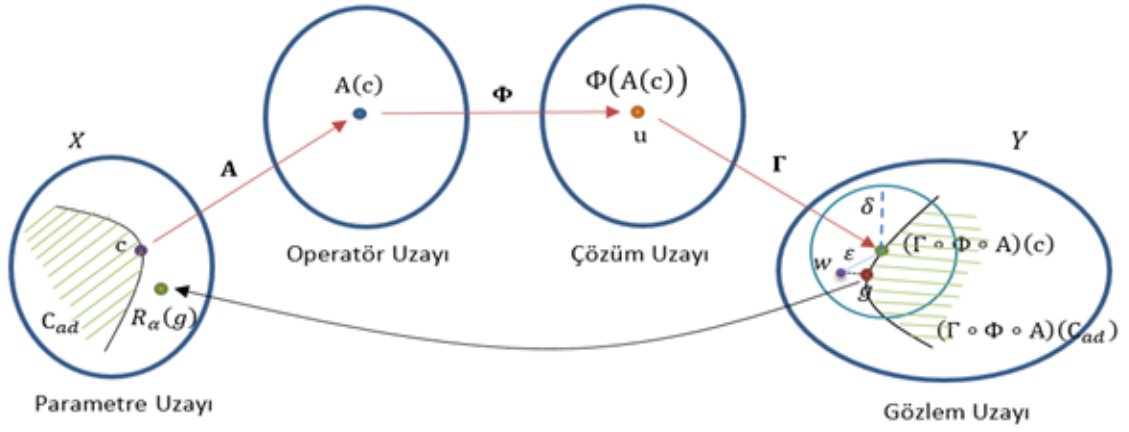
ise, o zaman $0 < \alpha < \infty$ olmak üzere $R_\alpha: Y \rightarrow X$ sürekli dönüşümüne bir düzgünleştirme stratejisi ve α parametresine de düzgünleştirme parametresi denir.

Ele aldığımız problem için $\delta > 0$ gürültü seviyesini göstermek üzere eğer $\delta \rightarrow 0$ iken $\alpha(\delta) \rightarrow 0$ ise ve her $c \in X$ için $\delta \rightarrow 0$ iken

$$\sup_g \{\|R_\alpha(g) - c\|\} \rightarrow 0$$

şartlarını sağlayan bir $\alpha = \alpha(\delta)$ seçimi düzgünleştirme parametresi olarak adlandırılabilir.

Ters problemin yapısı ve matematiksel ifadesi de şu şekildedir.



Şekil 12. Ters problemin şematik yapısı.

(1.1)-(1.4) ile ilişkili ters problemi aşağıdaki gibi ele alıyoruz:

Giriş fonksiyonlarının $a(x)$, $f(x, t)$, başlangıç fonksiyonlarının ve sınır fonksiyonunun (1.4)' teki koşullar altında bilineceğini varsayarsak ve $w = u(x, T)$ için son bir zaman gözlemi verildiğinde, $c(x)$ katsayı fonksiyonunu belirleme.

Bu doğrusal olmayan ters problemi aşağıdaki minimizasyon problemi ile çözüyoruz:

Gürültü $w \in Y$ verildiğinde,

$$F(c) = \|(\Gamma \circ \Phi \circ A)(c) - w\|^2 \quad (4.12)$$

fonksiyonunu minimum yapacak $c(x)$ elemanını bulma.

Ters katsayı belirleme probleminin çözümü

(4.12) ile verilen problemde, final anı gözleminden bilinmeyen katsayı fonksiyonunu elde etmek amaçlanmaktadır. Bundan dolayı

$$c^* = \underset{c}{\operatorname{argmin}} F(c(x)) \quad (4.13)$$

problemi için lineer olmayan en küçük kareler optimizasyonunu kullanacağız. Burada

$$F(c(x)) = \|u(x, T; c(x)) - w(x)\|^2 = \|r(c)\|^2 = \sum_{j=1}^{N_p} [u^h(P_j, T; c(P_j)) - w(P_j)]^2$$

ve $r(c)$ fonksiyonunun bileşenleri $r_j(c) = u^h(P_j, T; c(P_j)) - w(P_j)$, $j = 1, 2, \dots, N_p$ şeklinde tanımlanır.

Verilen başlangıç-sınır değer probleminde uyumluluk koşulları dikkate alınarak, ters problem ile belirlenmek istenen $c(x)$ fonksiyonu, içerisinde $c = \{c_1, c_2, \dots, c_m\}$ parametreleri olacak şekilde (4.13) şartı altında tahmin edilir. Bu yaklaşım, diferansiyel denklemler

teorisindeki belirsiz katsayılar metodunun veya yapay sinir ağları konusundaki ağırlık parametrelerinin kullanımı mantığı ile benzerdir.

Herhangi bir c noktası etrafında F fonksiyonelinin 2. derecesi ile Taylor yaklaşımını yazalım:

$$F(c + \Delta c) \cong L(\Delta c) = F(c) + (\Delta c)^T g + \frac{1}{2} (\Delta c)^T B (\Delta c).$$

Burada $L(\Delta c)$, $F(c)$ fonksiyonelinin c noktası etrafındaki ikinci dereceden yaklaşımıdır. g , c noktasında hesaplanan $F(c)$ fonksiyonelinin gradyanı, B , c noktasındaki $F(c)$ fonksiyonelinin H, Hessian, matrisinin yaklaşımıdır ve H , c noktasındaki $F(c)$ fonksiyonelinin gerçek Hessian matrisidir. r_i fonksiyonunun c_j noktasına göre kısmi türevi (Jacobian)

$$J(c) = [J_{ij}]_{N_p \times m} = \frac{\partial r_i}{\partial c_j}(c)$$

şeklindedir. Dolayısı ile, c noktasındaki gradyan $g(c) = J(c)^T r(c)$, c noktasındaki Hessian $H(c) = J(c)^T J(c) + \sum_{j=1}^{N_p} r_j(c) r_j''(c)$ ve c noktasındaki Hessian yaklaşımı $B(c) = J(c)^T J(c)$ şeklindedir.

2-norm ile $\|c\| = \sqrt{c_1^2 + \dots + c_m^2}$ olmak üzere, eğer c^* noktası local minimize eden nokta ise, o zaman $g(c^*) = 0$ (local minimize nokta için gerekli koşul) ve bazı $\delta > 0$ için $(\Delta c)^T B(c^*) (\Delta c) > \delta \|\Delta c\|^2$ (pozitif-kesinlik- local minimize nokta için yeterli koşul) olur. Bu nedenle, her zaman pozitif tanımlı B matrisini oluşturmalıyız.

Steepest descent metodundaki (en dik iniş metodu) Δc adım uzunluğu; $\Delta c = h_{sd} = -J(c)^T r(c)$ eşitliğiyle, Gauss-Newton metodundaki Δc adım uzunluğu ise $\Delta c = h_{gn} = -\left(J(c)^T J(c)\right)^{-1} J(c)^T r(c)$ eşitliğiyle hesaplanır. Steepest descent method, en dik iniş yöntemi olarak da bilinir, doğrusal olarak yerel minimuma yaklaşır. Bu nedenle genellikle yakınsama hızı yavaştır. Steepest Descent metodu, yavaş yakınsama ve bazı durumlarda zigzag hareketiyle karşılaşma gibi dezavantajlara sahiptir. Gauss-Newton yönteminde h_{gn} vektörü teorik olarak azalma yönünü gösterir, ancak bu yöntem, ill-posed (kötü tanımlanmış) problemlerde veya $J(c)$ Jakobiyen matrisinin rank yetersizliği durumlarında yakınsamayabilir

Bu sorun, adım uzunluğu $\Delta c = h_{lm} = -(J(c)^T J(c) + \lambda I)^{-1} J(c)^T r(c)$ olan Levenberg-Marquardt metodu (Levenberg 1944 ve Marquardt 1963) kullanılarak çözülebilir. λ parametresi Δc adımının hem yönünü hem de uzunluğunu etkiler. Fonksiyonel değerin azaldığı her iterasyon adımında, yön doğrudur ve yakınsama, λ değerini azaltarak $h_{lm} \cong h_{gn}$ ile hızlanır.

Fonksiyonel değerin arttığı iterasyon adımlarında ise, yön yanlıştır ve λ değeri arttırılarak $h_{lm} \cong h_{sd}$ ile steepest descent (en dik iniş) yönüne doğru döndürülür.

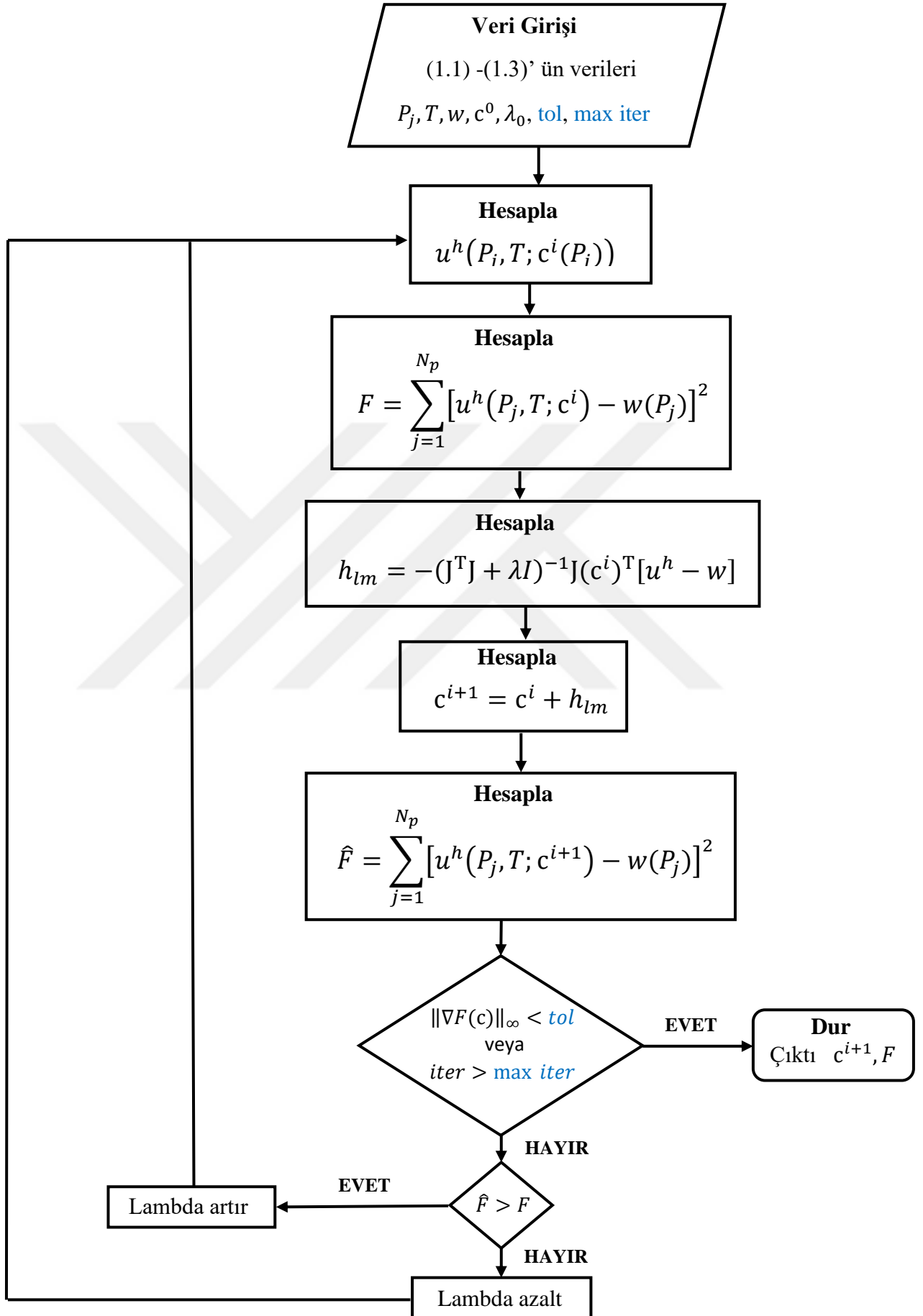
Bazı durumlarda, Hessian matrisinin kararsız veya tekillik eğiliminde olması nedeniyle optimizasyon sürecinde zorluklar yaşanabilir. Levenberg-Marquardt yönteminde, doğrusal olmayan bir problemde Hessian matrisini ve çözümü kararlı hale getirmek için şu düzgünleştirici terim kullanılır:

$$H = J^T J + \lambda I.$$

Burada J Jacobi (fonksiyon türevlerinin matrisi) matrisidir, λ damping(sönümlenme) parametresi ve I birim matristir. Bu ek terim, Hessian matrisinin tekillik durumunu engeller ve pozitif kesin hale getirir, bu da çözümün kararlılığını artırır.

İncelenen problem için Levenberg-Marquardt metodu akış şeması aşağıdaki gibidir.





Nümerik Örnekler

Şimdi, bu süreci aşağıdaki problemler üzerinden test ederek açıklayalım. Problemlerin çözümünde MATLAB2019 yazılımı kullanılmıştır (MathWorks 2019).

Örnek 1:

$\Omega = (0,1) \times (0,1)$ alanını kaplayan materyali düşünelim.

$t \in (0, 1]$ zaman aralığında,

$$\frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c(x)\nabla u) + (-4\pi^2 \cos \pi x \cos \pi y - 4\pi^2 - 1)u = 0, \quad (x, t) \in Q_1 \quad (4.14)$$

$$u(x, 0) = \sin \pi x \sin \pi y, \quad u_t(x, 0) = -\sin \pi x \sin \pi y, \quad x \in \Omega \quad (4.15)$$

$$u(x, t) = 0, \quad t \in (0, 1] \quad (4.16)$$

problemini ele alalım.

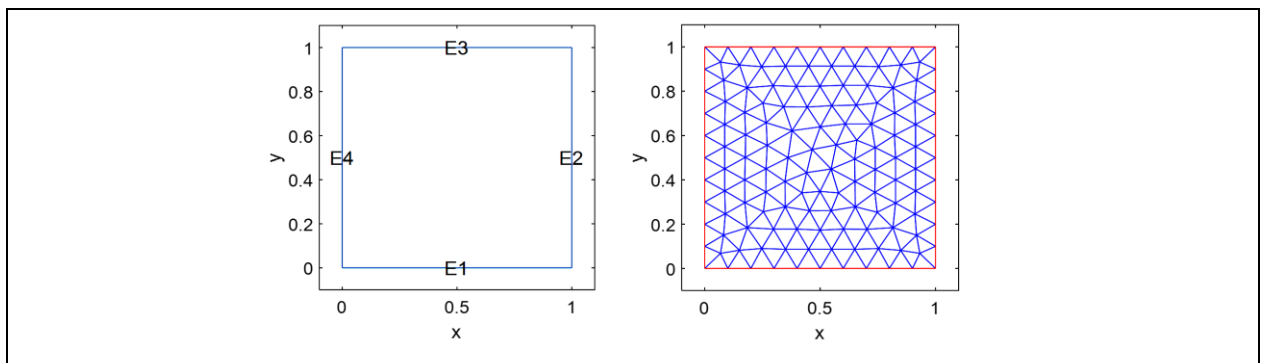
$$w(x) = u(x, 1) = e^{-1} \sin \pi x \sin \pi y \quad (4.17)$$

final anı konum bilgisinden $c(x)$ katsayı fonksiyonunun belirlenmesi isteniyor.

Gerçek katsayı fonksiyonu; $c(x) = 2 + \cos \pi x \cos \pi y$ şeklindedir.

İlk olarak $c(x) \rightarrow u(x, 1)$ direkt probleminin sonlu elemanlar yöntemi ile çözümünü hesaplayalım.

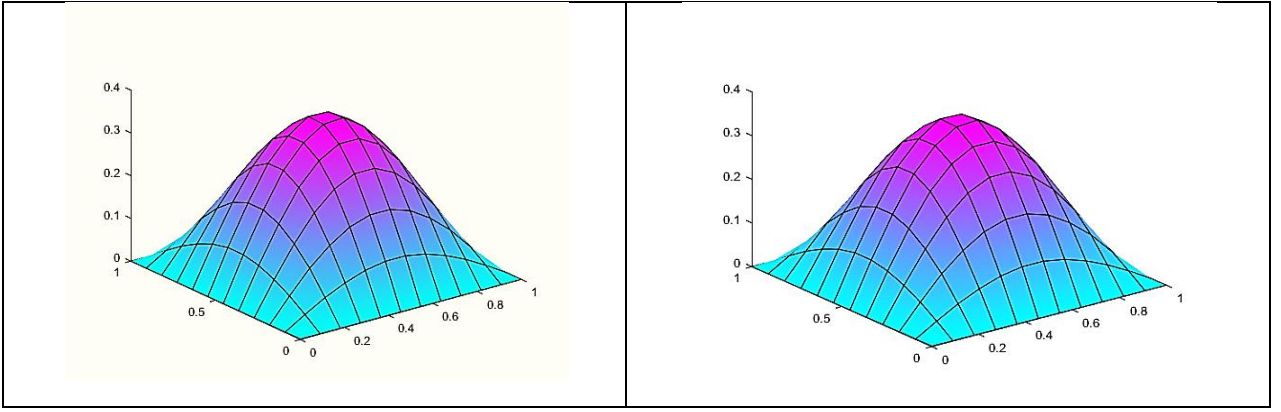
Problem için tanım kümesi ve sonlu eleman ağı şekli aşağıda verilmiştir;



Şekil 13. Örnek 1 için tanım kümesi ve $h = 0,1$ için sonlu eleman ağı.

Ağ boyutu $h = 0,1$ için eleman ve düğüm sayısı sırasıyla 228 ve 497'dir.

$u(x, 1)$ gerçek çözümü ve $u^h(x, 1)$ sonlu elemanlar yöntemi ile hesaplanan çözümü aşağıdaki şekillerde gösterilmiştir;



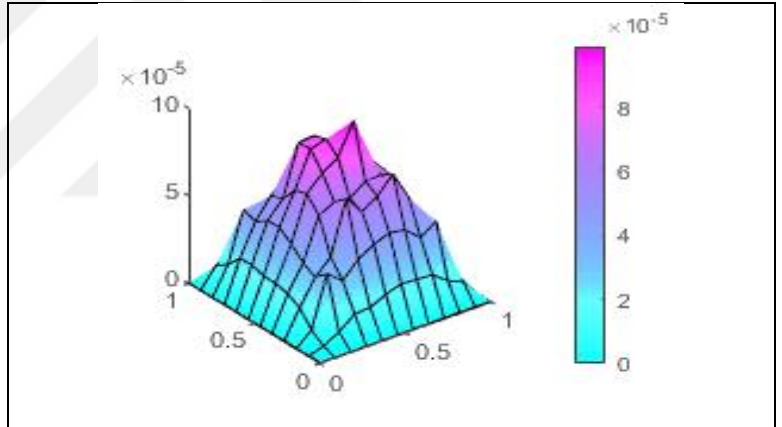
Şekil 14. Örnek 1 için $T = 1$ anında gerçek dalga fonksiyonu.

Şekil 15. Örnek 1 için $T = 1$ anında hesaplanan dalga fonksiyonu.

Farklı h maksimum ağ boyutuna karşılık gelen bazı hata norm değerleri Tablo 1'de verilmiştir. Kötü koşullama nedeniyle, Tablo 1'de $h = 0,1$ ve $h = 0,08$ deki değerlere bakılırsa daha küçük bir h değerinin daha büyük bir hata normu verdiği görülebilir. Dolayısıyla bu örnek için $h = 0,1$ ağ boyutunu alıyoruz. Şekil 16, bu ağ boyutu için hesaplama hatasını göstermektedir.

Tablo 1. Örnek 1 için Bazı h ve Karşılık Gelen Hata Norm Değerleri

h	$\ u(x, 1) - u^h(x, 1)\ $
0,2	$3,9822 \times 10^{-2}$
0,15	$1,2618 \times 10^{-2}$
0,1	$8,0195 \times 10^{-4}$
0,08	$5,482 \times 10^{-3}$
0,05	$1,744 \times 10^{-3}$



Şekil 16. Örnek 1 için $h = 0,1$ değerine karşılık hata değerleri.

İkinci olarak, doğrusal olmayan en küçük kareler optimizasyon teknikleri ile $w(x) \rightarrow c(x)$ ters tanımlama probleminin çözümüyle ilgileniyoruz.

Verilen problemin verileri ve final anı gözlem fonksiyonu kullanılarak katsayı fonksiyonu en uygun şekilde tahmin edilmelidir.

(4.14) ve (4.15) de verilen fonksiyonların yapısını göz önünde bulundurarak, $c(x)$ fonksiyonunu aşağıdaki formlardan biri olarak düşünebiliriz;

Tablo 2. Örnek 1 için Bazı Aday Parametre Fonksiyonları

P1	$c(x) = c$
P2	$c(x) = c_1 + c_2 \sin \pi x \sin \pi y$
P3	$c(x) = c_1 + c_2 \cos \pi x \cos \pi y$
P4	$c(x) = c_1 + c_2 \cos \pi x \cos \pi y + c_3 \sin \pi x \sin \pi y$

Bu formlardan en uygun olanı,

$$F(c) = \operatorname{argmin}_{c_i} \left\| u^{0,1}(P_j, 1; c(P_j)) - w(P_j) \right\|^2 \quad (4.18)$$

fonksiyonel değerini diğerlerinden daha küçük yapandır.

Bu fonksiyonelin minimizasyonu, $c^0 = [1, \dots, 0]$ başlangıç parametrelerinden başlayarak MATLAB kullanılarak Levenberg-Marquardt metoduyla gerçekleştirilirse, aşağıdaki sonuçlar elde edilir:

Tablo 3. Örnek 1 için Bazı Aday Parametre Fonksiyonları için Elde Edilen Sonuçlar

P1	$c(x) = 2,04135$	$F(c) = 2,08986 \times 10^{-1}$
P2	$c(x) = 1,52701 + 1,69518 \sin \pi x \sin \pi y$	$F(c) = 6,50539 \times 10^{-2}$
P3	$c(x) = 2,00001 + 0,999825 \cos \pi x \cos \pi y$	$F(c) = 3,01324 \times 10^{-8}$
P4	$c(x) = 2,00925 + 1,00784 \cos \pi x \cos \pi y - 2,55286 \times 10^{-2} \sin \pi x \sin \pi y$	$F(c) = 2,52699 \times 10^{-6}$

Bu çıktılar, EK-2' de verilen Matlab arayüzünün ekranlarından görülebilmektedir.

P3'te verilen fonksiyon, en küçük fonksiyonel değeri verdiği için (4.18) probleminin çözümü olarak alınabilir. Yani optimal çözüm;

$$c^*(x) = 2,00001 + 0,999825 \cos \pi x \cos \pi y$$

olur. Böylece, parametre fonksiyonunun Levenberg-Marquardt metoduyla elde edilebileceği ortaya çıkmıştır.

P3 probleminin hesaplama adımları aşağıdaki tabloda sunulmuştur;

Tablo 4. Örnek 1 için P3 Probleminde İterasyon Sonuçları

Iteration	Func-count	Residual	First-Order optimality	Lambda	Norm of step
0	5	217312	5.22e+05	0.01	
1	10	21846.7	7.2e+04	0.001	0.640571
2	15	1895.73	9.79e+03	0.0001	0.361556
3	20	123.598	1.38e+03	1e-05	0.261183
4	25	4.67058	195	1e-06	0.335156
5	30	0.670861	67.8	1e-07	0.437192
6	35	0.212061	37.2	1e-08	0.354979
7	40	0.115324	28.1	1e-09	0.312886
8	45	0.000559484	1.94	1e-10	0.0644938
9	50	2.50466e-07	0.039	1e-11	0.00485304
10	55	3.01324e-08	1.29e-08	1e-12	5.07699e-05

Bu hesaplamalarda, λ 'nin başlangıç değeri $\lambda_0 = 0,01$ olarak seçilmiş ve yinelemelerde bu parametre $\hat{F} < F$ iken $\lambda = \lambda/10$ azaltılmış ve $\hat{F} > F$ iken $\lambda = 10\lambda$ arttırılmıştır.

Ayrıca Levenberg-Marquardt algoritmasında,

$$\|\nabla F(c)\|_{\infty} \leq 1e - 10$$

durdurma kriteri kullanılmıştır. Hesaplama adımlarında $\|\nabla F(c)\|_{\infty} = 2.476602e - 14$ olarak bulunan değer durdurma kriterinden küçük olduğu için, algoritma 10. iterasyondan sonra durmuştur.

P3 tarafından verilen seçim kabul edildikten sonra, veriler üzerinde optimal çözümün sürekliliğini test etmek gerekmektedir. Bu süreçte istenilen bağıl hata payıyla rastgele gürültü eklenir ve ilgili çözümler elde edilir.

Aşağıdaki tabloda bazı gürültülü gözlemler ve bunlara karşılık gelen optimal çözümler yer almaktadır.

Tablo 5. Örnek 1 için P3 Problemindeki Gürültülere Karşılık Gelen Çözümlerin Bağıl Değişimi

Gürültü Yüzdesi	$\frac{\ w - w_{\text{gürültü}}\ }{\ w\ }$	$c^*(x; w_{\text{gürültü}})$	$\frac{\ c^*(x; w) - c^*(x; w_{\text{gürültü}})\ }{\ c^*(x; w)\ }$
%10	0,1	[2,00003 , 0,980146]	$5,10073 \times 10^{-3}$
%4	0,04	[2,00008 , 0,995556]	$1,10699 \times 10^{-3}$
%1	0,01	[1,99995 , 0,997524]	$5,97144 \times 10^{-4}$
%0,5	0,005	[2,00001 , 0,998932]	$2,31463 \times 10^{-4}$
%0,1	0,001	[2 , 0,99946]	$9,47348 \times 10^{-5}$
%0	0	[2,00001 , 0,999825]	0

Bu sonuçlar, EK-3'de verilen Matlab arayüzünün ekranlarından görülebilmektedir. Ayrıca bu arayüzün Matlab kodu da EK-4 ile verilmiştir.

Gürültü seviyesi $\delta \rightarrow 0$ iken

$$\sup_w \left\{ \frac{\|c^*(x; w) - c^*(x; w_{\text{gürültü}})\|}{\|c^*(x; w)\|} \right\} \rightarrow 0$$

olduğu için tablodaki Levenberg-Marquardt yöntemiyle elde edilen çözümün düzgünleştirilmiş çözüm olduğu görülebilir.

Örnek 2:

$y = \mp 1 \sqrt{2} x$ ve $y = \mp 1$ doğrularıyla sınırlanan alanı kaplayan malzemeyi düşünelim. (Subaşı et al. 2021)

$t \in (0, 2]$ zaman aralığında,

$$\frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c(x)\nabla u) + 2u = e^{\frac{t}{2}}(0,25x^2 + 5,5y^2), \quad (x, t) \in Q_2 \quad (4.19)$$

$$u(x, 0) = x^2 - 2y^2, \quad u_t(x, 0) = \frac{1}{2}(x^2 - 2y^2), \quad x \in \Omega \quad (4.20)$$

$$u(x, t) = \begin{cases} 0 & \text{on } y = \mp \frac{1}{\sqrt{2}}x, \quad t \in (0, 2] \\ e^{\frac{t}{2}}(x^2 - 2) & \text{on } y = \mp 1 \end{cases} \quad (4.21)$$

problemini ele alalım.

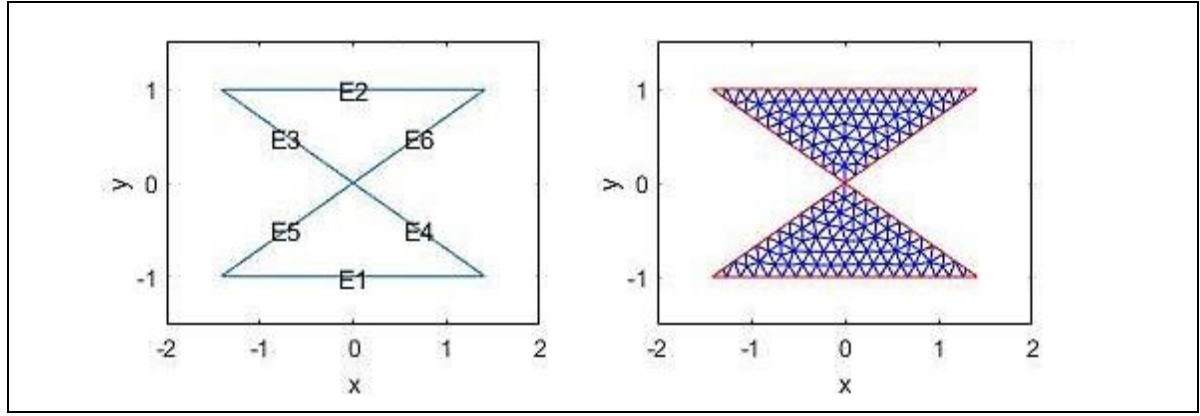
$$w(x) = u(x, 2) = e^1(x^2 - 2y^2) \quad (4.22)$$

final anı konum bilgisinden $c(x)$ katsayı fonksiyonunun belirlenmesi isteniyor.

Gerçek katsayı fonksiyonu; $c(x) = x^2 + y^2$ fonksiyonudur.

İlk olarak $c(x) \rightarrow u(x, 2)$ direkt probleminin sonlu elemanlar yöntemi ile çözümünü hesaplayalım.

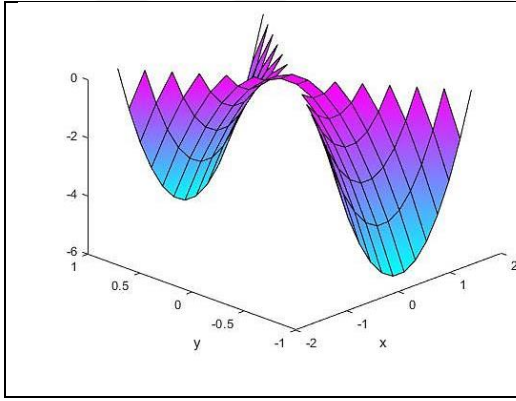
Problem için tanım kümesi ve sonlu eleman ağı şekli aşağıda verilmiştir;



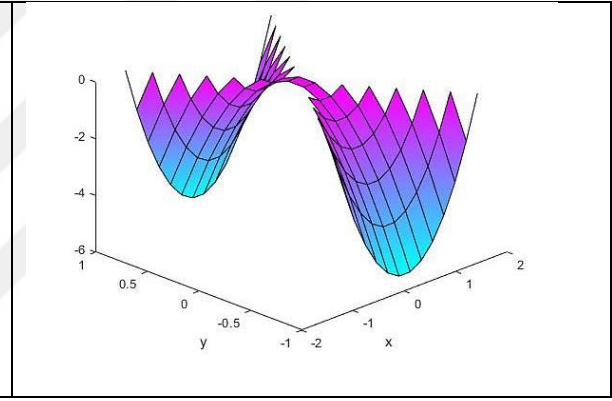
Şekil 17. Örnek 2 için tanım kümesi ve $h = 0,15$ için sonlu eleman ağı.

Ağ boyutu $h = 0,15$ için eleman ve düğüm sayısı sırasıyla 292 ve 671' dir.

$u(x, 2)$ gerçek çözümü ve $u^h(x, 2)$ sonlu elemanlar yöntemi ile hesaplanan çözümü aşağıdaki şekillerde gösterilmiştir;



Şekil 18. Örnek 2 için $T = 2$ anında gerçek dalga fonksiyonu.

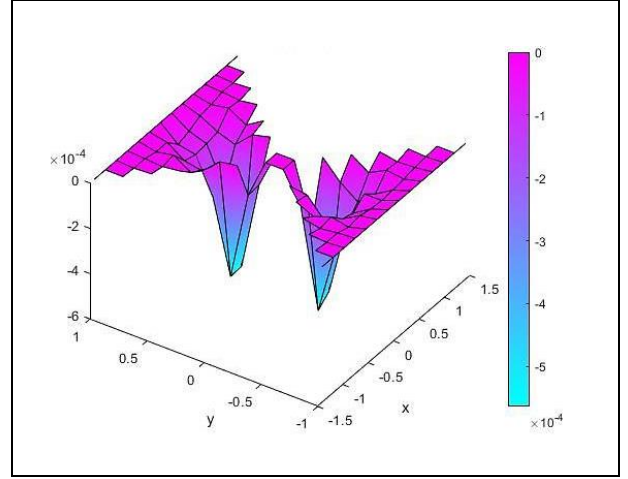


Şekil 19. Örnek 2 için $T = 2$ anında hesaplanan dalga fonksiyonu.

Farklı h maksimum ağ boyutuna karşılık gelen bazı hata norm değerleri Tablo 6' da verilmiştir. Kötü koşullama nedeniyle, Tablo 6' da $h = 0,15$ ve $h = 0,1$ deki değerlere bakılırsa daha küçük bir h değerinin daha büyük bir hata normu verdiği görülebilir. Dolayısıyla bu örnek için $h = 0,15$ ağ boyutunu alıyoruz. Şekil 20, bu ağ boyutu için hesaplama hatasını göstermektedir.

Tablo 6. Örnek 2 için Bazı h ve Karşılık Gelen Hata Norm Değerleri

h	$\ u(x, 2) - u^h(x, 2)\ $
0,3	$2,30 \times 10^{-2}$
0,25	$1,73 \times 10^{-2}$
0,2	$1,01 \times 10^{-2}$
0,15	$5,9 \times 10^{-3}$
0,1	$6,8 \times 10^{-3}$



Şekil 20. Örnek 2 için $h = 0,15$ değerine karşılık hata değerleri.

İkinci olarak, doğrusal olmayan en küçük kareler optimizasyon teknikleri ile $w(x) \rightarrow c(x)$ ters tanımlama probleminin çözümünü inceleyelim.

Verilen problemin verileri ve final anı gözlem fonksiyonu kullanılarak katsayı fonksiyonu en uygun şekilde tahmin edilmelidir.

(4.19) ve (4.20) de verilen fonksiyonların yapısını göz önünde bulundurarak, $c(x)$ fonksiyonunu aşağıdaki formlardan biri olarak düşünülebilir;

Tablo 7. Örnek 2 için Bazı Aday Parametre Fonksiyonları

P1	$c(x) = c_1x + c_2y + c_3$
P2	$c(x) = c_1x^2 + c_2xy + c_3y^2$
P3	$c(x) = c_1x^2 + c_2y^2$
P4	$c(x) = c_1x^2 + c_2xy + c_3y^2 + c_4x + c_5y + c_6$

Bu formlardan en uygun olanı,

$$F(c) = \operatorname{argmin}_{c_i} \left\| u^{0,15}(P_j, 2; c(P_j)) - w(P_j) \right\|^2 \quad (4.23)$$

fonksiyonel değerini diğerlerinden daha küçük yapandır.

Bu fonksiyonelin minimizasyonu, $c^0 = [0, \dots, 0]$ başlangıç parametrelerinden başlayarak MATLAB kullanılarak Levenberg-Marquardt metoduyla gerçekleştirilirse, aşağıdaki sonuçlar elde edilir:

Tablo 8. Örnek 2 için Bazı Aday Parametre Fonksiyonları için Elde Edilen Sonuçlar

P1	$c(x) = 1,34415x + 0,0076309y + 1,82528$	$F(c) = 1,84783$
P2	$c(x) = 0,998386x^2 + 1,25986e - 05xy + 0,999903y^2$	$F(c) = 3,38201 \times 10^{-5}$
P3	$c(x) = 0,998415x^2 + 0,999905y^2$	$F(c) = 3,38205 \times 10^{-5}$
P4	$c(x) = 0,998247x^2 + 1,54042e - 05xy + 0,999676y^2 - 0,180302x + 1,88356 \times 10^{-6}y + 0,000385914$	$F(c) = 4,03737 \times 10^{-5}$

Bu çıktılar Ek-5' te verilen Matlab arayüzünün ekranlarından görülebilmektedir.

P2'de verilen fonksiyon en küçük fonksiyonel değeri verdiği için, (4.23) probleminin çözümü olarak alınabilir. Yani optimal çözüm;

$$c^*(x) = 0,998386x^2 + 1,25986e - 05xy + 0,999903y^2$$

olur.

P2 probleminin hesaplama adımları aşağıdaki tabloda sunulmuştur.

Tablo 9. Örnek 2 için P2 Probleminde İterasyon Sonuçları

Iteration	Func-count	Residual	First-Order optimality	Lambda	Norm of step
0	7	30000.4	1.43e+06	0.01	
1	14	27515	5.97e+05	0.001	0.00143508
2	21	24471.4	4.92e+05	0.0001	0.00291776
3	28	16418.3	2.01e+05	1e-05	0.0155319
4	35	7257.16	4.89e+04	1e-06	0.0651355
5	42	2400.87	1.16e+04	1e-07	0.137261
6	49	446.689	2.01e+03	1e-08	0.27267
7	56	73.6973	396	1e-09	0.214946
8	63	6.29704	39.9	1e-10	0.230933
9	70	3.32318	8.62	1e-11	0.0798915
10	77	1.65345	3.17	1e-12	0.150742
11	84	0.544841	1.06	1e-13	0.27584
12	91	0.0700452	0.237	1e-14	0.444307
13	98	0.00326481	0.226	1e-15	0.345271
14	105	5.55952e-05	0.0472	1e-16	0.105951
15	112	3.38201e-05	0.000122	1e-17	0.00618693

Bu hesaplamalarda, λ 'nin başlangıç değeri $\lambda_0 = 0,01$ olarak seçilmiştir ve yinelemelerde bu parametre $\hat{F} < F$ iken $\lambda = \lambda/10$ azaltılmış ve $\hat{F} > F$ iken $\lambda = 10\lambda$ arttırılmıştır.

Ayrıca Levenberg-Marquardt algoritmasında,

$$\|\nabla F(c)\|_{\infty} \leq 1e - 10$$

durdurma kriteri kullanılmıştır. Hesaplama adımlarında $\|\nabla F(c)\|_{\infty} = 8.538889e - 11$ olarak bulunan değer durdurma kriterinden küçük olduğu için, algoritma 15. iterasyondan sonra durmuştur.

P2 tarafından verilen seçim kabul edildikten sonra, veriler üzerinde optimal çözümün sürekliliğini test etmek gerekmektedir. Bu süreçte istenilen bağıl hata payıyla rastgele gürültü eklenir ve ilgili çözümler elde edilir.

Aşağıdaki tabloda bazı gürültülü gözlemler ve bunlara karşılık gelen optimal çözümler yer almaktadır.

Tablo 10. Örnek 2 için P2 Problemindeki Gürültülere Karşılık Gelen Çözümlerin Bağıl Değişimi

Gürültü Yüzdesi	$\frac{\ w - w_{\text{gürültü}}\ }{\ w\ }$	$c^*(x; w_{\text{gürültü}})$	$\frac{\ c^*(x; w) - c^*(x; w_{\text{gürültü}})\ }{\ c^*(x; w)\ }$
%10	0,1	[1,11676 1,75095 × 10 ⁻¹ 1,02251]	1,08946 × 10 ⁻¹
%4	0,04	[0,830589 1,97917 × 10 ⁻⁴ 0,998188]	9,03616 × 10 ⁻²
%1	0,01	[0,964733 1,20856 × 10 ⁻² 0,999205]	1,91052 × 10 ⁻²
%0,5	0,005	[0,967433 4,48277 × 10 ⁻⁴ 0,999201]	1,68245 × 10 ⁻²
%0,1	0,001	[1,00406 1,37985 × 10 ⁻³ 1,00014]	3,19126 × 10 ⁻³
%0	0	[0,998386 1,25986 × 10 ⁻⁵ 0,999903]	0

Bu sonuçlar Ek-6' da verilen Matlab arayüzünün ekranlarından görülebilmektedir. Ayrıca bu arayüzün Matlab kodu da Ek-4 ile verilen kodun benzeridir.

Gürültü seviyesi $\delta \rightarrow 0$ iken

$$\sup_w \left\{ \frac{\|c^*(x; w) - c^*(x; w_{\text{gürültü}})\|}{\|c^*(x; w)\|} \right\} \rightarrow 0$$

olduğu için tablodaki Levenberg-Marquardt yöntemiyle elde edilen çözümün düzgünleştirilmiş çözüm olduğu görülebilir.

Bu tezdeki bulgular, Subaşı and Sendur (2024) tarafından bir ön baskı(preprint) olarak yayımlanmıştır.

TARTIŞMA VE SONUÇ

Hiperbolik sistemler, uygulamalı matematik ve mühendislik alanlarında karşılaşılan karmaşık problemlerden biridir. Özellikle çok değişkenli hiperbolik sistemlerde yeterli sayıda çalışma bulunmaması, düzgünleştirme tekniklerinin zorluklarından kaynaklanmaktadır. Bu durum, bu alanda önemli bir boşluk oluşturmaktadır. Bu çalışmanın amacı da, hiperbolik sistemlerdeki bu boşluğu hem teorik hem de pratik açıdan doldurmak ve mevcut bilgi birikimine katkı sağlamaktır. O halde düzgünleştirme tekniklerinin daha pratik bir şekilde kullanılabilirdiği metotları çok boyutlu hiperbolik problemlerle birleştirmenin yolları aranmalıdır. Hazırlanan bu tez çalışmasında da böyle bir tekniği bu tür denklemlere entegre etmenin yolları araştırılmıştır. Geliştirilen metotla çok boyutlu sistemler için teoriyi destekleyen nümerik yaklaşımlar oluşturup, bunlar hazırlanan yazılımlarla da kolay kullanılabilir hale getirilmiştir.

Bu tez çalışmasında, öncelikle mühendislik alanında da sıkça kullanılan iki boyutlu bir probleme ait modelleme gerçekleştirilmiştir. Burada modellenen sistemle bir ters problem oluşturulmuştur ve bu sistemin çözümü kısmi diferansiyel denklemler için kullanılan sayısal metotlarla gerçekleştirilmiştir. Çalışmanın esas katkısı, şimdiye kadar benzer problemlerde kullanılmamış bir optimizasyon tekniği olan Levenberg-Marquardt yöntemini probleme uyarlayarak kullanılmasıyla problemin çözümünün elde edilmesinin planlanmasıdır. Bu katkı hazırlanmış yazılım ve simülasyonlarla da desteklenmiştir.

Bu araştırmanın, literatüre önemli katkılar sağlayacağına inanıyorum. Yeni bir bakış açısı sunarak teorik bilgiyi genişletmenin yanı sıra, pratik uygulamalara yönelik değerli bilgiler de sunmaktadır. Bu, ilgili alanlarda çalışan araştırmacılar için önemli bir kaynak olacaktır.

Sonuç olarak, belirlenen optimizasyon tekniğinin hiperbolik başlangıç-sınır değer problemlerinde etkili bir şekilde kullanılabileceği ve bu yöntemin hiperbolik denklemlerle ilgili ters problemlerin çözümünde önemli bir araç olabileceği gösterilmiştir. Örneklerde gösterildiği gibi, ters problemlerde sıklıkla karşılaşılan kötü konumlanma (ill-posedness) sorunu, Levenberg-Marquardt yönteminin içerdiği düzgünleştirme tekniği sayesinde etkili bir şekilde ortadan kaldırılmıştır. Bu çalışmadan sonra, benzer ters problemler için diğer doğrusal olmayan optimizasyon tekniklerinin ne gibi sonuçlar verebileceğini incelemek amacıyla karşılaştırmalı analizler yapılabilir.

KAYNAKLAR

- Akhundov, A.Y. and Habibova, A.S., 2022. On an inverse problem for the hyperbolic equation. *Journal of Mathematical Sciences*, vol. 268, no.2.
- Bayraktar, M., 2017. *Fonksiyonel Analiz*. Korza Yayıncılık, 445 s, Ankara.
- Cheng, J. and Yamamoto, M., 2000. One new strategy for a priori choice of regularization parameters in Tikhonov's regularization. *Inverse Problems*, 16, L31-L38.
- Çağlıyan, M. ve Çelebi, O., 2013. *Kısmi Diferansiyel Denklemler*. Dora Yayınları, 276 s, Bursa.
- D'Alembert, J.L.R., 1747. Recherches sur la courbe que forme une corde tendue mise en vibration. *Memoires de l'Academie Royale des Sciences et Belles Lettres, Classe de Mathematique, Anné 1747 (Paris), Tome 3*, pp. 214-219.
- Deiveegan, A., Prakash, P. and Nieto, J. J., 2017. Optimization method for identifying the source term in an inverse wave equation. *Electronic Journal of Differential Equations*, vol. 2017, no. 200, 1-15.
- Engl, H. W., Scherzer, O. and Yamamoto, M., 1994. Uniqueness and stable determination of forcing terms in Linear Partial Differential Equations with overspecified boundary data. *Inverse Problems*, 10, 1253-1276.
- Evans, L. C., 2002. *Partial Differential Equations*, 2nd edn. Graduate Studies in Mathematics, American Mathematical Society, vol. 19.
- Farlow, S. J., 1993. *Partial Differential Equations for Scientists and Engineers*. Dover Publications, Inc., New York.
- Feng, X., Sutton, B., Leuhart, S., et al., 2003. Identification problem for the wave equation with neumann data input and dirichlet data abservating. *Nonlinear Analysis*, vol. 52, no. 7, pp. 1777–1795.
- Goncharkii, A.V. and Romanov, S. Yu., 2012. Two approaches to the solution of coefficient inverse problems for wave equations. *Computational Mathematics and Mathematical Physics*, vol. 52, no. 2, pp. 245-251.
- Graff, K. F., 2012. *Wave Motion in Elastic Solids*. Courier Corporation.
- Hadamard, J., 1908. Théorie des équations aux dérivées partielles linéaires hyperboliques et du problème de Cauchy. *Acta Mathematica* 31, 333–380. <https://doi.org/10.1007/BF02415449>.
- Hadamard, J., 1923. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Yale University Press.
- Ivanov, D.A. and Potapov, M. M., 2015. Approximate solution to a time optimal boundary control problem for the wave equation. *Proceedings of the Steklov Institute of Mathematics*, vol. 291, pp. 102-117.
- Kabanikhin, S. I., Satybaev, A. D. and Shishlenin, M. A., 2005. *Direct Methods of Solving Multidimensional Inverse Hyperbolic Problems*. VSP Science Press, Utrecht.
- Larson, M. G. and Bengzon, F., 2010. *The Finite Element Method: Theory, Implementation, and Practice*. Berlin: Springer.
- Levenberg, K., 1944. A method for the solution of certain nonlinear problems in least squares, *Qart. Appl. Math.* 2,164-166.

- Li, J.- R., 2019. Solving pdes using the finite element method with the matlab pde toolbox. ENSTA Paris, https://perso.ensta-paris.fr/~jing-rebecca.li/frenchvietnammaster2_files/2019/Lectures_JRL/lectures_2019.pdf (10.07.2024).
- Maciag, A., 2011. The usage of wave polynomials in solving direct and inverse problems for two-dimensional wave equation. *International Journal for Numerical Methods in Biomedical Engineering*, 27 (7):1107–1125.
- Marquardt, D. W., 1963. An algorithm for least-squares estimation of nonlinear inequalities, *SIAM J. Appl. Math.* 11, 431-441.
- MathWorks, 2019. Partial Differential Equation Toolbox™ User’s Guide, R2019b, MATLAB.
- Pişkin, E., 2018. Teori ve Çözümlü Problemlerle Diferansiyel Denklemler. Seçkin Yayıncılık, 384 s, Ankara.
- Pişkin, E., 2018. Teori ve Çözümlü Problemlerle Kısmi Diferansiyel Denklemler. Seçkin Yayıncılık, 316 s, Ankara.
- Rosenbrock, H. H., 1960. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3), 175–184.
- Safarova, Z. R., 2020. On a finding the coefficient of one nonlinear wave equation in the mixed problem. *Archives of Control Sciences*, vol. 30(LXVI), no. 2, p. 199-212.
- Saraç Y. and Subaşı M., 2012. On the regularized solutions of optimal control problem in a hyperbolic system. *Abstract and Applied Analysis*, 2012(2012), p. 12.
- Serovajsky, S., 2013. Optimal control for systems described by hyperbolic equation with strong nonlinearity. *Journal of Applied Analysis and Computation*, 3(2), 183–195.
- Subaşı, M. and Araz, S., 2019. Numerical regularization of optimal control for the coefficient function in a wave equation. *Iranian Journal of Science and Technology, Transactions A: Science*, vol 43, pp. 2325–2333.
- Subasi, M., Sendur, F. D. and Yasar, C., (2021). An inverse source identification by nonlinear optimization in a two-dimensional hyperbolic problem. *Inverse Problems in Science and Engineering*, vol. 29, no. 12, 2110-2130.
- Subasi, M. and Sendur, F. D., (2024). Inverse coefficient identification by nonlinear optimization in a two dimensional hyperbolic problem. *Authorea*, Doi: 10.22541/au.170668765.54842164/v1.
- Şahin, Y. ve Yıldırım, M., 2018. Mühendisler ve Fen Bilimciler için Fizik. Korza Yayıncılık, 446 s, Erzurum.
- Tagiyev, R. K., 2012. On optimal control of the hyperbolic equation coefficients. *Automation and Remote Control*, vol. 73, no. 7, pp. 1145-1155.
- Tekin, İ., 2018. Existence and uniqueness of an inverse problem for a second order hyperbolic equation. *Universal Journal of Mathematics and Applications*, 1 (3), 178-185.
- Tikhonov A. N., 1963. On the solution of ill-posed problems and the method of regularization, *Dokl. Akad. Nauk SSSR*, Volume 151, Number 3, 501–504.
- Yamamoto, M., 1996. On ill-posedness and a tikhonov regularization for a multidimensional inverse hyperbolic problem. *J. Math Kyoto University*, 36, 825-856.
- Yamamoto, M., 1999. Uniqueness and stability in multidimensional hyperbolic inverse problem. *Journal De Mathematiques Pures Et Appliquees*, 78, p. 65-98.

Yükselen, M.A., 2009. HM504 Uygulamalı sayısal yöntemler ders notları kısmi diferansiyel denklemlerin sayısal çözümü. Uçak ve Uçak Bilimleri Fakültesi, İstanbul Teknik Üniversitesi, <https://web.itu.edu.tr/yukselen/HM504/05-%20K%FDsmi%20diferansiyel%20denklemler.pdf> (23.06.2021).



EKLER

EK-1. Direkt Problemin Çözümü için Yazılmış Matlab Arayüzünün Kodu

```
function varargout = direct_problem(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',      gui_Singleton, ...
                  'gui_OpeningFcn',     @direct_problem_OpeningFcn, ...
                  'gui_OutputFcn',      @direct_problem_OutputFcn, ...
                  'gui_LayoutFcn',      [] , ...
                  'gui_Callback',       []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function direct_problem_OpeningFcn(hObject, eventdata,
handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = direct_problem_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
hmax=0.15;
domain;
pdegplot(model,'edgeLabels','on');
axis([-0.1 1.1 -0.1 1.1]);

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton2_Callback(hObject, eventdata, handles)
axes(handles.axes2);
hmax=str2double(get(handles.edit1,'String'));
```

```

domain
pdeplot(model);
axis([-0.1 1.1 -0.1 1.1]);

function edit2_Callback(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)

function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

```

```

function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)

function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton3_Callback(hObject, eventdata, handles)
axes(handles.axes3);

hmax=str2double(get(handles.edit1,'String'));
domain;
data;
specifyCoefficients(model,'m',m,'d',d,'c',ccoeff,'a',a,'f',f);
setInitialConditions(model,u0,ut0);
applyBoundaryCondition(model,'Edge',[1,2,3,4],'u',bc);
result=solvepde(model,tlist);
u=result.NodalSolution;
pdeplot(model,'xydata',u(:,N),'zdata',u(:,N),'mesh','on','xygrid',
'id','on','ColorBar','on');
title('Observation at Final Time');
axes(handles.axes4);

title('Coefficient Function');
w=exp(-T)*(sin(pi*mesh.Nodes(1,:)).*sin(pi*mesh.Nodes(2,:)));
w=w';
pdeplot(model,'xydata',u(:,N)-w,'zdata',u(:,N)-
w,'mesh','on','xygrid','on','ColorBar','on');
set(handles.text18,'String',num2str(norm(u(:,N)-w)))

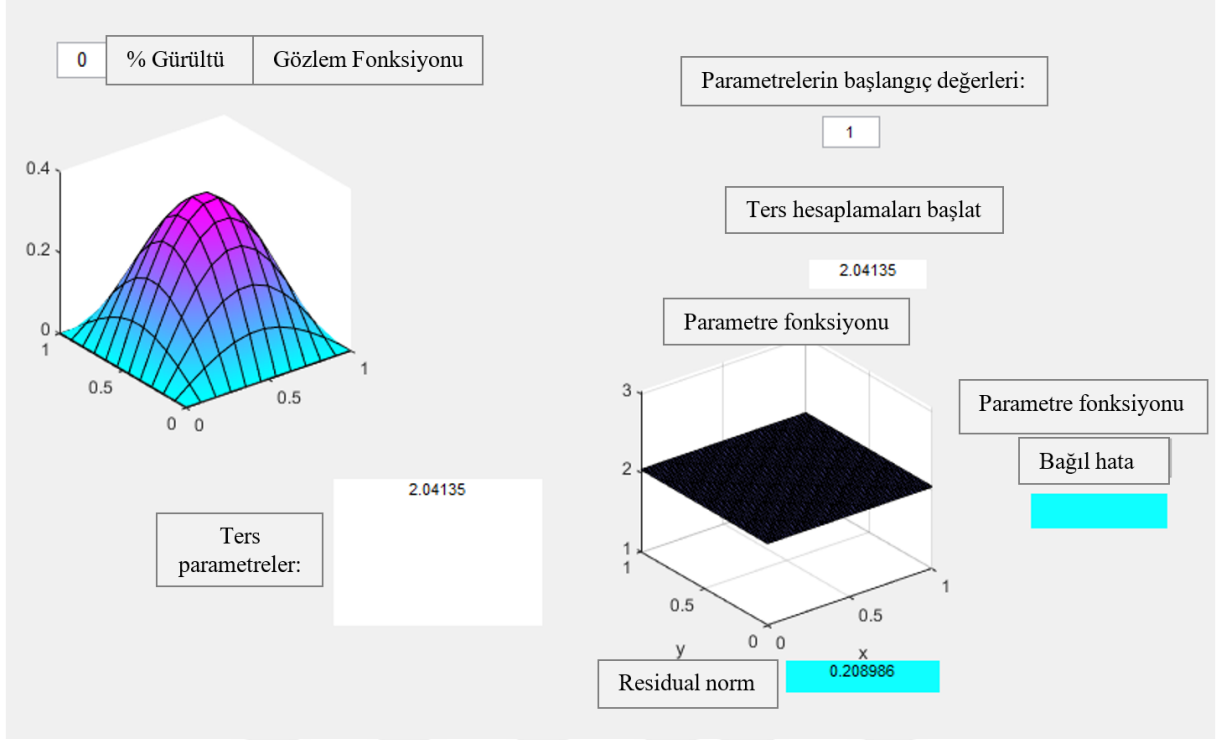
function edit10_Callback(hObject, eventdata, handles)

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

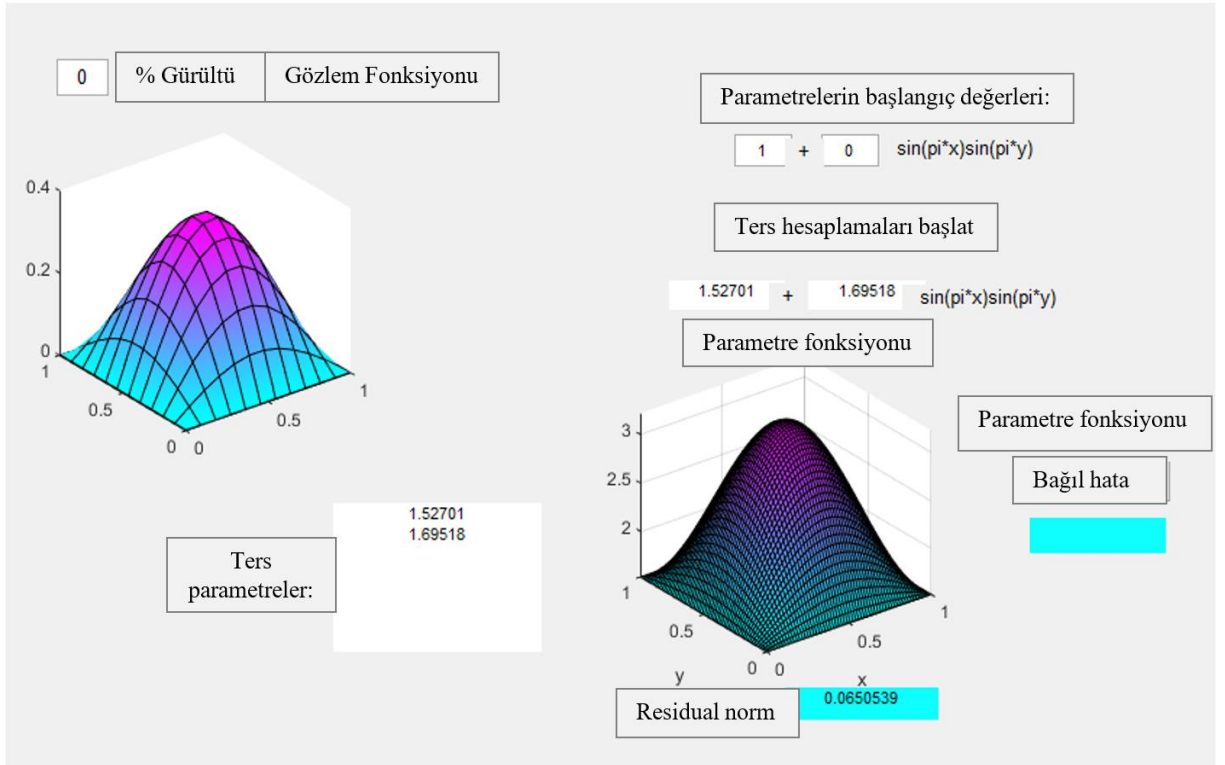
```

EK-2. Örnek 1 ile Verilen Seçenekler için Gürültüsüz Gözlemlere Ait Matlab Arayüz Ekranları

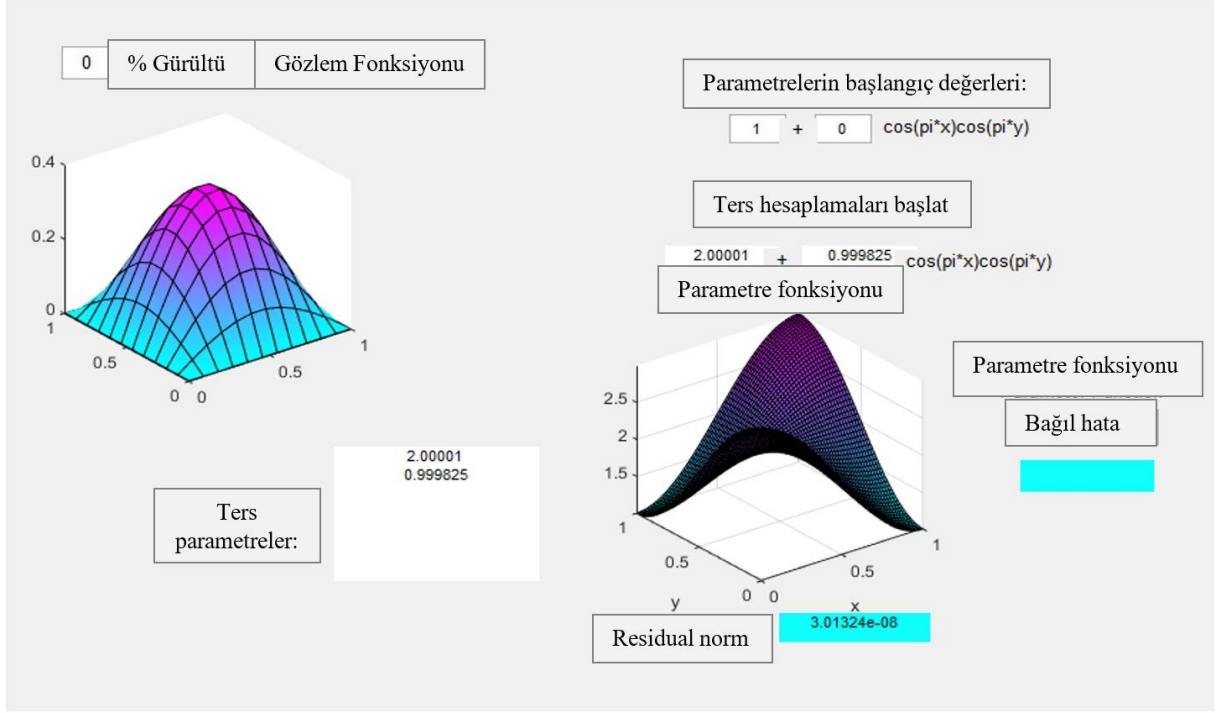
Örnek 1 ile verilen P1 seçeneği için gürültüsüz gözlem için Matlab arayüz ekranı.



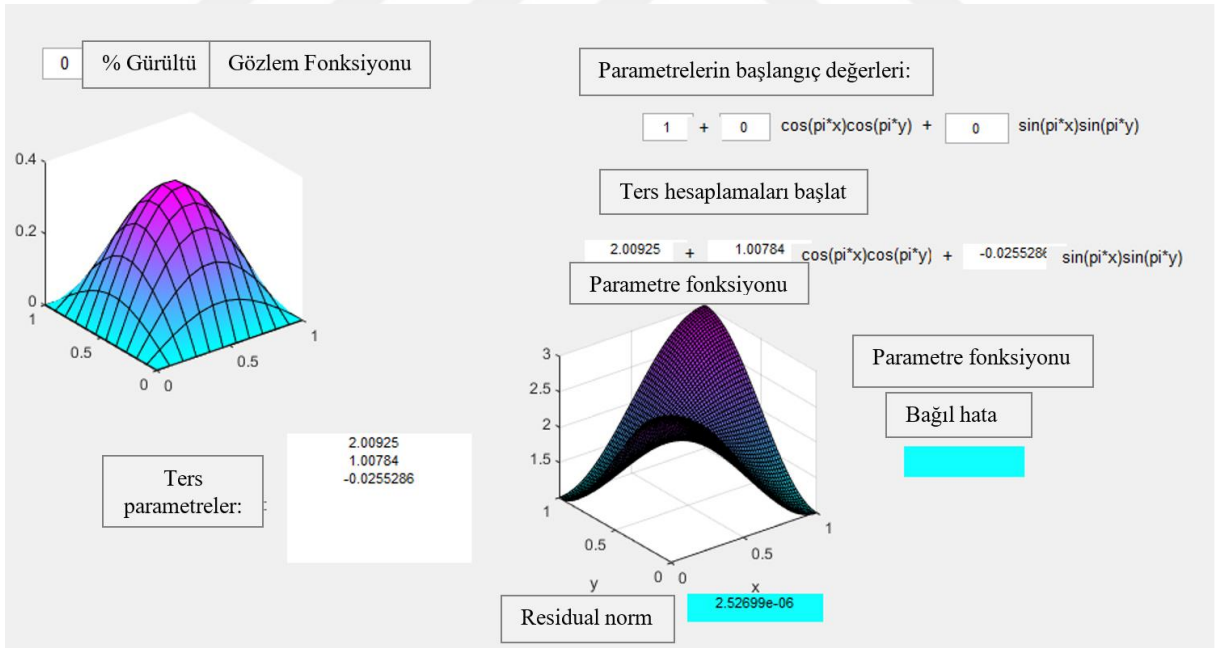
Örnek 1 ile verilen P2 seçeneği için gürültüsüz gözlem için Matlab arayüz ekranı.



Örnek 1 ile verilen P3 seçeneği için gürültüsüz gözlem için Matlab arayüz ekranı.

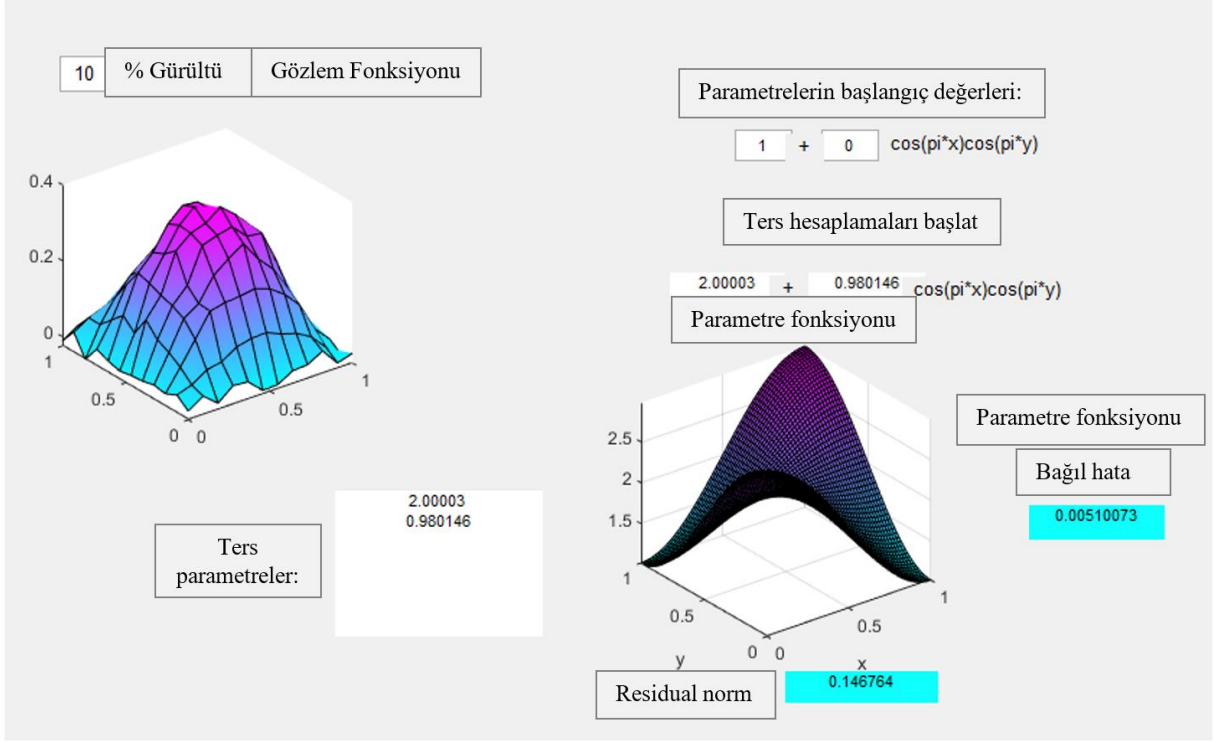


Örnek 1 ile verilen P4 seçeneği için gürültüsüz gözleme ait Matlab arayüz ekranı.

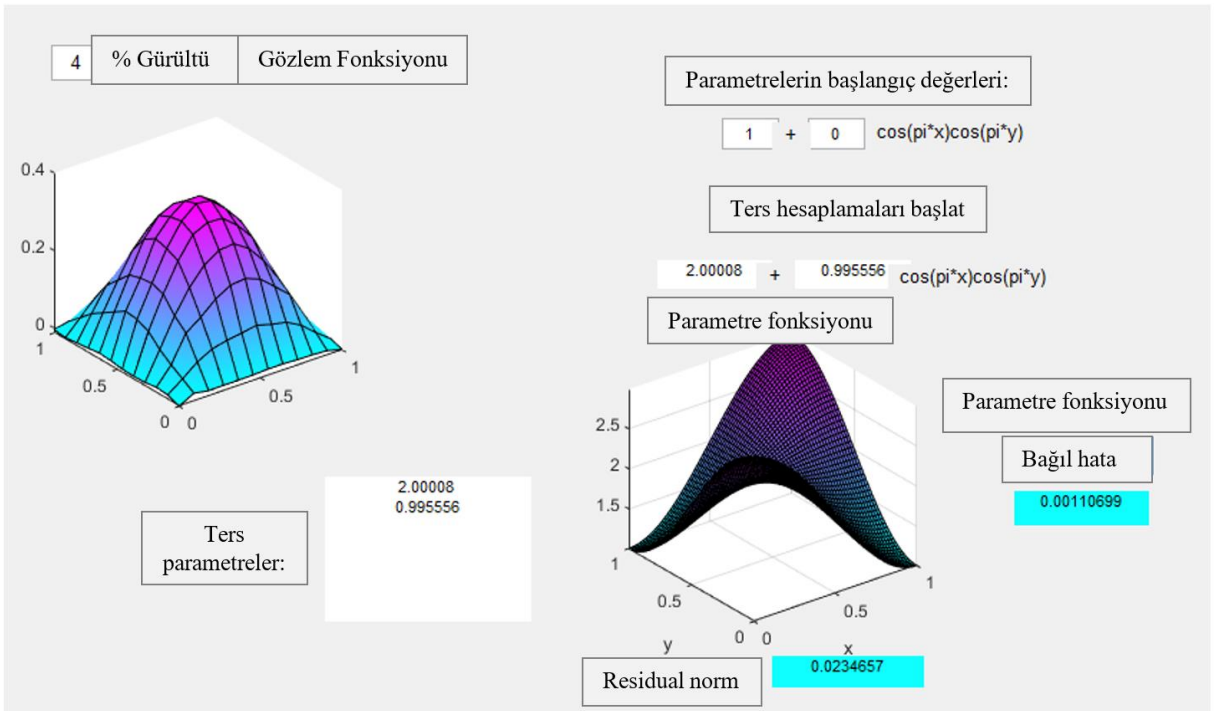


EK-3. Örnek 1 ile Verilen P3 Seçeneği için Bazı Gürültülü Gözlemlere Ait Matlab Arayüz Ekranları

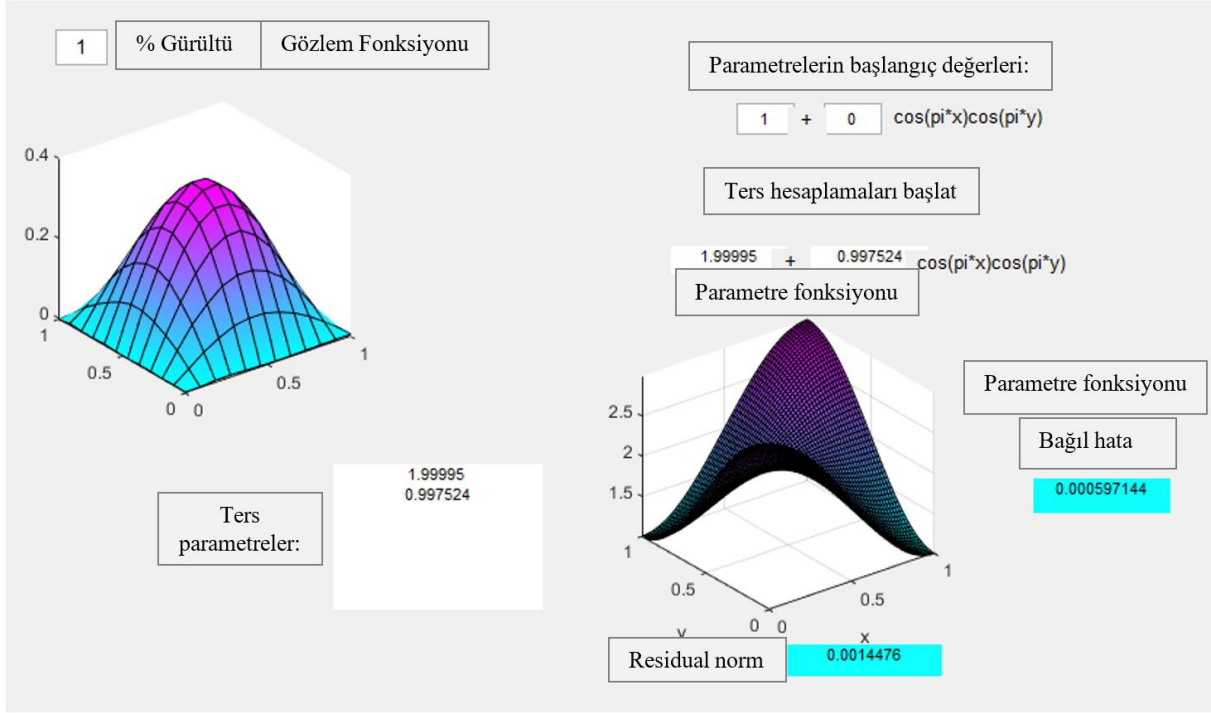
Örnek 1 ile verilen P3 seçeneği için %10 gürültülü gözleme ait Matlab arayüz ekranı.



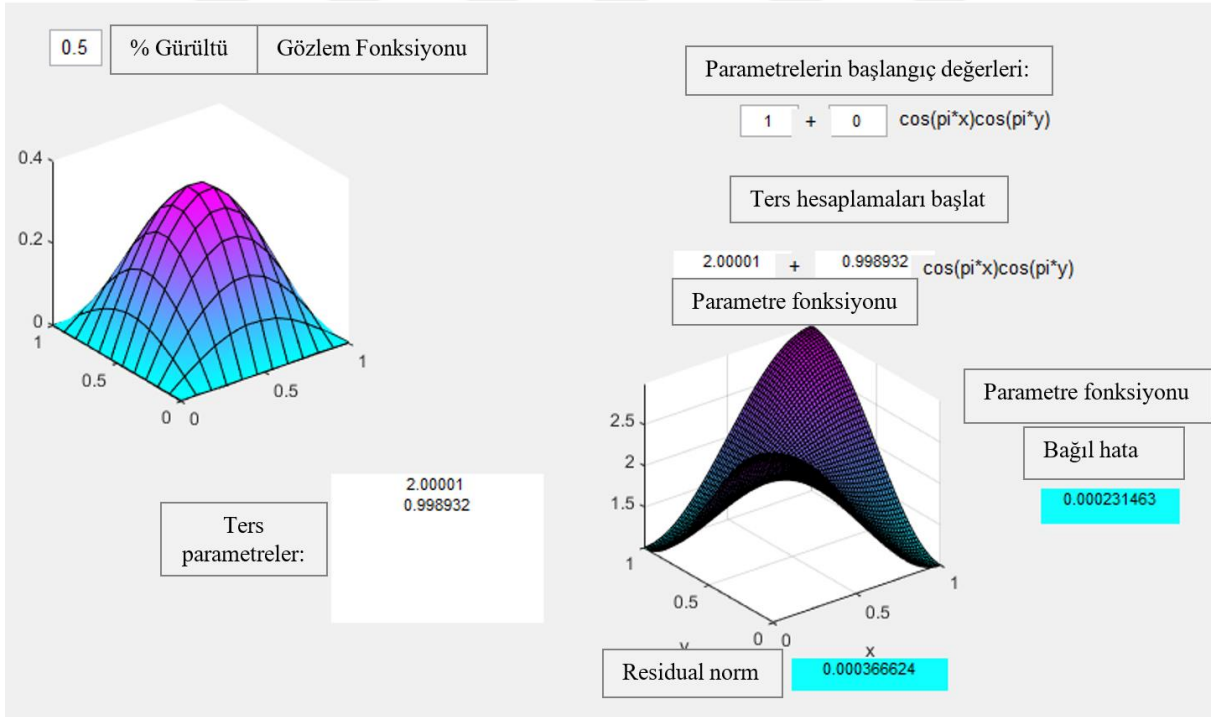
Örnek 1 ile verilen P3 seçeneği için %4 gürültülü gözleme ait Matlab arayüz ekranı.



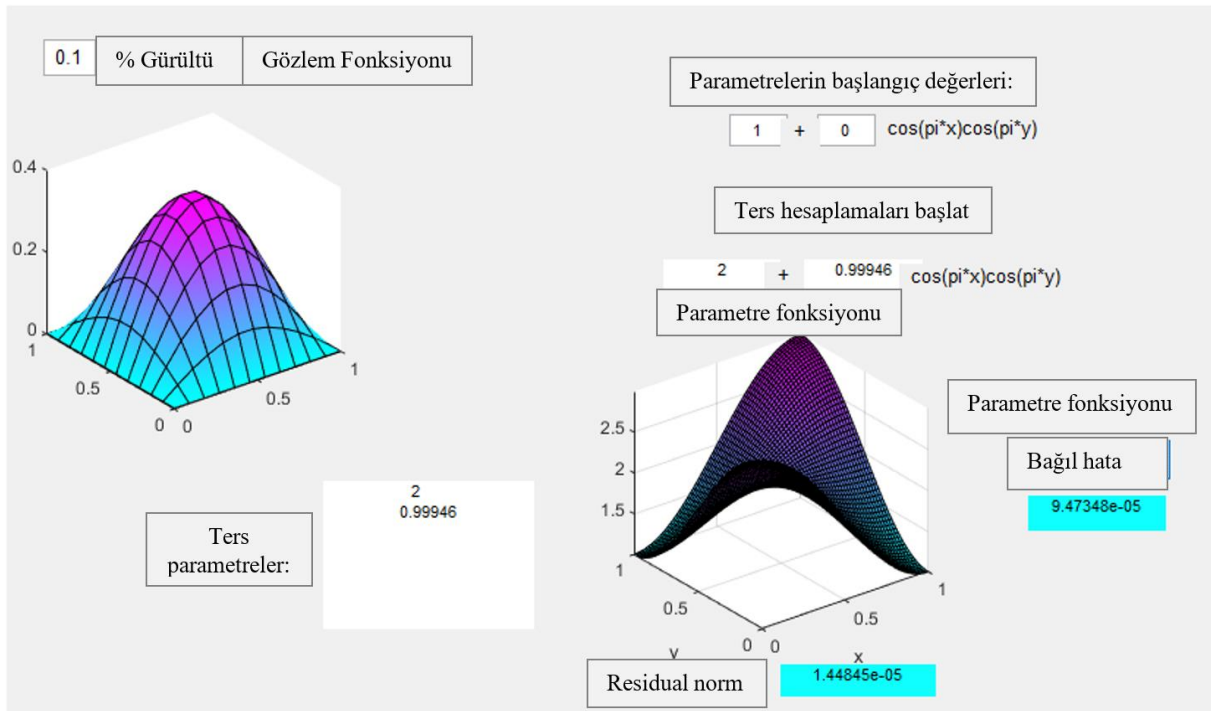
Örnek 1 ile verilen P3 seçeneği için %1 gürültülü gözleme ait Matlab arayüz ekranı.



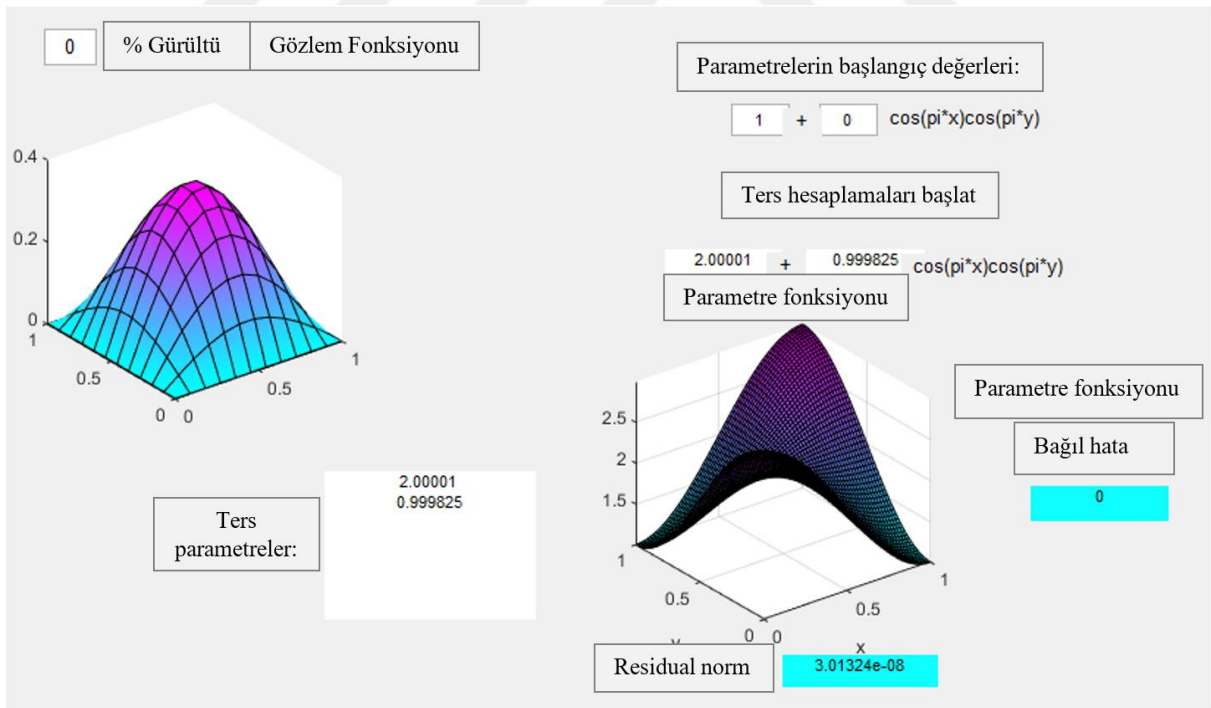
Örnek 1 ile verilen P3 seçeneği için %0.5 gürültülü gözleme ait Matlab arayüz ekranı.



Örnek 1 ile verilen P3 seçeneği için %0.1 gürültülü gözleme ait Matlab arayüz ekranı.



Örnek 1 ile verilen P3 seçeneği için %0 gürültülü gözleme ait Matlab arayüz ekranı.



EK-4. Örnek 1 ile Verilen Ters Problemden Gürültülü Gözlemlere Karşılık Gelen Çözümü Bulma için Yazılmış Matlab Arayüzünün Kodu

```
function varargout = inverse_problem1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',      gui_Singleton, ...
                  'gui_OpeningFcn',     @inverse_problem1_OpeningFcn, ...
                  'gui_OutputFcn',     @inverse_problem1_OutputFcn, ...
                  'gui_LayoutFcn',     [] , ...
                  'gui_Callback',      []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function inverse_problem1_OpeningFcn(hObject, eventdata,
handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = inverse_problem1_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
p=str2double(get(handles.edit1,'String'));
hmax=0.1;
domain
w=exp(-1)*(sin(pi*mesh.Nodes(1,:)).*sin(pi*mesh.Nodes(2,:)));
if p==0
    wp=w;
else
    no=normrnd(0,p,[1,length(w)]);
    eps=no/norm(no)*p/100*norm(w);
```

```

    wp=w+eps;
end
pdeplot(model, 'xydata', wp, 'zdata', wp, 'mesh', 'on', 'xygrid', 'on'
, 'ColorBar', 'off');

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton2_Callback(hObject, eventdata, handles)
c1=str2double(get(handles.edit5,'String'));
c2=str2double(get(handles.edit7,'String'));

c=[c1 c2];
options = optimoptions('lsqnonlin','Algorithm','levenberg-
marquardt','FiniteDifferenceType','central','Display','iter');
[opt_cont, resnorm, residual, exitflag, output, lambda, jacobian]
=lsqnonlin(@fonksiyonel,c,[],[],options);
set(handles.text3,'String',opt_cont)
set(handles.text4,'String',resnorm)
set(handles.text19,'String',opt_cont(1));
set(handles.text21,'String',opt_cont(2));

axes(handles.axes2);
syms x y;
ezsurf(opt_cont(1)+opt_cont(2)*(cos(pi*x)*cos(pi*y)), [0 1 0
1]);
title('Parameter Function');

function edit4_Callback(hObject, eventdata, handles)

function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)

```

```

function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

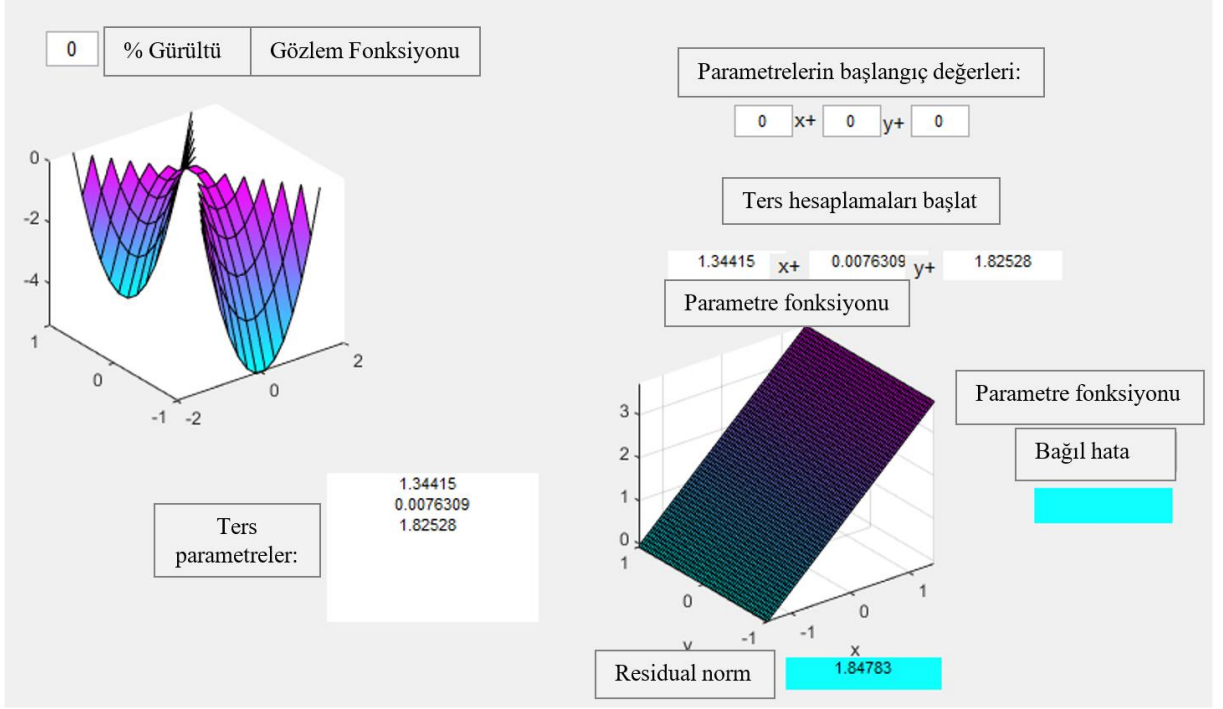
function pushbutton4_Callback(hObject, eventdata, handles)
domain;
cc1=str2double(get(handles.text19,'String'));
cc2=str2double(get(handles.text21,'String'));

c=2.00001+0.999825*cos(pi*mesh.Nodes(1,:)).*cos(pi*mesh.Nodes(
2,:));
cw=cc1*ones(1,length(c))+cc2*cos(pi*mesh.Nodes(1,:)).*cos(pi*m
esh.Nodes(2,:));
bh=norm(c-cw)/norm(c);
set(handles.text30,'String',bh);

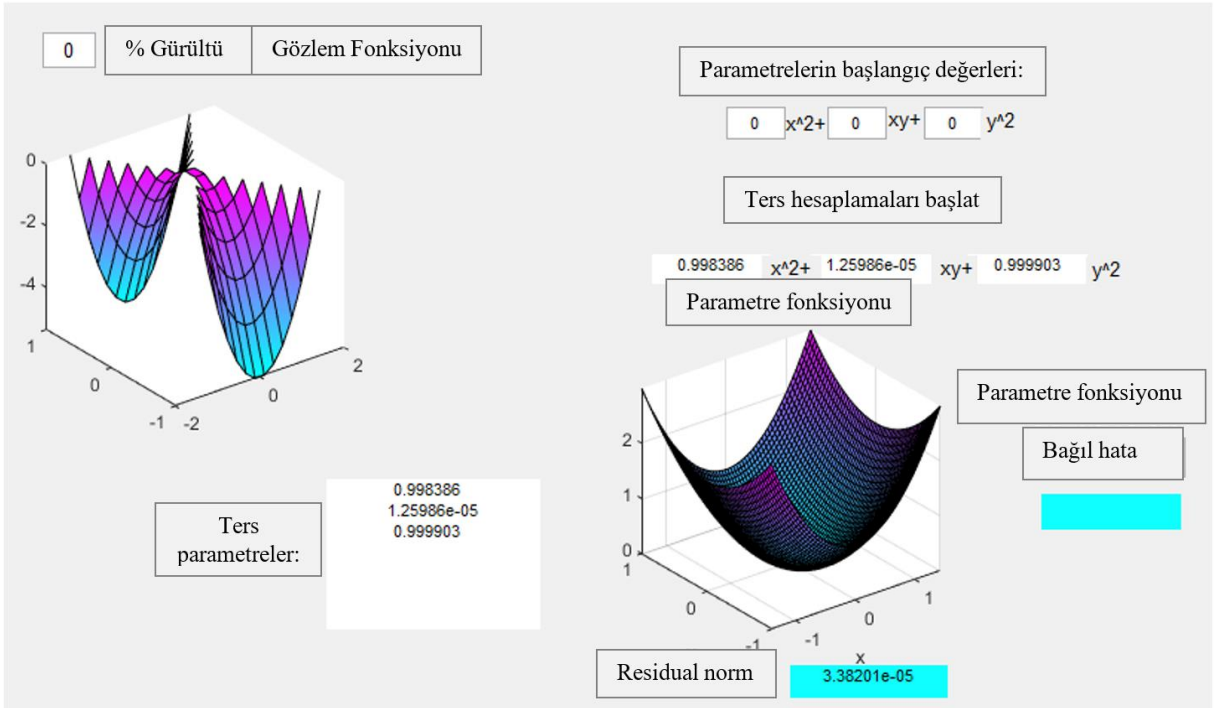
```

EK-5. Örnek 2 ile Verilen Seçenekler için Gürültüsüz Gözlemlere Ait Matlab Arayüz Ekranları

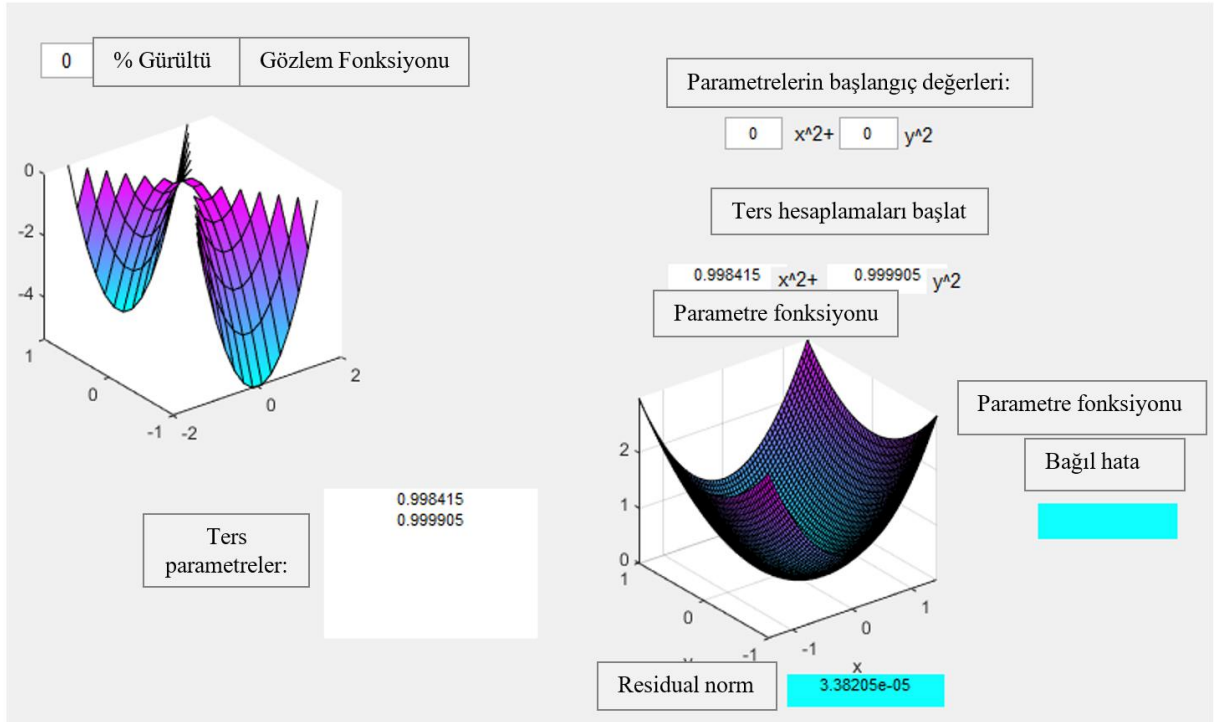
Örnek 2 ile verilen P1 seçeneği için gürültüsüz gözleme ait Matlab arayüz ekranı.



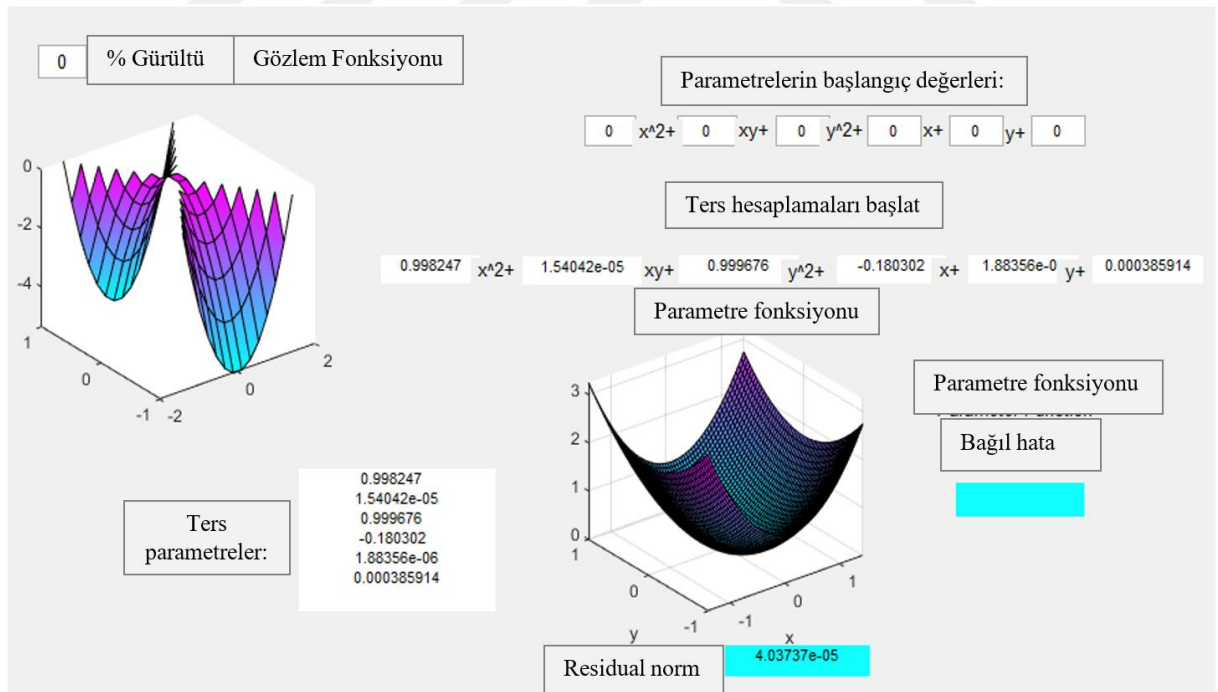
Örnek 2 ile verilen P2 seçeneği için gürültüsüz gözleme ait Matlab arayüz ekranı.



Örnek 2 ile verilen P3 seçeneği için gürültüsüz gözleme ait Matlab arayüz ekranı.

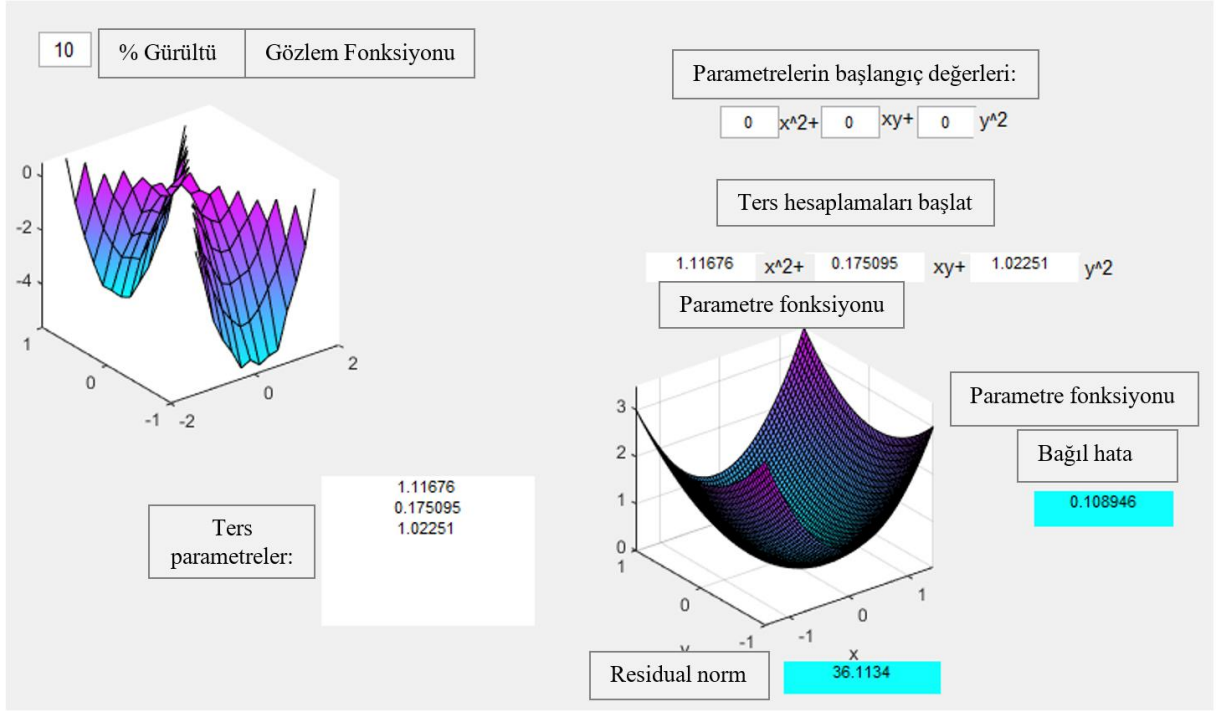


Örnek 2 ile verilen P4 seçeneği için gürültüsüz gözleme ait Matlab arayüz ekranı.

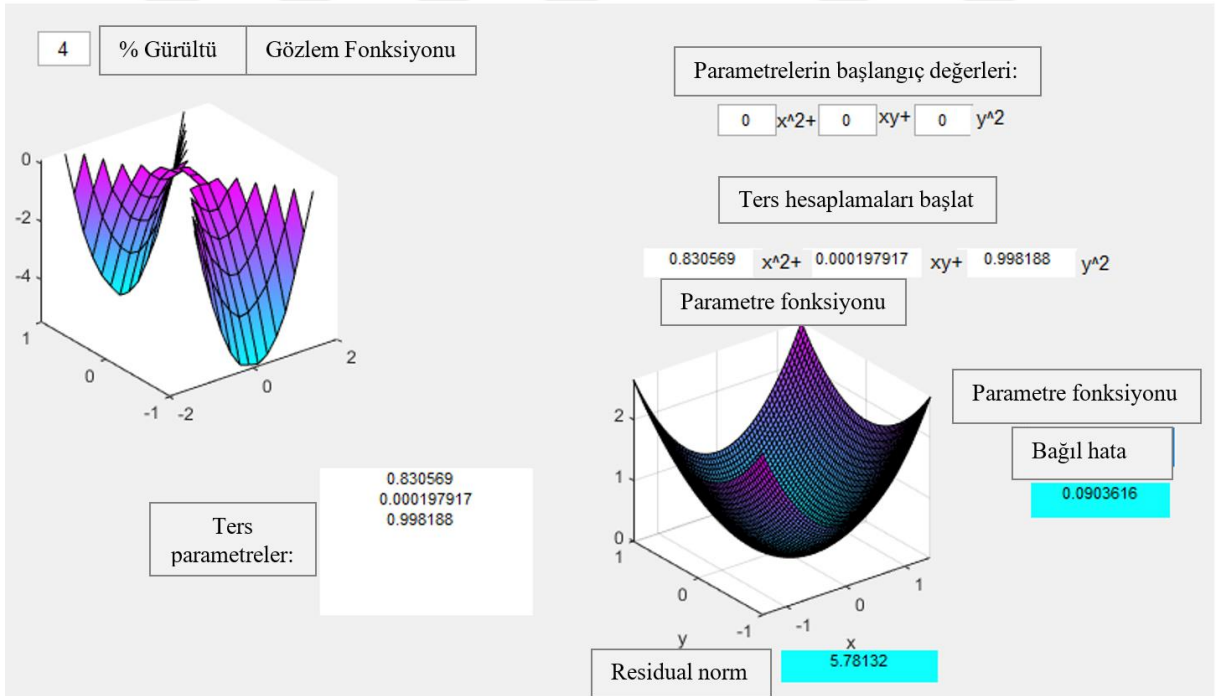


EK-6. Örnek 2 ile Verilen P2 Seçeneği için Bazı Gürültülü Gözlemlere Ait Matlab Arayüz Ekranları

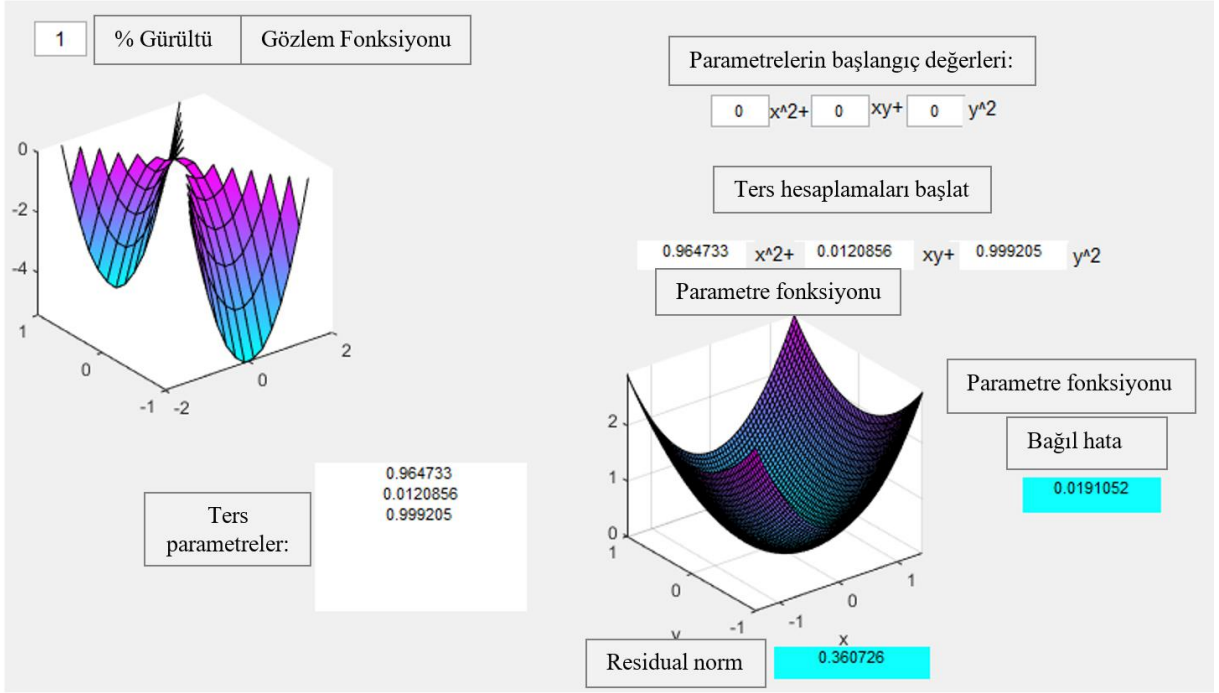
Örnek 2 ile verilen P2 seçeneği için %10 gürültülü gözleme ait Matlab arayüz ekranı.



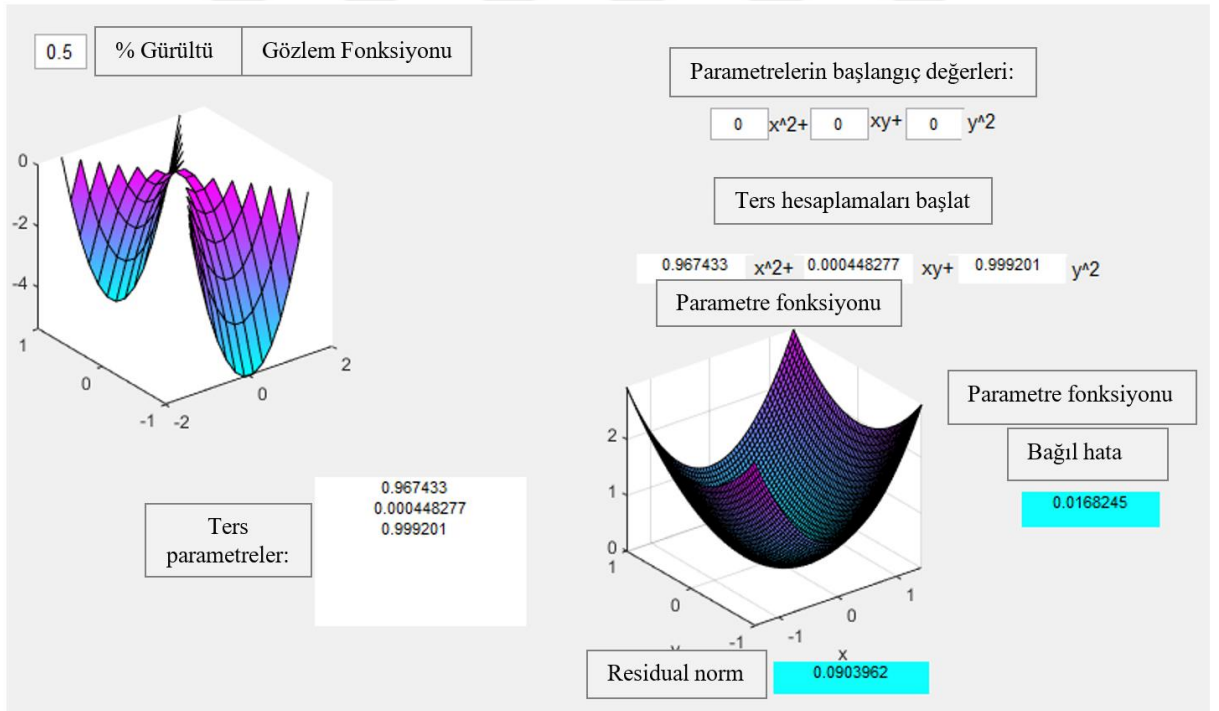
Örnek 2 ile verilen P2 seçeneği için %4 gürültülü gözleme ait Matlab arayüz ekranı.



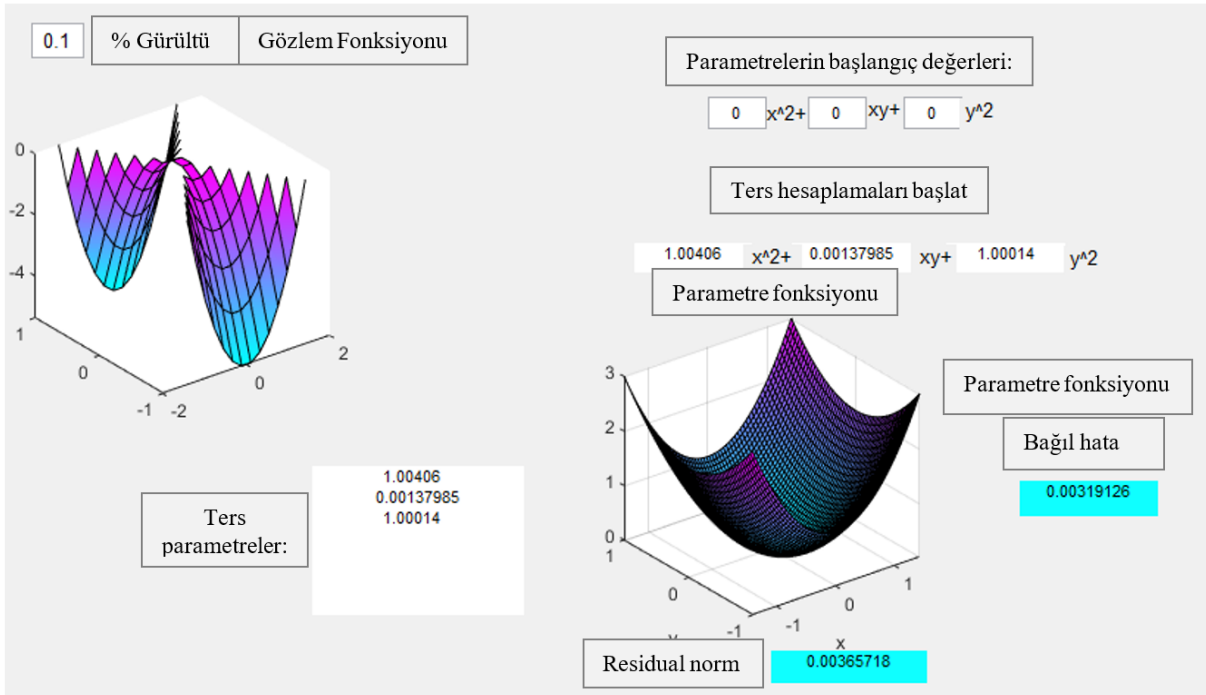
Örnek 2 ile verilen P2 seçeneği için %1 gürültülü gözleme ait Matlab arayüz ekranı.



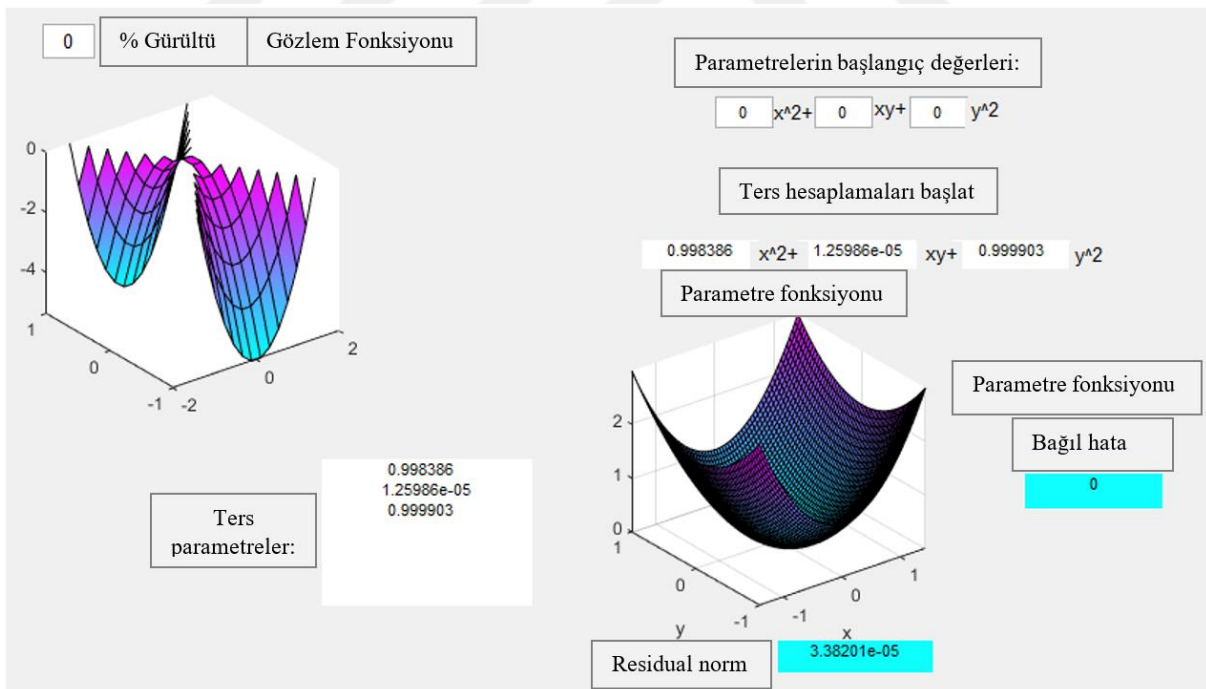
Örnek 2 ile verilen P2 seçeneği için %0.5 gürültülü gözleme ait Matlab arayüz ekranı.



Örnek 2 ile verilen P2 seçeneği için %0.1 gürültülü gözleme ait Matlab arayüz ekranı.



Örnek 2 ile verilen P2 seçeneği için %0 gürültülü gözleme ait Matlab arayüz ekranı.



ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Faika Derya ŞENDUR
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
E-mail:	
Eğitim	
Lisans:	Atatürk Üniversitesi, Kazım Karabekir Eğitim Fakültesi, OFMA Matematik Öğretmenliği (2015)
Yüksek lisans:	Akdeniz Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı (2017)
Doktora:	Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Matematik Anabilim Dalı (2018-2019) Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı
Yabancı Dil Bilgisi	
İngilizce:	İyi
Üye Olunan Mesleki Kuruluşlar	
-	
Tezden Üretilmiş Yayınlar	
1. An inverse source identification by nonlinear optimization in a two-dimensional hyperbolic problem, Inverse Problems in Science and Engineering. (SCI-Exp.)	