

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

JEOFİZİK MÜHENDİSLİĞİ ANABİLİM DALI

**FARKLI AĞIRLIKLANDIRMA FONKSİYONLARI İLE YIĞMA SONRASI EN
KÜÇÜK KARELER GÖÇÜ VE GÖÇ DEKONVOLÜSYONU**

DOKTORA TEZİ

Jeofizik Yüksek Mühendisi Recep GÜNEY

**NİSAN-2017
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

JEOFİZİK MÜHENDİSLİĞİ ANABİLİM DALI

**FARKLI AĞIRLIKLANDIRMA FONKSİYONLARI İLE YIĞMA SONRASI EN KÜÇÜK
KARELER GÖÇÜ VE GÖÇ DEKONVOLÜSYONU**

Jeofizik Yüksek Mühendisi Recep GÜNEY

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“DOKTOR (JEOFİZİK MÜHENDİSİ)”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 14 / 03 / 2017

Tezin Savunma Tarihi : 13 / 04 / 2017

Tez Danışmanı : Prof. Dr. Hakan KARSLI

Trabzon 2017

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**Jeofizik Mühendisliği Anabilim Dalında
Recep GÜNEY Tarafından Hazırlanan**

**FARKLI AĞIRLIKLANDIRMA FONKSİYONLARI İLE YIĞMA SONRASI EN
KÜÇÜK KARELER GÖÇÜ VE GÖÇ DEKONVOLÜSYONU**

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 14 /03/2017 gün ve 1693 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
DOKTORA TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

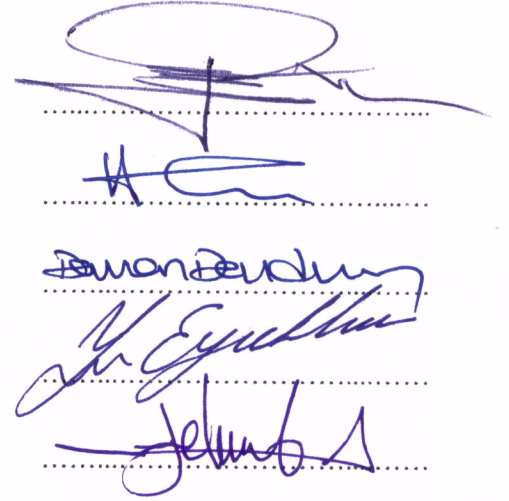
Başkan : Prof. Dr. Hüseyin TUR

Üye : Prof. Dr. Hakan KARSLI

Üye : Prof. Dr. Derman DONDURUR

Üye : Prof. Dr. Yener EYÜBOĞLU

Üye : Doç. Dr. Aysel ŞEREN



Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

ÖNSÖZ

Bu tez çalışması 11546 kod numaralı “Farklı Ağırlıklandırma Fonksiyonları İle Yığma Sonrası En Küçük Kareler (EKKG) Göçü” isimli proje kapsamında Karadeniz Teknik Üniversitesi Bilimsel Araştırma Projeleri (BAP) Birimi tarafından desteklenmiştir. Tez çalışmalarım sırasında ve daha öncesinde tüm mesleki hayatım boyunca ilgisiyle, bilgisiyle ve desteğiyle yanımda olan danışmanım Prof. Dr. Hakan KARSLI’ ya, 2014-2015 yıllarında tez çalışmalarının bir bölümünü gerçekleştirdiğim Miskolc Üniversitesi Jeofizik Bölümü personeline ve yardımlarını esirgemeyen Prof. Dr. Mihaly DOBROKA’ ya, çalışma arkadaşım Mustafa ŞENKAYA, Hamdi ALKAN, Erdem BAYRAK ve Zeynep ÖĞRETMEN’ e, gerçek veri ihtiyacımızı her daim karşılayan Dokuz Eylül Üniversitesi Deniz Bilimleri ve Teknolojileri Enstitüsü SeisLab ekibine ve sayın Prof. Dr. Derman DONDURUR’ a, 2009-2016 yılları arasında Araştırma Görevlisi olarak görev yaptığım KTÜ Jeofizik ailesine ve bu süreçte yanımda olan aileme ve eşime sonsuz teşekkür ederim.

Recep GÜNEY
Trabzon-2017

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduđum “Farklı Ađırlıklandırma Fonksiyonları ile yıđma sonrası G Dekonvolsyonu ve En K Kareler G Dekonvolsyonu ” bařlıklı bu alıřmayı bařtan sona kadar danıřmanım Prof. Dr. Hakan KARSLI ’nın sorumluluđunda tamamladıđımı, SEG/EAGE model verisini (url1) internet adresinden, arazi atıř verilerini Dokuz Eyllnn Deniz Bilimleri ve Teknolojileri Enstits’nden aldıđımı, testleri yaparken kullandıđım kodların bizzat bizim tarafımızdan yazıldıđını, bařka kaynaklardan aldıđım bilgileri metinde ve kaynaka da eksiksiz olarak gsterdıđımı, alıřma sresi boyunca bilimsel arařtırma ve etik kurallara uygun olarak davrandıđımı ve aksinin ortaya ıkması durumunda her trl sonucunu kabul ettiđimi beyan ederim. 01/06/2017


Recep GNEY

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	XX
KISALTMALAR DİZİNİ	XXI
1. GİRİŞ.....	1
1.1. Genel Bilgiler.....	1
1.2. Sismik Yansıma Verisinin Önemi	6
2. YÖNTEM	11
2.1. Göç İşlemi.....	11
2.2. Kirchhoff Göçü Matris Formu.....	13
2.3. En Küçük Kareler Göçü (EKKG).....	21
2.4. Göç Dekonvolüsyonu (Netleştirme)	23
2.5. En Küçük Kareler Göç Dekonvolüsyonu (EKKG D).....	24
2.6. Ağırlıklandırma Fonksiyonları	24
3. UYGULAMALAR.....	27
3.1. Model Veriler Üzerinde Göç Uygulamaları	27
3.1.1. Nokta Saçılma Modeli Üzerinde Göç Uygulamaları.....	27
3.1.2. Papyon-Kravat Modeli Üzerinde Göç Uygulamaları	48
3.1.3. Fay Modeli Üzerinde Göç Uygulamaları	69
3.1.4. SEG/EAGE Tuz Modeli Üzerinde Göç Uygulamaları	91
3.2. Arazi Verileri Üzerinde Göç Uygulamaları.....	105
3.2.1. Arazi Verisi-1 Üzerinde Göç Uygulamaları	106
3.2.2. Arazi Verisi-2 üzerinde Göç Uygulamaları	119
4. BULGULAR	132
4.1. Hız Fonksiyonunun Göç Algoritmalarına Etkisi	132
4.2. Yarı Operatör Boyunun Göç Algoritmalarına Etkisi.....	132

4.3.	Beklenen Çıkışa En Fazla Yakınsamayı Sağlayan Göç Algoritmaları.....	133
4.4.	Netleştirme Fonksiyonlarının Yineleme Sayılarına Etkisi	134
4.5.	Zaman Ekonomisi Sağlayan Fonksiyonlar	134
4.6.	EKKG ve GD Karşılaştırması	136
4.7.	Netleştirme Fonksiyonlarını Kullanmanın Etkisi	136
4.8.	EKKG ve EKKGD Yöntemleri için Yinelemeli (Eşlenik-Gradyent ve Gauss-Seidel) Çözüm Algoritmalarının Etkisi	137
4.9.	Düzenleme (Regularization) Parametresinin Etkisi.....	138
4.9.	Maliyet ve Fayda açısından en uygun çıkışları üreten netleştirme fonksiyonları.	142
5.	TARTIŞMALAR VE SONUÇLAR.....	163
KAYNAKLAR.....		167
EKLER		171
ÖZGEÇMİŞ		

Doktora Tezi

ÖZET

FARKLI AĞIRLIKLANDIRMA FONKSİYONLARI İLE YIĞMA SONRASI EN KÜÇÜK KARELER GÖÇÜ VE GÖÇ DEKONVOLÜSYONU

Recep GÜNEY

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Jeofizik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Hakan KARSLI
2017, 170 Sayfa, 10 Sayfa Ek

Sismik veri işlem adımlarının son ve en önemli aşaması olan sismik göç işlemi yansıma genliklerini gerçek yansıma pozisyonuna taşıyarak yeraltı yapısının doğru görüntüsünü elde etmek için yaygın şekilde kullanılmaktadır. Aynı zamanda bir ters çözüm işlemi olarak kabul edilen sismik göç çıkış kesitleri göç işlem hatalarını da beraberinde getirir. Bu işlem hatalarını bastırmak için En Küçük Kareler Göçü (EKKG), Göç Dekonvolüsyonu (GD) veya bu iki yöntemin birleşik kullanımını kapsayan En Küçük Kareler Göç Dekonvolüsyonu (EKKGD) yöntemleri kullanılmaktadır. Bu yöntemlerin ortak dezavantajı, işlem maliyetini (zamanını) arttırmalarıdır. Bu çalışmada; göç operatöründen (L) türetilen geleneksel ağırlıklandırma (netleştirme) fonksiyonu ($(L^T L)^{-1}$) ile giriş verisinden (d) oluşturulan farklı ağırlıklandırma fonksiyonları kullanılarak işlem maliyeti azaltılmaya çalışılmıştır. Farklı sentetik ve gerçek veri modelleri üzerinde yapılan uygulamalardan en az hata oranını veren netleştirme fonksiyonları $cov((dd^T))^{-1}$ ve $cov((L^T L)^{-1})$ olurken, $cov(dd^T)^{-1}$ ağırlıklandırma fonksiyonu $cov(L^T L)^{-1}$ netleştirme fonksiyonuna göre yaklaşık %60 (GD için %80, EKKGD için %40) işlem zamanını azalttığı görülmüştür. Bunun yanında $(dd^T)^{-1}$ fonksiyonu işlem zamanını yaklaşık %45 (GD için %18, EKKGD için %65) maliyetle geleneksel netleştirme fonksiyonu ile aynı hata oranlarına sahip olduğu görülmüştür. Ayrıca EKKGD işlemi, GD işlemine göre hata oranlarını yaklaşık %15 daha azalttığı görülmüştür.

Anahtar Kelimeler: Sismik Göç, En Küçük Kareler Göçü, Göç Dekonvolüsyon, Ağırlıklandırma

PhD. Thesis

SUMMARY

POST STACK LEAST SQUARES MIGRATION AND MIGRATION DECONVOLUTION WITH DIFFERENT WEIGHTENING FUNCTIONS

Recep GÜNEY

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Geophysical Engineering Department
Supervisor: Prof. Dr. Hakan KARSLI
2017, 170 Pages, 10 Pages Appendix

The seismic migration process, which is the last and most important step of the seismic data processing steps, is widely used to obtain the correct image of the subsurface structure by moving the reflection amplitudes to the actual reflection positions. In this scope, least square migration (LSM) is recently used a linearized inversion process which improve spatial resolution of post-stack seismic data, but also represents blurred images of earth's reflectivity distribution because of migration artifacts. To partly alleviate this blurring, Migration Deconvolution (MD) are proposed and applied to migrated images. The main disadvantage of the method is that it increases the computational time cost. In this study, in addition to the deblurring function which is derived from migration operator (L), it is tried to decrease processing cost by using different deblurring functions which are derived from input data (d). According to the applications on models, the deblurring functions which have the least error rate are $cov((dd^T))^{-1}$ and $cov((L^T L)^{-1})$, and it is seen that $cov(dd^T)^{-1}$ deblurring function reduces processing time %60 (for MD %80, for LSMD %40) respect to $cov(L^T L)^{-1}$. Besides, $(dd^T)^{-1}$ has been found to have the same error rates as the traditional deblurring function $(L^T L)^{-1}$ at a cost of %45 (for MD %18, for LSMD %65). It was also found that the LSMD procedure reduced the error rates by about 15% compared to the MD procedure.

Key Words: Seismic Migration, Least Squares Migration, Migration Deconvolution, Weightening

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1. 1.	Yüzey sismik verisi kaydetmek için (a) arazi (ConocoPhillips izniyle) ve (b) deniz (open-learn.open.ac.uk izniyle) ölçüm geometrileri.	6
Şekil 1. 2.	(a) yol kenarlarında ki yamaçlarda ve (b) denizel sismik kesitinde açıkça görülen jeolojik faylar (Schuster, 2010'dan düzenlenerek).....	7
Şekil 1. 3.	Sismik göç işleminin şekilsel gösterimi.	8
Şekil 1. 4.	a) Yeraltında 100 m derinlikte saçıcı bir kütle, b) sıfır ofset zaman kesiti (hız=1000 m/sn).....	8
Şekil 1. 5.	Yapısal karmaşıklık ve yanal hız değişimlerine göre öne çıkan göç teknikleri (Gürel, 2007' den uyarlanarak).....	10
Şekil 1. 6.	Sahada kayıt edilen verinin gerçek yansıma noktasına taşınması (Dondurur, 2009). Aşamalı olarak NMO-DMO ve göç uygulaması (solda), yığma öncesi ve sonrası göç uygulama aşamalarının temsili gösterimi (sağda).....	10
Şekil 2. 1.	A noktasında bulunan bir yansıtıcının saçılma geometrisi.....	11
Şekil 2. 2.	a) Göç işlemi için hesaplanan birkaç operatör, b) saçılma toplama ile göç çıkışı.	13
Şekil 2. 3.	Giriş verisi ve sıfır eklenmiş veri.	17
Şekil 2. 4.	Hesaplanan operatörler. 1. (sol sütun), 26. (orta sütun) ve 51. (sağ sütun) izlerin $t=1$ ms (üst satır), 165 ms (orta satır) ve 344 ms (alt satır) için hesaplanmış operatörler.	18
Şekil 2. 5.	26. İz için oluşturulan (L^T) matrisinin sıfır olmayan elemanları.	19
Şekil 2. 6.	1., 26. ve 51 izler için ters çözüm ($L^T L$) ⁻¹ Matrisleri.....	19
Şekil 2. 7.	Her bir iz için kullanılan giriş verisi. a) 1. iz için giriş verisi, b) 26. iz için giriş verisi ve c) 51. iz için giriş verisi.	20
Şekil 2. 8.	Kirchhoff göç çıkışı verisi ve göç işlem hataları.....	20
Şekil 2. 9.	Bu çalışmada kullanılan göç yöntemlerinin akış şemaları.	26
Şekil 3. 1.	a) Nokta saçıcı sismik veri modeli ve b) beklenen göç çıkışı.	28
Şekil 3. 2.	Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=75 m).....	29

Şekil 3. 3. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=75 m).....	30
Şekil 3. 4. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKKG çıkışları (yarı operatör boyu=75 m).....	31
Şekil 3. 5. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=75 m).....	32
Şekil 3. 6. Nokta saçılma modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=75 m).....	33
Şekil 3. 7. Nokta saçılma modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=75 m).....	33
Şekil 3. 8. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).....	35
Şekil 3. 9. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).....	36
Şekil 3. 10. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).....	37
Şekil 3. 11. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (hız=1000 m/sn).....	38
Şekil 3. 12. Nokta saçılma modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).....	39
Şekil 3. 13. Nokta saçılma modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).....	39
Şekil 3. 14. Nokta saçıcı model için (a) Kirchhoff göç, (b) eşlenik-gradyent yöntemi ile EKKG (c) Gauss-seidel yöntemi ile EKKG çıkışları.....	41

Şekil 3. 15. Nokta saçıcı model için $(L^T L)^{-1}$ netleştirme fonksiyonu ile (a) GD, (b) eşlenik-gradyent yöntemi ile EKKGD (c) Gauss-seidel yöntemi ile EKKGD çıkışları.	41
Şekil 3. 16. Nokta saçıcı model için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıkışları.	42
Şekil 3. 17. Nokta saçıcı model için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıkışları.	42
Şekil 3. 18. Nokta saçıcı model için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) gauss-seidel yöntemi ile EKKGD çıkışları.	43
Şekil 3. 19. Nokta saçıcı model için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıkışları.	43
Şekil 3. 20. Nokta saçıcı model için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıkışları.	44
Şekil 3. 21. Nokta saçıcı model için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıkışları.	44
Şekil 3. 22. Nokta saçıcı model için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD çıkışı c) Gauss-seidel yöntemi ile EKKGD çıkışları.	45
Şekil 3. 23. Nokta saçıcı model için uygulanan göç yöntemlerinin çözüm zamanları.....	47
Şekil 3. 24. Nokta saçıcı model için uygulanan göç yöntemlerinin karekök ortalama hata (kko) miktarları.....	47
Şekil 3. 25. Nokta saçılma modeli için eşlenik-gradyent ile uygulanan EKKGD yöntemi için ortalama yineleme sayıları.....	48
Şekil 3. 26. a) Papyon-kravat modeli verisi ve b) beklenen göç çıkışı.....	49
Şekil 3. 27. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=250 m).....	50
Şekil 3. 28. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit	

hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=250 m).....	51
Şekil 3. 29. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKKG çıkışları (yarı operatör boyu=250 m).....	52
Şekil 3. 30. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=250 m).....	53
Şekil 3. 31. Papyon-kravat modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=250 m).....	54
Şekil 3. 32. Papyon-kravat modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=250 m).....	54
Şekil 3. 33. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).....	56
Şekil 3. 34. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).....	57
Şekil 3. 35. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).....	58
Şekil 3. 36. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (hız=1000 m/sn).....	59
Şekil 3. 37. Papyon-kravat modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).....	60
Şekil 3. 38. Papyon-kravat modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).....	60
Şekil 3. 39. Papyon-kravat modeli için a) Kirchhoff göçü, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.....	62
Şekil 3. 40. Papyon-kravat modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	62

Şekil 3. 41. Papyon-kravat modeli için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	63
Şekil 3. 42. Papyon-kravat modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	63
Şekil 3. 43. Papyon-kravat modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	64
Şekil 3. 44. Papyon-kravat modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	64
Şekil 3. 45. Papyon-kravat modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	65
Şekil 3. 46. Papyon-kravat modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	65
Şekil 3. 47. Papyon-kravat modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	66
Şekil 3. 48. Papyon-kravat modeli için uygulanan göç yöntemlerinin çözüm zamanları. ..	68
Şekil 3. 49. Papyon-kravat modeli için uygulanan göç yöntemlerinin kök ortalama kare hata miktarları.	68
Şekil 3. 50. Papyon-kravat modeli için uygulanan eşlenik-gradyent (EKKGD) yöntemi için ortalama yineleme sayıları.	69
Şekil 3. 51. a) Fay modeli verisi ve b) beklenen göç çıkışı.	70
Şekil 3. 52. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=50 m).	72
Şekil 3. 53. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=50 m).	73

Şekil 3. 54. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKKG çıkışları (yarı operatör boyu=50 m).....	74
Şekil 3. 55. Fay saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=50 m).....	75
Şekil 3. 56. Fay modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=50 m). 76	
Şekil 3. 57. Fay modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=50 m).....	76
Şekil 3. 58. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).....	78
Şekil 3. 59. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).....	79
Şekil 3. 60. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).....	80
Şekil 3. 61. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (hız=1000 m/sn).....	81
Şekil 3. 62. Fay modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).....	82
Şekil 3. 63. Fay modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).....	82
Şekil 3. 64. Fay modeli için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.....	84
Şekil 3. 65. Fay modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	84
Şekil 3. 66. Fay modeli için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	85
Şekil 3. 67. Fay modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	85

Şekil 3. 68. Fay modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) Gauss-seidel yöntemi ile EKKGD çıkışları.	86
Şekil 3. 69. Fay modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	86
Şekil 3. 70. Fay modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	87
Şekil 3. 71. Fay modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışı....	87
Şekil 3. 72. Fay modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	88
Şekil 3. 73. Fay modeli için farklı netleştirme fonksiyonları ile uygulanan göç yöntemlerinin çözüm zamanları.	90
Şekil 3. 74. Fay modeli için farklı netleştirme fonksiyonları uygulanan göç yöntemlerinin kko hata miktarları.	90
Şekil 3. 75. Fay modeli için farklı netleştirme fonksiyonları ile uygulanan eşlenik-gradyent (EKKGD) yöntemi için ortalama yineleme sayıları.	91
Şekil 3. 76. a) SEG/EAGE tuz modeli, b) sıfır ofset kesiti.	92
Şekil 3. 77. Tuz modeli için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.....	94
Şekil 3. 78. Tuz modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD , b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	95
Şekil 3. 79. Tuz modeli için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	96
Şekil 3. 80. Tuz modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	97
Şekil 3. 81. Tuz modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	98
Şekil 3. 82. Tuz modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	99

Şekil 3. 83. Tuz modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	100
Şekil 3. 84. Tuz modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	101
Şekil 3. 85. Tuz modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	102
Şekil 3. 86. Tuz modeli için uygulanan göç yöntemlerinin çözüm zamanları.	104
Şekil 3. 87. Tuz modeli için uygulanan eşlenik-gradyent yöntemi ile EKKGD işlemi için ortalama yineleme sayıları.	104
Şekil 3. 88. a) Arazi verisi-1 sıfır ofset kesiti, b) karekök ortalama hız (rms) hız kesiti... 106	
Şekil 3. 89. Arazi verisi-1 için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.....	108
Şekil 3. 90. Arazi verisi-1 için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	109
Şekil 3. 91. Arazi verisi-1 için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	110
Şekil 3. 92. Arazi verisi-1 için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	111
Şekil 3. 93. Arazi verisi-1 için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	112
Şekil 3. 94. Arazi verisi-1 için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	113
Şekil 3. 95. Arazi verisi-1 için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.....	114
Şekil 3. 96. Arazi verisi-1 için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	115

Şekil 3. 97. Arazi verisi-1 için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	116
Şekil 3. 98. Arazi verisi-1 için uygulanan göç yöntemlerinin çözüm zamanları.....	118
Şekil 3. 99. Arazi verisi-1 için uygulanan eşlenik-gradyent yönteminin ortalama yineleme sayıları.....	118
Şekil 3. 100. a) Arazi verisi-2 sıfır ofset kesiti, b) karekök ortalama hız (rms) hız kesiti.	119
Şekil 3. 101. Arazi verisi-2 için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.....	121
Şekil 3. 102. Arazi verisi-2 için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	122
Şekil 3. 103. Arazi verisi-2 için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	123
Şekil 3. 104. Arazi verisi-2 için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	124
Şekil 3. 105. Arazi verisi-2 için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	125
Şekil 3. 106. Arazi verisi-2 için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	126
Şekil 3. 107. Arazi verisi-2 için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	127
Şekil 3. 108. Arazi verisi-2 için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	128
Şekil 3. 109. Arazi verisi-2 için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.	129
Şekil 3. 110. Arazi verisi-2 için uygulanan göç yöntemlerinin çözüm zamanları.....	131

Şekil 3. 111. Arazi verisi-2 için uygulanan eşlenik-gradyent yönteminin ortalama yineleme sayıları.....	131
Şekil 4. 1. Papyon-kravat modeli için eşlenik-gradyent yöntemi ile, a) en fazla yineleme sayısı 20, b) en fazla yineleme sayısı 30 ve c) en fazla yineleme sayısı 766, ortalama yineleme sayısı 540 olan EKKG çıkışları.....	136
Şekil 4. 2. Fay modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile (a) sonlandırma kriteri $\varepsilon = 2.2204e - 16$ için (b) sonlandırma kriteri $\varepsilon = 4.9304e - 32$ için elde edilen göç çıkışları.	138
Şekil 4. 3. Düzenleme ve sönüm parametresi testleri. Nokta saçılma modeli için a) $(L^T L)^{-1}$, b) $(L^T L + I)^{-1}$, c) $(L^T L + I * 1000)^{-1}$, d) $(L^T L + Q)^{-1}$, e) $(L^T L + Q * 10000^2)^{-1}$, f) $(dd^T)^{-1}$, g) $(dd^T + I)^{-1}$, h) $(dd^T + I * 1000)^{-1}$, i) $(dd^T + Q)^{-1}$ ve j) $(dd^T + Q * 10000^2)^{-1}$ ile elde edilen EKKGD çıkışları.	140
Şekil 4. 4. Nokta saçılma modeli için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.	141
Şekil 4. 5. Tuz modeli için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.	141
Şekil 4. 6. Arazi verisi-1 için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.	142
Şekil 4. 7. Arazi verisi-2 için a) kazanç uygulamadan önce $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı ve kazanç uygulaması, b) kazanç uygulamadan sonra $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı.....	143
Şekil 4. 8. Tuz modeli için a) Giriş verisi, b) a' nin $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı ve kazanç uygulaması, c) kazanç uygulamasından sonra giriş verisi d) c' nin $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı.	144
Şekil 4. 9. Tuz modeli için a) Kazanç uygulanmadan elde edilen göç çıkışının (Şekil 4.8b) 12.5 km deki izi, b) kazanç uygulandıktan sonra elde edilen göç çıkışının (Şekil 4.8d) 12.5 km deki izi.	144
Şekil 4. 10. Nokta saçılma modeli için, a) Kirchhoff göç çıkışı, (b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve (e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.	145
Şekil 4. 11. Nokta saçılma modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.	146

- Şekil 4. 12. Nokta saçılma modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 147
- Şekil 4. 13. Papyon-kravat modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları. 148
- Şekil 4. 14. Papyon-kravat modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları. 149
- Şekil 4. 15. Papyon-kravat modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 150
- Şekil 4. 16. Fay modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları..... 151
- Şekil 4. 17. Fay modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları..... 152
- Şekil 4. 18. Fay modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 153
- Şekil 4. 19. Tuz modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları..... 154
- Şekil 4. 20. Tuz modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları..... 155
- Şekil 4. 21. Tuz modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 156
- Şekil 4. 22. Arazi verisi-1 için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları..... 157
- Şekil 4. 23. Arazi verisi-1 Modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları. 158

- Şekil 4. 24. Arazi verisi-1 için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 159
- Şekil 4. 25. Arazi verisi-2 için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları..... 160
- Şekil 4. 26. Arazi verisi-2 Modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları. 161
- Şekil 4. 27. Arazi verisi-2 için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları..... 162

TABLolar DİZİNİ

Sayfa No

Tablo 1. 1.	Literatürde sıklıkla kullanılan göç yöntemleri.	9
Tablo 3. 1.	Nokta saçılma modeli için Netleştirme fonksiyonlarına göre çözüm zamanları açısından maliyet karşılaştırması. (İz sayısı: 51, zaman örnek sayısı: 344, yarı operatör boyu: 75 m, 15 iz).....	45
Tablo 3. 2.	Nokta saçılma modeli için Netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.....	46
Tablo 3. 3.	Papyon-kravat modeli için netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 100, zaman örnek sayısı: 766, yarı operatör boyu: 250 m, 50iz)...	66
Tablo 3. 4.	Papyon-kravat modeli için netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.....	67
Tablo 3. 5.	Fay modeli için netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 100, zaman örnek sayısı: 390, yarı operatör boyu: 100 m, 10iz).....	88
Tablo 3. 6.	Fay modeli için netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.	89
Tablo 3. 7.	Tuz modeli için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 801, zaman örnek sayısı: 350, yarı operatör boyu: 3000 m, 75 iz).....	103
Tablo 3. 8.	Arazi verilerinin veri toplama parametreleri.....	105
Tablo 3. 9.	Arazi verilerine uygulanan veri-işlem adımları.	105
Tablo 3. 10.	Arazi verisi-1 için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 801, zaman örnek sayısı: 801, yarı operatör boyu: 468.75 m, 75 iz)	117
Tablo 3. 11.	Arazi verisi-2 için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 986, zaman örnek sayısı: 701, yarı operatör boyu: 625 m, 50iz)	130
Tablo 4. 1.	EKKG ve $(LTL) - 1$, $(ddT) - 1$, $cov((ddT) - 1)$ ve $cov((LTL) - 1)$ netleştirme fonksiyonlarının kullanılması ile GD ve EKKGD çıkışı hatalarının Kirchhoff göç çıkışı hatalarına oranları. Kirchhoff göç çıkışı hatası 1 birim alınarak oranlar hesaplanmıştır.	133
Tablo 4. 2.	Tüm veri setleri için Kirchhoff integral göçü, Matris ortamında Kirchhoff göçü ve maliyet oranları.....	135
Tablo 4. 3.	EKKG ve $(LTL) - 1$, $(ddT) - 1$, $cov(ddT - 1)$ ve $cov(LTL - 1)$ netleştirme fonksiyonlarının kullanılması ile GD ve EKKGD çıkışı çözüm zamanlarının Kirchhoff göç çıkışı çözüm zamanına oranları. Kirchhoff göç çıkışı toplam zamanı 1 birim alınarak diğer oranlar hesaplanmıştır.....	135

SEMBOLLER DİZİNİ

EKK	: En Küçük Kareler
EKKG	: En Küçük Kareler Göçü
EKKGD	: En Küçük Kareler Göç Dekonvolüsyonu
F	: Netleştirme fonksiyonu
GD	: Göç Dekonvolüsyonu
J(m)	: Amaç fonksiyonu
L	: Göç operatörü
m	: Model
m_d	: Göç Dekonvolüsyonu çıkışı
m_{ekk}	: En Küçük Kareler Göç çıkışı
m_{mig}	: Göç çıkışı
m_{sekk}	: Sönümlü En Küçük Kareler Göç çıkışı
n	: Gürültü bileşeni
N	: Veri sayısı
N_t	: Bir izdeki zaman örnek sayısı
N_x	: Bir verideki uzaklık örnek sayısı
N_{px}	: Pencere boyu
N_z	: Veriye eklenecek sıfır sayısı
nl	: Göç yarı operatör Boyu
t	: Zaman
P_{in}	: Göç işlemine giriş değeri
P_{out}	: Göç işleminden çıkış değeri
r	: Uzaklık
τ	: Başlangıç zamanı
V_{rms}	: Kök ortalama hız
$R(m)$: Düzenleme (regularizasyon) normu
Q	: Düzenleme (regularizasyon) parametresi
x	: Uzaklık (sismik hat boyunca)
Θ	: Yüzeyle yapılan açı
Z	: Derinlik
γ	: Sönüm katsayısı

- β : Ölçekleme parametresi
 ϵ : Düzenleme sabiti
 δ : Düzenleme parametresi katsayısı
 ε : Eşlenik-gradyent yöntemi sonlandırma kriteri.



1. GİRİŞ

1.1. Genel Bilgiler

Yansıma sismiği yöntemi; yüzeyden kaydedilen sismik kayıtlardan yer içinin yapısal ve stratigrafik özelliklerini detaylı olarak belirlemenin ve dolayısıyla hidrokarbon (petrol, doğal gaz ve kömür) araştırmalarının vazgeçilmez ve en güçlü yöntemidir. Bununla birlikte, sismik verinin jeolojik olarak yorumlanabilir hale getirilmesi için sayısal olarak işlenmesi gerekir. Sismik göç (seismic migration) bu işlemlerin en son aşamasında uygulanan bir işlem olup, bu işlemin kullanılmasındaki temel amaç sismik dalga yayılım teorisini göz önüne alarak sismik olayları doğru yansıma noktalarına taşımaktır. Yeraltı yansıma noktalarının oluşturduğu yansıtıcı yüzeyler aynı zamanda sismik hız sınırlarıdır ve bu hız sınırları jeolojik yapının görüntüsünü temsil eder. Bununla birlikte, bu hız sınırları sismik veri işlem sırasında birbirine paralel ve düzlem olduğu kabul edilmektedir. Ancak doğada yatay tabakalanma pek mümkün olmadığından dolayı, ortak derinlik noktası (ODN) yığma kesitleri gerçek tabaka sınırlarını göstermez. Çünkü yeraltındaki kıvrımlı, intrüsif (volkanik, tuz domları, şeyl diapirler, vb.) kırıklı (normal, ters, doğrultu atımlı faylar) ve eğimli karmaşık jeolojik ortamlardaki yansıtıcılardan dolayı, sismik yığma kesitlerinde gerçek konumlarından uzakta olan yansıma genlikleri, yorumlanması zor sismik yığma kesitlerine neden olurlar. Bu nedenle, yansıma genlikleri gerçek yansıma noktalarına taşınması gerekmektedir. Bu sebeple sismik yığma kesitlerine göç işlemi uygulanarak yansıma genlikleri gerçek yeraltı konumlarına taşınır. Böylece gerçek yeraltı jeolojisi sismik olarak görüntülenebilir.

Günümüzde, sismik göç, sismik veri işlem akışında temel bir adımdır. Ancak yaklaşık 30 yıl öncesine kadar sismik veri işlemin son aşamasında çoğunlukla isteğe bağlı olarak kullanılan bir işlem idi. Ölçekleme, dekonvolüsyon, statik düzeltme, hız analizi, eğimli kayma zamanı (Dip moveout-DMO) düzeltmeleri ve ortak orta nokta (common midpoint-CMP) gibi işlemlerden sonra elde edilen yığma kesitine (stacked section) yorum için bir yapısal görüntü sağlama amacıyla göç uygulanmaktadır. Yığma kesitleri üzerindeki saçılmalar, ince tabakalar (pinch-outs) ve resif (reefs, sığ kayalık) kenarları gibi detay hedefleri örttüğü durumlarda, stratigrafik görüntüleme için sismik göç kullanılması gerekmektedir. Günümüzde, tüm sismik verilere yığma öncesi veya sonrası göç işlemi

uygulanarak genliklerin ve diğ er niteliklerin daha detaylı analizleri yapılmaktadır. İlk olarak 1920'lerde kullanılmaya başlanan sismik gö ç 1960 yılına kadar grafiksel olarak yapılmaktaydı. Bu, saçılma toplaması ve ışın teorisine bağı lı olarak dalga cephesi gö çü ile devam ettirilmiştir. 1970 ve sonrasında sismik gö ç uygulamalarında, geliş en teknoloji ile birçok parametreyi kapsayan sayısal ç ö züm yöntemleri kullanılmaya (sonlu farklar, frekans-dalgasayısı, Fourier, Kirchhoff saçılma toplamı, ters zaman gö çü) baş lanmıştır (Gazdag, 1978; Stolt, 1978; Gray ve diğ ., 2001). 1960'larda sismik veride sayısal sinyal işleme tekniklerinin uygulanmasının geliş mesiyle birlikte, sayısal saçılma yığı ma işlemini (Schinder, 1971) içeren, dalga denklemi tabanlı ilk sayısal sismik gö ç işlemini tasarlanmıştır (Mayne 1962). Bu fikir ilk olarak 1970'li yıllarda Stanford Exploration Project'de Jon Claerbout ve onun ö ğrencileri tarafından açıklanmıştır. Onlar sismik gö ç uygulaması olarak yaklaşık dalga denkleminin sonlu farklar (finite difference) ç ö zümünü üretmişlerdir (Claerbout ve Doherty, 1972). Kirchhoff dalga denklemi gö çü (Schinder, 1978) ve frekans dalgasayısı gö ç uygulamaları (Gazdag, 1978; Stolt, 1978) daha sonra kısa sürede ortaya çı kmıştır. Bu yöntemler ilk olarak zaman gö çü olarak geliştirilmiş, daha sonra yanal hız de ğişimi durumunda sismik gö çün doğ ruluğunun artırılma ihtiyacına istinaden zaman gö çü teknikleri birkaç yıl içinde derinlik gö çüne dönüştürülmüştür. Bu yıllarda, ters zaman gö çü (reverse time) geliştirilmiştir (Baysal ve diğ ., 1983; McMechan, 1983; Whitmore, 1983; Gray ve diğ ., 2001). Son 20 yıldır, yöntemlerle ilgili gelişmeler üç boyutta (3B) ve yığı ma öncesi sismik gö ç alanında onların doğ rulu ğu ve etkinli ği de göz önüne alınarak hızla devam etmektedir. Sismik gö ç kesitleri yorumlanacak kesitler olduğ undan, yeraltının sismik görüntüsünü verece ği göz önüne alındığında bir çeş it ters ç ö züm işleminin (Tarantola, 1984), yani "sismik gö ç=gerçek yeraltı görüntüsü" veya yerin yansıtabilirlik dağı lımının bir görünümü olarak da de ğerlendirilmektedir. Bununla birlikte sismik gö ç işleminin kendi hatasını (artifacts) de oluşturmaktadır. Bu hatalardan kurtulmak için, son yıllarda sismik gö ç algoritmaları en küçük kareler (EKK) hatasının analizine dayalı olarak geliştirilmiş ve En Küçük Kareler Gö ç (EKKG) metodu geliştirilmiştir (Nemeth ve diğ ., 1999). Bunun yanında EKKG ile aynı amaç için kullanılan Gö ç Dekonvolüsyonu (GD) işleminin (Hu ve Schuster, 1998) geliştirilmiştir. EKKG ve GD işlemleri birleş ik olarak kullanılarak (En Küçük Kareler Gö ç Dekonvolüsyonu, EKKGD) iterasyon sayıları azaltılabilmektedir (Aoki ve Schuster 2009) . Bu olay teknolojinin geliş imi ile birlikte siyah-beyaz televizyonlardan renkli televizyonlara ve günümüzde yüksek ç ö zünürlüklü (High Definition-HD) televizyonlara geçiş e benzerdir.

Standart sismik göçten türetilen bir görüntü genel olarak tam doğru değildir. Ancak, standart bir sismik göçten başlayarak bir görüntü EKK anlamında yinelemeli olarak en iyileştirilebilir. EKKG sismik verinin görüntülenmesi için standart göç tekniklerine bir alternatif olarak literatürde yer almıştır (Nemeth ve diğ., 1999). Kuhl ve Sacchi (2003) hız modelinde küçük hatalar olduğu zamanda bile EKKG' nün yüksek kaliteli genlik bilgisini sağlayabildiğini göstermişlerdir. Bununla birlikte, Nemeth ve diğ. (1999) tamamlanmamış sismik ve yer radarı verisinin görüntü kalitesini ve Clapp (2005), Valenciano ve diğ. (2006), Valenciano (2008) karmaşık örtü tabakası altındaki haznelerin görüntü kalitesinin EKKG sayesinde, standart sismik göç ile görüntülemeye göre çok daha güvenilir ve kaliteli sonuçlar sağladığını göstermişlerdir.

EKKG algoritmasının avantajı uygun sismik kesit üretebilmek için bir düzenleme (regularization) parametresine sahip olmasıdır (Robello ve Sacchi, 2010). Örneğin örnekleme problemi olan bir veride bu etkiyi giderebilir (Kuhl ve Sacchi, 2003). EKKG uygulaması yığma öncesi sismik verilere uygulandığında elde edilen sonuçların kalitesinin son derece başarılı olduğu gösterilmiştir (Nemeth ve diğ., 1999). Aoki ve Schuster (2009) netleştirme (deblurring) süzgeçleri ile az sayıda yineleme ile en uygun çözüme ulaşabileceğini belirtmiştir. Bunun yanında ağırlıklandırma sabitleri düşey çözünürlüğü de artırılabilir (Wang ve Sacchi, 2007). Rebollo ve Sacchi (2010) daha az karmaşık ortamlarda eğim bağımlı sabitler ile yüksek çözünürlük sağlanabileceğini göstermiştir. Kaplan (2010), EKKG yönteminde Bayes teoremini kullanarak bir yakınsama sağlamaya çalışmıştır. Drineas ve diğ. (2011) farklı bir dönüşüm tekniği (Randomized Hadamard Transform) uygulayarak iki farklı metot ile iki aşamada EKKG işleminin maliyetini azaltmışlardır. Statton (2013) önkestirim parametresi ile EKK derinlik göçü çıkışının göç hatalarını azaltmıştır. Wei ve Zhou (2014) stokastik eşlenik-gradyent metodu ile büyük boyutlu sismik veriler için EKKG yönteminin genlik düzenlemesini yaparak maliyet bakımından uygulanabilir hale getirmişlerdir. Luo ve Hale (2014) tahmin edilen genlik ile gözlenen genlik arasındaki farktan elde ettikleri hata fonksiyonu ile EKKG işleminde hatalı hız fonksiyonlarında dahi odaklamayı başarabilmişlerdir. Ganssle (2015) sabitleştirilmiş (Stabilized) EKKG yöntemi ile hem yineleme sayısını düşürmüş hem de hata oranını 1/10 a kadar düşürmüştür. Huo (2015) ağırlıklandırılmış eşlenik-gradyent yöntemi ile EKKG işlemini hızlandırmıştır. Wu ve diğ (2016) l_0 düzenleme parametresi ile EKKG çıkışının ayrımlılığını arttırmıştır. EKK yönteminde kullanılacak farklı ağırlıklandırma (veya şartlandırma) fonksiyonları sayesinde yineleme sayısı ve dolayısıyla işlem zamanı ve

maliyeti azaltılabilir (Whitmore, 1983; Tibshirani, 1994; Sacchi, 1997; Schmidt, 2005; Aoki ve Schuster, 2009; Kaplan 2010).

EKKG yöntemine alternatif olarak gösterilen GDişlemi göç hatalarını bastırmak ve yüksek ayrımlılığa sahip göç çıkışları üretmek için kullanılmaktadır. Hu ve Schuster (1998) GD işlemleri ile veri toplama gürültülerini (acquisition footprint) kısmen dekonvole ederek uzaklık boyutunda ayrımlılığını arttırmış, göç hatalarını zayıflatmış ve göç çıkışının kalitesini arttırmışlardır. Hu ve diğ., (2001) yığma sonrası sismik verilerde GD uygulayarak göç çıkışının ayrımlılığını ve kalitesini arttırmışlardır. Yu ve Schuster (2003) GD işleminin EKKG işlemine oranla daha az maliyetli olduğunu belirtmiştir. Yu (2004) GD işlemini 3 boyutlu arazi verisi üzerinde test ederek ayrımlılığını arttırdığını ve göç hatalarını bastırdığını göstermiştir. GD yönteminin amacı veri toplama geometrisinin örneklenememesi, sınırlı bir kayıt aralığına sahip olması ve yüksek hız farklılıkları yeraltının net bir şekilde görüntülenmesinin önüne geçmektedir (Yu ve diğ., 2006). Luo (2006) 3-B Deniz verileri üzerinde GD işlemini uygulayarak yapısal olayları yüksek ayrımlılıkta görüntülemiştir. Wang ve diğ., (2009) eşlenik-gradyent metodunun düzenlenmiş (regularization) yinelemeli versiyonunu GD işleminin uyguladıklarıdır. Wang ve Yang (2010) düzenlenmiş En Küçük Kareler metodunun yerine Nonmonotone gradyent metodunu kullanarak GD işleminin maliyetini azaltmışlardır. Li ve diğ., (2011) Memoryless Quasi-Newton-Simulated Annealing (MQN-SA) algoritması ile çözüm maliyetini düşürmüşlerdir. Genel olarak En Küçük Kareler Göç işlemi ile GD işlemi aynı amaçlar için kullanılmaktadır. Aoki ve Schuster (2009) EKKG işleminde netleştirme fonksiyonları kullanarak EKKGD işlemini kullanmıştır. Yani EKKG ile GD işlemini birleşik olarak çözmüştür.

Sismik göç yöntemleri yığma öncesi ve sonrası olmak üzere, derinlik ve zaman ortamı göç şeklinde gruplandırılabilir. Bununla birlikte, yığma sonrası sismik göç teknikleri yaygın olarak hidrokarbon araştırmasında kullanılmaktadır ve karmaşık yeraltı yapılarının görüntülenmesinde önemli rol oynar. Bir göç algoritmasından beklenen ilk özellik ekonomik (işlem zamanının az) olmasıdır. Bu çalışma kapsamında ele alınan En Küçük Kareler Göç (EKKG) yöntemi bilinen EKK ters çözüm algoritmasını kullandığından ve en uygun çözüme yinelemeli olarak ulaştığından dolayı, diğer göç algoritmalarına göre çok daha yavaş fakat göç hatalarını en küçükleğinden dolayı çözünürlüğü çok daha yüksektir. Aynı şekilde, GD işlemi de netleştirme (ağırlıklandırma) fonksiyonu kullanarak göç çıkışındaki hataları minimize ederek ayrımlılığını arttırmaktadır.

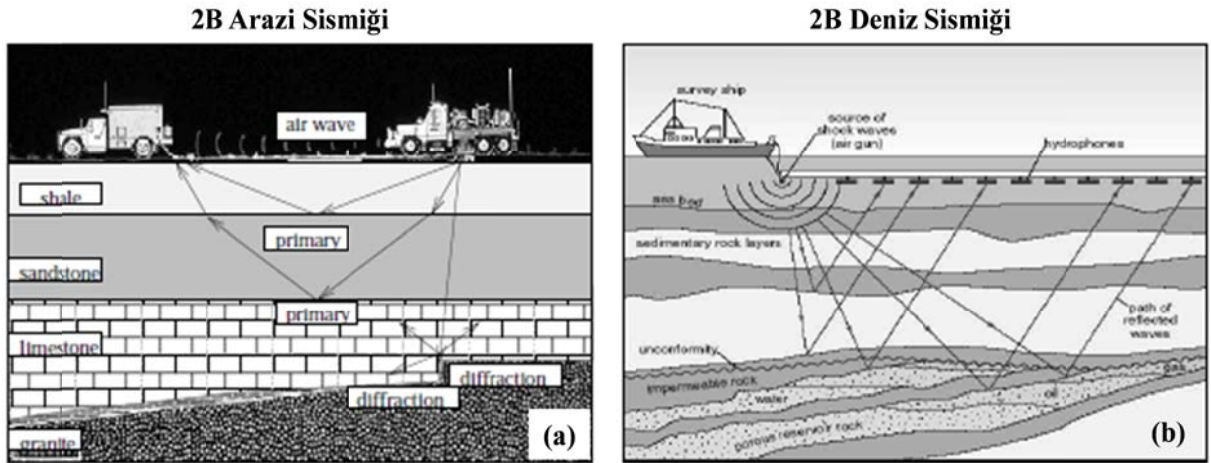
Yukarıda kısaca özetlendiği gibi EKKG ve GD işlemleri kaliteli çıkışlar ürettiğinden dolayı son yıllarda sıklıkla tercih edilen yöntemler olmuştur. Yöntemlerin uygulamada eksiklikleri ve üstünlükleri mevcut olup, her kullanıcı yöntemi farklı parametreler ve uygulama algoritmaları geliştirerek kullanmıştır (Gray ve diğ., 2001). Bu yöntemlerin çözüm zamanları maliyet ile doğru orantılıdır (çözüm zamanı = maliyet). Bu nedenle hem maliyet, hem de ürettiği sonuçlarının doğruluğu açısından halen incelenmekte ve irdelenmektedir. Göç hatalarını gidermeye yönelik EKKG çalışmaları son beş yılda Kirchhoff göç tekniğinden Ters Zaman Göçü (Reverse Time Migration) göç tekniğine ve özellikle yığma öncesi uygulamalara yönelmiştir. Ancak, genel olarak yığma öncesi sismik göç, en hızlı çalışan algoritmalar kullanılsa dahi, göç işleminin doğasından kaynaklı olarak hem veri hacmi hem de hesaplama zamanı açısından oldukça maliyetlidir. Bu maliyetlerin azaltılması amacıyla, çoğunlukla yığma sonrası sismik göç uygulamaları daha çok tercih edilmektedir, ancak kullanılan göç algoritmaları ne kadar güçlü olsa da yığma öncesine göre elde edilen sonuçların kaliteleri düşük olabilir. EKKG ve GD uygulamalarının yığma sonrası verilere uygulamasındaki etkinliği artırılarak yığma öncesinde ulaşılan başarılı sonuçlar elde edilebilir. Bunun için EKKG, GD ve En Küçük Kareler Göç Dekonvolüsyonu (EKKGD) algoritmalarının etkinliklerini artıracak ve hesaplama maliyetlerini azaltacak uygulamaların (çözüme ağırlıklandırma ve/veya şartlandırma fonksiyonlarının eklenmesi) araştırılması ve en iyilerinin belirlenerek kıyaslanması önemli bir araştırma konusu olmuştur. Bu nedenle, bu çalışma kapsamında, yığma sonrası Kirchhoff göç tekniği ile EKKG yönteminde çözüme farklı ağırlıklandırma fonksiyonları eklenmesi durumunda, göç sonuçlarının kalitesinin nasıl etkilendiği araştırılması amaçlanmıştır. Bu amaca ulaşmak için,

- GD, EKKG ve EKKGD yöntemlerinin genel uygulaması ve sismik göç işleminde kullanım ilkeleri belirlenmiştir,
- GD, EKKG ve EKKGD algoritması oluşturulmuş ve MatLab programlama dilinde yazılmıştır (EK-C),
- Yazılan programın basit yer modelleri (nokta saçıcı, eğimli yansıtıcı, süreksizlikler) üzerinde testleri yapılmıştır,
- Göç algoritmasında modelleme operatörünün göç sonuç kalitesine olan katkısı irdelenmiştir,
- EKKG algoritmasında matris yapılarının tersinin hesaplanmasında eşlenik-gradyent (EK-A,EK-D) ve gauss-seidel (EK-B,EK-F) teknikleri değerlendirilmiştir,

- Tüm göç algoritmaları için ağırlıklandırma (deblurring) fonksiyonlarının kullanılması durumundaki karşılaştırmalar yapılmıştır,
- Yukarıdaki tüm testlerden sonra, algoritmalar tam olarak güvenli çalıştığını test etmek için bilinen EAGE/SEG tuz modeli üzerinde denemiştir,
- Son olarak algoritmalar iki farklı arazi verisine uygulanmıştır.

1.2. Sismik Yansıma Verisinin Önemi

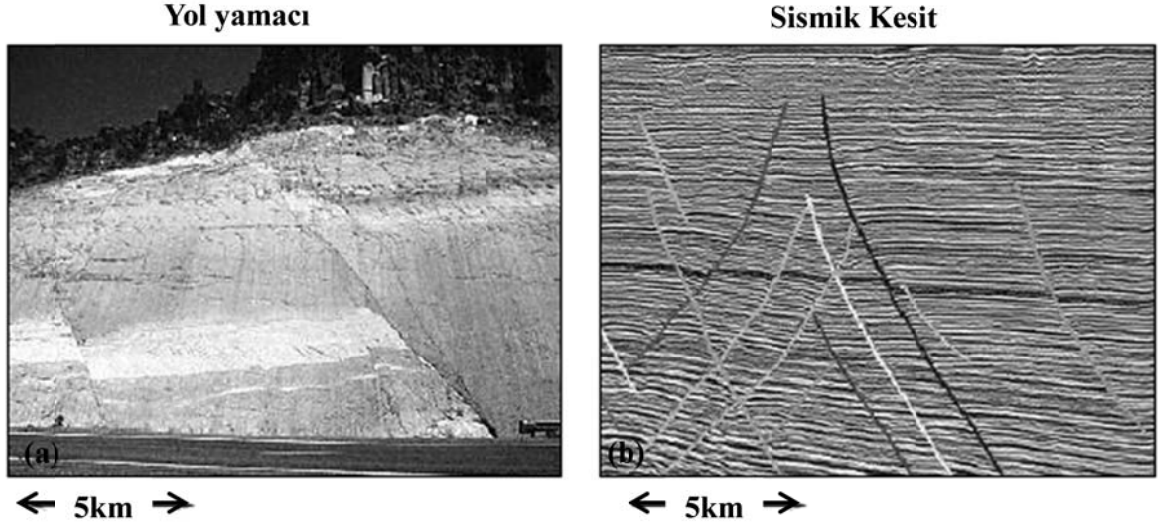
Araştırma sismolojisinin birincil amacı sismik olarak yerin yansıtılabilirlik dağılımını görüntüleyerek petrol ve doğal gaz haznelerinin yeraltı konumlarını belirlemektir. Bunun için ideal olarak Şekil 1.1' deki benzer bir sismik ölçüm gerçekleştirilir. Şekilde görüldüğü üzere kaynaklar sismik dalgaları oluşturur ve birincil yansımalar bir hat boyunca dizilen alıcılarda kaydedilir ve Ortak Orta Nokta (OON, CMP) grupları oluşturulur. Daha sonra ofset etkisini gidermek için Normal Kayma Zamanı (NKZ, NMO) düzeltmesi uygulanır ve aynı CMP grubuna ait sismik sinyaller zaman eksenini boyunca toplanarak sismik yığıma kesitleri oluşturulur.



Şekil 1. 1. Yüzeysel sismik verisi kaydetmek için (a) arazi (ConocoPhillips izniyle) ve (b) deniz (open-learn.open.ac.uk izniyle) ölçüm geometrileri.

Yeraltının yatay ve homojen tabakalardan oluştuğu varsayılırsa sismik yığıma kesitleri yeraltı görüntüsünü temsil ettiği söylenebilir. Aksi durumda yeraltı yapısı eğimli tabakalanma, yanal hız farklılıkları, süreksizlikler içeriyorsa, sismik yığıma kesiti yeraltının gerçek bir görüntüsünü sunamayacaktır. Şekil 1.2 de yol yamacı ve bu yol yamacına

paralel olarak denizde alınmış bir sismik kesit görülmektedir. Buradan da anlaşılacağı gibi doğada homojen yatay tabakalanma ile karşılaşmak çok güçtür. Bu nedenle sismik göç işlemi, sismik görüntüleme için hayati önem taşımaktadır.



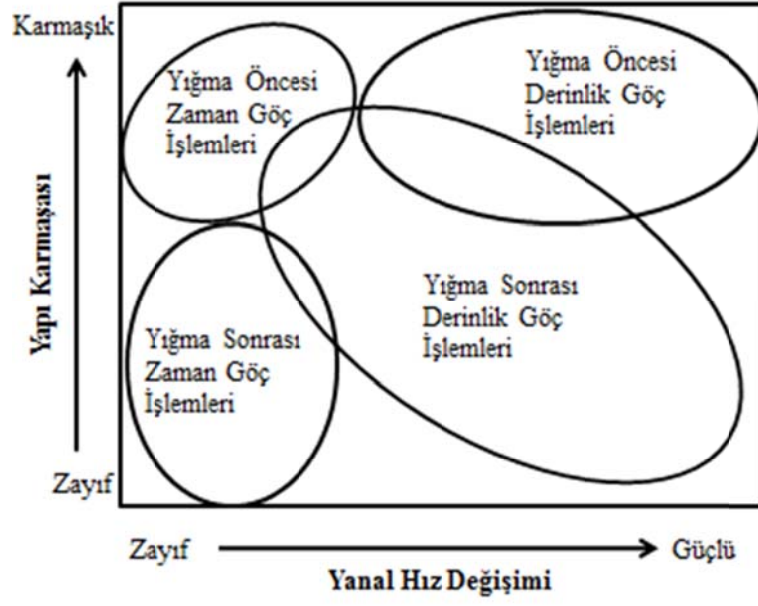
Şekil 1. 2. (a) yol kenarlarında ki yamaçlarda ve (b) denizel sismik kesitinde açıkça görülen jeolojik faylar (Schuster, 2010'dan düzenlenerek).

Şekil 1.3 yeraltında $|CD|$ eğimli yansıtıcı olması durumunda yeryüzünden kaydedilen sismik zaman kesitinde $|C'D'|$ şeklinde görünmektedir. Bu yansıtıcının gerçek yeraltı konumuna taşınması için sismik göç işlemi uygulanır. Sismik göç işlemi hız fonksiyonu yardımıyla hesaplanan hiperbolik yollar boyunca sismik yansıma genliklerini $|C'D'|$ konumundan, gerçek yeraltı konumuna $|CD|$ taşınması işlemidir. Görüldüğü gibi yansıtıcı seviye hiperbolik yollar boyunca eğim yukarı yönde taşınması gerekir. Sismik göç işleminin bir diğer önemli özelliği de saçılma dalgalarını bir noktaya dönüştürmesidir. Şekil 1.4 yeraltında bir saçıcı noktayı ve bunun sismik tepkisini göstermektedir. Şekil üzerinde görüldüğü gibi saçıcı noktanın ürettiği saçılma dalgası hiperbolik bir olay gibi görülmektedir. Bu hiperbolik olay ancak hiperbolik yol boyunca taşınarak gerçek yeraltı pozisyonuna yani saçıcı noktaya dönüştürülebilir. Bu hiperbolik yol ise hızın bir fonksiyonudur.

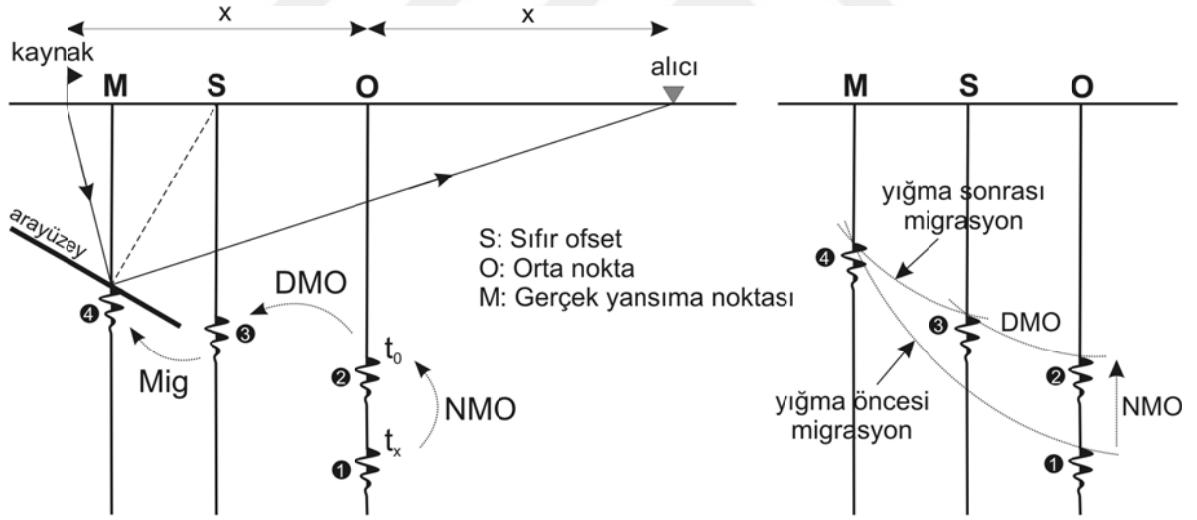
gerek kalmadan yansıma genliğini gerçek konumuna taşıyabilmektedir. Bu nedenle NMO ve DMO işlemlerinin hatalarını gidermiş olur ve karmaşık jeolojiye sahip ortamlarda daha doğru sonuçlar üretebilir. Hangi yol ile göç işlemi gerçekleştirileceğine karar verildikten sonra göç tekniği seçilmelidir. Tablo 1 sıklıkla kullanılan göç yöntemlerini göstermektedir. Göç yöntemlerinin de birbirlerine göre avantaj ve dezavantajları vardır. Literatürde en çok kullanılan göç yöntemi Kirchhoff göçüdür ve bütün paket programlarda mevcuttur.

Tablo 1. 1. Literatürde sıklıkla kullanılan göç yöntemleri.

GÖÇ YÖNTEMLERİ
Saçılma Toplaması (Diffraction Summation)
Kirchhoff Göçü (Kirchoff Migration)
Sonlu Farklar (Finite Difference Migration)
Ters Zaman Göçü (Reverse Time Migration, RTM)
Faz Kayması Göçü (Phase Shift Migration)
Faz Kayması Artı Enterpolasyon Göçü (Phase Shift Plus Interpolation, PSPI Migration)
Işın Yolu Göçü (Beam Migration)
Frekans-Uzaklık Göçü (Frequency-Space Migration)
Frekans-Dalgasayısı Göçü (Frequency-Wavenumber Migration)
Stolt Göçü (Stolt Migration)
Fourier Sonlu Farklar Göçü (Fourier Finite Difference Migration)



Şekil 1. 5. Yapısal karmaşıklık ve yanal hız değişimlerine göre öne çıkan göç teknikleri (Gürel, 2007' den uyarlanarak).



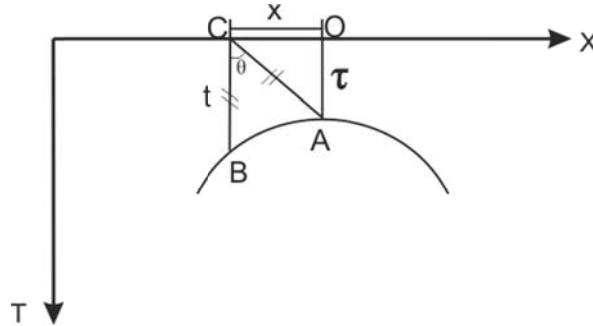
Şekil 1. 6. Sahada kayıt edilen verinin gerçek yansıma noktasına taşınması (Dondurur, 2009). Aşamalı olarak NMO-DMO ve göç uygulaması (solda), yığma öncesi ve sonrası göç uygulama aşamalarının temsili gösterimi (sağda).

2. YÖNTEM

2.1. Göç İşlemi

Yatay tabakalı durumdan farklı olan temel jeolojik yapılar; eğimli yansıtıcılar, saçıcılar (intrüsif, resifal, dom, dayklar, fay kenarları,vb.), antiklinal ve senklinaller olarak verilebilir. Bu tür yapılardan gelen yansıma genlikleri sismik veri üzerinde doğru yerlerinde görülmezler ve sismik veriyi karmaşıktırır. Dolayısıyla sismik göç zorunlu bir uygulama olarak karşımıza çıkar. Sismik göç işlemi saçılmaları tek bir noktaya toplar, eğimli olayları eğim yönünde yukarıya taşır, antiklinalleri tepe noktasına doğru daraltır, senklinalleri genişletir (papyon kravat etkisini giderir), böylece birbirini kesen olayları birbirinden ayırır, yansıtıcının geometrisinden dolayı sismik kesitteki genlik azalmasını veya artmasını düzeltir ve ayrıca dalgacığın genlik, frekans ve faz bilgilerini korur (Baysal ve diğ., 1983). Buna ek olarak, kırıklı tabakalar ve nokta yansıtıcılardan oluşmuş saçılmaları birer noktaya dönüştürür (pratikte göç, hiperbollerini bir odağa dönüştürür) ve Fresnel bölgesi üzerinde yayılmış enerjiyi odaklayarak (yani Fresnel yarıçapını küçültür) sismik kesitin uzaysal (yanal) çözünürlüğünü artırır (Kanbur, 2002).

Sismik göç işlemi için öncelikle saçılma olayının geometrisi önem taşımaktadır. Yeraltında herhangi bir A noktası saçıcı nokta olarak düşünülürse; x uzaklıktaki alıcıya ulaşma zamanı B noktası olacaktır ve bu saçıcı nokta sismik hat boyunca bir saçılma hiperbolü oluşturacaktır (Şekil 2.1). Bu saçılma hiperbolü; uzaklık ve ortamın hızının bir fonksiyonudur.



Şekil 2. 1. A noktasında bulunan bir yansıtıcının saçılma geometrisi.

Burada, C: Atış ve alıcının bulunduğu nokta, O: yansıtıcı noktanın (A) yeryüzündeki izdüşümü, B: yansıtıcı noktadan (A) yansıyarak alıcıda (C) kaydedilen yansıma genliğinin sismik kesitte atandığı nokta, t: varış zamanı, τ : yansıtıcının derinliğini gösteren t_0 zamanı. Şekil 2.4 deki CAO üçgeninden yola çıkılırsa, saçılma hiperbolü denklemi:

$$t^2 = \tau^2 + \frac{4x^2}{V_{rms}^2} \quad (1)$$

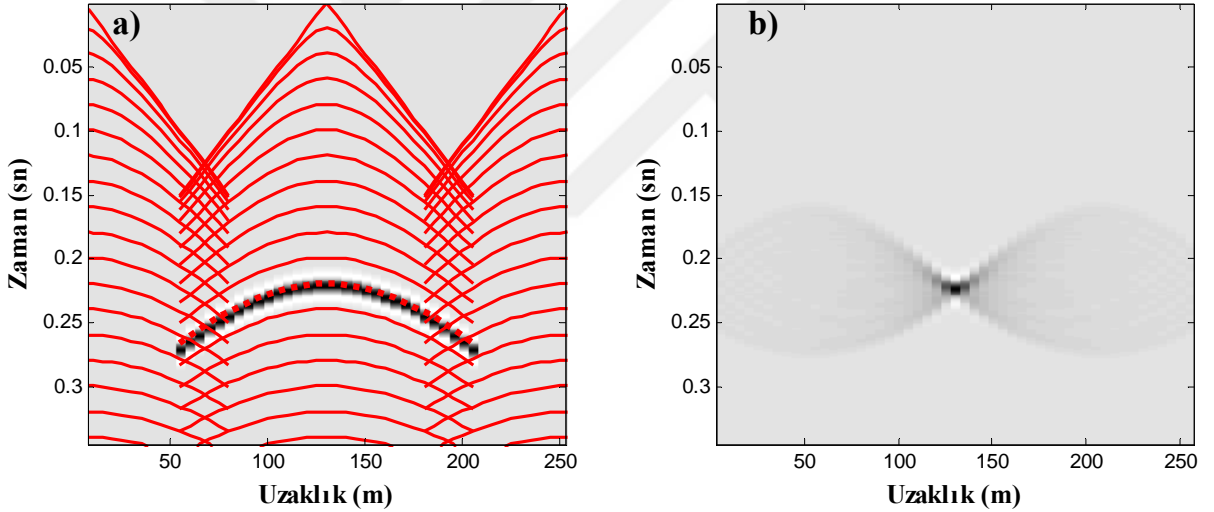
şeklinde düzenlenebilir. Burada, t : göç zamanı, τ : tepe noktası zamanı, x : uzaklık, V_{rms} ise karekök ortalama hız bilgisini temsil etmektedir. Bu saçılma hiperbolü formülü aynı zamanda göç işlemi için de kullanılmaktadır. Hesaplanacak hiperbolik yol boyunca toplama işlemi gerçekleştirilir ve toplanan genlik hiperbolün tepe noktasına (A) atanır. Bu işlem veri setindeki her bir tepe noktası için tekrarlanarak göç işlemi gerçekleştirilir. Göç işleminin doğru sonuçlar verebilmesi için hız bilgisinin doğru olması gerekir. Ayrıca yarı operatör boyunun yeterli uzunlukta olması da gerekir.

Schneider (1978), Berryhill (1979), ve Berkhout (1980) göç yönteminin matematiksel gelişimi için önemli referanslardır. Sayısal dalga denkleminin integral çözümü üç terim doğurur; $(1/r, \text{genlik azalımı})$ ile orantılı olan uzak alan terimi ve $(1/r^2, \text{dalga enerjisinin azalımı})$ ile orantılı olan bundan farklı iki terim. Böylelikle; toplamaya en büyük katkıyı yapan uzak alan terimi Khirchhoff göç uygulamasında pratik olarak kullanılır. Sadece uzak alan terimi kullanılarak yeraltında (x_0, z) konumundaki bir yeraltı çıkış görüntüsü $P_{out}(x_0, z = \frac{v\tau}{2}, t = 0)$ aşağıdaki uzamsal bir açıklık üzerinden toplama ile yüzeyde $(z = 0)$ ölçülen 2B lu sıfır ofset dalga alanından $P_{in}(x, z = 0, t)$ hesaplanan uzak alan terimi ile hesaplanır (Yılmaz, 2001 den uyarlanarak).

$$P_{out}(\tau, x) = \frac{\Delta x}{2\pi} \sum_{x=-l}^{+l} \left[\frac{\cos\theta}{\sqrt{V_{rms}r}} P_{in}(t, x) \right] \quad (2)$$

Burada, τ : tepe noktası zamanı, t : göç zamanı, θ : tepe noktası ile göç zamanı arasındaki açı, r : giriş ile çıkış arasındaki uzaklık V_{rms} ise karekök ortalama hız bilgisidir. Pratikte bir sismik kesitin N_x adet sismik izi ve her bir sismik izin de N_t adet zaman örnekleme olduğu varsayılırsa, bu sismik kesitin $N_x * N_t$ adet tepe noktası olması gerekir. Yani $N_x * N_t$ adet hiperbol hesaplanır ve bu hiperboller boyunca toplama işlemi

gerçekleştirilir. Şekil 2.2 göç işlemi için hesaplanan birkaç saçılma hiperbolünü göstermektedir. Sırasıyla 1., 26. ve 51. izlerin, her 0.02 sn de bir olmak üzere toplam 54 saçılma (göç) operatörü gösterilmektedir. Söz konusu örneğin göç işlemi için toplamda (zaman örnek sayısı * iz sayısı = operatör sayısı) $344*51=17544$ adet operatör hesaplanmaktadır. Pratikte bu hiperboller üzerindeki genlikler toplanıp hiperbolün tepe noktasına atanarak göç işlemi gerçekleştirilmiş olur. Sismik kesitte saçılma hiperbolü ile en fazla eşleşmeyi sağlayan hiperbolün (kesikli çizgili) tepe noktası çıkış verisinde en büyük genliğe sahip olacaktır. Fakat diğer hiperbollerin de bu genlikleri kestiği görülmektedir ve bu genlikleri kendi tepe noktalarına taşıyacak olduklarından dolayı göç işleminin hatası (artifact) olarak sonuç kesitinde görünecektir. Çıkış kesitinde saçıcı noktaya tam bir odaklanma sağlanamamıştır, çünkü göç algoritmaları sismik kesitin her bir noktasında saçılma hiperbolü boyunca toplamaya odaklanmıştır.



Şekil 2. 2. a) Göç işlemi için hesaplanan birkaç operatör, b) saçılma toplama ile göç çıkışı.

2.2. Kirchhoff Göçü Matris Formu

Kirchhoff göçü olarak bilinen ve saçılmaların toplaması (diffraction summing) işlemine dayanan göç işlemi matris ortamında da gerçekleştirilebilir. Bunun için öncelikle bir saçılma hiperbolü modellemek gerekir (Nemeth, 1999; Yousefzadeh ve Bancroft, 2012a,b).

$$d = Lm \quad (3)$$

Burada, d : veriyi, m : yansıtabilirliği ve L : göç (saçılma) operatörünü temsil etmektedir. Buna göre, yansıma katsayısı ile göç (saçılma) operatörü matrislerinin çarpılması sonucu veri elde edilir.

m : yansıtabilirlik modeli

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$L_{4,4}$: saçılma hiperbolü ($m_{4,4}$ noktası için)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Bu geometri ile üretilen veri aşağıdaki matrise benzer olacaktır. $d_{4,4} = L_{4,4}r_{4,4} = L_{4,4}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Kirchhoff göçünün saçılma hiperbolleri boyunca toplama işlemi yaptığı düşünüldüğünde; saçılma hiperbolünü temsil eden göç operatörü (L matrisi) ile giriş verisi (d) çarpılır ve elde edilen matrisin ($L^T d$) köşegeni üzerindeki değerlerin toplamı, hesap

yapılan tepe noktası (i. satır, j. sütun veya i. zaman örneği j. iz) için Kirchhoff göçü çıkışı verir. Bu durumda

$$m_{i,j} = L^T d \quad (4)$$

işlemi Kirchhoff göç işleminin matris ortamındaki ifadesidir. $L_{4,4} = d_{4,4}$, $L_{4,4}$ ve $d_{4,4}$ aynı boyutta matrislerdir (7x7). $r_{4,4}$ ise, 1x1 lik bir matrisdir. Bu nedenle $L_{4,4}$ ve $d_{4,4}$ aşağıdaki gibi vektör olarak düzenlenebilirler.

$$L_{v4,4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{ve} \quad d_{v4,4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$L_{v4,4}$ ve $d_{v4,4}$ birer vektör olarak düzenlendikten sonra. Her bir noktanın göç işlemi için bu vektörlerin çarpımını kullanılır.

$$L_{v4,4} m_{v4,4} = d_{v4,4} \quad (5)$$

Bu işlem her bir izin içerdiği örnek sayısı kadar tekrarlanarak göç çıkışı hesaplanır.

$$L_{1,j} m_{1,j} = d_{1,:} \quad (6)$$

Burada $L_{1,:}$ ilk iz için hesaplanan ilk operatörü temsil ederken, $m_{1,j}$ modelin giriş penceresini ve $d_{1,j}$ ise veri giriş penceresini temsil etmektedir. $L_{1,j}$ in boyutu $(\mathbf{1}, N_t * N_{px})$ olacaktır.

$$L = [L_{v1,1} \ L_{v1,2} \ \dots \ L_{v1,N_x} \ L_{v2,1} \ L_{v2,2} \ \dots \ L_{vN_t,N_{px}}] \quad (7)$$

N_{px} bir penceredeki iz sayısıdır. Bunun yanında model matrisini de vektör haline getirmek gerekir.

$$m_v = \begin{bmatrix} m_{v1,j} \\ m_{v2,j} \\ m_{v1,j} \\ \vdots \\ m_{vN_z,j} \end{bmatrix} \quad (8)$$

L matrisinin her bir kolonu göç işlemi tamamlanmış bir noktayı temsil eder. Yani L matrisi $N_t * N_{px}$ satır, N_t sütundan oluşmaktadır. Dolayısıyla L matrisinin $(N_t, N_t * N_{px})$, m_v vektörü $(N_t * N_{px}, 1)$ ile çarpımı, d_v vektörünü $(N_t * N_{px}, 1)$ verecektir.

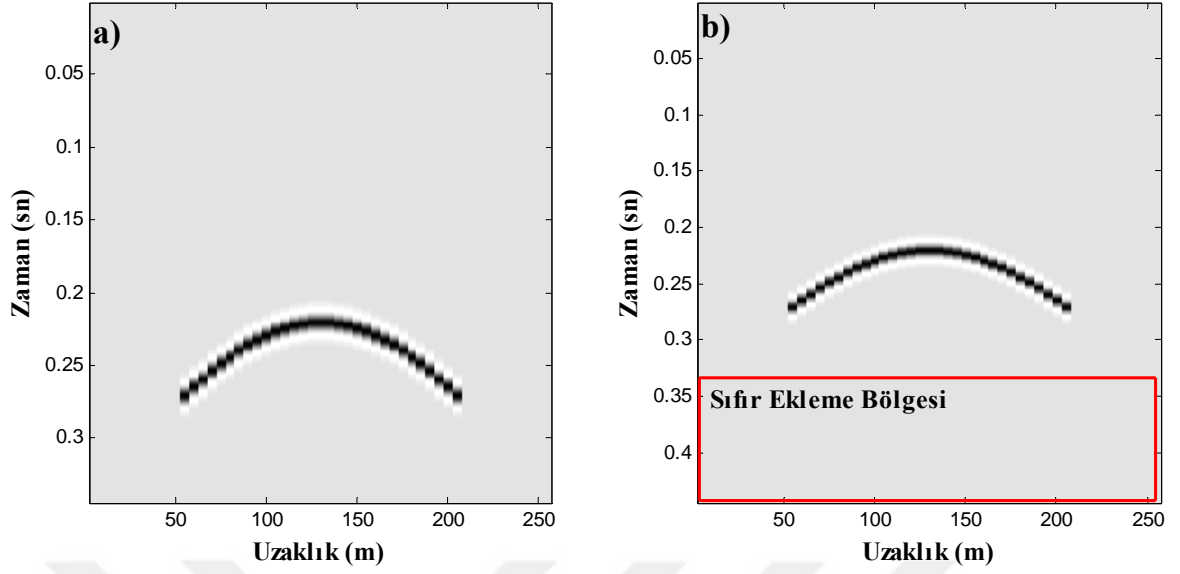
$$L m_v = d_v \rightarrow \begin{matrix} N_t \\ * \\ N_{px} \end{matrix} \left\{ \begin{matrix} \dots & \dots & \dots \\ \vdots & \ddots & \vdots \\ \dots & \dots & \dots \end{matrix} \right\} \begin{matrix} N_t \\ * \\ N_{px} \end{matrix} \left\{ \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} \right\} = \begin{matrix} N_t \\ * \\ N_{px} \end{matrix} \left\{ \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} \right\} \quad (9)$$

Bu aşamadan sonra d_v vektörü yeniden şekillendirilerek (N_t, N_{px}) boyutlu matrise dönüştürülür. Böylece her bir operatör, bir izin zaman örnek sayısı kadar satıra sahip olacaktır ve göç işlemi izden ize uygulanabilecektir.

$$m_{mig} = L^T d \quad (10)$$

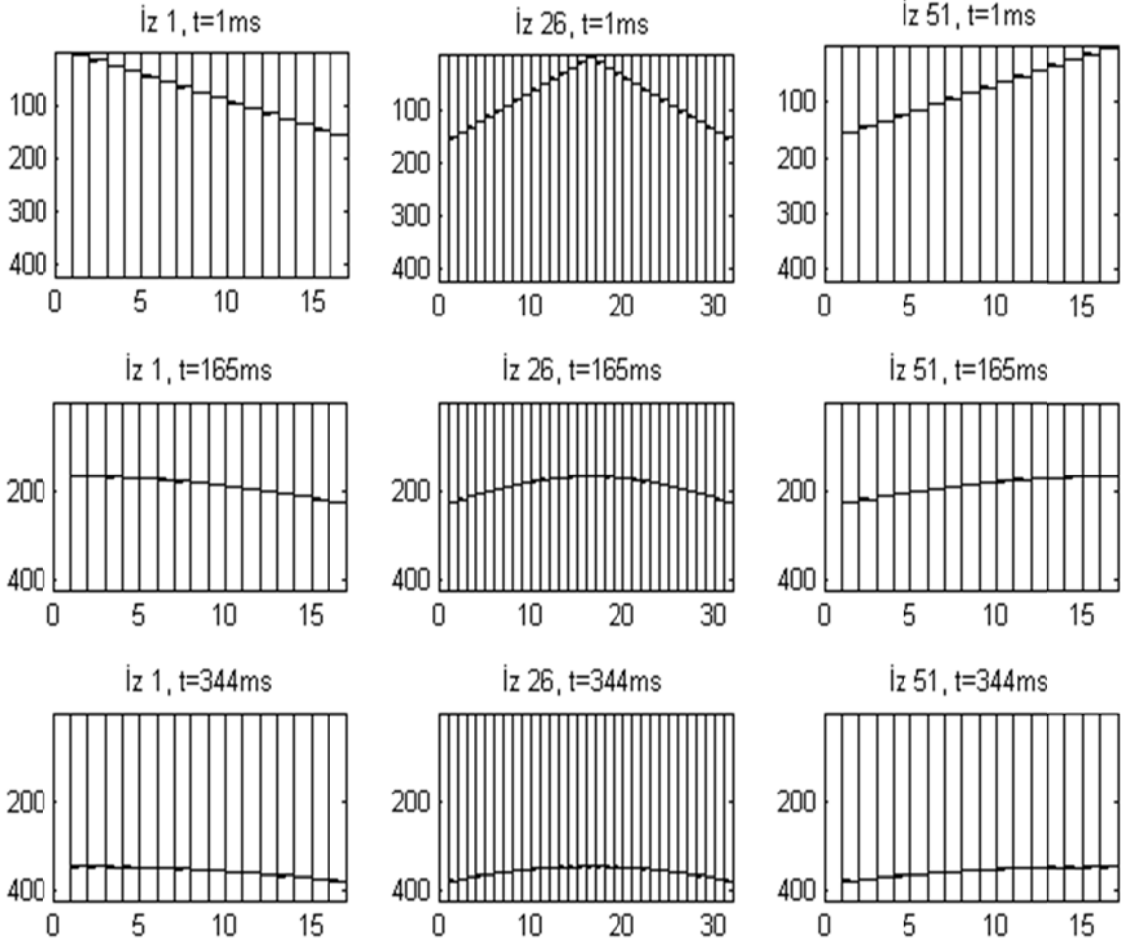
Bu şekilde göç işlemi birkaç aşamada gerçekleştirilir.

- 1- Giriş verisinin sonuna sıfır eklenir (hesaplamalarda kolaylık sağlaması amacıyla): Sismik kaydın son zamanlarında saçılma hiperbollerinin kanatlarının veri dışına çıkmaması için kaydın sonuna sıfır eklenir (Şekil 2.3). Eklenecek sıfır sayısı $N_z = \max(\text{operatör boyu}) / \text{ort}(hız)$ formülü ile hesaplanabilir.



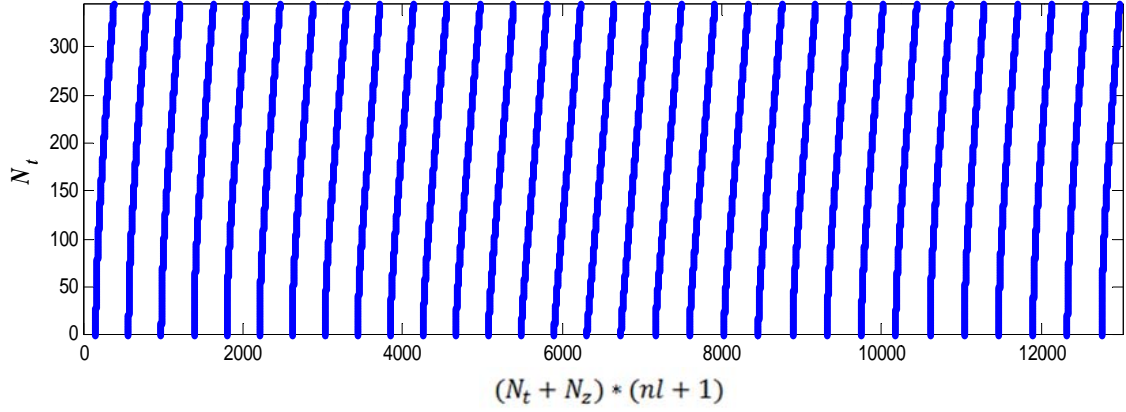
Şekil 2. 3. Giriş verisi ve sıfır eklenmiş veri.

- 2- Her bir izin her bir zaman adımı için Denklem-1 ve 2 yardımıyla operatör hesaplanır. Şekil 2.3 deki giriş örneğinde 51 iz ve her bir ize ait 344 örnek vardır. Yani her bir izi hesaplamak için 344 adet (zaman örnek sayısı kadar) operatör hesaplanmaktadır. Bu operatörler her bir zaman adımı için saçılma hiperbollerini temsil ederler ve saçılma hiperbolü o zamandaki hız bilgisi ile ilişkilidir. Şekil 2.4 de basit saçılma modeli (Şekil 2.3) için hesaplanan operatörler gösterilmektedir. Bu operatörler 1, 26 ve 51. izlerin 1 ms, 165 ms ve 344 ms için hesaplanan operatörleridir. Daha sonra bu operatörler kullanılarak her bir iz için çekirdek matrisleri oluşturulur.

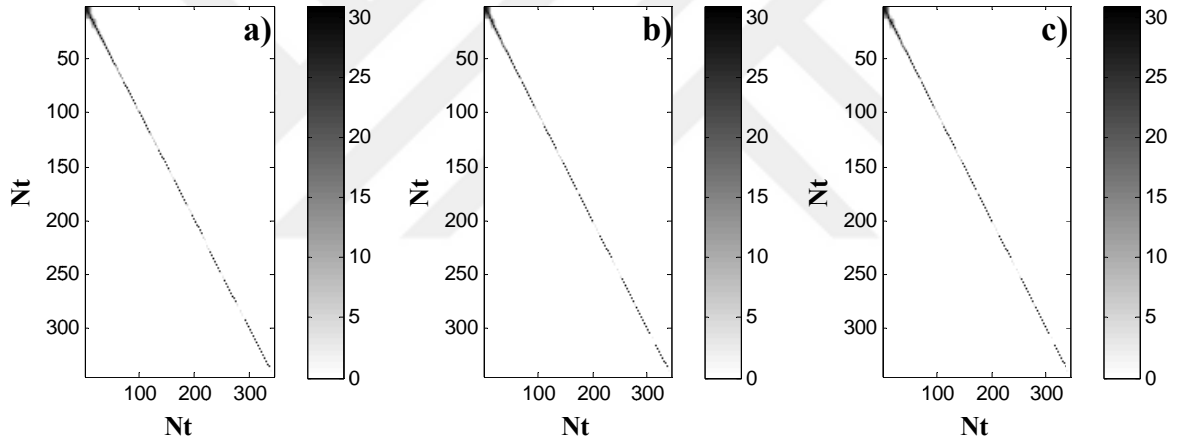


Şekil 2. 4. Hesaplanan operatörler. 1. (sol sütun), 26. (orta sütun) ve 51. (sağ sütun) izlerin $t=1$ ms (üst satır), 165 ms (orta satır) ve 344 ms (alt satır) için hesaplanmış operatörler.

- 3- Her bir izin tüm örnekleri için hesaplanan operatörler çekirdek matrisinin bir satırını oluşturur. Yani çekirdek matrisinin satır sayısı giriş verisinin zaman örnek sayısına eşittir. Söz konusu örnek için satır sayısı 419 (sıfır eklendikten sonra) olmalıdır. Şekil 2.4' de gösterilen operatörlerin her biri bir satır vektörüne dönüştürülür ve operatör matrisi oluşturulur. Sütun sayısı ise yarı operatör (aperture) boyu (nl) ile ilişkili olarak değişkenlik gösterir. Buna göre ilk ve son izlerde saçılma hiperbolleri tek taraflı olduğu için (Şekil 2.4 sol ve sağ sütun) az sütun sayısı içerirken orta sütun sayısı $(N_t + N_z) * (2 * nl + 1)$ şeklinde hesaplanır. Söz konusu örneğin 1. ve 51 (son) izi için $(344+75)*(15+1)$ olarak hesaplanırken 26. iz için $(344+75)*(30+1)=12989$ olarak hesaplanır (Şekil 2.5). Hesaplanan ters çözüm matrisleri $(L^T L)^{-1}$ de Şekil 2.6' da gösterilmektedir. Görüldüğü gibi köşegen ağırlıklı bir matristir.

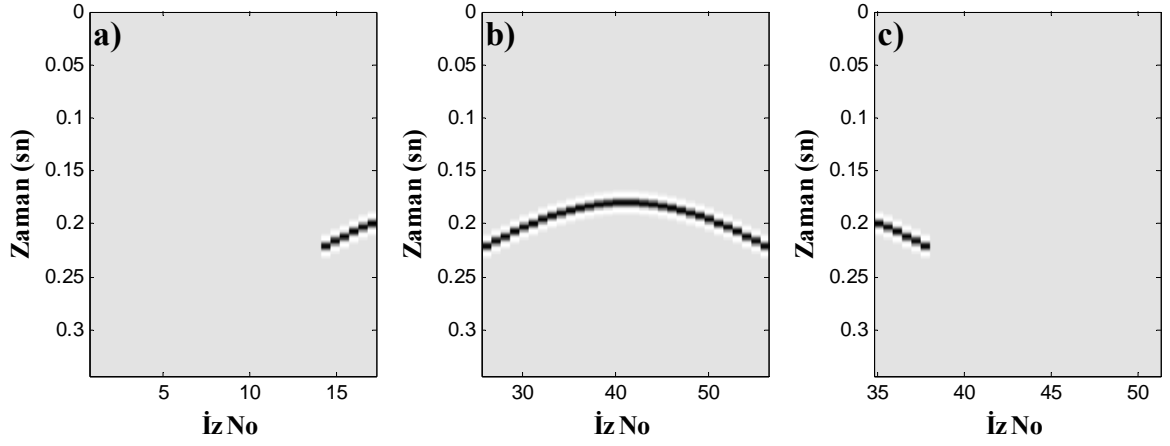


Şekil 2. 5. 26. İz için oluşturulan (L^T) matrisinin sıfır olmayan elemanları.



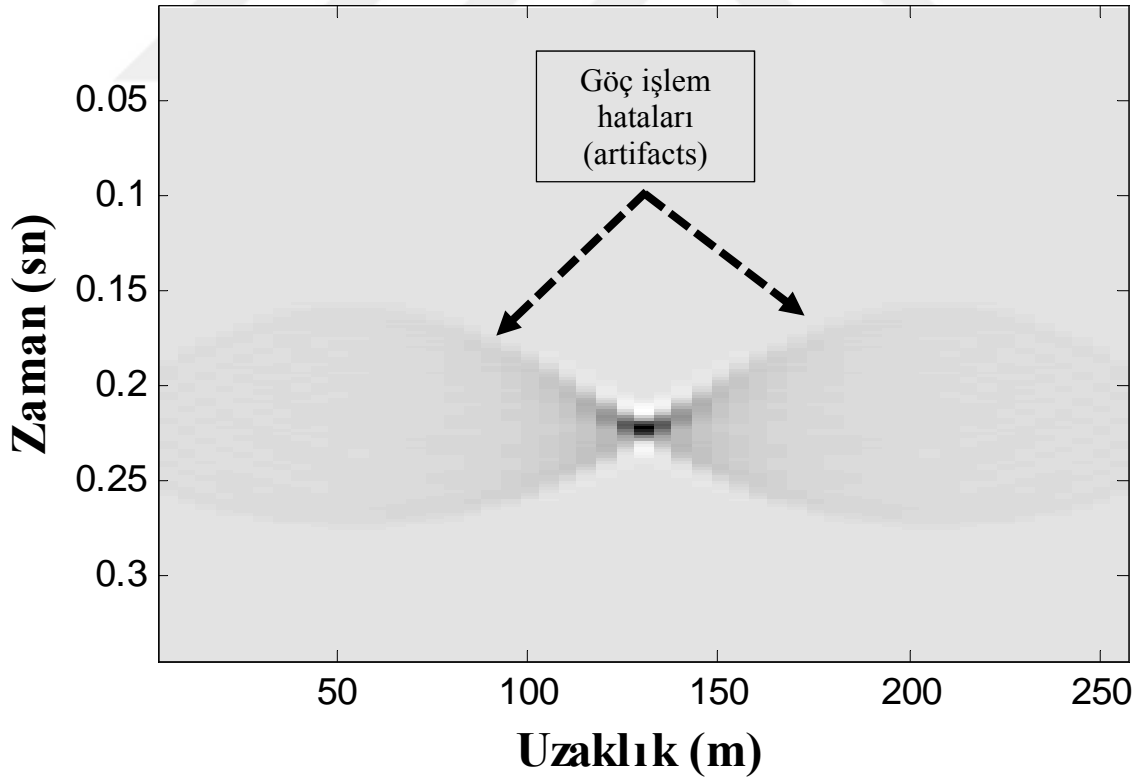
Şekil 2. 6. 1., 26. ve 51 izler için ters çözüm $(L^T L)^{-1}$ Matrisleri.

- 4- Her bir iz için giriş verisinin bir kısmı (pencere) kullanılır. Sıfır eklenmiş verinin operatör boyu kadar kısmı kullanılır. Operatör boyu hat boyunca değişkenlik gösterdiğinden pencere boyu da buna uygun olarak değişkenlik gösderecektir. Şekil 2.10 da 1., 26. ve 51. izler için kullanılan giriş verisi (d) gösterilmektedir. Buna göre 1. ve 51. iz için sütun sayısı 16 iken 26. iz için 31 dir.



Şekil 2. 7. Her bir iz için kullanılan giriş verisi. a) 1. iz için giriş verisi, b) 26. iz için giriş verisi ve c) 51. iz için giriş verisi.

5- Her bir iz için ters çözüm işlemi gerçekleştirildikten sonra çıkış verisi üretilmiş olur (Şekil 2.11).



Şekil 2. 8. Kirchhoff göç çıkışı verisi ve göç işlem hataları.

2.3. En Küçük Kareler Göçü (EKKG)

EKK ters çözüm algoritması ilk olarak Schuster (1993) tarafından karşılıklı kuyu verilerinin göçü için uygulanmış ve daha sonra Nemeth ve diğ. (1999) tarafından bu teknik yüzey sismik yansıma verisine uygulanmıştır. Bu çalışmalar EKKG yönteminin göç işlemi hatalarını önemli derecede indirdiğini ve sismik verinin uzaysal (veya yanal) ayrımlığını artırdığını göstermiştir. EKKG hem model uzayı ortamında (Tang, 2008; Aoki ve Schuster, 2009; Dai ve diğ., 2011) hem de zaman ortamında (Tang ve Biondi, 2009; Dai ve diğ., 2010; Zhan ve Schuster, 2010) uygulanabilmektedir. Eğer bir sismik verinin ileri modelleme operatörü L bilinirse, sismik veriden yansıtılabilirlik modeli, m , elde edilebilir. Bu işlem bir ters çözüm işlemine işaret etmektedir. Bununla birlikte, veri içindeki saçılmalar saçıcı noktalarına tekrar taşınabilir. Bunun için, ileri modelleme operatörünün tersi, L^{-1} , kullanılabilir. Ancak genellikle görüntüleme için L^{-1} bir eşlenik (adjoint) operatöre L^T yaklaşır. Eşlenik, makul yapısal görüntüler üretirken, üretilen modelin genlikleri gerçek değildir, çünkü ileri modelleme operatörü kabul edilebilir bir derecede kaydedilen veriyi tekrar geri kazanamaz, özellikle de, eğer veri toplama geometrisi önemli düzensiz geometri içeriyorsa (Nemeth ve diğ., 1999). EKKG model veriyi m sismik veriden elde etmeyi amaçlar. İleri modelleme operatörü EKK anlamında gerçek veriyi yeniden üretir. Sismik veriler her zaman bir miktar gürültü içerir ve dolayısıyla daha iyi yansıtılabilirlik görüntüsü elde etmek için, en küçük kareler ters çözümü tercih edilir. Bu çözüm bir amaç fonksiyonu $J(m)$ 'in en küçüklenmesi ile elde edilir. $J(m)$ ileri model veri (hesaplanan veya kuramsal) ($d_0=Lm$) ve kaydedilen (ölçülen veya gerçek) veri, d arasındaki en küçük kareler hata (L_2 norm) olarak tanımlanır. Aslında $J(m)$ hataları içeren bir tanımlayıcı fonksiyondur.

$$J(m) = \|Lm - d\|^2 = (Lm - d)^T(Lm - d) \quad (11)$$

$J(m)$ 'in m 'ye göre türevleri alınır ve en küçüklenirse (sıfıra eşitleme), gerekli düzenlemelerden sonra, EKKG çözümü, aşağıdaki gibi elde edilebilir.

$$m_{ekk} = (L^T L)^{-1}(L^T d) \quad (12)$$

Denklem (12) genelleştirilmiş EKKG çözümdür. Bunun için $(L^T L)$ matrisinin sağlıklı bir tersinin olması gerekir. Aksi durumda $(L^T L)$ matrisinin özdeğerlerinin sıfırdan farklı değerler alabilmesini sağlamak adına matrisin köşegenine sabit bir sönüm parametresi eklenerek çözülebilir. Bu durumda, Sönümlü En Küçük Kareler yöntemi;

$$m_{sekk} = (L^T L + \gamma I)^{-1} (L^T d) \quad (13)$$

şeklinde tanımlanır. Burada I birim matrisi, γ ise sabit sönüm katsayısını temsil eder. Fakat EKKG yönteminin avantajı düzenleme (regularization) parametresine sahip olmasıdır. Sabit sönüm parametresi (γ) yerine düzenleme parametresi kullanılarak istenen göç çıkışları elde edilebilir. Bu şekilde Düzenlenmiş EKK yöntemi;

$$m_{dekk} = (L^T L + Q)^{-1} (L^T d) \quad (14)$$

şeklinde tanımlanabilir. Burada Q düzenleme (regülerizasyon) terimidir ve farklı şekillerde tanımlanabilir. Bu çalışmada Cauchy normu (Yousefzadeh, 2013):

$$R(m) = \sum_{i=1}^N \ln(1 + m_i^2 / \beta^2) \quad (15)$$

dikkate alınarak elde edilen düzenleme fonksiyonu:

$$Q_{ii} = \left(\ln \left(1 + \frac{m_i^2}{\beta^2} \right) + \epsilon \right)^{-1} * \delta^2 \quad (16)$$

kullanılmıştır. Burada m_i giriş modelini (Kirchoff göç çıkışı), β ölçekleme parametresini, δ sabit katsayı, ϵ ise denklemin sonucunun sonsuz çıkmasını engellemek için kullanılan bir kararlılık sabitidir. Bu çalışmadaki tüm işlemlerde $\beta = \max(|m_i|)$, $\epsilon = 0.001$ ve $\delta = 10000$ olarak kullanılmıştır. Bu parametrelerin farklı kullanılması ve farklı normlardan (L1 veya L2) türetilen düzenleme parametrelerinin EKKGD yöntemine etkileri Bölüm 4.8' de tartışılmıştır.

2.4. Göç Dekonvolüsyonu (Netleştirme)

Şekil 2.11'den görüldüğü üzere göç çıkışları işlem hatalarını içerir ve bu sismik göç kesit görüntüsünde bulanıklık (blurring) olarak değerlendirilir. Bu nedenle göç uygulanmış kesitlere dekonvolüsyon tekniği uygulanması ile gerçekleştirilen işleme Göç Dekonvlüsyonu (GD) işlemi denilir ve EKKG işlemine benzer olarak göç görüntülerinin yanal çözünürlüğünü arttırmak, işlem hatalarını azaltmak ve veri toplamadaki yetersiz geometriden kaynaklanan problemleri gidermek için uygulanan bir işlemdir. Böylece görüntü işlemede netleştirme (deblurring) olarak kullanılan GD göç çıkışı kesitlerinin görüntülerini de netleştirmek için kullanılabilir. İleri modelleme denklem sistemi ($Lm = d$) göz önüne alınırsa, bu denklemin her iki tarafı L^T ile çarpıldığında:

$$L^T L m = L^T d \quad (17)$$

denklemin sağ tarafı Kirchhoff göçü çıkışına karşılık gelmektedir. Sol taraftaki ($L^T L$) ifadesi ise Kirchhoff göçünün bozucu (blurring) etkisidir. GD işleminde, Kirchhoff göç çıkışı ($L^T d$) netleştirme fonksiyonu (F) ile çarpıldığında çözünürlük artmalıdır. Bu fonksiyon (F), göç operatörü (L) kullanılarak hesap edilebilir. Birçok çalışmada netleştirme fonksiyonu ($F = (L^T L)^{-1}$) olarak kullanılmaktadır (Hu ve Schuster, 1998; Schuster ve Hu, 2000; Hu ve diğ., 2001; Aoki ve Schuster, 2009). Bu durumda GD işlemi;

$$m_d = F L^T d \quad (18)$$

olarak ifade edilir. Ayrıca tersi alınacak F fonksiyonuna düzenleme parametreside eklenerek çözünürlük daha da arttırılabilir. Bu durumda F fonksiyonu

$$F = (L^T L + Q)^{-1} \quad (19)$$

şeklinde düzenlenebilir.

2.5. En Küçük Kareler Göç Dekonvolüsyonu (EKKGD)

EKKGD işlemi ilk olarak EKKG işleminde yineleme sayısını azaltmak için Aoki ve Schuster (2009) tarafından kullanılmıştır. Burada netleştirme fonksiyonu bir ağırlık fonksiyonu gibi davranır. Yani hem Kirchhoff göç çıkışına uygulanabilir hem de EKKG algoritması için kullanılabilir. EKKGD için başlangıç modeli olarak GD çıkışı ($\mathbf{m}_d = \mathbf{F}^T \mathbf{L}^T \mathbf{d}$) kullanılabilir. Bu durumda EKKGD algoritması;

$$\begin{aligned} m_{ekk} &= (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{m}_d \\ m_{dekk} &= (\mathbf{L}^T \mathbf{L} + \mathbf{Q})^{-1} \mathbf{F}^T \mathbf{L}^T \mathbf{d} \end{aligned} \quad (20)$$

şeklinde ifade edilebilir. EKKG ve EKKGD algoritmaları (1) eşlenik-gradyent (bakınız EK-A) ve (2) gauss-seidel (bakınız EK-B) yinelemeli çözüm yöntemleri ile çözülmüştür.

2.6. Ağırlıklandırma Fonksiyonları

Ağırlıklandırma fonksiyonları mühendislik problemlerinde çözümleri odaklamak için sıklıkla kullanılmaktadır. Genelde veri setinde anlamsız değerler olduğunda çözümler bu anlamsız değerlerin etkisinde kalarak veri yoğunluğu olan bölgeye odaklanamaz, bu nedenle ağırlıklandırma uygulanarak daha makul çözümler üretilebilir. Bu çalışmada farklı ağırlıklandırma (veya deblurring) fonksiyonları hem Kirchhoff göç çıkışına hem de EKKG algoritmasına uygulanarak çözüme odaklama sağlanması amaçlanmıştır. Çeşitli ağırlık (netleştirme) fonksiyonları ile birlikte EKKG algoritmasını eşlenik-gradyent ve gauss-seidel yöntemleri ile birlikte test ederek GD yöntemi ile karşılaştırıp zaman ve çözünürlük açısından en uygun çözüm yönteminin belirlenmesi amaçlanmaktadır. Uygun görülen netleştirme fonksiyonları Tablo 2.1’de gösterilmektedir. Burada gösterilen $(\mathbf{L}^T \mathbf{L})^{-1}$ fonksiyonu literatürde kullanılan geleneksel fonksiyondur. Bu fonksiyon GD işlemi için bozucu etkiyi (blur effect) gideren netleştirme (deblurring) fonksiyonudur. $(\mathbf{L}^T \mathbf{L})$ fonksiyonu ise Kirchhoff göç çıkışında hataya neden olan fonksiyondur. Bilinmektedir ki standart sapma değeri ağırlıklandırma fonksiyonlarında sıklıkla kullanılmaktadır. Bu nedenle, bu fonksiyonların kovaryansları (cov) da kullanılmıştır. Ayrıca, sismik veride saçılma olması durumunda $(\mathbf{L}^T = \mathbf{d})$ ifadesinden yola çıkarak giriş verisinden türetilen fonksiyonlar da kullanılmıştır.

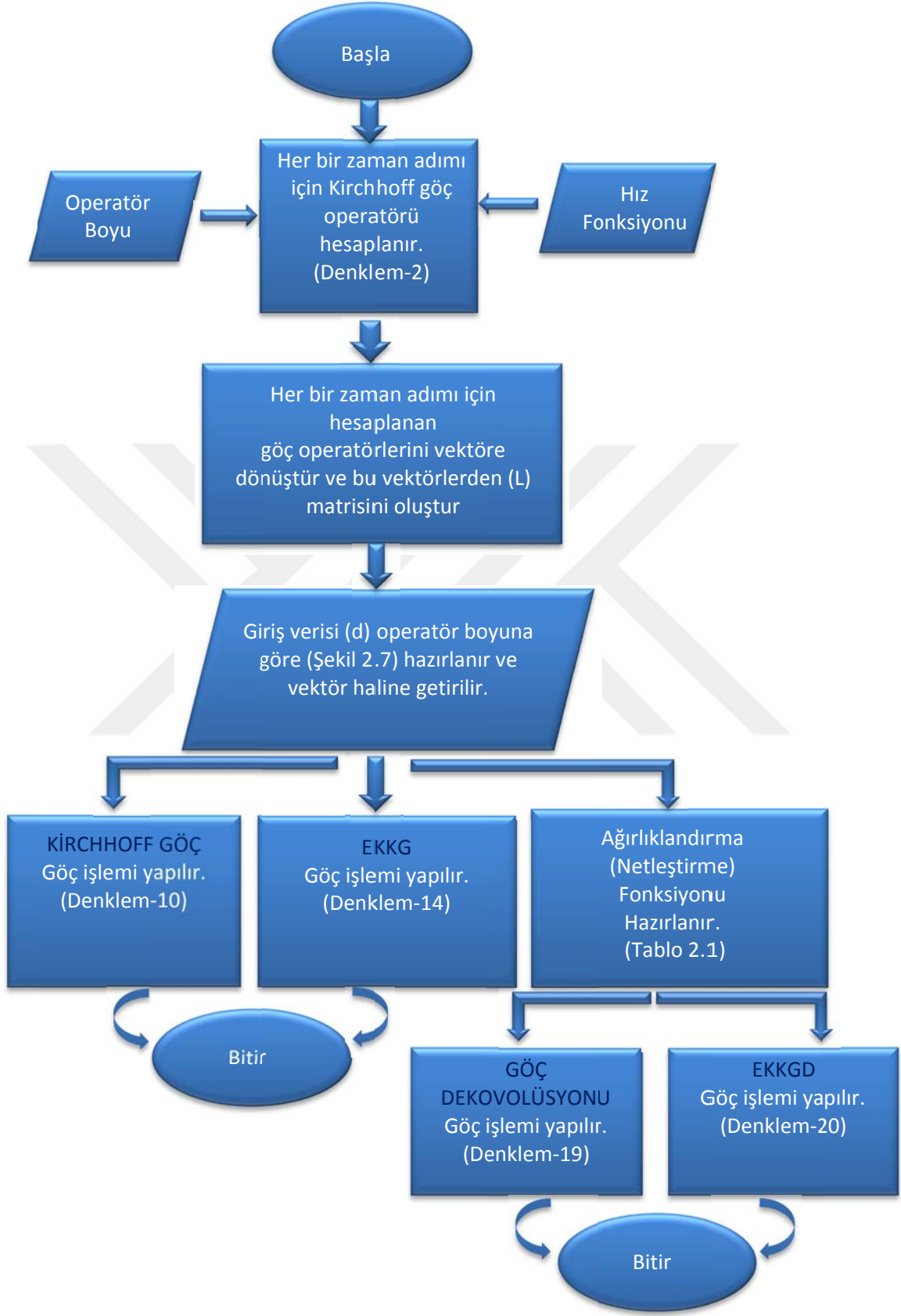
Tablo 2. 1. Bu çalışmada kullanılan ağırlıklandırma (netleştirme) fonksiyonları.

Göç Dekonvolüsyonu	EKKGD Eşlenik-Gradyent	EKKGD Gauss-seidel
Yok (Kirchhoff göçü)	Yok (EKKG)	Yok (EKKG)
$(L^T L)^{-1}$	$(L^T L)^{-1}$	$(L^T L)^{-1}$
$(L^T L)$	$(L^T L)$	$(L^T L)$
$cov(L^T L)$	(dd^T)	(dd^T)
$cov((L^T L))^{-1}$	$(dd^T)^{-1}$	$(dd^T)^{-1}$
$(dd^T)^{-1}$	$cov(dd^T)$	$cov(dd^T)$
(dd^T)	$cov((dd^T)^{-1})$	$cov((dd^T)^{-1})$
$cov(dd^T)$	$cov(L^T L)$	$cov(L^T L)$
$cov((dd^T)^{-1})$	$cov((L^T L))^{-1}$	$cov((L^T L))^{-1}$

Yukarıdaki tabloda L göç operatörünü, d giriş verisini (pencere boyunda, N_{px}), cov ise kovaryans matrisi temsil etmektedir.

Yukarıda bahsi geçen dört farklı göç işlemi için akış şeması Şekil 2.9 da gösterilmektedir. Buna göre her bir işlem için Kirchhoff göç hiperbollerini (Denklem-2) hesaplanır. Bu hiperbollerin doğru hesaplanabilmesi için doğru hız bilgisine ihtiyaç duyulur. Ayrıca hiperbollerin bir kanat uzunluğunun belirlendiği yarı operatör boyu (aperture length) da önemlidir. Hesaplanan hiperboller ne kadar doğru olursa, göç çıkışları da o kadar doğru olacaktır. Her bir tepe noktası için operatörler hesaplandıktan sonra L matrisi oluşturulur. Daha sonra giriş verisi (d) yarı operatör boyu bilgisine göre ayarlanarak vektör haline getirilir. Kirchhoff göçü için Denklem-10, EKKG için Denklem-14 kullanılarak göç çıkışları üretilir. GD veya EKKGD için netleştirme fonksiyonları hesaplanır. GD işlemi için Denklem-19, EKKGD için Denklem-20 kullanılarak göç çıkışları elde edilir.

Bu tez çalışması, göç işlem hatalarını gidermek için kullanılan GD, EKKG ve EKKGD işlemlerinin maliyetlerini azaltmak amacıyla literatürde kullanılan $(L^T L)^{-1}$ geleneksel fonksiyonuna alternatif olabilecek farklı ağırlıklandırma fonksiyonlarının kullanılması üzerine kurulmuştur.



Şekil 2. 9. Bu çalışmada kullanılan göç yöntemlerinin akış şemaları.

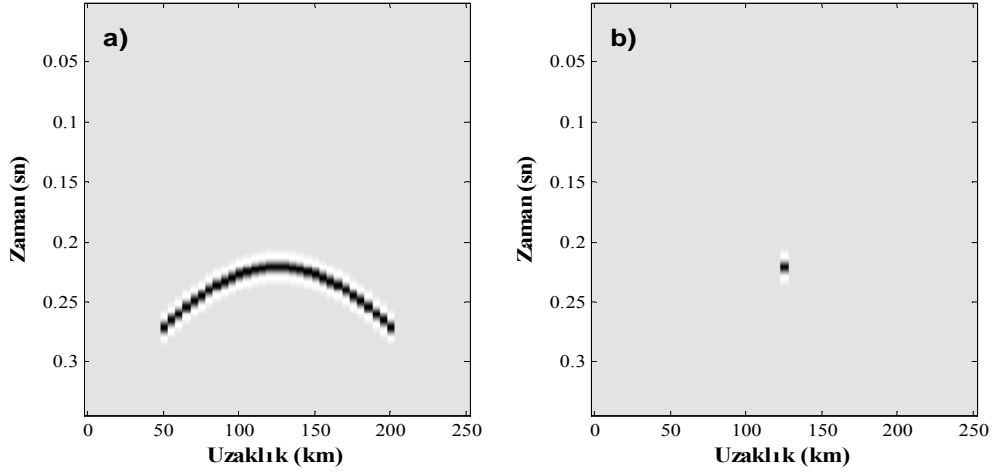
3. UYGULAMALAR

3.1. Model Veriler Üzerinde Göç Uygulamaları

Sismik veri işlemede model çalışmaları veri işlem adımlarının testleri için kullanılmaktadır. Bunun için testi yapılacak veri-işlem adımına uygun problemler tanımlanır ve bu doğrultuda sismik modeller oluşturulur. İlk olarak akla gelen saçıcı noktalar, eğimli yeraltı yapısı antiklinal ve senklinaller ve yüksek hız farklılığından kaynaklanan tuz yapılarıdır. Bu sebeplerden dolayı nokta saçılma, senklinal, fay modelleri üretilmiş ve bunlar üzerinde uygulamalar yapılmıştır. Öncelikle, göç işlemi için oldukça önemli olan hız bilgisi ve operatör uzunlukları; literatürde kullanılan Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ kullanılarak GD ve EKKGD işlemleri test edilmiştir. Göç çıkışlarının, beklenen göç çıkışları ile farklarının karekök ortalamaları (kko) hesaplanarak hata oranları türetilmiştir. Bunun yanında çözüm zamanları da karşılaştırılarak en uygun yöntem belirlenmeye çalışılmıştır. Hız bilgisi ve yarı operatör boyunun etkisi incelendikten sonra farklı netleştirme fonksiyonlarının göç çıkışları üzerinde etkileri incelenmiştir. Daha sonra sismik göç testlerinde sıklıkla kullanılan SEG/EAGE tuz modeli üzerinde farklı netleştirme fonksiyonları ile testler yapılmış, çözüm zamanı ve göç çıkışı kalitesi incelenmiştir.

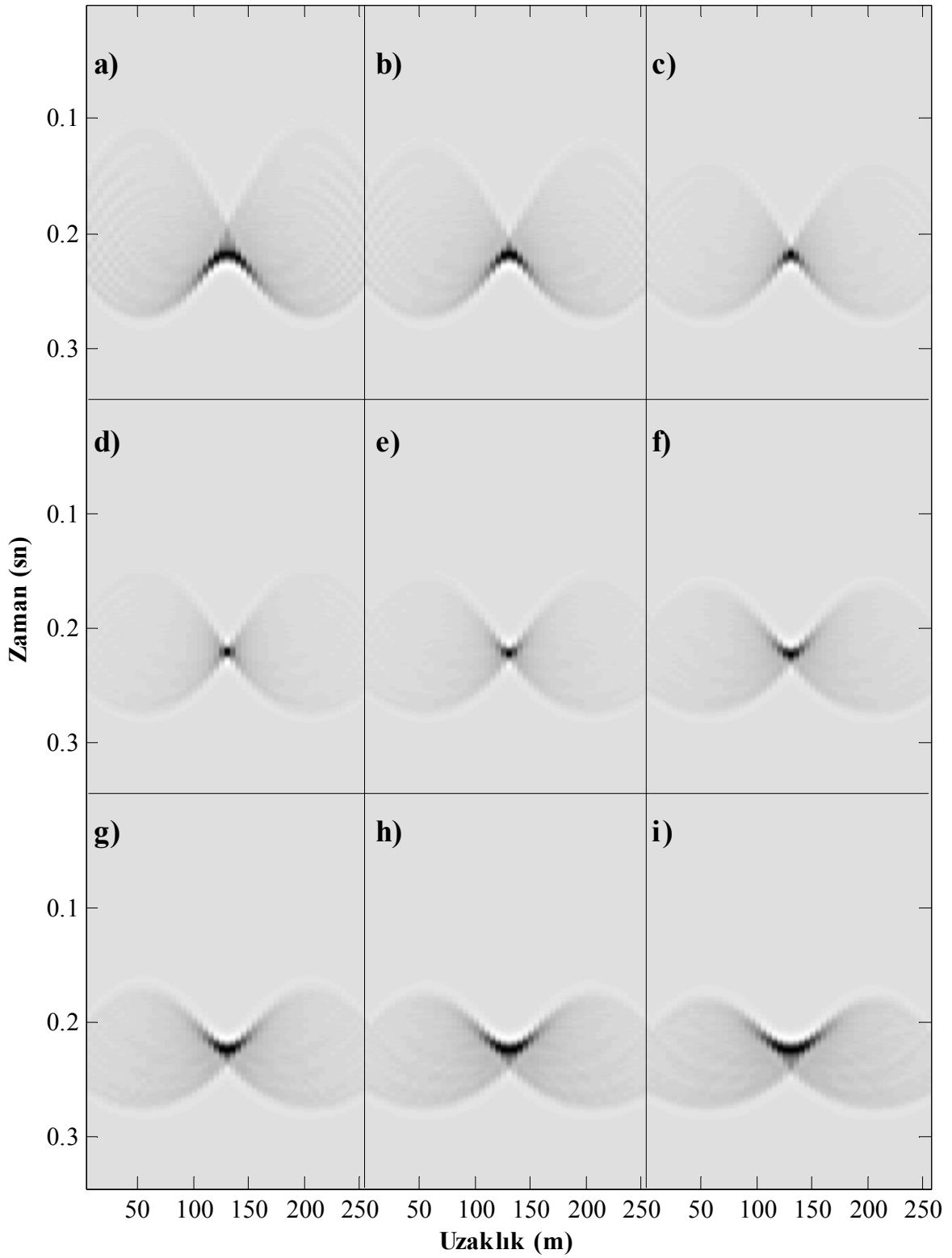
3.1.1. Nokta Saçılma Modeli Üzerinde Göç Uygulamaları

Nokta saçıcı yer modellerinin karakteristik sismik veri modelleri hiperbol şeklindedir (Şekil 3.1a) ve göç algoritmalarını test için hayati önem taşımaktadır. Dolayısıyla başlangıç model olarak kabul edilebilir. Söz konusu model yeraltında 100 m derinlikte bir nokta saçıcıyı içermektedir (Şekil 3.1b). Model hızı 1000 m/sn, örnekleme aralığı (dt) 0.001 sn, kayıt süresi (T) 0.344 sn, iz aralığı (dx) 5 m ve toplam kesit uzunluğu 250 m dir. Beklenen göç çıkışı Şekil 3.1b de gösterilmektedir.

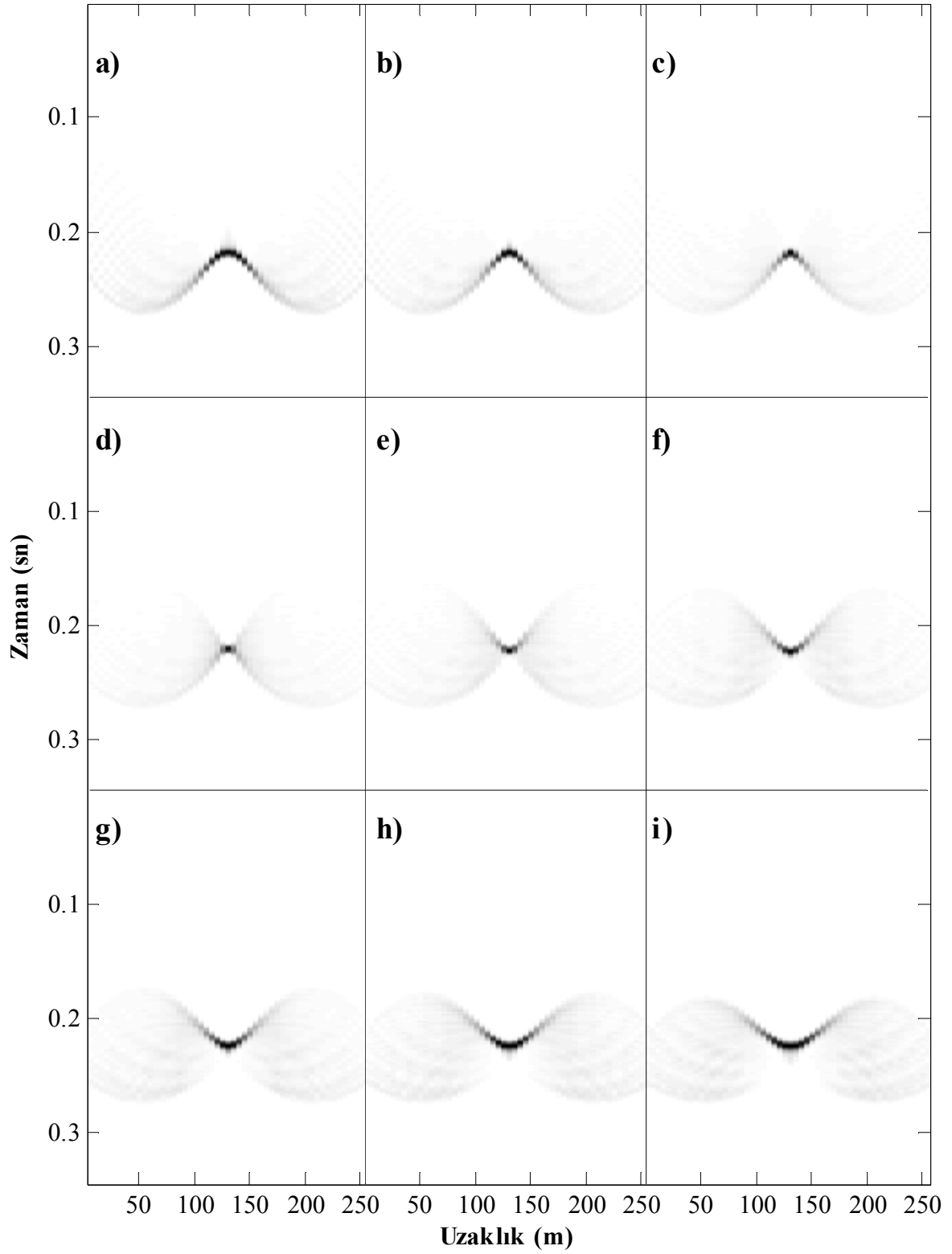


Şekil 3. 1. a) Nokta saçıcı sismik veri modeli ve b) beklenen göç çıkışı.

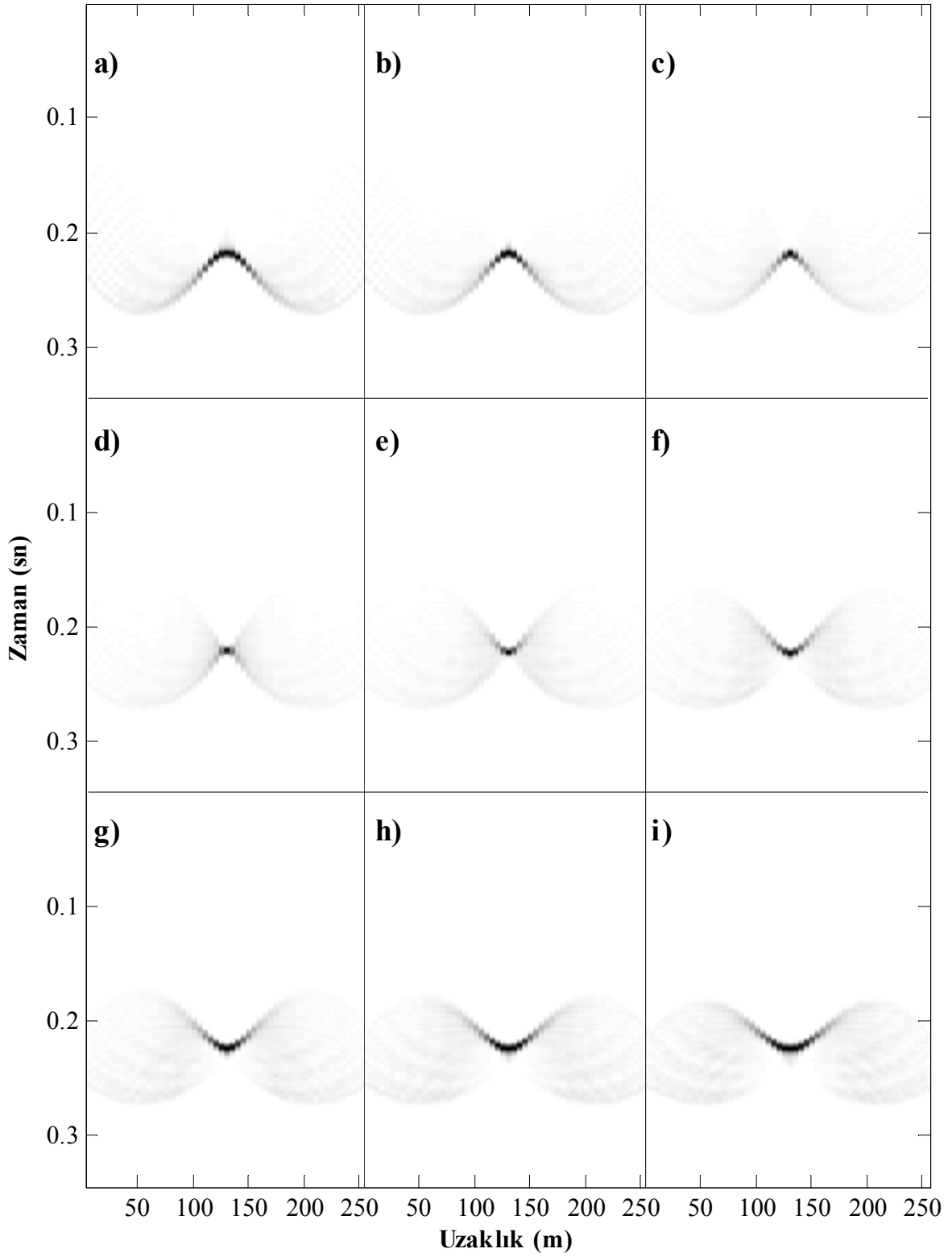
Nokta saçıcı model için yaklaşık %20 eksik hız fonksiyonundan (800 m/sn) %20 fazla hız fonksiyonuna (1200 m/sn) Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile GD ve EKKGD işlemleri uygulanmıştır. Şekil 3.2 farklı hız fonksiyonları ile gerçekleştirilen Kirchhoff göç çıkışlarını göstermektedir. Düşük hız bilgisi saçılma hiperbolünün alt kısımlarını toplayamamıştır (Şekil 3.2a-c). Bu olay literatürde eksik göç (undermigration) olarak yer almaktadır. Yüksek hız bilgisi saçılma hiperbolünü yukarıya taşımıştır (Şekil 3.2g-i). Bu olay literatürde aşırı göç (overmigration) olarak yer almaktadır. %5 eksik hız fonksiyonu (950 m/sn) ile elde edilen göç çıkışı (Şekil 3.2d) istenen göç çıkışına oldukça yaklaşıırken, %5 fazla hız fonksiyonu (1050 m/sn) kullanılarak elde edilen Kirchhoff göç çıkışı (Şekil 3.2e) daha fazla hatalı (950 m/sn hıza göre) göç çıkışı üretmiştir. Şekil 3.3 aynı hız fonksiyonları ile uygulanan GD çıkışlarını göstermektedir. GD çıkışları göç hatalarını sönmüleyerek görüntü kalitesini arttırmıştır. Şekil 3.4 aynı hız fonksiyonları ile uygulanan EKKG çıkışlarını göstermektedir. EKKG çıkışları ile GD çıkışları arasında herhangi bir fark görülememiştir. Şekil 3.5 aynı hız fonksiyonları ile uygulanan EKKGD çıkışlarını göstermektedir. EKKGD çıkışları, GD ve EKKG çıkışlarına göre ayrımlığı yüksek çıkışlar üretmiştir. Şekil 3.6 dört farklı yöntemin çözüm zamanlarını göstermektedir. Buna göre tüm hız fonksiyonları için Kirchoff göçü en hızlı yöntemdir. EKKG ve GD işlemlerinin çözüm zamanları yaklaşık eşittir. En maliyetli yöntem Kirchhoff göç çıkışının yaklaşık 6 katı maliyetle EKKGD işlemidir. Fakat hata oranını en aza (Şekil 3.7) düşüren yöntem de EKKGD yöntemidir. Şekil 3.7 incelendiğinde, GD ile EKKG işlemlerinin aynı hata oranlarını ürettiği görülmektedir. Hız fonksiyonu doğru hız fonksiyonuna (1000 m/sn) yaklaştıkça hata oranı düşmektedir.



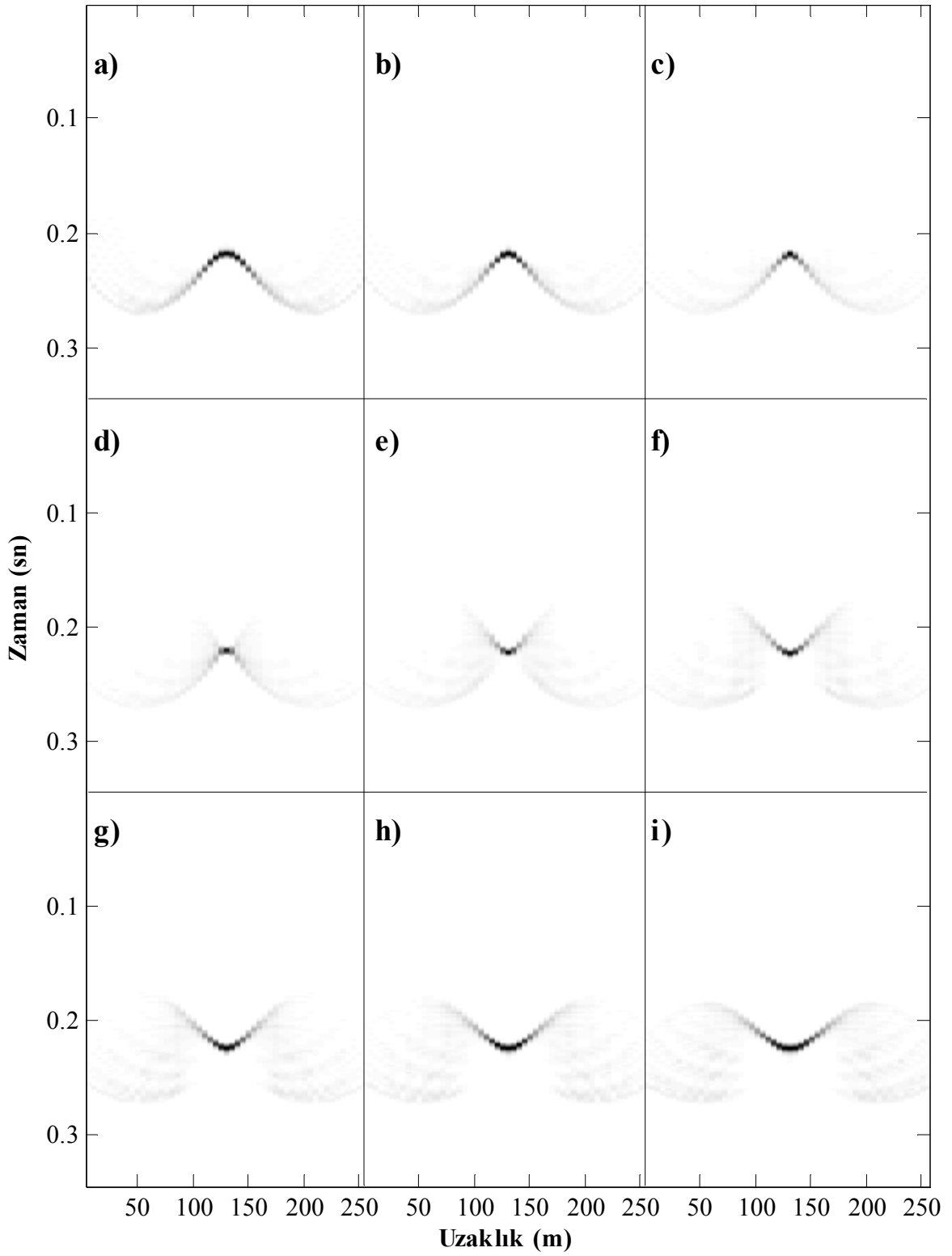
Şekil 3. 2. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=75 m).



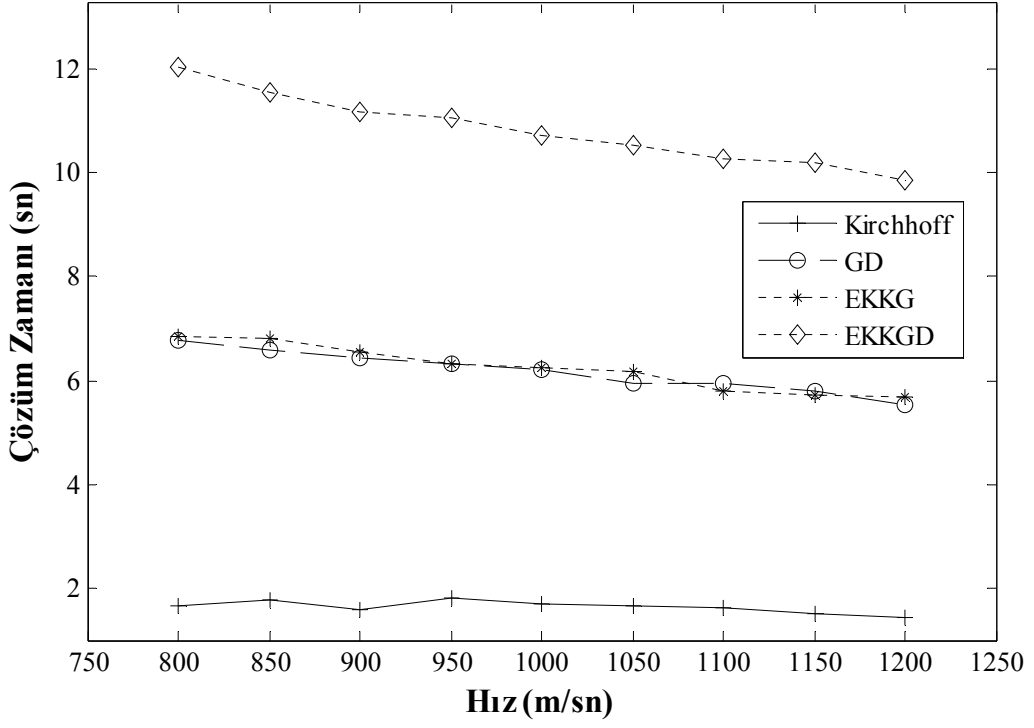
Şekil 3. 3. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=75 m).



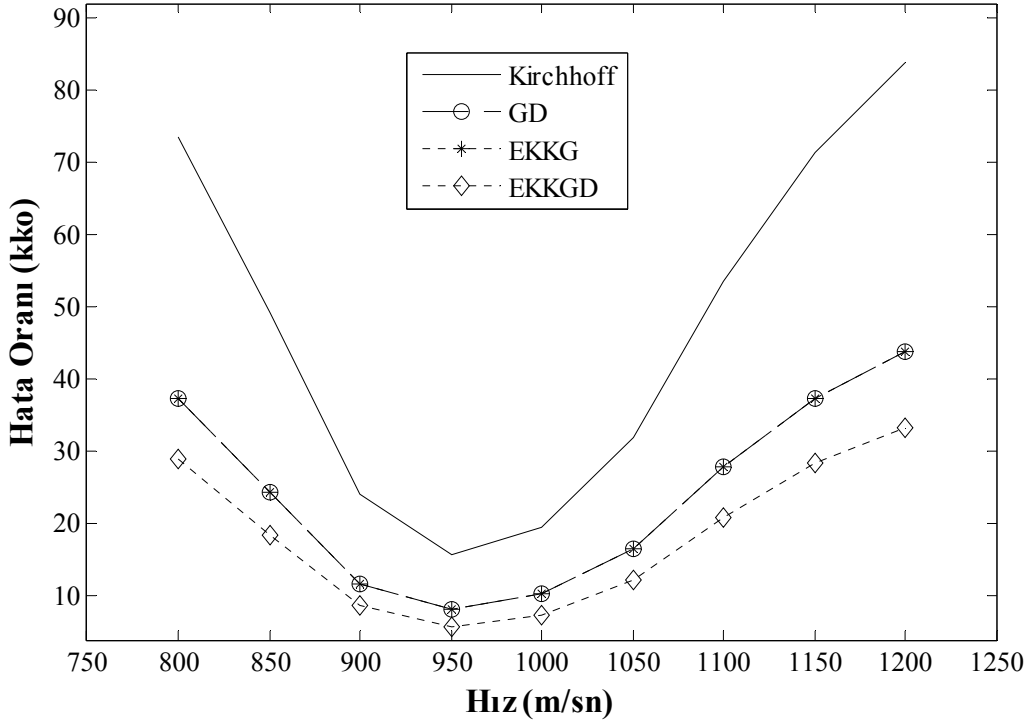
Şekil 3. 4. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKGK çıkışları (yarı operatör boyu=75 m).



Şekil 3. 5. Nokta saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=75 m).

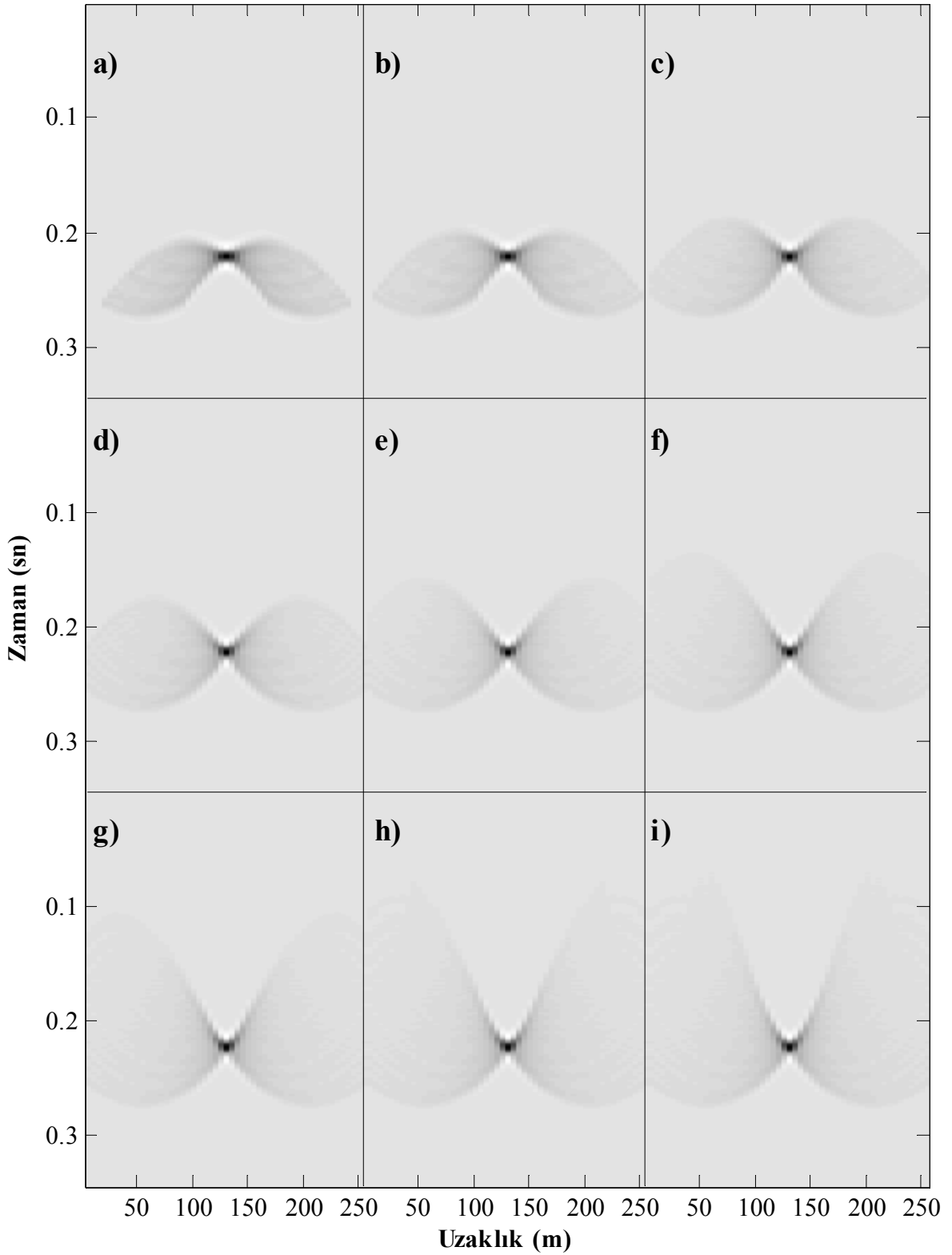


Şekil 3. 6. Nokta saçılma modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=75 m).

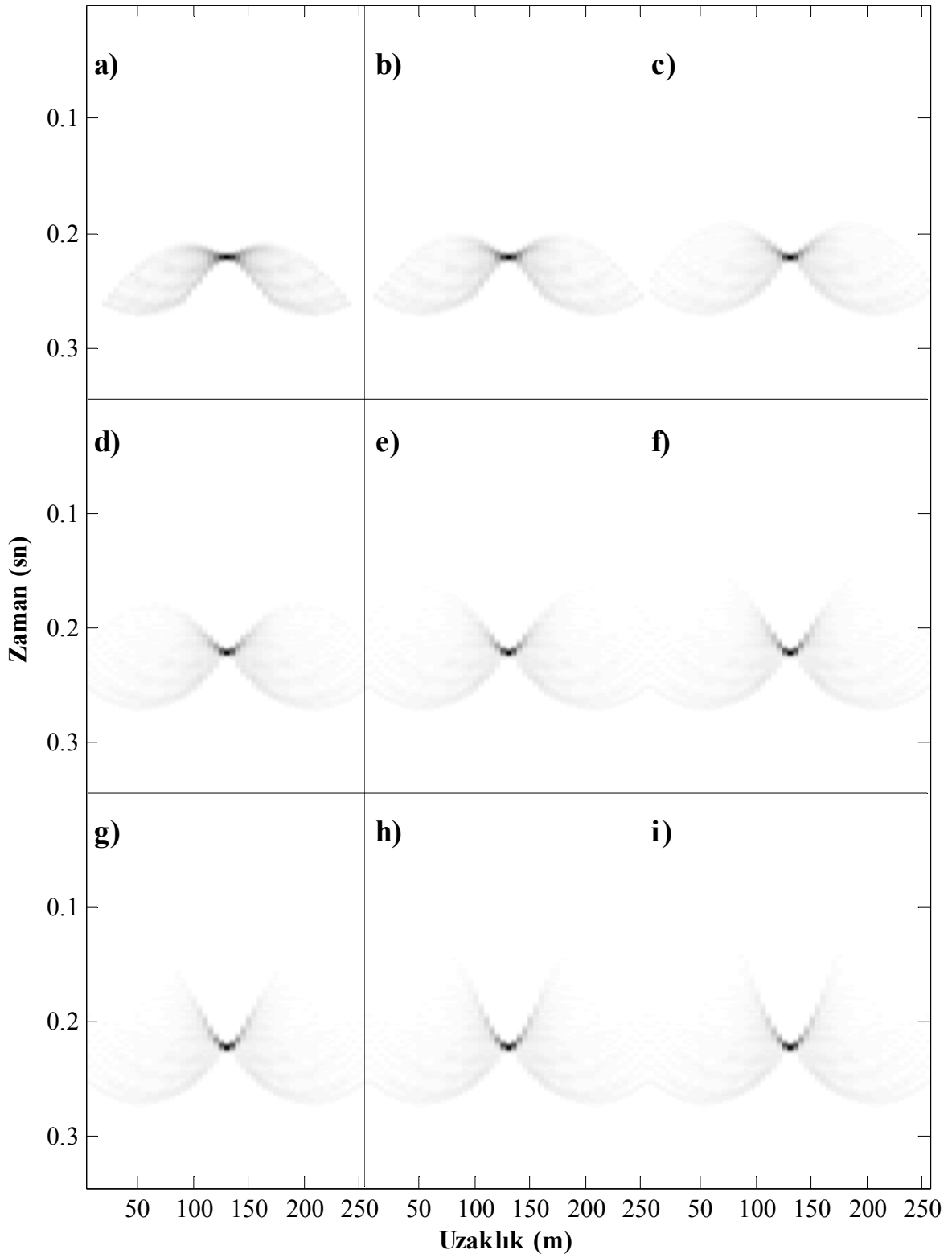


Şekil 3. 7. Nokta saçılma modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=75 m).

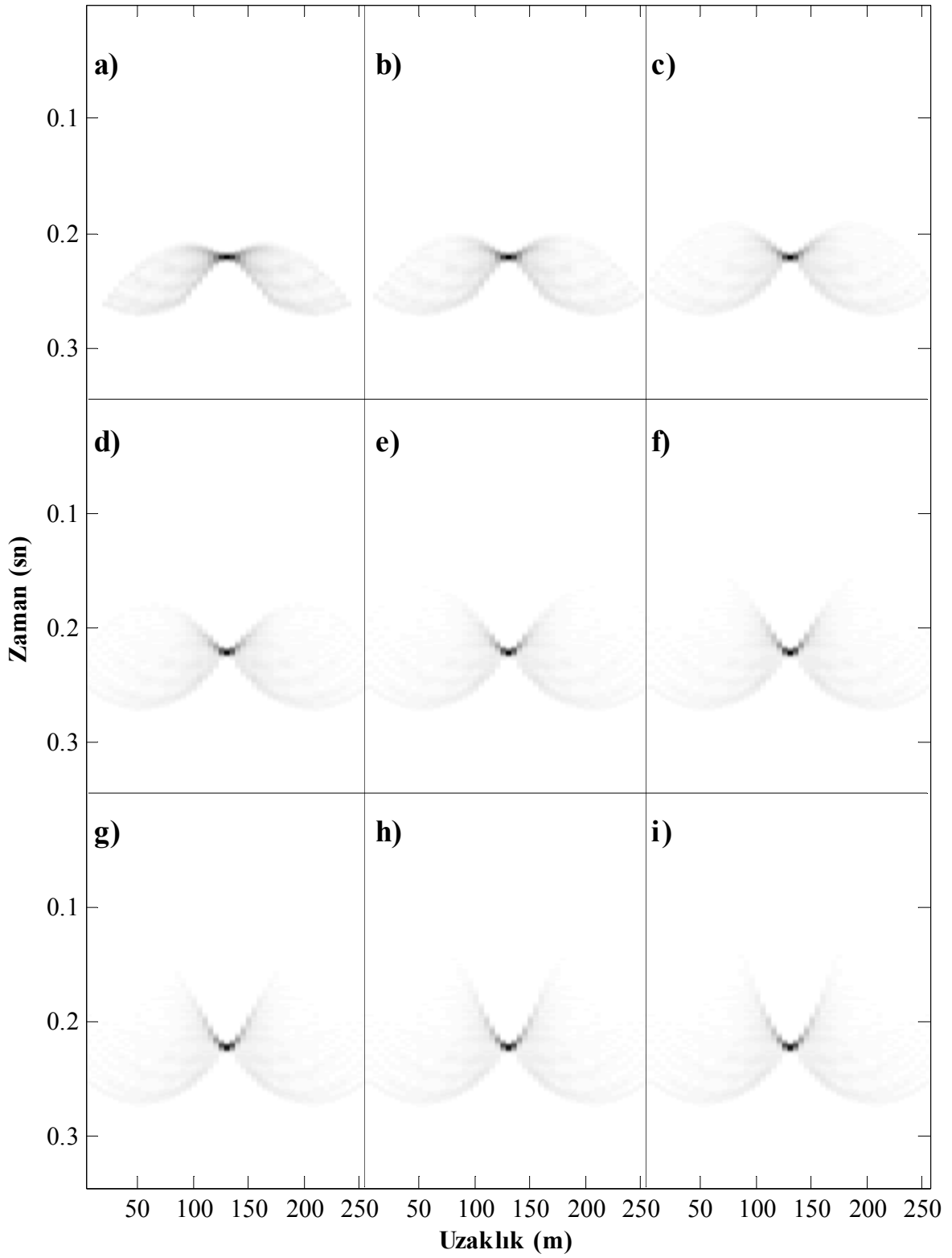
Nokta saçıcı model için farklı yarı operatör boyları (35 m, 45 m, 55 m, 65 m, 75 m, 85 m, 95 m, 105 m ve 115 m) ile Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile GD ve EKKGD işlemleri uygulanmıştır. Şekil 3.8 farklı yarı operatör boyları ile gerçekleştirilen Kirchhoff göç çıkışlarını göstermektedir. Model verisinde saçılma hiperbolünün boyu 150 m olduğunda yarı operatör boyu 75 m olmalıdır. Yarı operatör boyunun küçük (35 m, 45 m, 55 m ve 65 m) seçilmesi (Şekil 3.8a-d) tam bir odaklanma sağlayamamıştır. Aksine, yarı operatör boyunun büyük (85 m, 95 m, 105 m ve 115 m) seçilmesi saçılma hiperbolünde tam bir odaklanma sağlamıştır fakat göç hatalarını daha fazla yaymıştır. Şekil 3.9 yine aynı operatör yarı boyları ile uygulanan GD çıkışlarını göstermektedir. GD işleminin hata oranını azalttığı görülmektedir. Bunun yanında EKKG işleminin de hata oranını GD işlemi kadar azalttığı ve görüntü kalitesini arttırdığı görülmektedir (Şekil 3.10). Hız testleri için en küçük hata oranını veren EKKGD işlemi aynı etkiyi yarı operatör boyu testleri için de sağlayarak çözünürlüğü en iyi göç çıkışlarını üretmiştir (Şekil 3.11). EKKGD en iyi çözünürlüğü sağlamasına rağmen en maliyetli yöntem olmuştur (Şekil 3.12). EKKG ile GD işlemlerinin maliyetleri yaklaşık olarak eşittir. Kirchhoff göçü, çözüm zamanı en düşük olan yöntemdir. Ayrıca yarı operatör boyu arttıkça tüm yöntemler için çözüm zamanı artmaktadır. Hata oranlarına bakıldığında (Şekil 3.13) en düşük hata oranını EKKGD işlemi vermiştir. GD ve EKKG işlemleri aynı hata oranlarını üretmiştir. Kirchhoff göçü çözüm zamanı açısından en iyi yöntem olmasına rağmen en fazla göç hatasını üreten yöntem olmuştur. Çözüm zamanlarına (Şekil 3.12) ve hata oranlarına (Şekil 3.13) bakıldığında yarı operatör boyunun olması gereken değerden (75 m) küçük olması hata oranını yükseltirken, büyük olması hata oranını çok fazla yükseltmese de çözüm zamanını oldukça arttırmaktadır. Bu nedenle yarı operatör boyunun seçimi oldukça önemlidir.



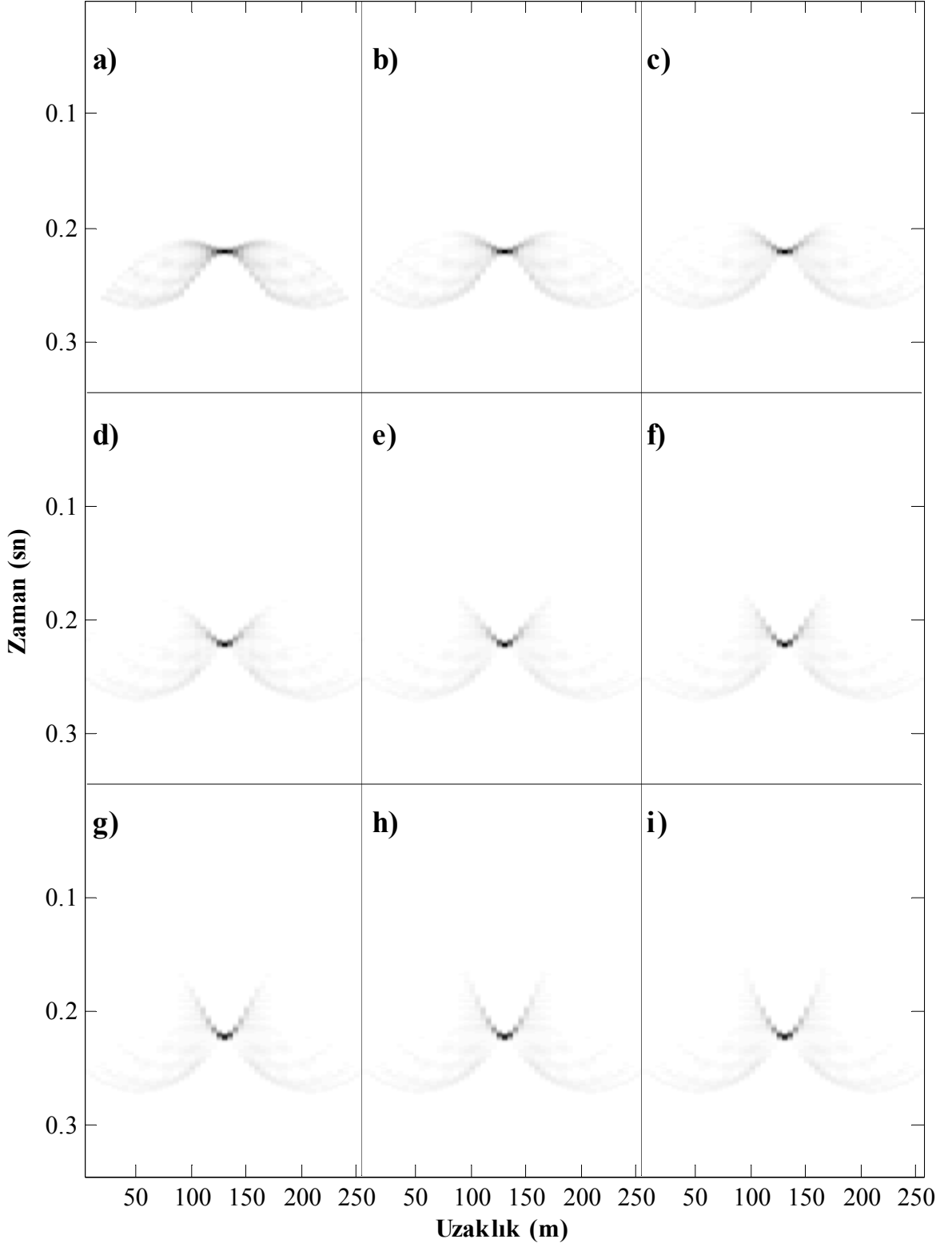
Şekil 3. 8. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).



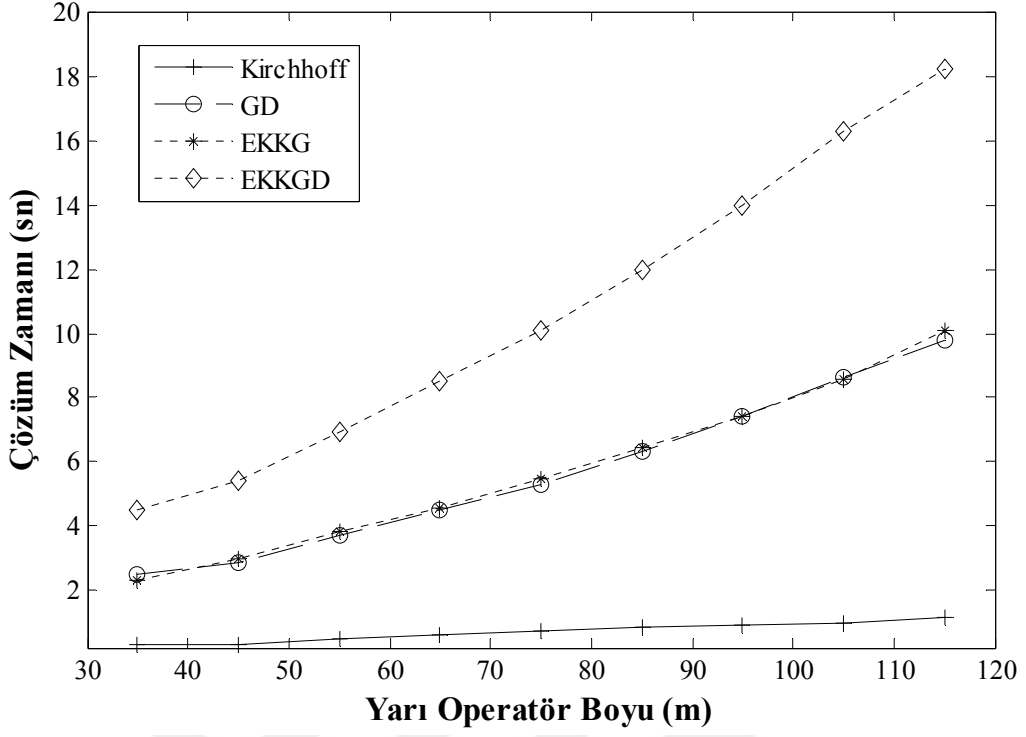
Şekil 3. 9. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).



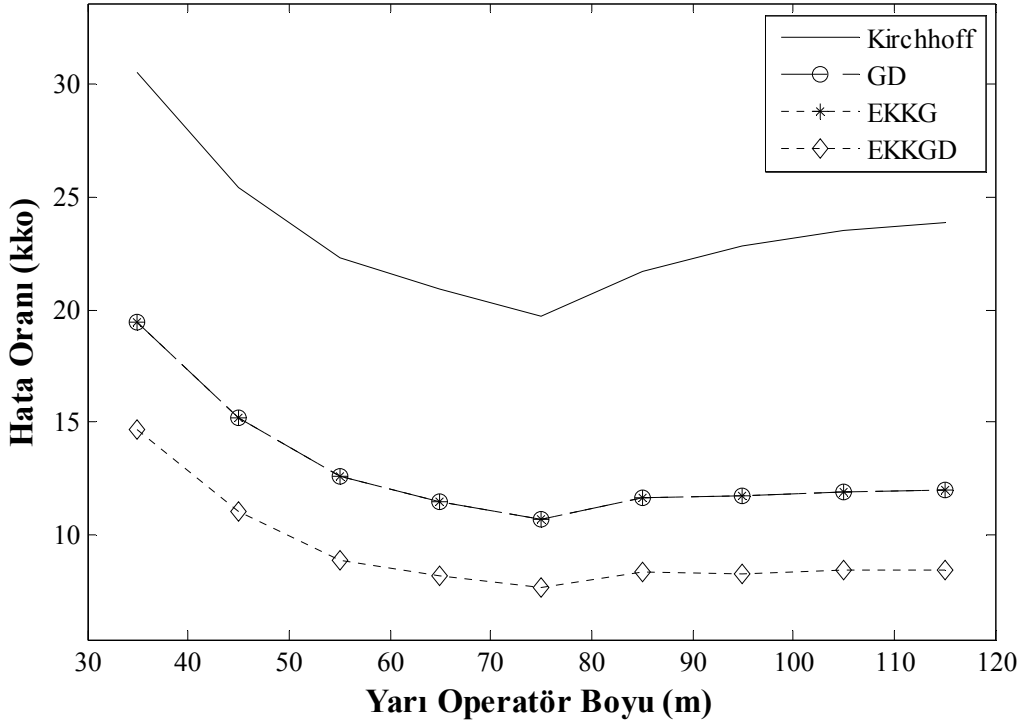
Şekil 3. 10. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).



Şekil 3. 11. Nokta saçılma modeli için a) 35 m, b) 45 m, c) 55 m, d) 65 m, e) 75 m, f) 85 m, g) 95 m, h) 105 m ve i) 115 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (hız=1000 m/sn).

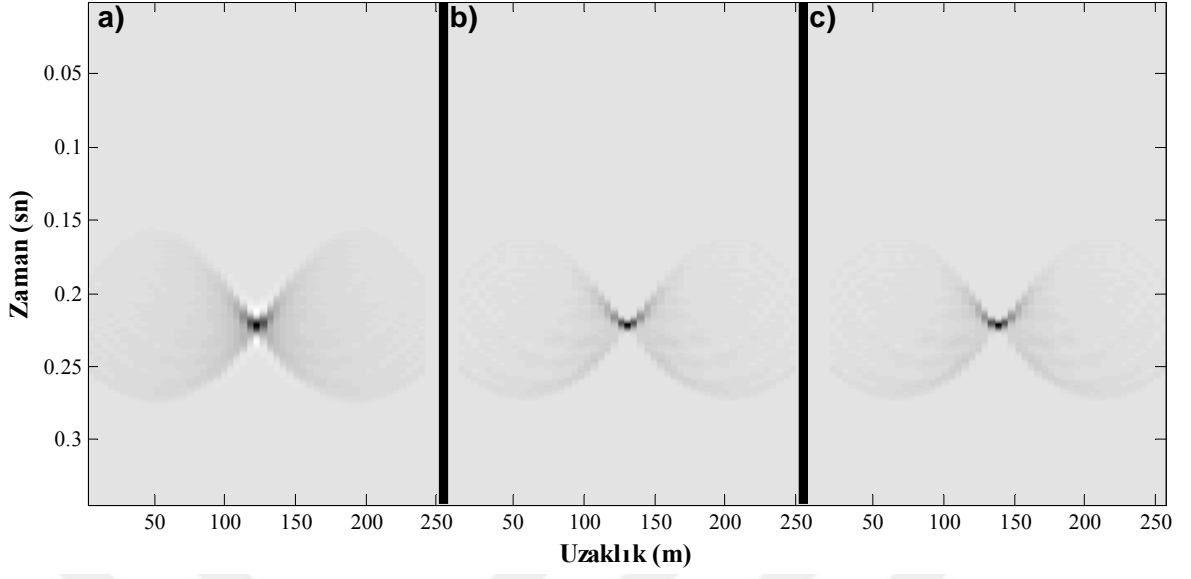


Şekil 3. 12. Nokta saçılma modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).

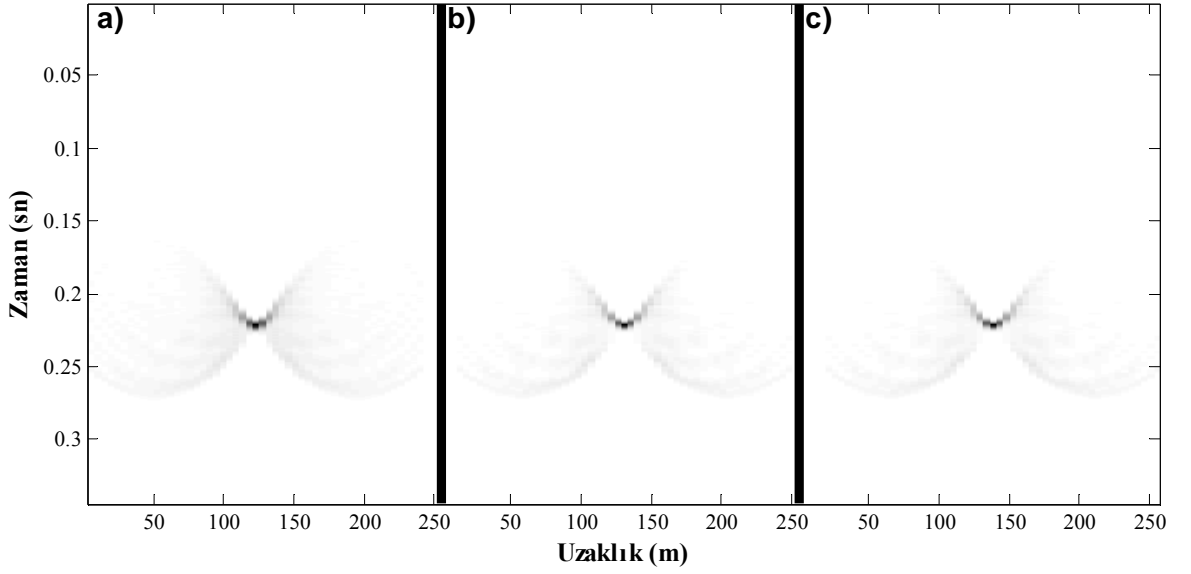


Şekil 3. 13. Nokta saçılma modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).

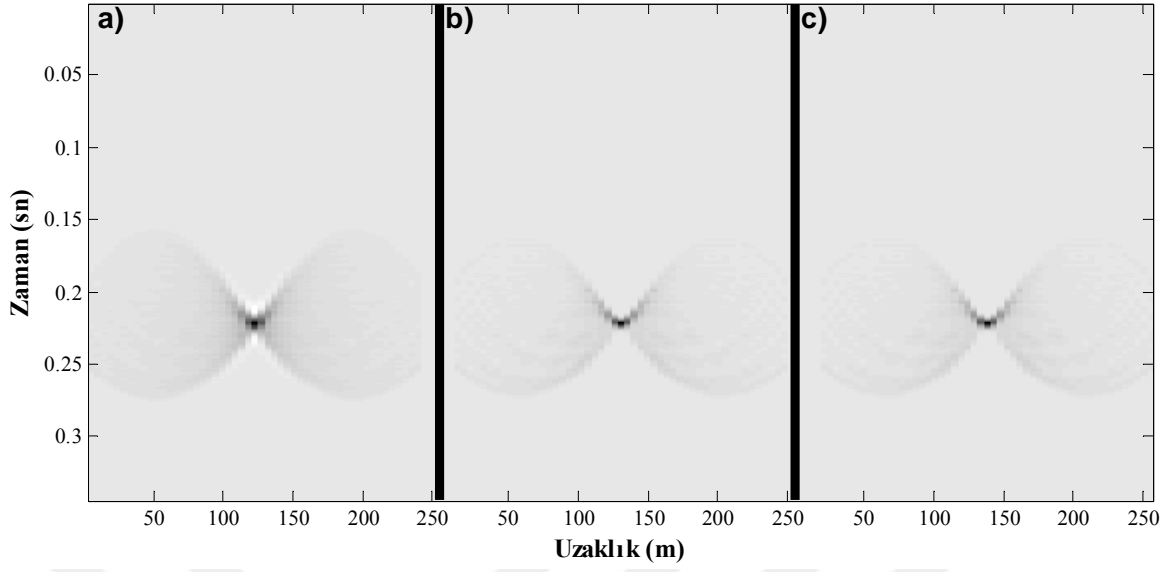
Nokta saçıcı model için yarı operatör boyu 75 m ve hız 1000 m/sn kullanılarak 27 farklı göç çıkışı üretilmiştir. Bunun yanında beklenen göç çıkışı (Şekil 3.1b) ile olan farklarının karelerinin toplamları (karekök ortalama, kko) da hesaplanmıştır. Tablo 3.1 nokta saçıcı model için tüm göç çıkışlarının çözüm zamanlarını, Tablo 3.2 ise nokta saçıcı model için tüm göç yöntemlerinin hata oranlarını göstermektedir. Nokta saçıcı model için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve Gauss-seidel yöntemi ile EKKG çıkışı Şekil 3.14 de gösterilmektedir. Buna göre, EKKG yöntemleri (Şekil 3.14b, c) Kirchhoff göç çıkışına (Şekil 3.14a) göre hata oranı daha düşük (Tablo 3.1) çıkışlar üretmiştir. Şekil 3.15 ile Şekil 3.22 arasında sırasıyla $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$ ve $cov((dd^T)^{-1})$, netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve Gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. Geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile gerçekleştirilen EKKGD çıkışları (Şekil 3.15b,c) GD çıkışına (Şekil 3.15a) göre daha düşük hata oranına sahiptirler (Tablo 3.2). $(L^T L)$ netleştirme fonksiyonu (Şekil 3.16) Kirchhoff göçüne göre hata oranını düşürmesine rağmen $(L^T L)^{-1}$ netleştirme fonksiyonundan daha fazla hata oranına sahiptir (Tablo 3.2). $cov(L^T L)$ netleştirme fonksiyonu (Şekil 3.17) yüksek hata oranında (Tablo 3.2) göç çıkışlarını yine yüksek maliyetle (Tablo 3.1) üretmiştir. $cov((L^T L)^{-1})$ netleştirme fonksiyonu (Şekil 3.18) son derece düşük hata oranlarını (Tablo 3.2) çok yüksek maliyetle (Tablo 3.1) üretmiştir. $(dd^T)^{-1}$ netleştirme fonksiyonu (Şekil 3.19) geleneksel netleştirme fonksiyonu ile aynı hata oranına ve görüntü kalitesine sahip çıkışlara GD işlemi için %35, EKKGD için %61 zamanda (Tablo 3.1) ulaşmıştır. (dd^T) netleştirme fonksiyonu (Şekil 3.20) çok düşük hata oranlarına (Tablo 3.2) oldukça düşük zamanlarda (Tablo 3.1) ulaşmıştır, fakat görüntü kalitesi istenen göç çıkışından oldukça uzaktır. $cov(dd^T)$ netleştirme fonksiyonu düşük maliyetle (Tablo 3.1) hata oranını en küçüklemiştir, ancak görüntü kalitesi istenen göç çıkışına yaklaşamamıştır (Şekil 3.21). $cov((dd^T)^{-1})$ netleştirme fonksiyonu (Şekil 3.22) $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile aynı hata oranlarını (Tablo 3.2) GD için %33, EKKGD için %63 zamanda elde etmiştir.



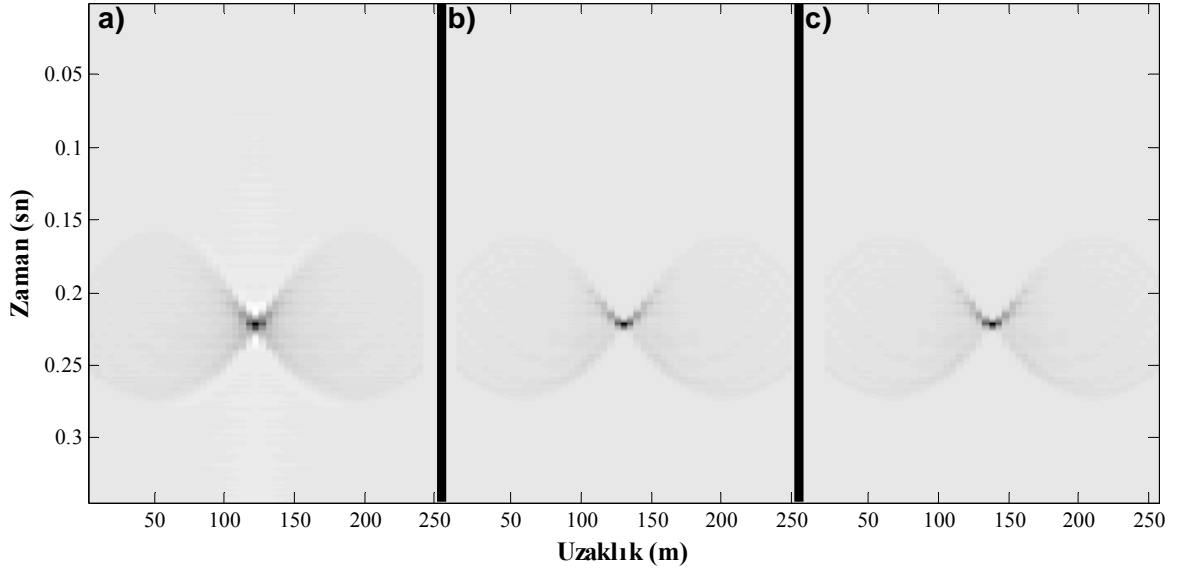
Şekil 3. 14. Nokta saçıcı model için (a) Kirchhoff göç, (b) eşlenik-gradyent yöntemi ile EKKG (c) Gauss-seidel yöntemi ile EKKG çıktıları.



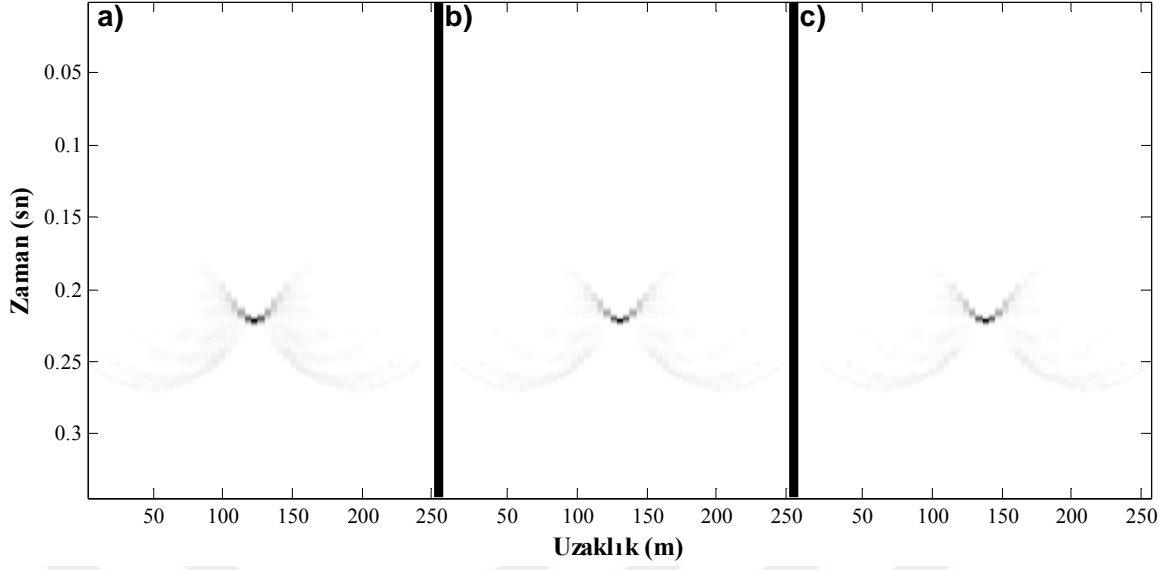
Şekil 3. 15. Nokta saçıcı model için $(L^T L)^{-1}$ netleştirme fonksiyonu ile (a) GD, (b) eşlenik-gradyent yöntemi ile EKKGD (c) Gauss-seidel yöntemi ile EKKGD çıktıları.



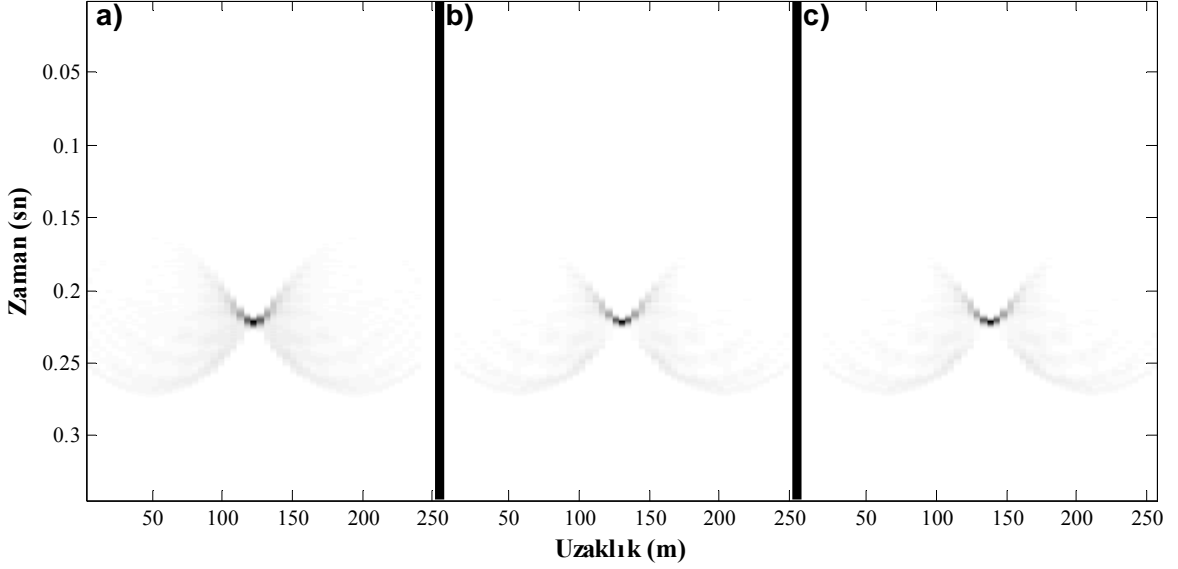
Şekil 3. 16. Nokta saçıcı model için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıktıları.



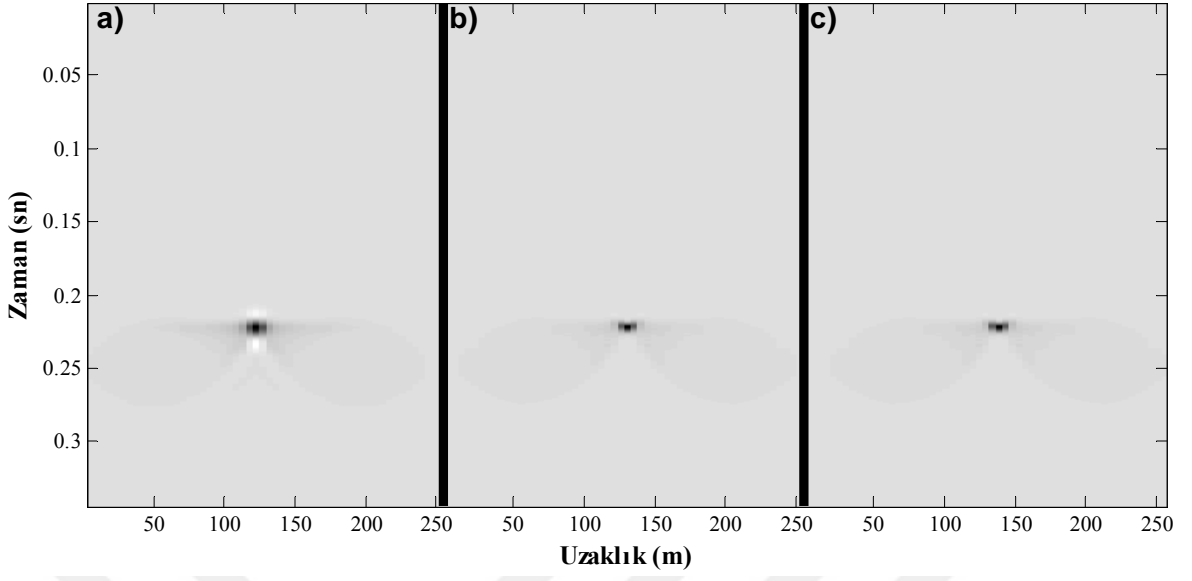
Şekil 3. 17. Nokta saçıcı model için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıktıları.



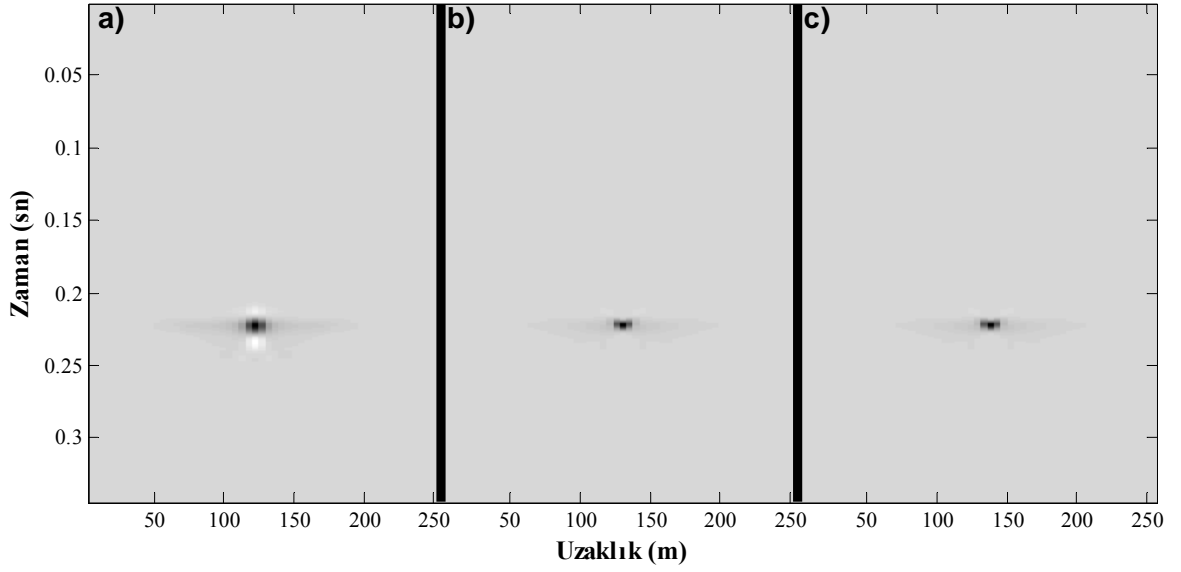
Şekil 3. 18. Nokta saçıcı model için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) gauss-seidel yöntemi ile EKKGD çıktıları.



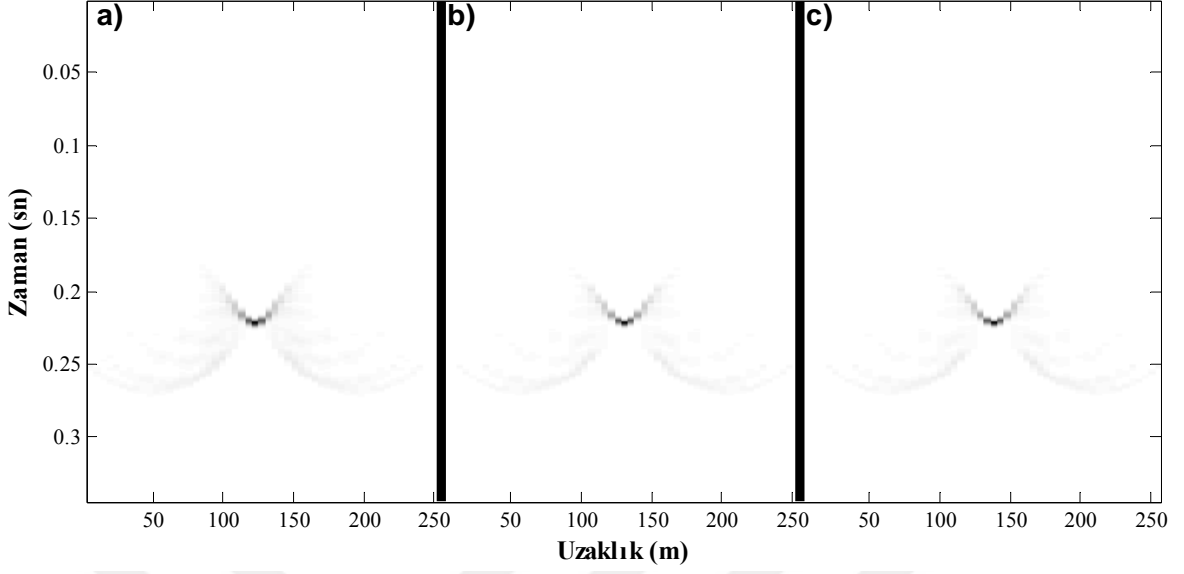
Şekil 3. 19. Nokta saçıcı model için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 20. Nokta saçıcı model için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 21. Nokta saçıcı model için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD c) Gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 22. Nokta saçıcı model için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD çıkışı c) Gauss-seidel yöntemi ile EKKGD çıkışları.

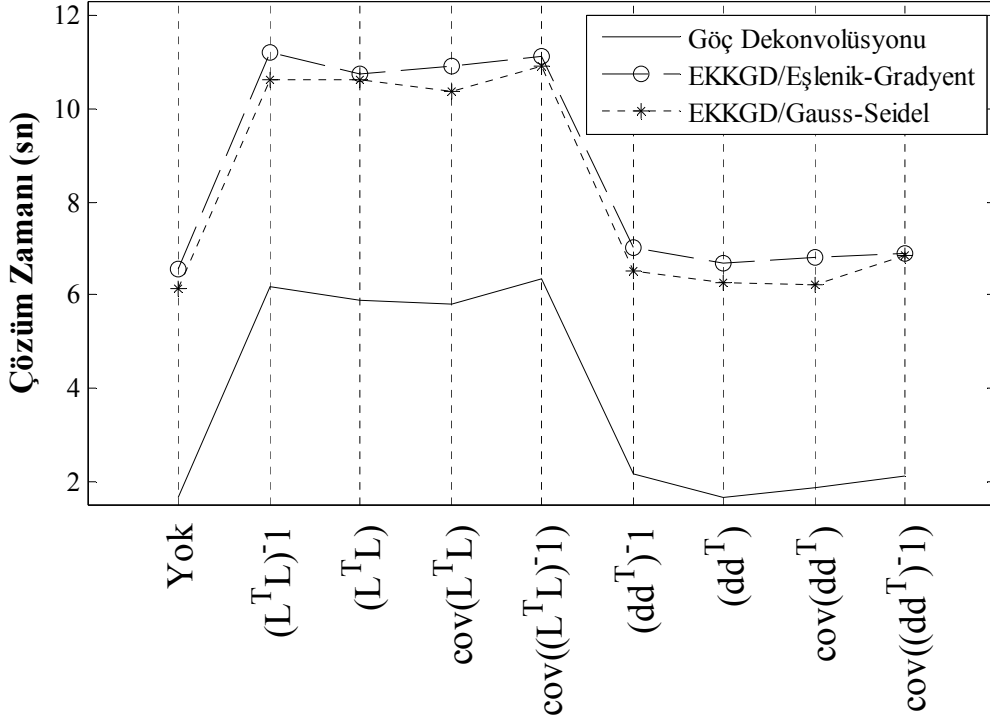
Tablo 3. 1. Nokta saçılma modeli için Netleştirme fonksiyonlarına göre çözüm zamanları açısından maliyet karşılaştırması. (İz sayısı: 51, zaman örnek sayısı: 344, yarı operatör boyu: 75 m, 15 iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD (sn)	EKKGD Eşlenik- Gradyent (sn)	EKKGD Gauss- Seidel (sn)	Netleştirme Fonk. Maliyeti
Yok Kirchhoff göçü/EKKG	1,67	6,54	6,15	En Düşük Maliyet
$(L^T L)^{-1}$	6,16	11,20	10,61	Yüksek Maliyet
$(L^T L)$	5,88	10,73	10,61	Yüksek Maliyet
$cov(L^T L)$	5,79	10,94	10,36	Yüksek Maliyet
$cov((L^T L)^{-1})$	6,35	11,12	10,94	Yüksek Maliyet
$(dd^T)^{-1}$	2,17	7,00	6,51	Düşük Maliyet
(dd^T)	1,76	6,66	6,26	Düşük Maliyet
$cov(dd^T)$	1,86	6,80	6,22	Düşük Maliyet
$cov((dd^T)^{-1})$	2,12	6,90	6,85	Düşük Maliyet

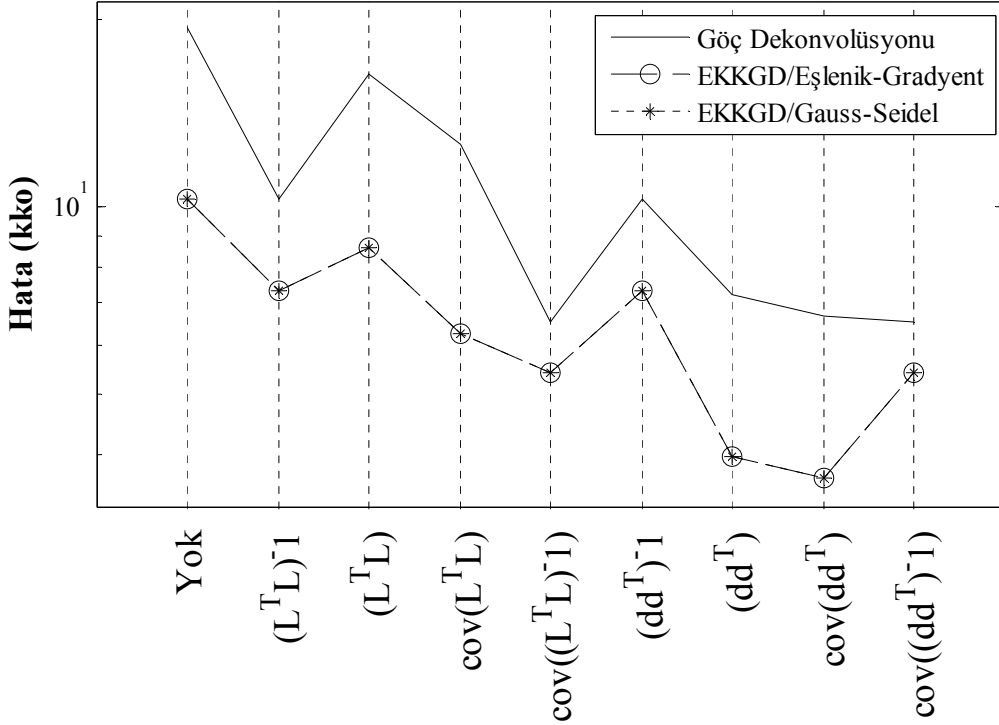
Tablo 3. 2. Nokta saçılma modeli için Netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.

Hata Oranı (<i>kko</i>)/ Netleştirme Fonksiyonu	GD	EKKGD Eşlenik- Gradyent	EKKGD Gauss- Seidel	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	19,46	10,29	10,29	Orta
$(L^T L)^{-1}$	10,29	7,31	7,31	Düşük
$(L^T L)$	16,37	8,60	8,60	Yüksek
$cov(L^T L)$	12,60	6,23	6,23	Yüksek
$cov((L^T L)^{-1})$	6,54	5,41	5,41	Çok Düşük
$(dd^T)^{-1}$	10,29	7,31	7,31	Düşük
(dd^T)	7,25	3,97	3,97	Çok Düşük
$cov(dd^T)$	6,67	3,67	3,67	Çok Düşük
$cov((dd^T)^{-1})$	6,54	5,41	5,41	Çok Düşük

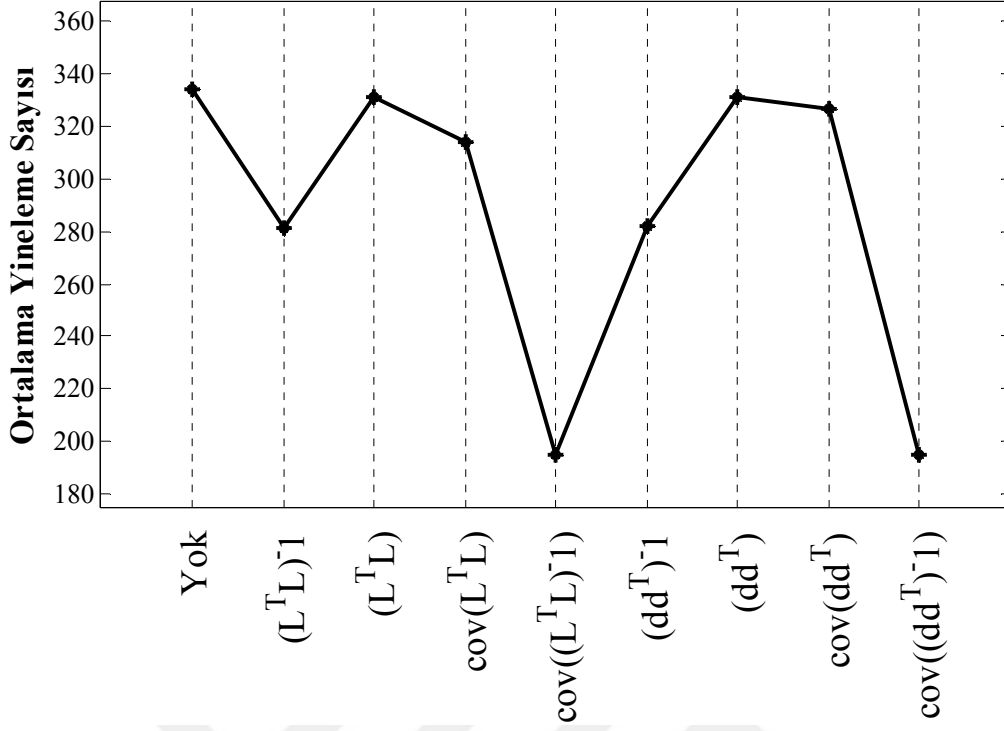
Şekil 3.23 nokta saçıcı model için tüm yöntemlerin çözüm zamanlarını grafik üzerinde göstermektedir. Giriş verisi (d) kullanılarak elde edilen göç çıkışları, GD için yaklaşık %72 daha az zamanda, EKKGD ise %46 daha az zamanda çözüme ulaşmışlardır. Şekil 3.24 göç çıkışlarının toplam karekök hatalarını göstermektedir. Buna göre, $(d^T d)$ ve $cov(d^T d)$ netleştirme fonksiyonları kullanılarak gerçekleştirilen göç işlemleri istenen çıkışa (nokta saçıcı, Şekil 3.1b) en fazla yaklaşan yöntemlerdir. Şekil 3.25 eşlenik-gradyent yöntemi için ortalama yineleme sayılarını göstermektedir. Buna göre $(L^T L)^{-1}$, (dd^T) , $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları yineleme sayılarını azaltabilen netleştirme fonksiyonları olduğu görülmektedir. Fakat yineleme sayısını en aza düşüren $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ fonksiyonlarıdır.



Şekil 3. 23. Nokta saçıcı model için uygulanan göç yöntemlerinin çözüm zamanları.



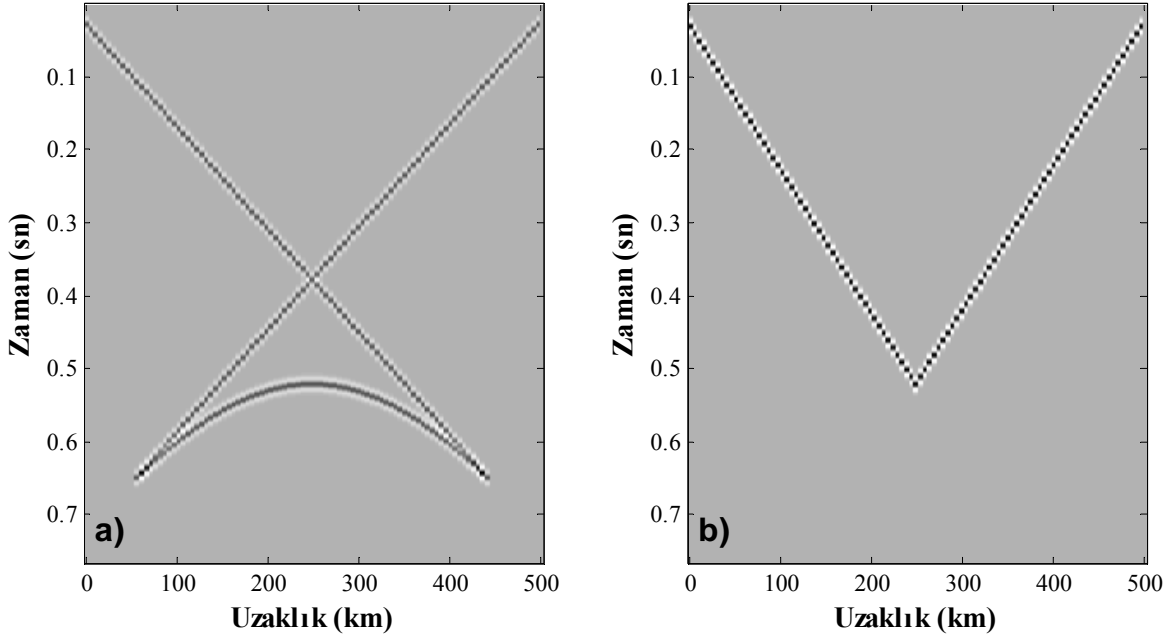
Şekil 3. 24. Nokta saçıcı model için uygulanan göç yöntemlerinin karekök ortalama hata (kko) miktarları.



Şekil 3. 25. Nokta saçılma modeli için eşlenik-gradyent ile uygulanan EKKGD yöntemi için ortalama yineleme sayıları.

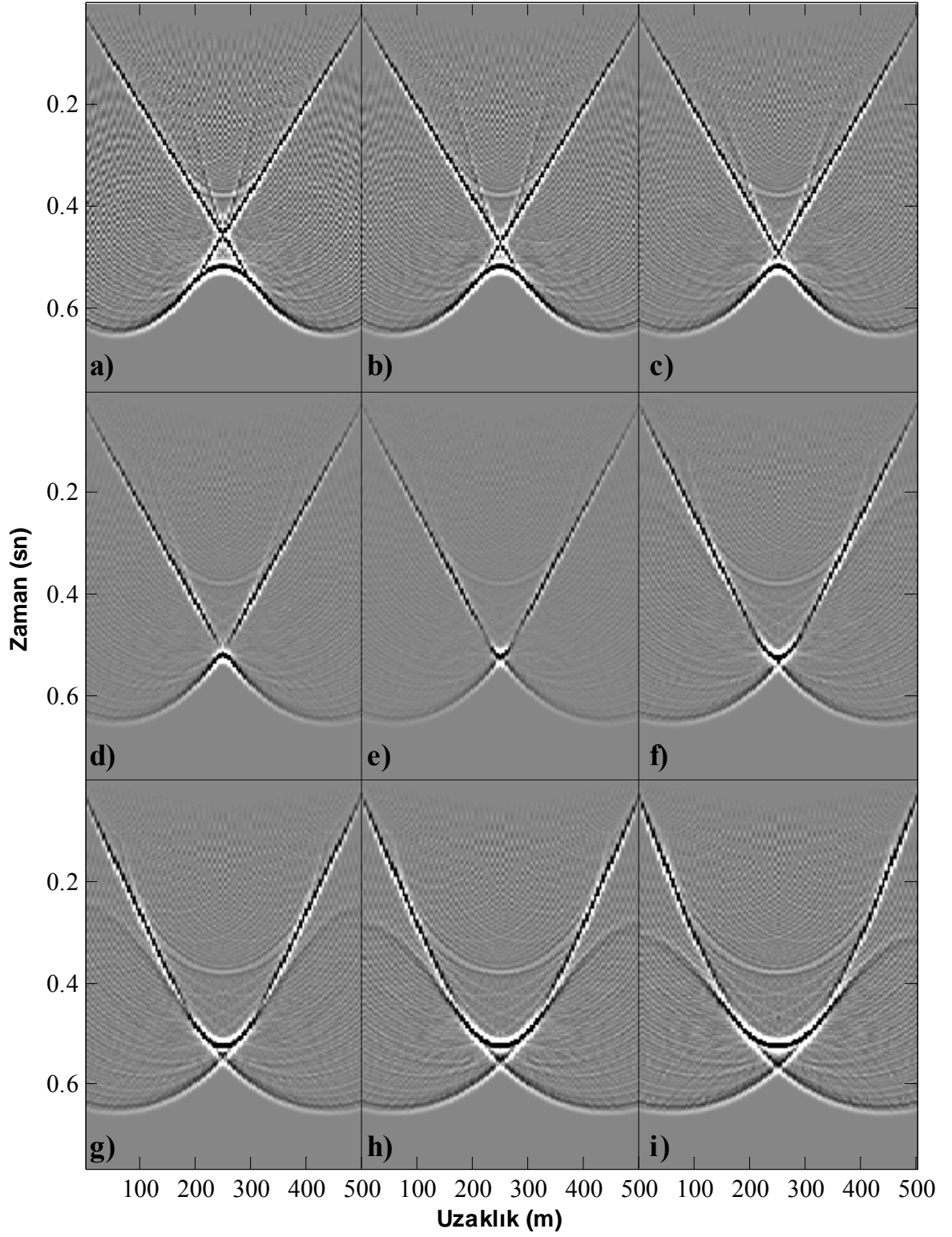
3.1.2. Papyon-Kravat Modeli Üzerinde Göç Uygulamaları

Papyon-kravat (bow-tie) modeli göç algoritmalarını test için önemli bir modeldir ve senklinal tipi yansıtıcılardan kaynaklanırlar. Şekil 3.26a'daki model kesit hem eğimli bir yansıtıcı hem de senklinal etkilerini içermektedir. Model hızı 1000 m/sn, örnekleme aralığı (dt) 0.001 sn, kayıt süresi (T) 0.766 sn, iz aralığı (dx) 5 m ve toplam kesit uzunluğu 500 m dir. Beklenen göç çıkışı veya gerçek yeraltı yansıtıcı modeli Şekil 3.26b de gösterilmektedir. Papyon-kravat modeli üzerinde farklı hız fonksiyonları ve farklı yarı operatör boyları ile testler gerçekleştirilmiş ve daha sonra farklı netleştirme fonksiyonları ile göç çıkışları elde edilmiştir.

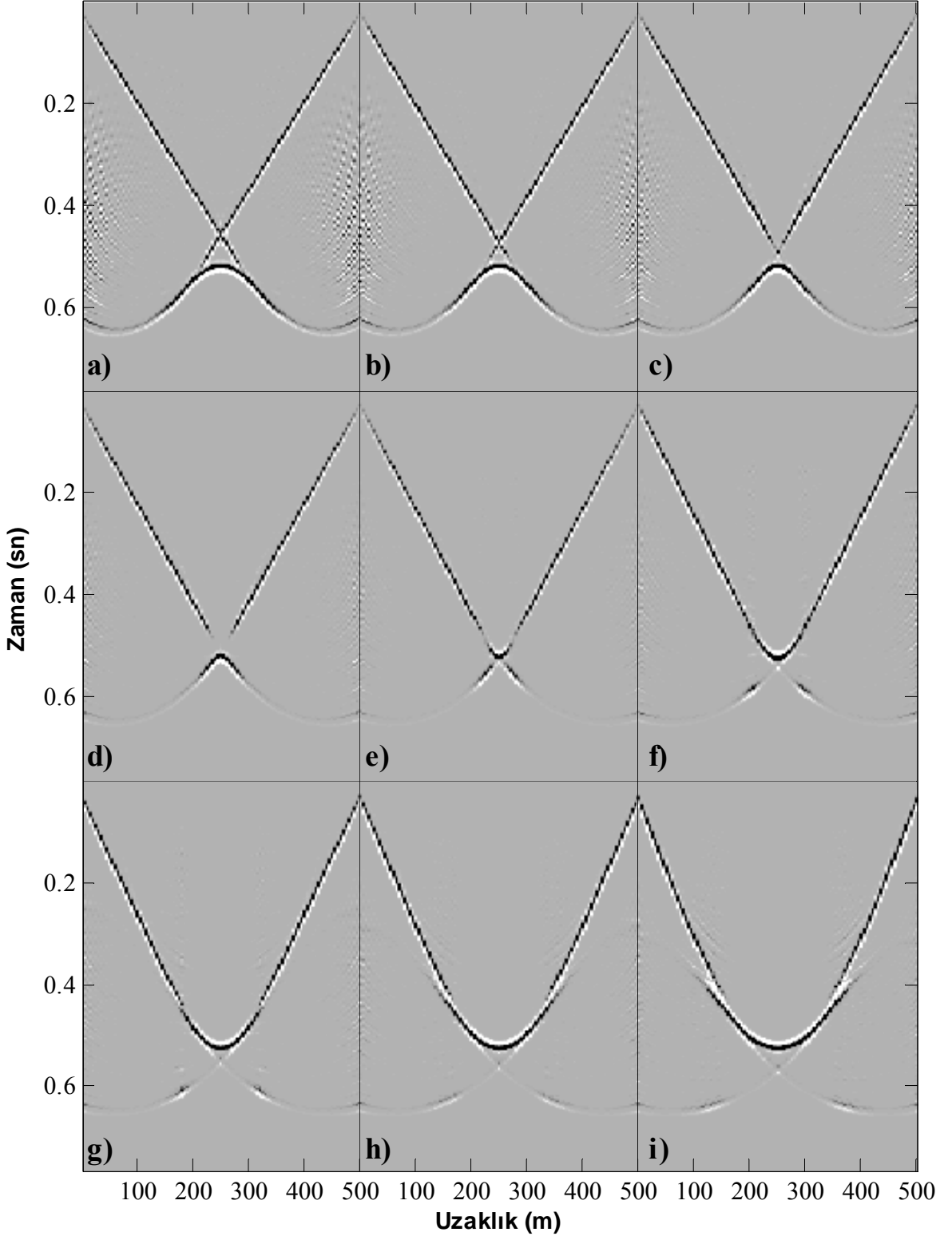


Şekil 3. 26. a) Papyon-kravat modeli verisi ve b) beklenen göç çıkışı.

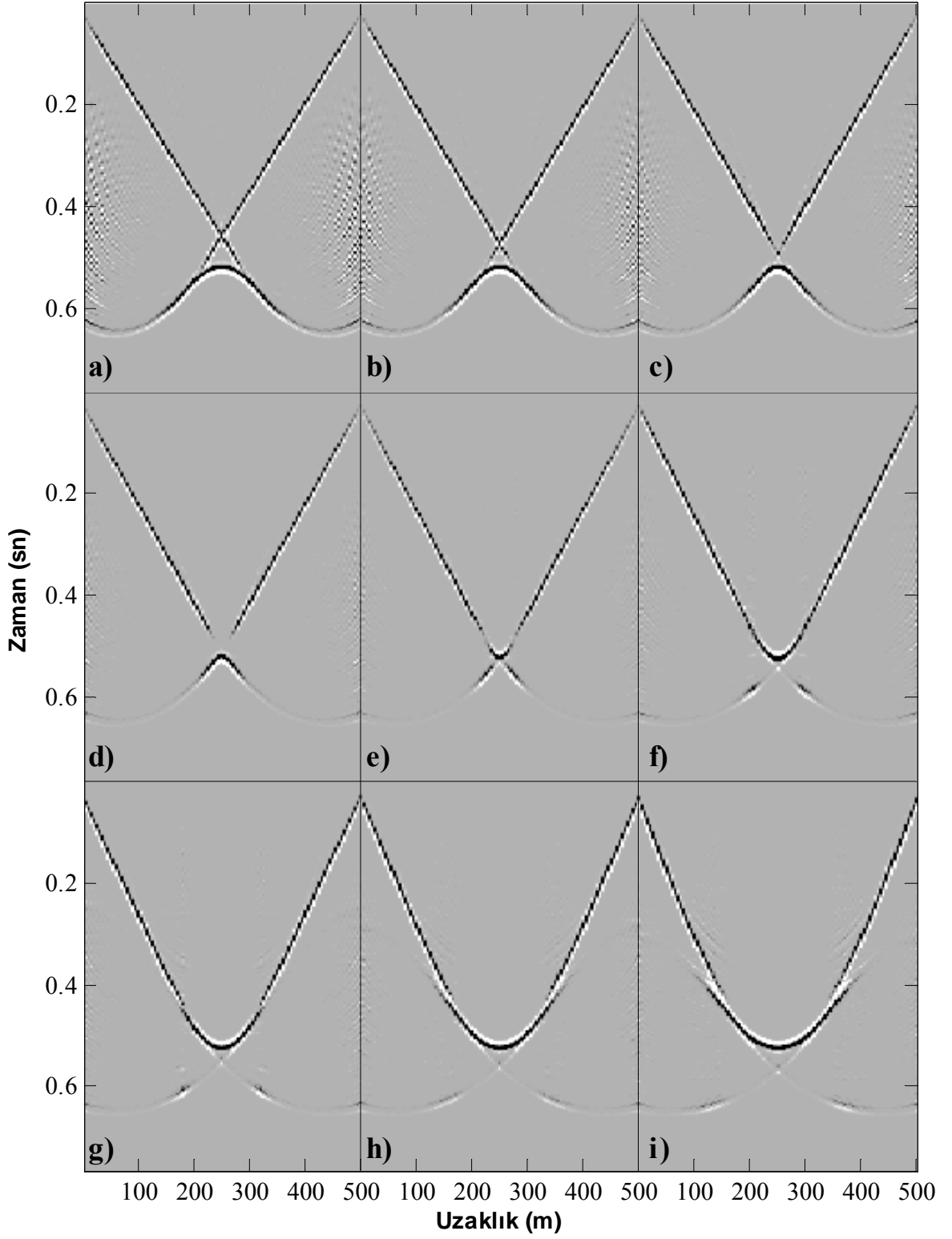
Papyon-kravat model için %20 eksik hız fonksiyonundan (800 m/sn) %20 fazla hız fonksiyonuna (1200 m/sn) Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile GD ve EKKGD işlemleri uygulanmıştır. Şekil 3.27 Kirchhoff göç çıkışlarını göstermektedir. Düşük hız bilgisi papyon-kravat yapısını çözmemiştir (Şekil 3.27a-c). Yüksek hız bilgisi senklinalin alt kısmına aşırı göç uygulayarak ovalleştirmiştir (Şekil 3.27g-i). %5 eksik hız fonksiyonu ile elde edilen göç çıkışı (Şekil 3.27d) istenen göç çıkışına oldukça yaklaşıırken, %5 fazla hız fonksiyonu kullanılarak elde edilen Kirchhoff göç çıkışı (Şekil 3.27e) daha fazla hatalı göç çıkışı üretmiştir. Şekil 3.28 GD çıkışlarını göstermektedir. GD çıkışları göç hatalarını sönümleyerek görüntü kalitesini arttırsa da eksik veya aşırı hız bilgisinin neden olduğu hataları giderememiştir. Şekil 3.29 EKKG çıkışlarını göstermektedir. Şekil 3.30 aynı hız fonksiyonları ile uygulanan EKKGD çıkışlarını göstermektedir. EKKGD çıkışları, GD ve EKKG çıkışlarına göre ayrımlığı yüksek çıkışlar üretmiştir. Şekil 3.31 dört farklı yöntemin çözüm zamanlarını göstermektedir. Buna göre tüm hız fonksiyonları için Kirchhoff göçü en hızlı yöntemdir. EKKG ve GD işlemlerinin çözüm zamanları yaklaşık eşittir. En maliyetli yöntem Kirchhoff göç çıkışının yaklaşık 18 katı maliyetle EKKGD işlemidir. Fakat hata oranını en aza (Şekil 3.32) düşüren yöntem de EKKGD yöntemidir. Şekil 3.32 incelendiğinde, GD ile EKKG işlemlerinin aynı hata oranlarını ürettiği görülmektedir. Hız fonksiyonu doğru hız fonksiyonuna (1000 m/sn) yaklaştıkça hata oranının düştüğü görülmektedir.



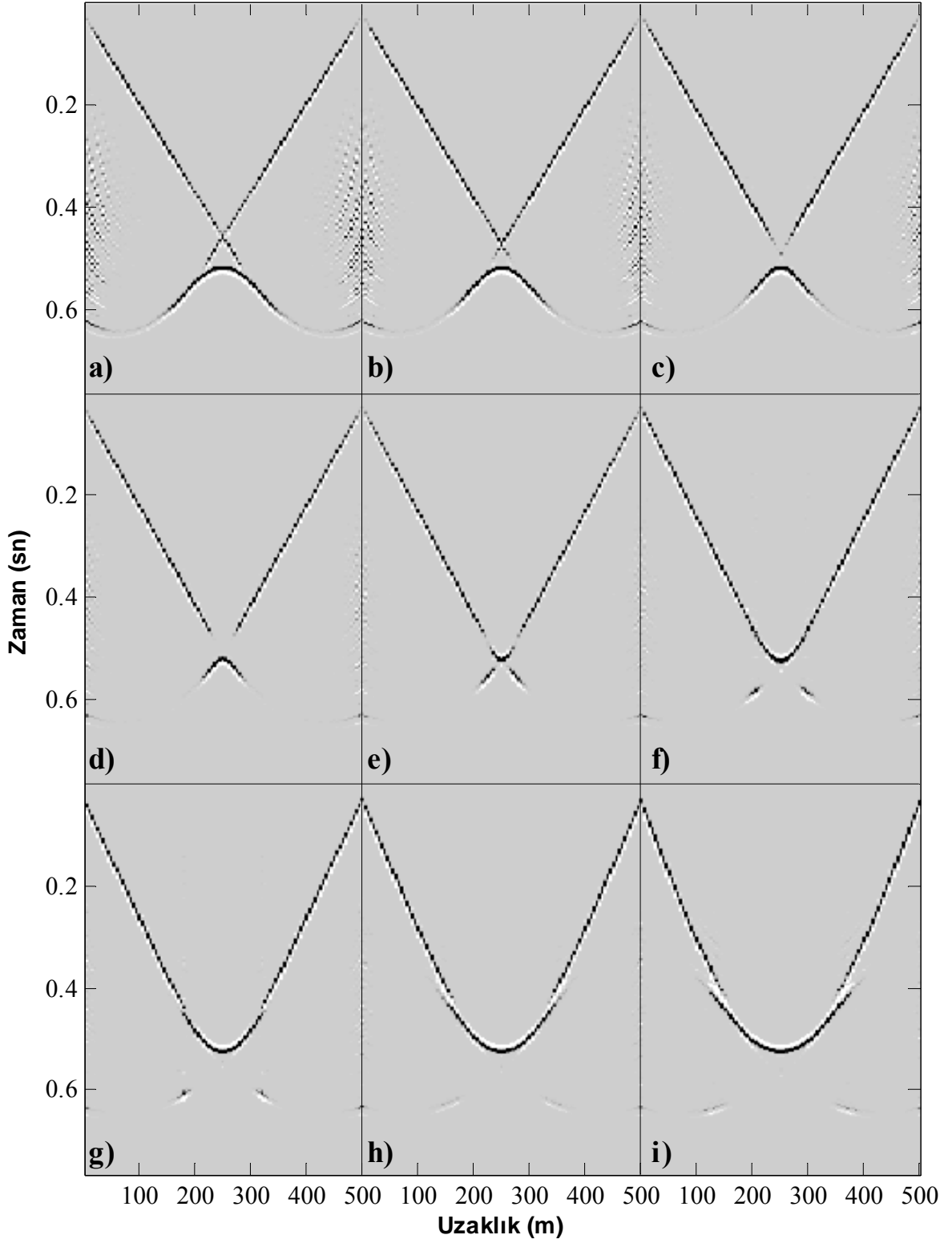
Şekil 3. 27. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=250 m).



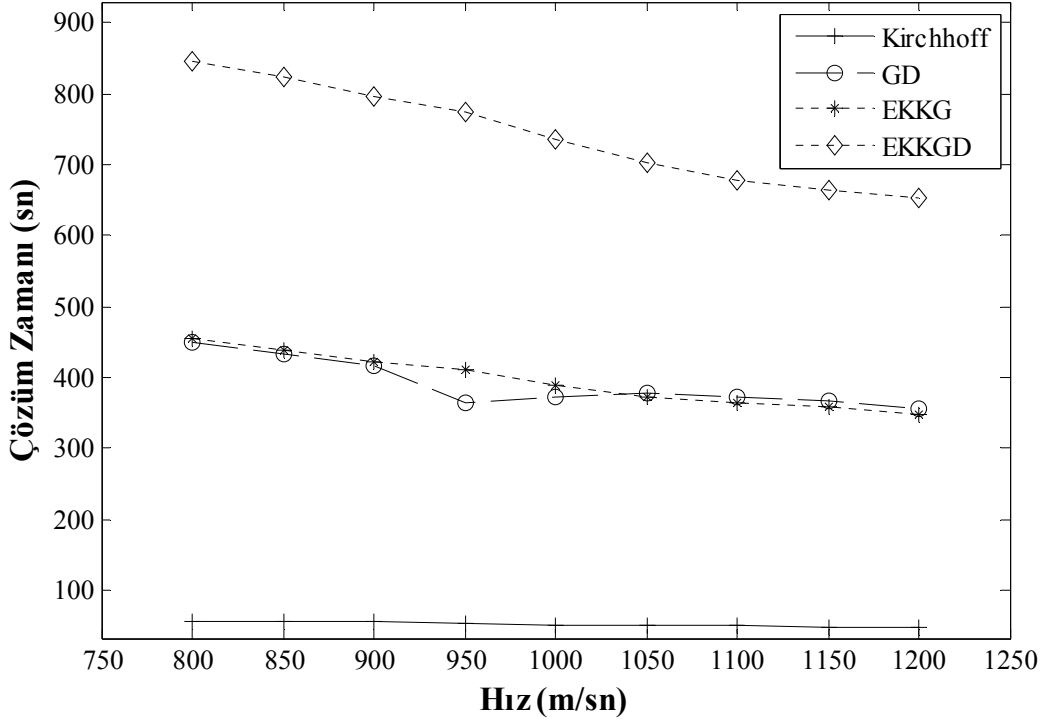
Şekil 3. 28. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=250 m).



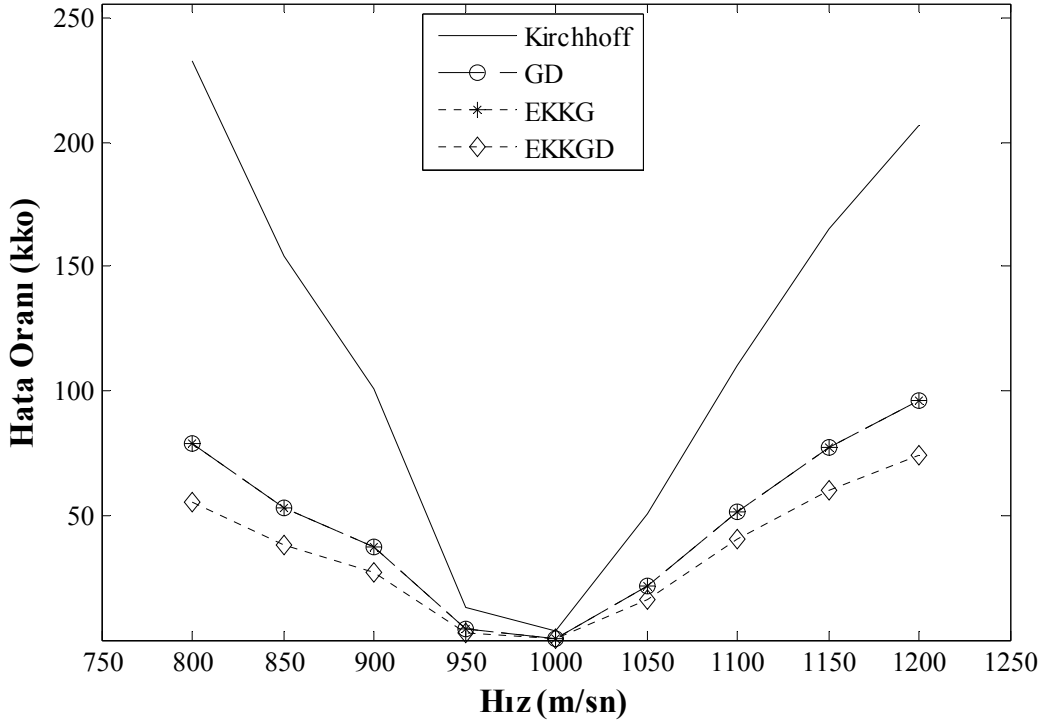
Şekil 3. 29. Papyon-krvat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKKG çıkışları (yarı operatör boyu=250 m).



Şekil 3. 30. Papyon-kravat modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=250 m).

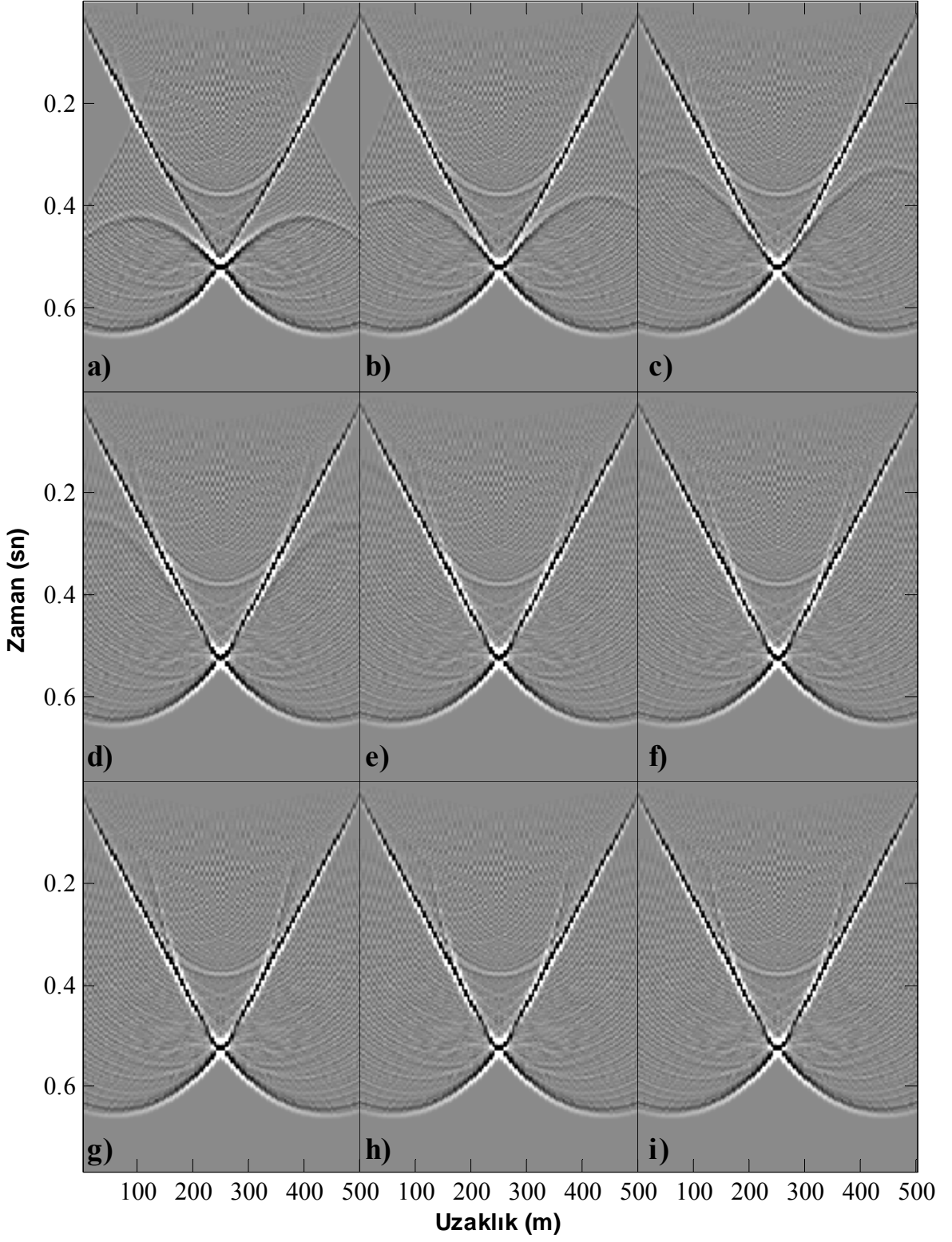


Şekil 3. 31. Papyon-kravat modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=250 m).

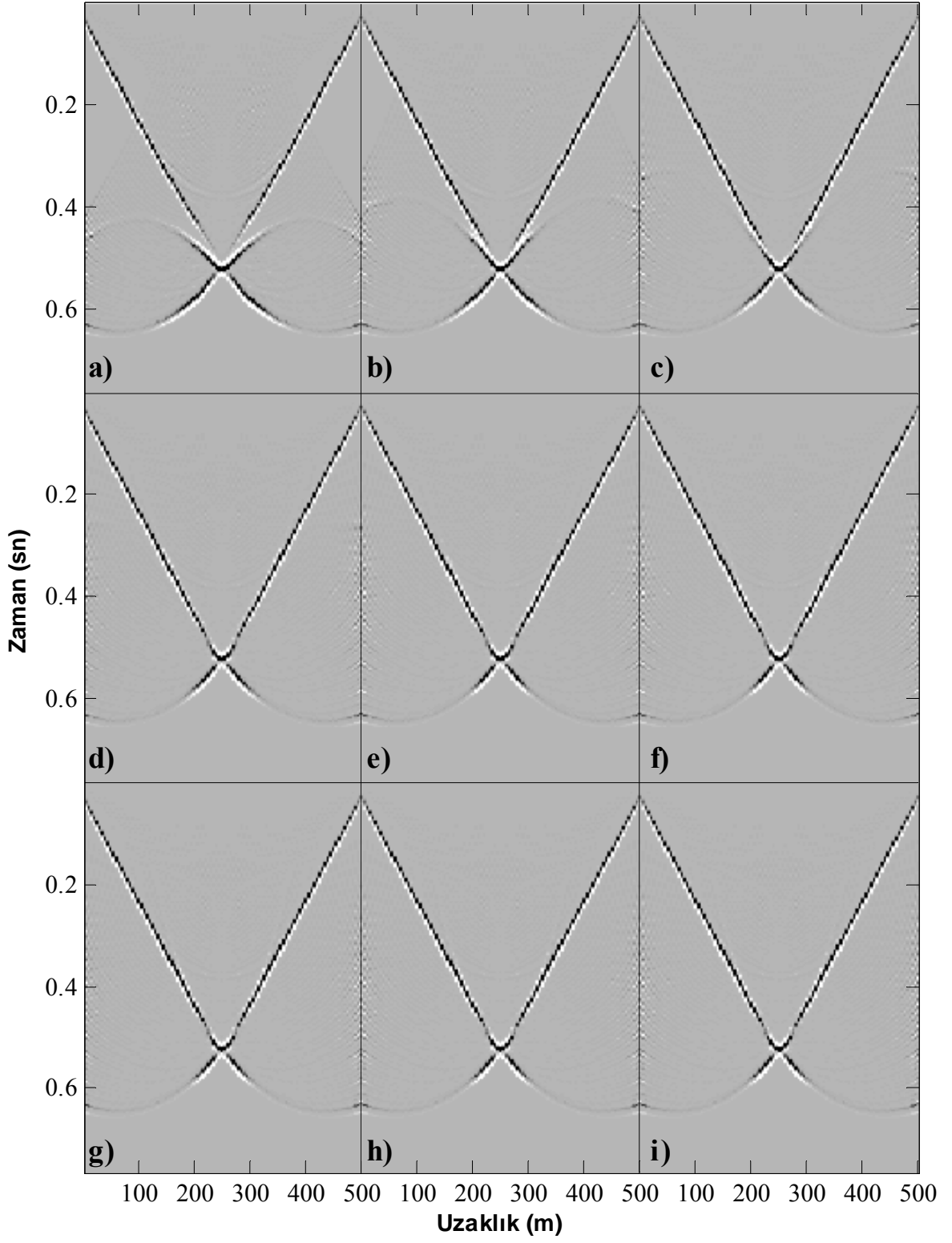


Şekil 3. 32. Papyon-kravat modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=250 m).

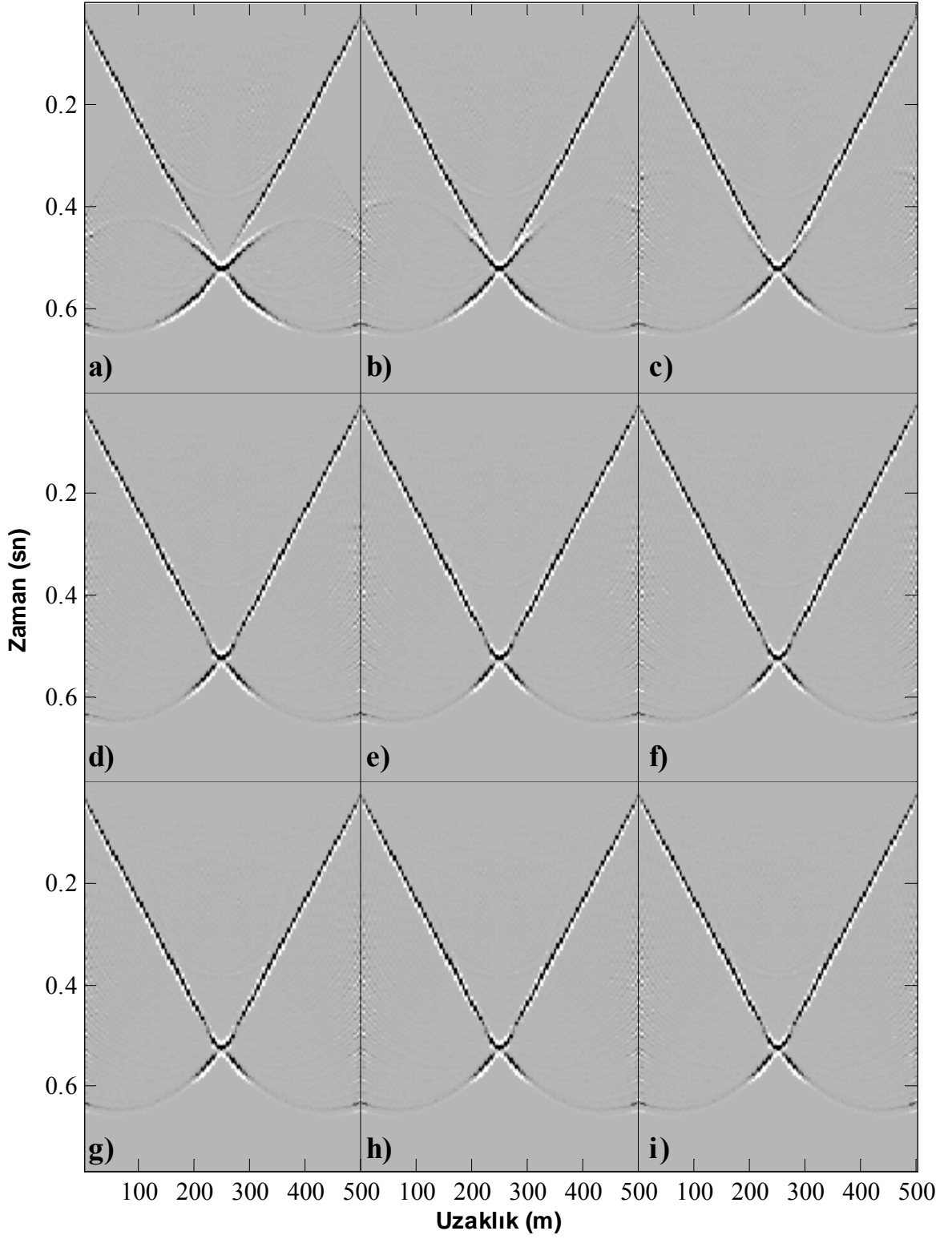
Papyon-kravat modeli için farklı yarı operatör boyları (150 m, 175 m, 200 m, 225 m, 250 m, 275 m, 300 m, 325 m ve 350 m) ile Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile GD ve EKKGD işlemleri uygulanmıştır. Şekil 3.33 farklı yarı operatör boyları ile gerçekleştirilen Kirchhoff göç çıkışlarını göstermektedir. Yarı operatör boyunun küçük (150 m, 175 m, 200 m, 225 m) seçilmesi (Şekil 3.33a-d) senklinalin eğimli kanatlarını düzeltse de çukur noktasını düzeltmekte başarılı olamamıştır. Aksine, yarı operatör boyunun büyük (275 m, 300 m, 325 m ve 350 m) seçilmesinin çıkışlar üzerinde herhangi bir etkisi olmamıştır. Şekil 3.34 yine aynı operatör yarı boyları ile uygulanan GD çıkışlarını göstermektedir. GD işleminin hata oranını azalttığı görülmektedir. Bunun yanında EKKG işleminin de hata oranını GD işlemi kadar azalttığı ve görüntü kalitesini arttırdığı görülmektedir (Şekil 3.35). EKKGD işlemi yarı operatör boyu testleri için de en küçük hata oranını sağlayarak çözünürlüğü en iyi göç çıkışlarını üretmiştir (Şekil 3.36). EKKGD en iyi çözünürlüğü sağlamasına rağmen en maliyetli yöntem olmuştur (Şekil 3.37). EKKG ile GD işlemlerinin maliyetleri yaklaşık olarak eşittir. Kirchhoff göçü, çözüm zamanı en düşük olan yöntemdir. Ayrıca yarı operatör boyu arttıkça tüm yöntemler için çözüm zamanı artmaktadır. Hata oranlarına bakıldığında (Şekil 3.38) en düşük hata oranını EKKGD işlemi vermiştir. GD ve EKKG işlemleri aynı hata oranlarını üretmiştir. Kirchhoff göçü çözüm zamanı açısından en iyi yöntem olmasına rağmen en fazla göç hatasını üreten yöntem olmuştur. Şekil 3.38 hata oranlarını göstermektedir. Özellikle GD, EKKG ve EKKGD eğrilerine bakıldığında 150 m den 250 m yarı operatör boyuna kadar hata oranlarında bir azalım gözlenirken, 250 m yarı operatör boyundan sonra sabit kalmıştır. Çözüm zamanlarına ve hata oranlarına bakıldığında yarı operatör boyunun olması gereken değerden (250 m) küçük olması hata oranını yükseltirken, büyük olması hata oranını çok fazla yükseltmese de çözüm zamanını oldukça arttırmaktadır.



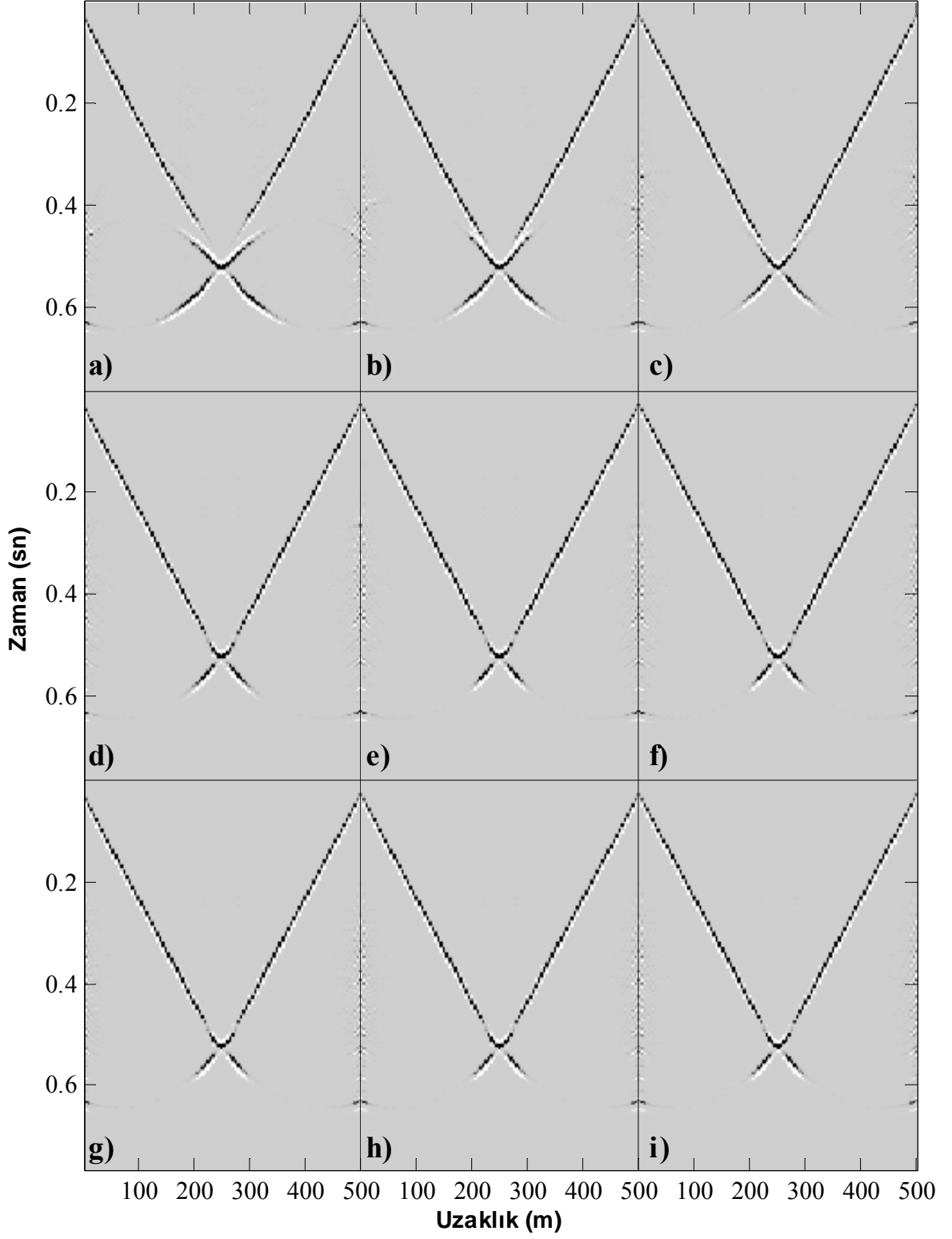
Şekil 3. 33. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).



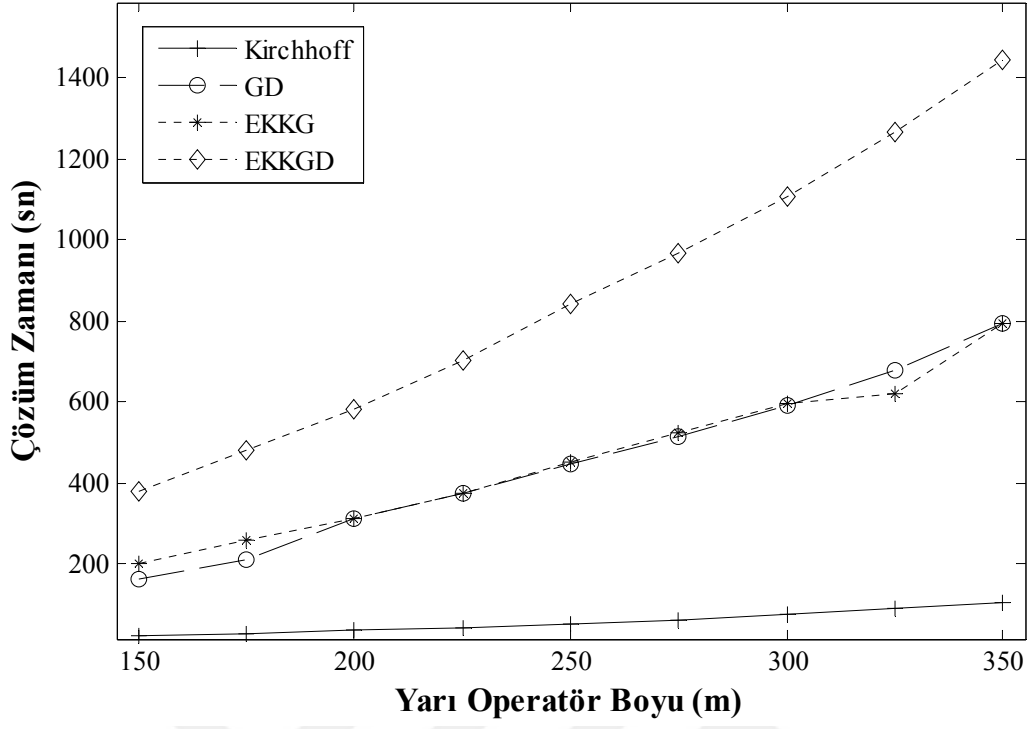
Şekil 3. 34. Papyon-kravat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).



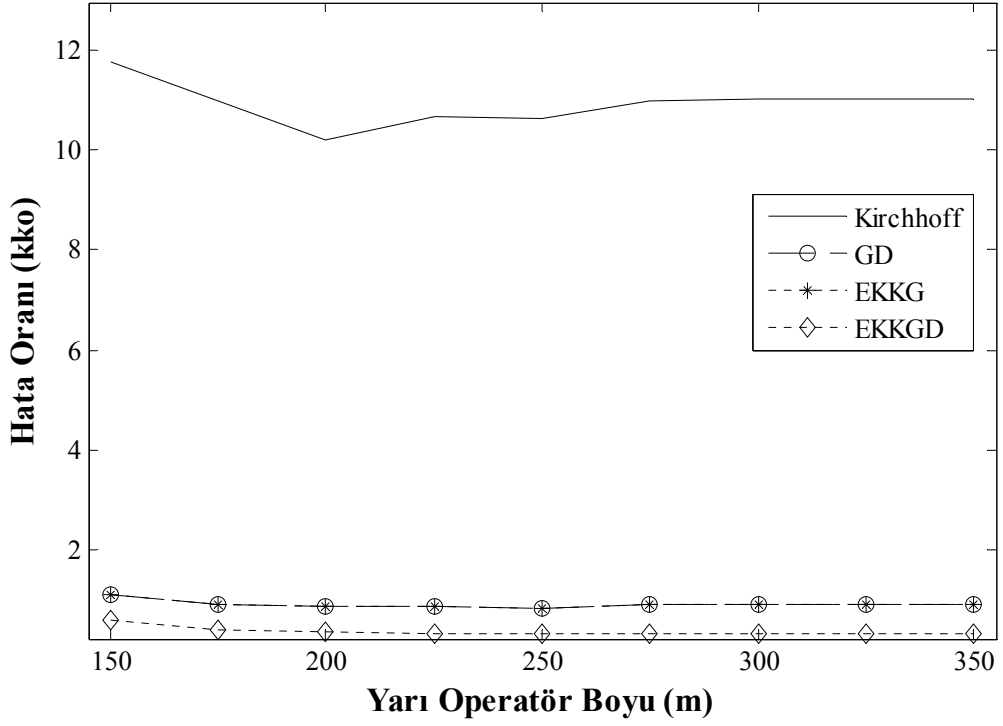
Şekil 3. 35. Papyon-kıavat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).



Şekil 3. 36. Papyon-kıavat modeli için a) 150 m, b) 175 m, c) 200 m, d) 225 m, e) 250 m, f) 275 m, g) 300 m, h) 325 m ve i) 350 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıktıları (hız=1000 m/sn).

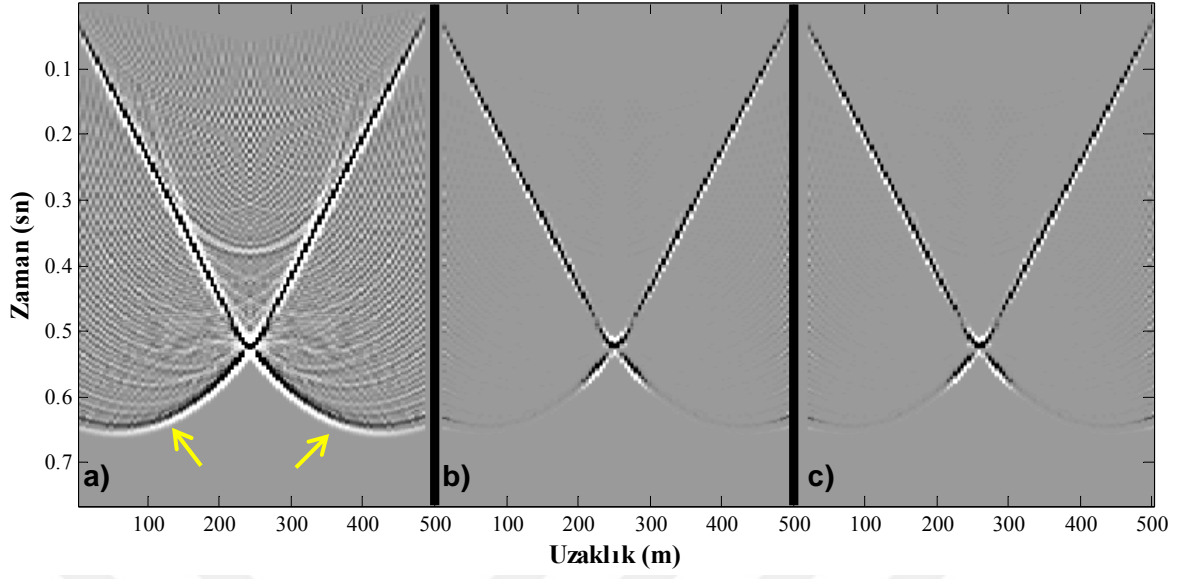


Şekil 3. 37. Papyon-kravat modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).

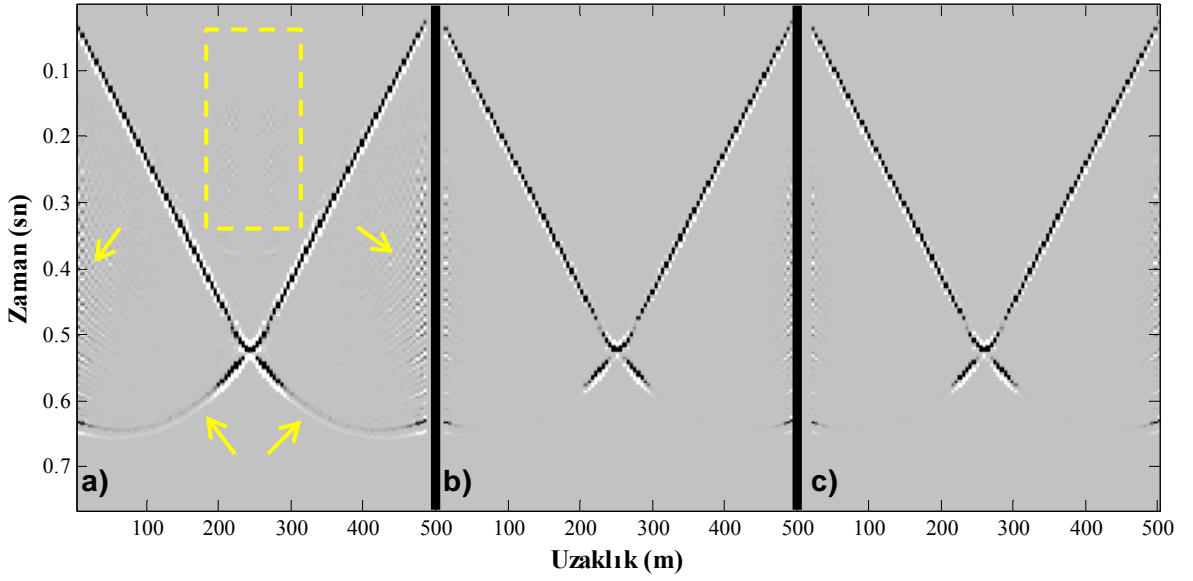


Şekil 3. 38. Papyon-kravat modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).

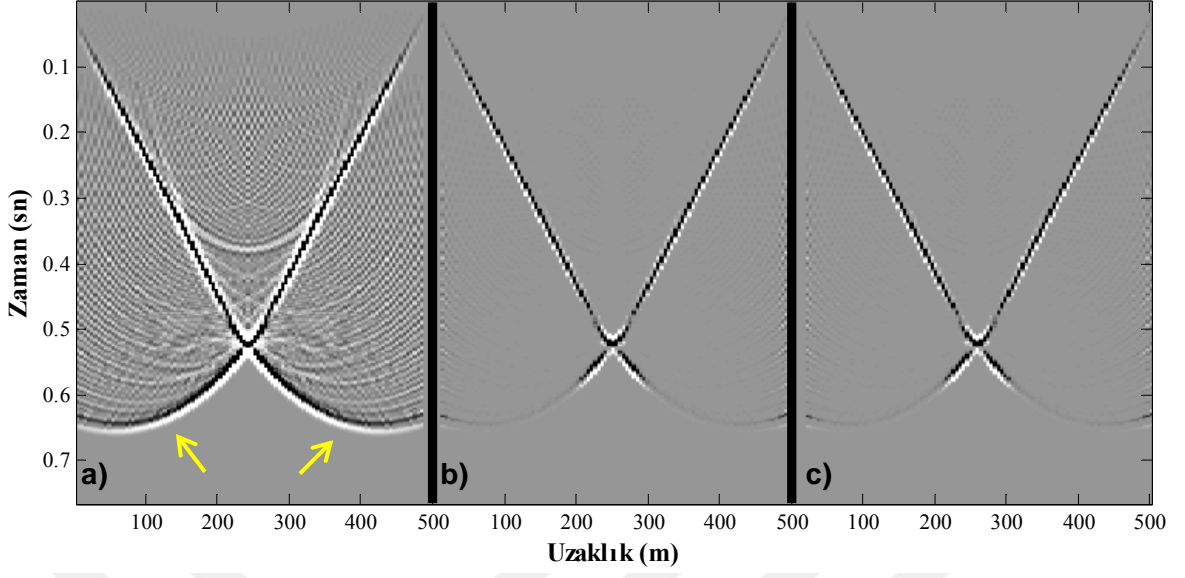
Tablo 3.3 papyon-kravat modeli için tüm göç çıkışlarının çözüm zamanlarını, Tablo 3.4 ise papyon-kravat modeli için tüm göç yöntemlerinin hata oranlarını göstermektedir. Papyon-kravat modeli için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışı Şekil 3.39’ da gösterilmektedir. EKKG yöntemleri Kirchhoff göçüne göre daha kaliteli, hata oranı daha düşük (Tablo 3.4) çıkışlar ürettiği açıkça görülebilmektedir. Senklinalin iç kısmı ve oklar ile gösterilen göç hataları oldukça belirgindir. Şekil 3.40 ile Şekil 3.47 arasında sırasıyla papyon-kravat modeli için $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. Şekil 3.40 üzerinden oklar ile gösterilen ve kesikli çizgiler ile belirtilen alan takip edilerek, geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile gerçekleştirilen EKKGD çıkışları (Şekil 3.40b,c) GD çıkışına (Şekil 3.40a) göre daha düşük hata oranına sahiptirler (Tablo 3.2). $(L^T L)$ netleştirme fonksiyonu (Şekil 3.41) Kirchhoff göçüne göre hata oranını düşürmesine rağmen çok yüksek genlikli göç hataları ürettiği görülmektedir. $cov(L^T L)$ netleştirme fonksiyonu yüksek hata oranında (Tablo 3.4) yüksek maliyetle (Tablo 3.3) göç çıkışları (Şekil 3.42) üretmiştir. $cov((L^T L)^{-1})$ netleştirme fonksiyonu çok düşük hata oranında (Tablo 3.4) yüksek maliyetle (Tablo 3.3) göç çıkışları (Şekil 3.43) üretmiştir ancak oklar ile gösterilen kenar kısımlardaki hataları sönümleyememiştir. Şekil 3.44 üzerinde kesikli çizgiler ve oklar ile gösterilen alanlara bakıldığında $(dd^T)^{-1}$ netleştirme fonksiyonu geleneksel netleştirme fonksiyonundan GD işlemi için yaklaşık %85, EKKGD için yaklaşık %30 daha az zamanda (Tablo 3.3) aynı hata oranlarına (Tablo 3.4) sahip çıkışlar üretmiştir. Şekil 3.45 kesikli çizgiler ve oklar ile işaretlenmiş alanlar incelendiğinde (dd^T) netleştirme fonksiyonunun EKKGD çıkışları için (Şekil 3.45b, c) çok düşük hata oranlarına (Tablo 3.4) sahip olmasına rağmen özellikle GD çıkışında (Şekil 3.45a) oldukça kusurlu sonuçlar ürettiği açıkça görülmektedir. Şekil 3.46 üzerinde kesikli çizgilerle ve oklar ile gösterilen alanlara bakıldığında açıkça görülebilir ki $cov(dd^T)$ netleştirme fonksiyonu hata oranını en küçüklemesine (Tablo 3.4) rağmen (dd^T) netleştirme fonksiyonu gibi kusurlu sonuçlar üretmiştir. Şekil 3.47 üzerinde oklar ile gösterilen alanlara bakıldığında $cov((dd^T)^{-1})$ netleştirme fonksiyonu $cov((L^T L)^{-1})$ fonksiyonundan GD işlemi için yaklaşık %86, EKKGD için yaklaşık %30 daha az zamanda (Tablo 3.3) aynı hata oranlarına (Tablo 3.4) sahip çıkışlar üretmiştir. Bunun yanında $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ ile elde edilen göç çıkışları göç hatalarını daha sönümlediği iyi Tablo 3.4 üzerinden açıkça görülmektedir.



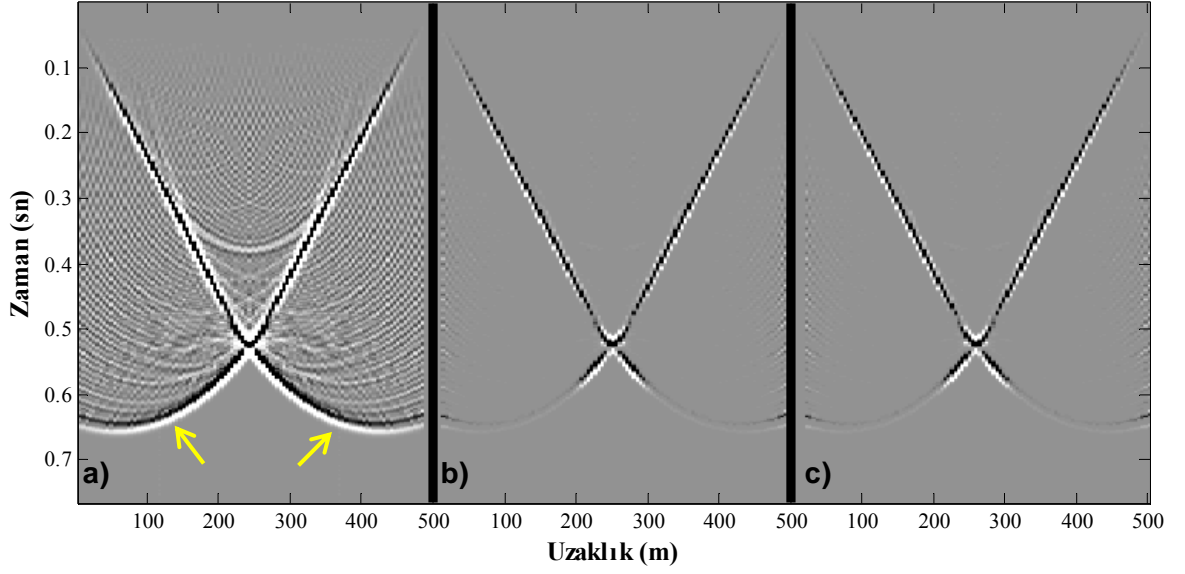
Şekil 3. 39. Papyon-kravat modeli için a) Kirchoff göçü, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıktıları.



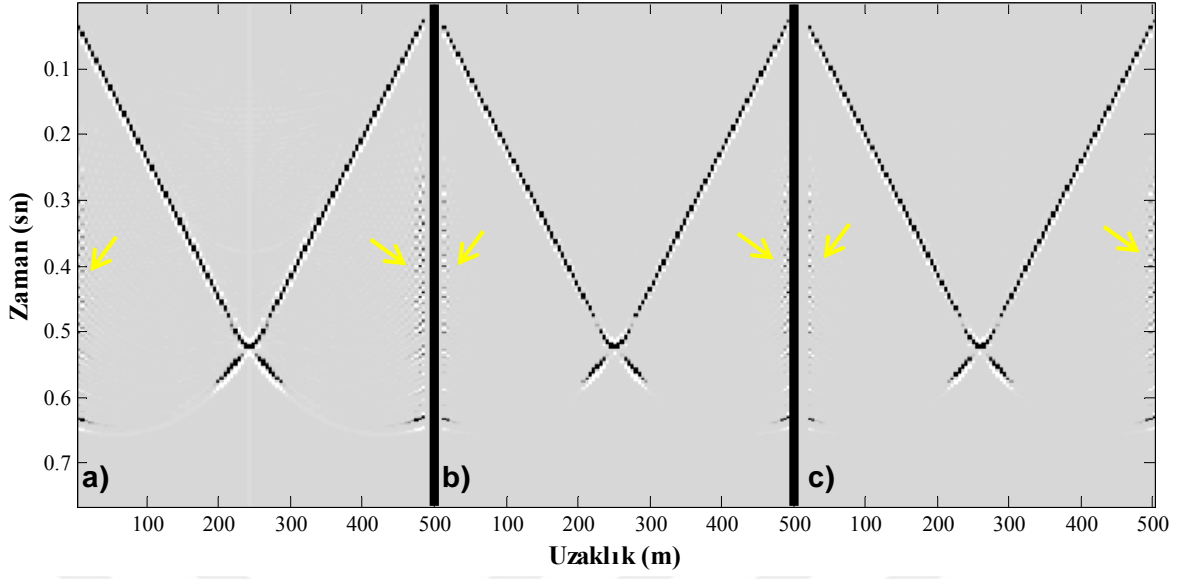
Şekil 3. 40. Papyon-kravat modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



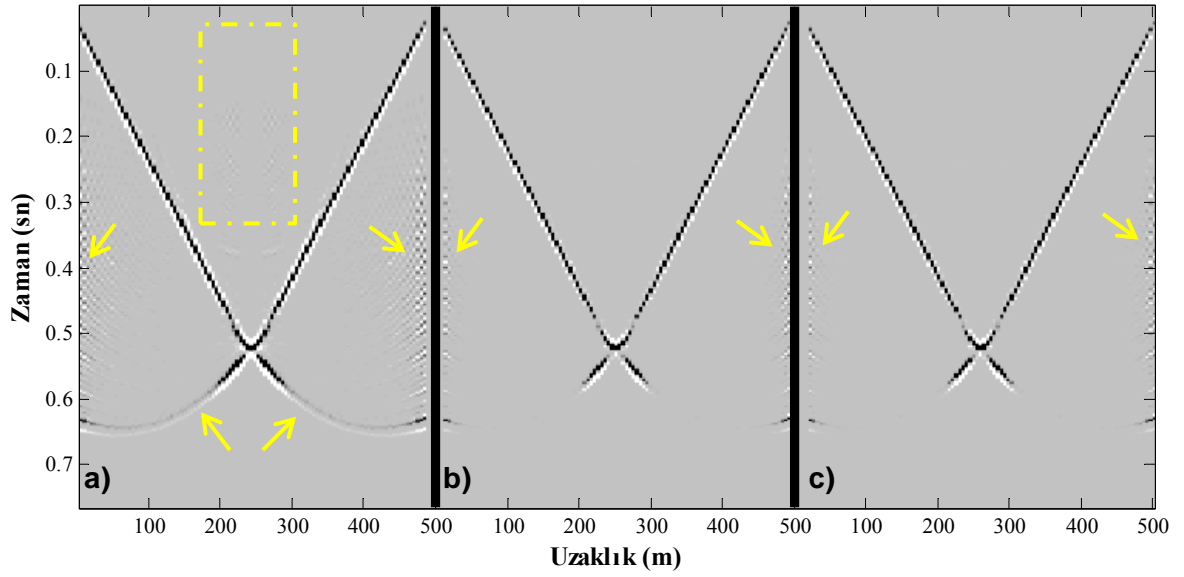
Şekil 3. 41. Papyon-kıavat modeli için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



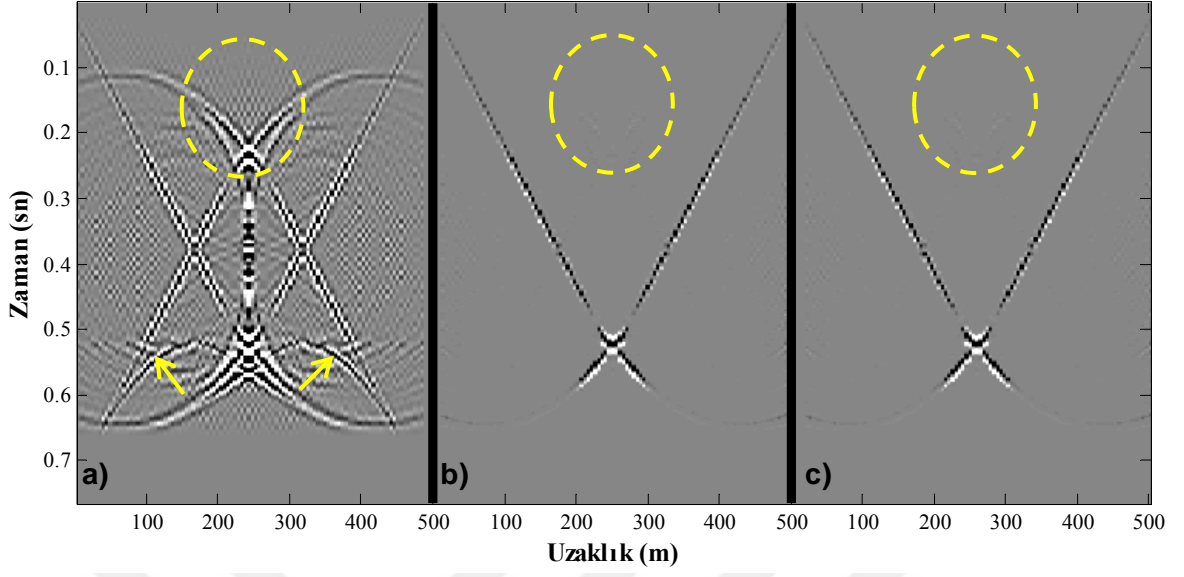
Şekil 3. 42. Papyon-kıavat modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



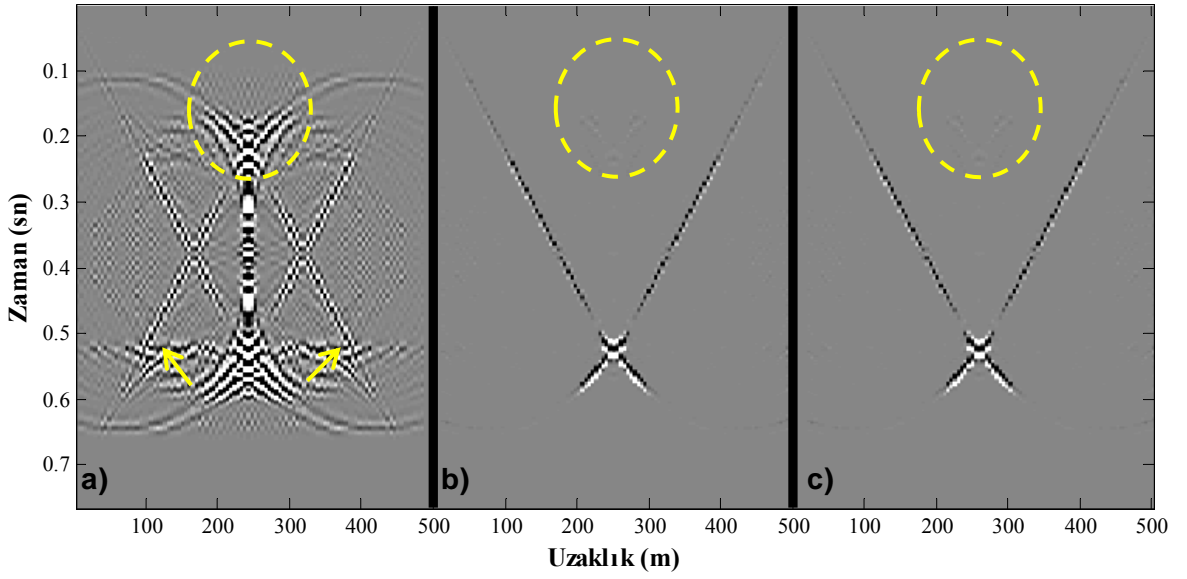
Şekil 3. 43. Papyon-kravat modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



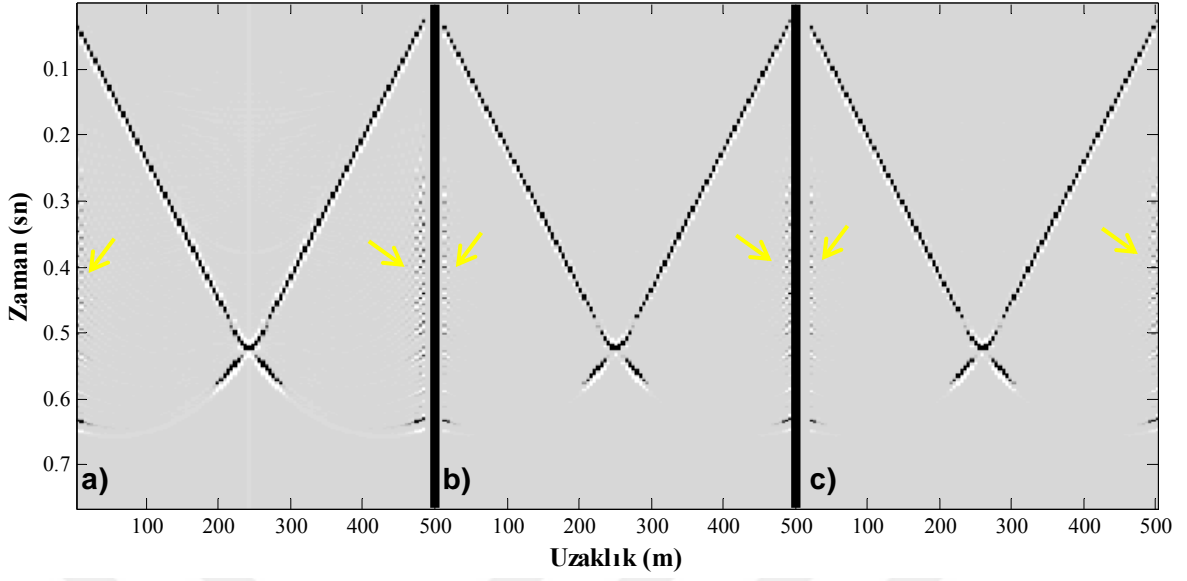
Şekil 3. 44. Papyon-kravat modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 45. Papyon-kravat modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 46. Papyon-kravat modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 47. Papyon-krawat modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

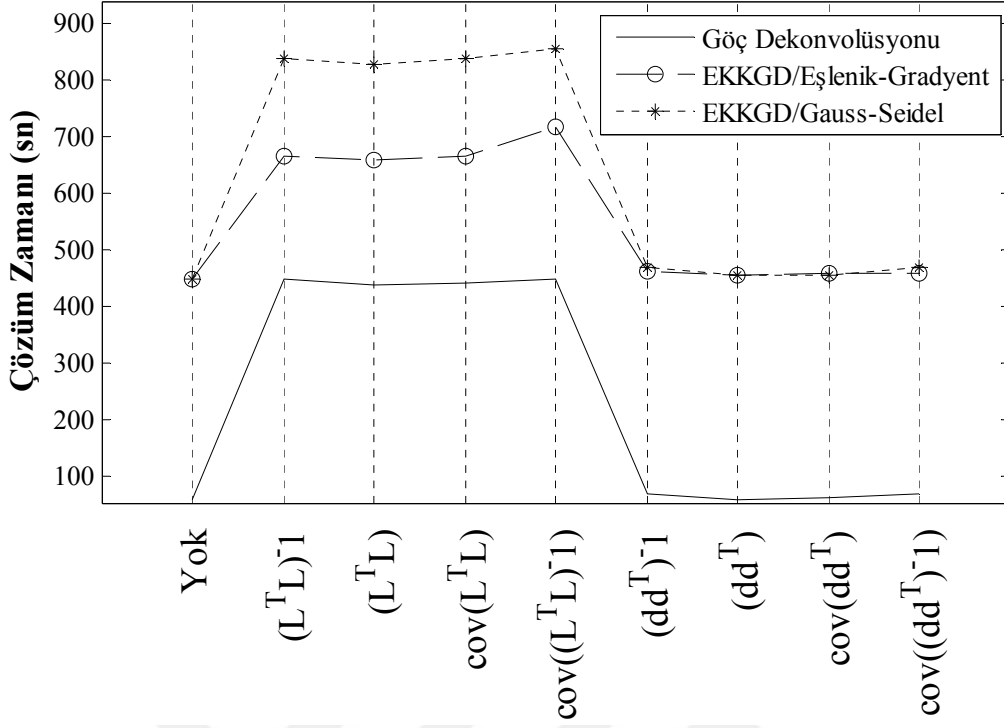
Tablo 3. 3. Papyon-krawat modeli için netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 100, zaman örnek sayısı: 766, yarı operatör boyu: 250 m, 50iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD(sn)	EKKGD Eşlenik- Gradyent(sn)	EKKGD Gauss- seidel(sn)	Netleştirme Fonk. Değerlendirme
Yok	58,95	446,88	448,58	En Düşük Maliyet
Kirchhoff göçü/EKKG $(L^T L)^{-1}$	448,23	666,11	838,39	Yüksek Maliyet
$(L^T L)$	437,75	658,25	827,15	Yüksek Maliyet
$cov(L^T L)$	442,05	666,15	840,40	Yüksek Maliyet
$cov((L^T L)^{-1})$	449,52	719,22	585,64	Yüksek Maliyet
$(dd^T)^{-1}$	69,16	462,34	468,43	Düşük Maliyet
(dd^T)	58,58	454,92	475,04	Düşük Maliyet
$cov(dd^T)$	62,07	458,40	457,02	Düşük Maliyet
$cov((dd^T)^{-1})$	68,84	458,29	468,02	Düşük Maliyet

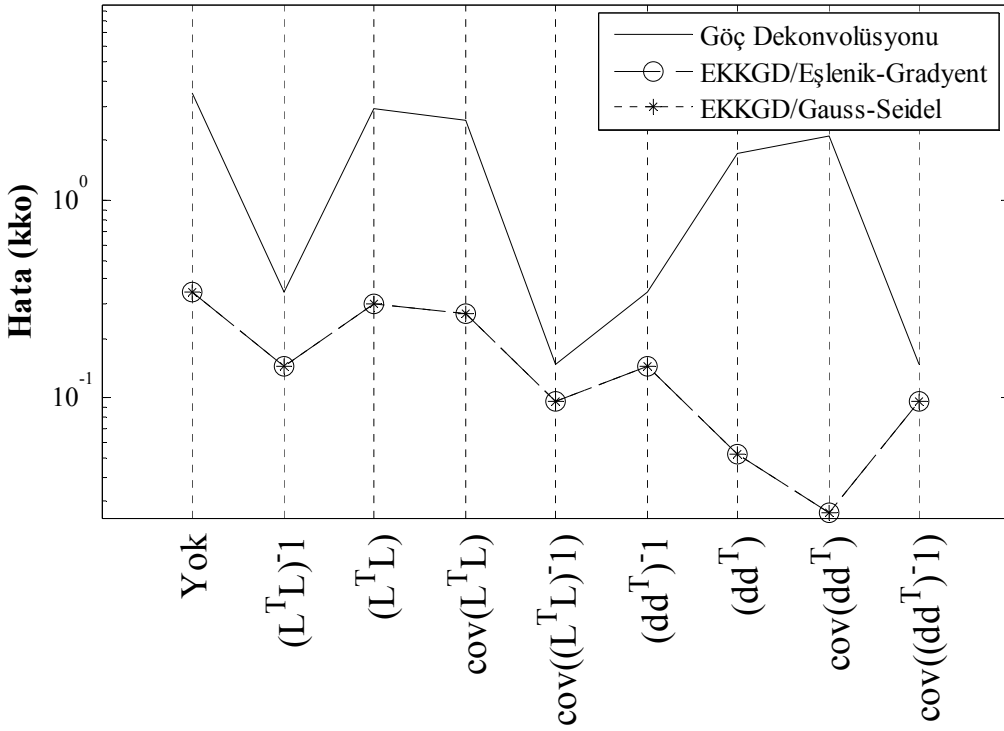
Tablo 3. 4. Papyon-kravat modeli için netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.

Hata Oranı (<i>kko</i>)/ Netleştirme Fonksiyonu	GD	EKKGD Eşlenik- Gradyent	EKKGD Gauss- seidel	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	3,50	0,35	0,35	Orta
$(L^T L)^{-1}$	0,35	0,15	0,15	Düşük
$(L^T L)$	2,90	0,30	0,30	Yüksek
$cov(L^T L)$	2,53	0,27	0,27	Yüksek
$cov((L^T L)^{-1})$	0,15	0,10	0,10	Çok Düşük
$(dd^T)^{-1}$	0,35	0,15	0,15	Düşük
(dd^T)	1,73	0,05	0,05	Çok Düşük (kusurlu)
$cov(dd^T)$	2,10	0,03	0,03	Çok Düşük (kusurlu)
$cov((dd^T)^{-1})$	0,15	0,10	0,10	Çok Düşük

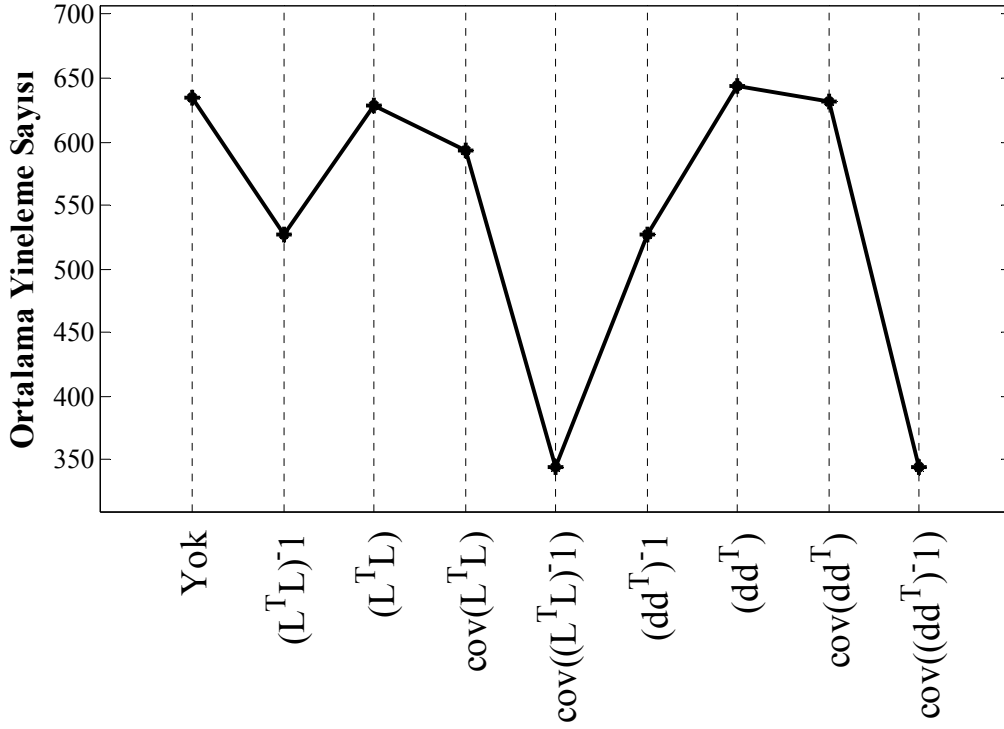
Şekil 3.48 papyon-kravat model için tüm yöntemlerin çözüm zamanlarını grafik üzerinde göstermektedir. Giriş verisi (*d*) kullanılarak elde edilen göç çıkışları, GD yaklaşık %86 daha az zamanda, EKKGD için yaklaşık %33 daha az zamanda çözüme ulaşmışlardır. Şekil 3.49 göç çıkışlarının toplam karekök hatalarını göstermektedir. Buna göre, $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları kullanılarak gerçekleştirilen göç işlemleri istenen çıkışa en fazla yaklaşan yöntemlerdir. Şekil 3.50 eşlenik-gradyent yöntemi için ortalama yineleme sayılarını göstermektedir. Buna göre $(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları yineleme sayılarını azaltabilen netleştirme fonksiyonları olduğu görülmektedir. Fakat yineleme sayısını en aza düşüren $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ fonksiyonlarıdır.



Şekil 3. 48. Papyon-kravat modeli için uygulanan göç yöntemlerinin çözüm zamanları.



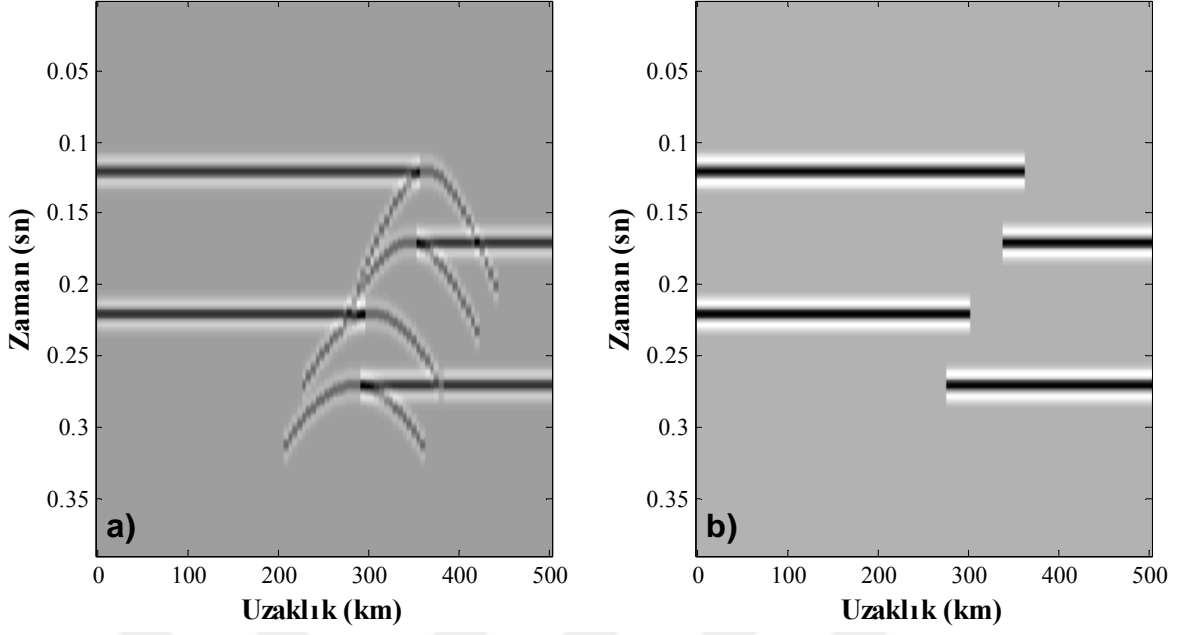
Şekil 3. 49. Papyon-kravat modeli için uygulanan göç yöntemlerinin kök ortalama kare hata miktarları.



Şekil 3. 50. Papyon-krawat modeli için uygulanan eşlenik-gradyent (EKKGD) yöntemi için ortalama yineleme sayıları.

3.1.3. Fay Modeli Üzerinde Göç Uygulamaları

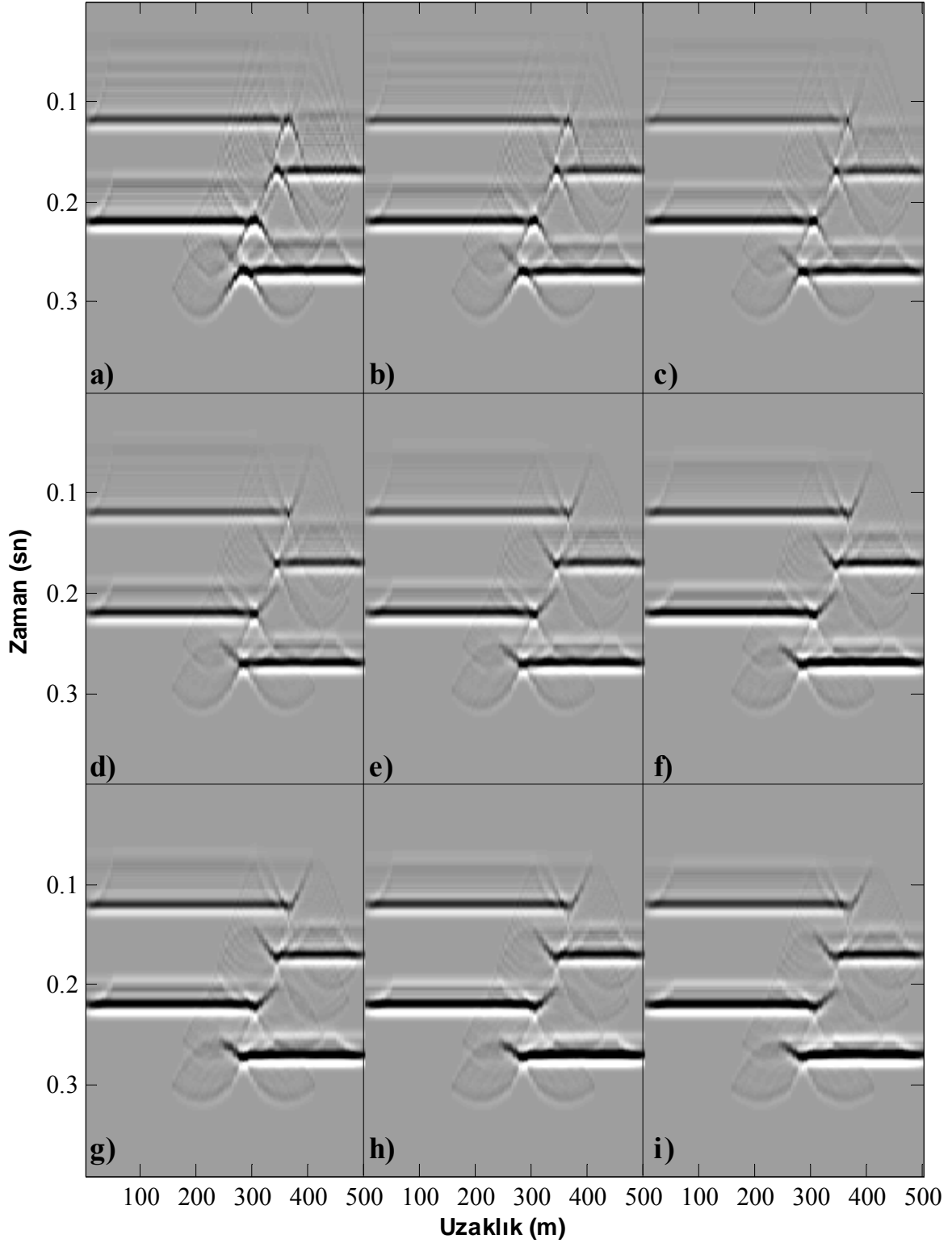
Faylar jeolojik olarak yanal yöndeki süreksizlik yapılarıdır ve göç algoritmalarının testleri için sıklıkla kullanılmaktadır. Hidrokarban tuzaklarının oluşmasında ve tektonik havzalarda en sık rastlanan jeolojik yapılarıdır. Genellikle oluşum mekanizmaları açısından normal, ters ve doğrultu atımlı olmak üzere üç ana grupta olmak üzere, bunların bileşimleri şeklinde de oluşabilirler. Bununla birlikte fay yüzeyleri en bilinen saçıcıdırlar (Şekil 3.51a). Şekil 3.51a basit bir normal fay modeli için hız sabit 1000 m/sn, örnekleme aralığı (dt) 0.001 sn, kayıt süresi (T) 0.390 sn, iz aralığı (dx) 5 m ve toplam kesit uzunluğu 500 m olacak şekilde elde edilmiştir. Beklenen göç çıkışı Şekil 3.51b de gösterilmektedir.



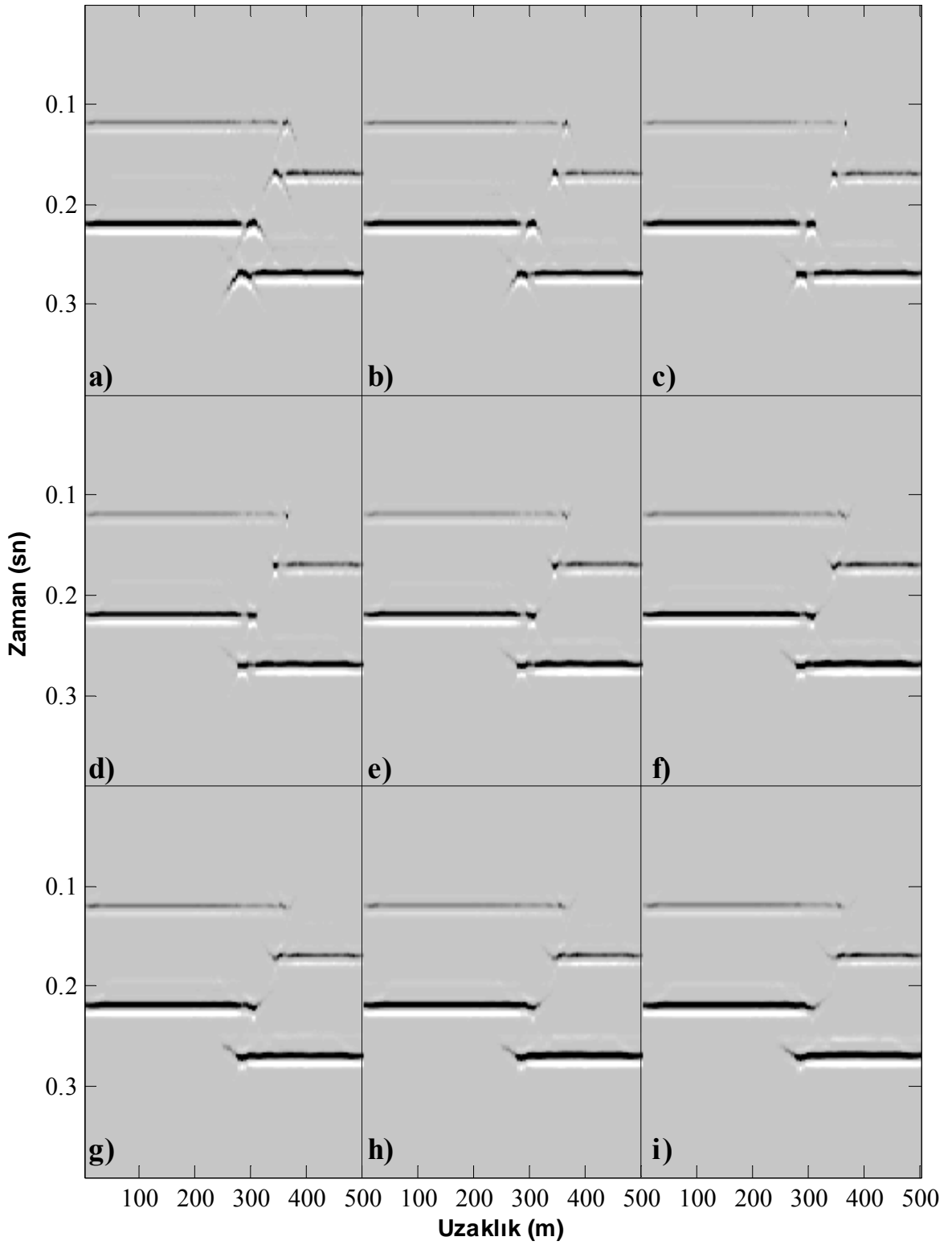
Şekil 3. 51. a) Fay modeli verisi ve b) beklenen göç çıkışı.

Fay modeli için yarı operatör boyu 100 m kullanılarak 27 farklı göç çıkışı ve beklenen göç çıkışı (Şekil 3.27b) ile olan farklarının karelerinin toplamları (h_{rms}) hesaplanmıştır. Tablo 3.5 fay modeli için tüm göç çıkışlarının çözüm zamanlarını, Tablo 3.6 ise fay modeli için tüm göç çıkışlarının hata oranlarını göstermektedir. Fay modeli için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışı Şekil 3.28’ de gösterilmektedir. EKKG yöntemleri (Şekil 3.28b, c) Kirchhoff göçüne (Şekil 3.28a) göre daha kaliteli, hata oranı daha düşük (Tablo 3.6) çıkışlar ürettiği görülmektedir. Şekil 3.28 üzerinde oklar ile belirtilen göç hatalarının EKKGD ile minimize edildiği açıktır. Şekil 3.29 ile Şekil 3.36 arasında sırasıyla fay modeli için $(L^T L)^{-1}$, $(L^T L)$, (dd^T) , $(dd^T)^{-1}$, $cov(dd^T)$, $cov((dd^T)^{-1})$, $cov(L^T L)$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. $(L^T L)^{-1}$ netleştirme fonksiyonu ile elde edilen EKKGD çıkışları (Şekil 3.29b,c) GD çıkışına (Şekil 3.29a) göre daha düşük hata oranına sahiptirler (Tablo 3.2). Buradan hareketle netleştirme fonksiyonunun hata oranını düşürdüğü EKKG üzerinde uygulandığında hata oranının daha çok düşürülebildiği (Tablo 3.6) görülmektedir. $(L^T L)$ netleştirme fonksiyonu (Şekil 3.30), bu model için, Kirchhoff göçüne göre hata oranını yükseltmiştir. (dd^T) netleştirme fonksiyonu çok düşük hata oranlarına sahip olmamasına (Tablo 3.6) rağmen fay modeli için beklenen çıkış verisine (Şekil 3.27b) çok yakın sonuçlar (Şekil 3.31) üretmiştir. $(dd^T)^{-1}$ netleştirme fonksiyonu (Şekil 3.32)

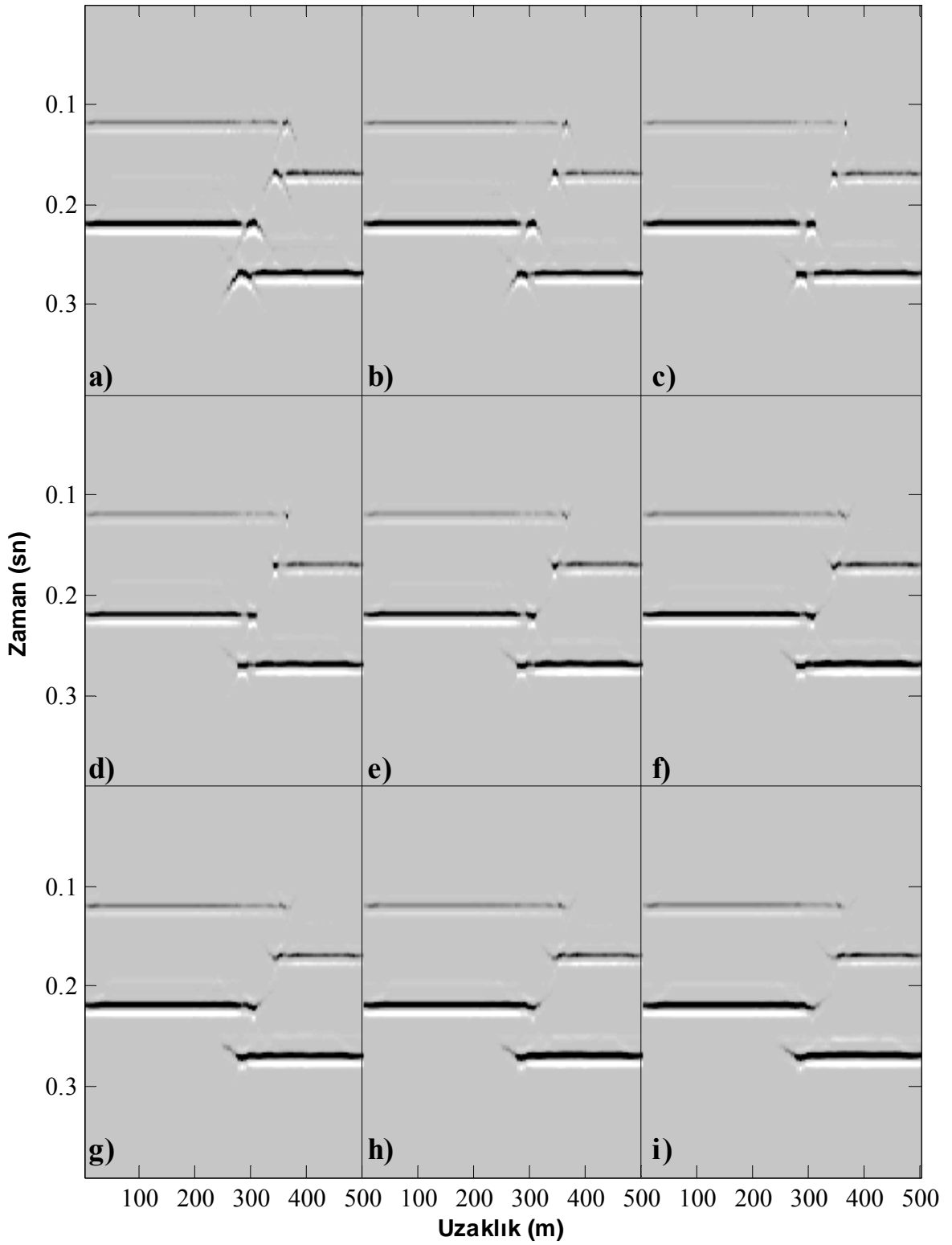
geleneksel netleştirme fonksiyonunun yaklaşık %61 maliyetiyle (Tablo 3.5) aynı hata oranlarına (Tablo 3.6) ve görüntü kalitesine sahip çıkışlar üretmiştir (Şekil 3.29, 3.32). $cov(dd^T)$ netleştirme fonksiyonu hata oranını yükseltmesine (Tablo 3.4) rağmen (dd^T) netleştirme fonksiyonu gibi beklenen göç çıkışına oldukça yakın sonuçlar üretmiştir (Şekil 3.33). $cov((dd^T)^{-1})$ netleştirme fonksiyonu (Şekil 3.34) en düşük hata oranına düşük maliyet ile ulaşmıştır. $cov(L^T L)$ netleştirme fonksiyonu EKKGD çıkışları (Şekil 3.35b, c) için düşük hata oranlarında (Tablo 3.6) kaliteli göç çıkışları üretse de GD çıkışı (Şekil 3.35a) yüksek hata oranına ve yüksek maliyete (Tablo 3.5) sahiptir. $cov((L^T L)^{-1})$ netleştirme fonksiyonu (Şekil 3.36) önceki modellerde olduğu gibi $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile aynı oranda en düşük hata oranlarını (Tablo 3.4) çok yüksek maliyetle (Tablo 3.5) üretmiştir. Nokta saçılma modeli ve papyon-kravat modelinde olduğu gibi $(dd^T)^{-1}$ ile $(L^T L)^{-1}$ netleştirme fonksiyonları ile elde edilen göç çıkışları (Şekil 3.29, 3.32) ve $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile elde edilen göç çıkışları (Şekil 3.34, 3.36) aynı hata oranlarına ve aynı görüntü kalitesine sahiptir. Bunun yanında $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonlarının göç hatalarını daha iyi sönümlendiği hem şekiller üzerinden (Şekil 3.29, 3.32, 3.34, 3.36) hem de Tablo 3.6 üzerinden açıkça görülmektedir. Ayrıca EKKGD çıkışları tüm netleştirme fonksiyonları için daha düşük hata oranlarına sahiptirler.



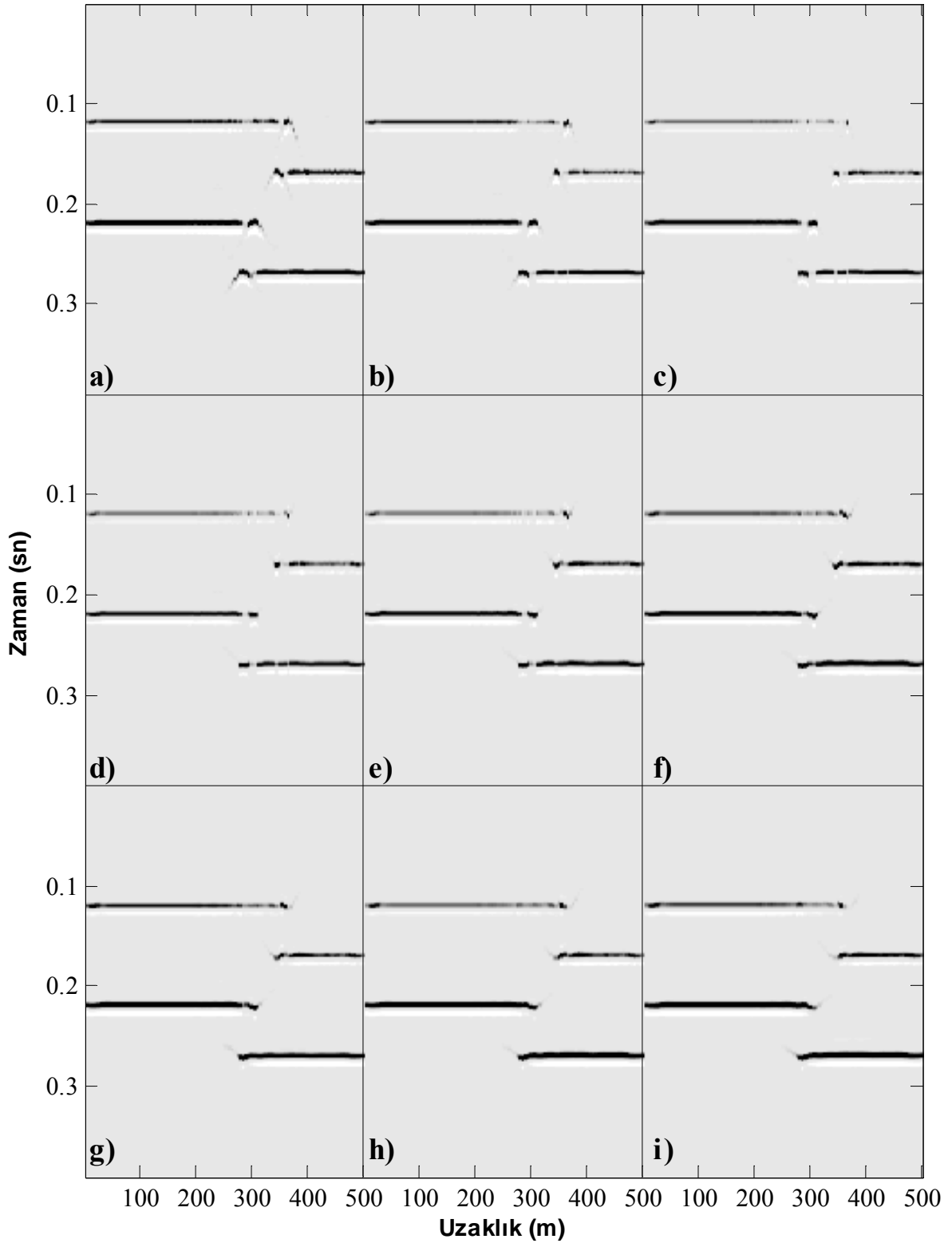
Şekil 3. 52. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan Kirchhoff göç çıkışları (yarı operatör boyu=50 m).



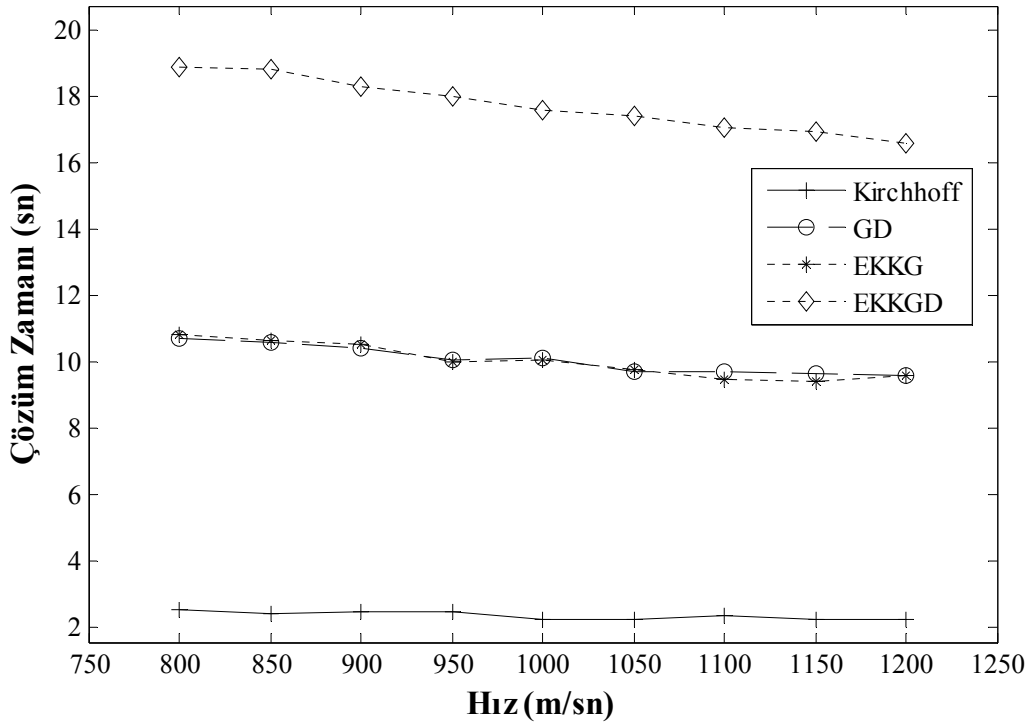
Şekil 3. 53. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (yarı operatör boyu=50 m).



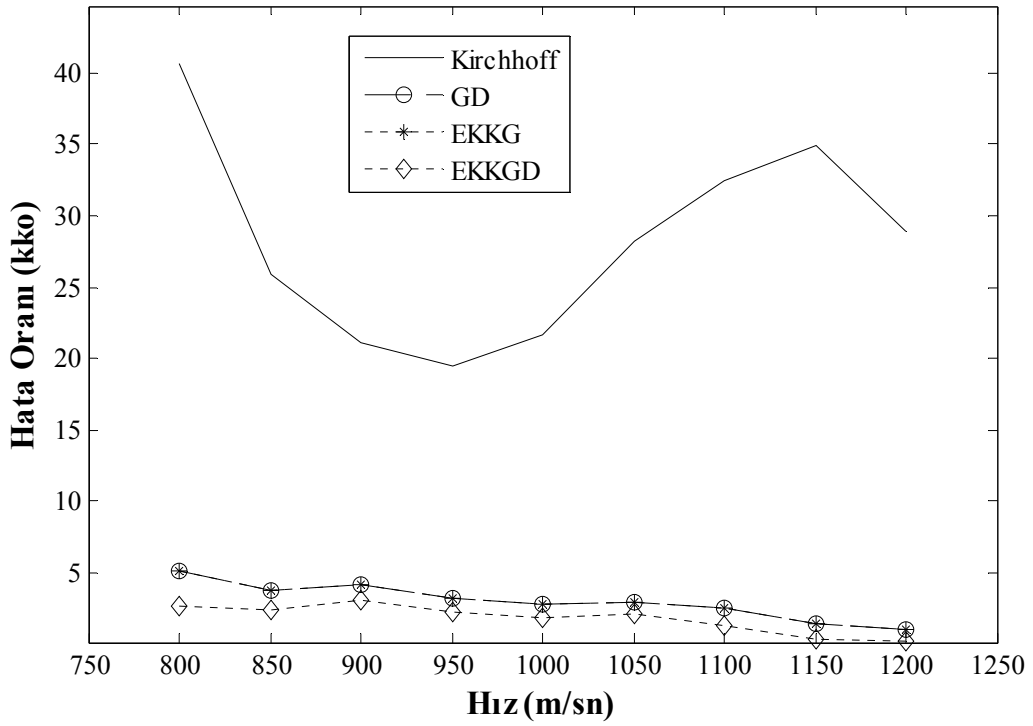
Şekil 3. 54. Fay modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ile uygulanan EKKG çıkışları (yarı operatör boyu=50 m).



Şekil 3. 55. Fay saçılma modeli için a) 800 m/sn, b) 850 m/sn, c) 900 m/sn, d) 950 m/sn, e) 1000 m/sn, f) 1050 m/sn, g) 1100 m/sn, h) 1150 m/sn ve i) 1200 m/sn sabit hızlar ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (yarı operatör boyu=50 m).

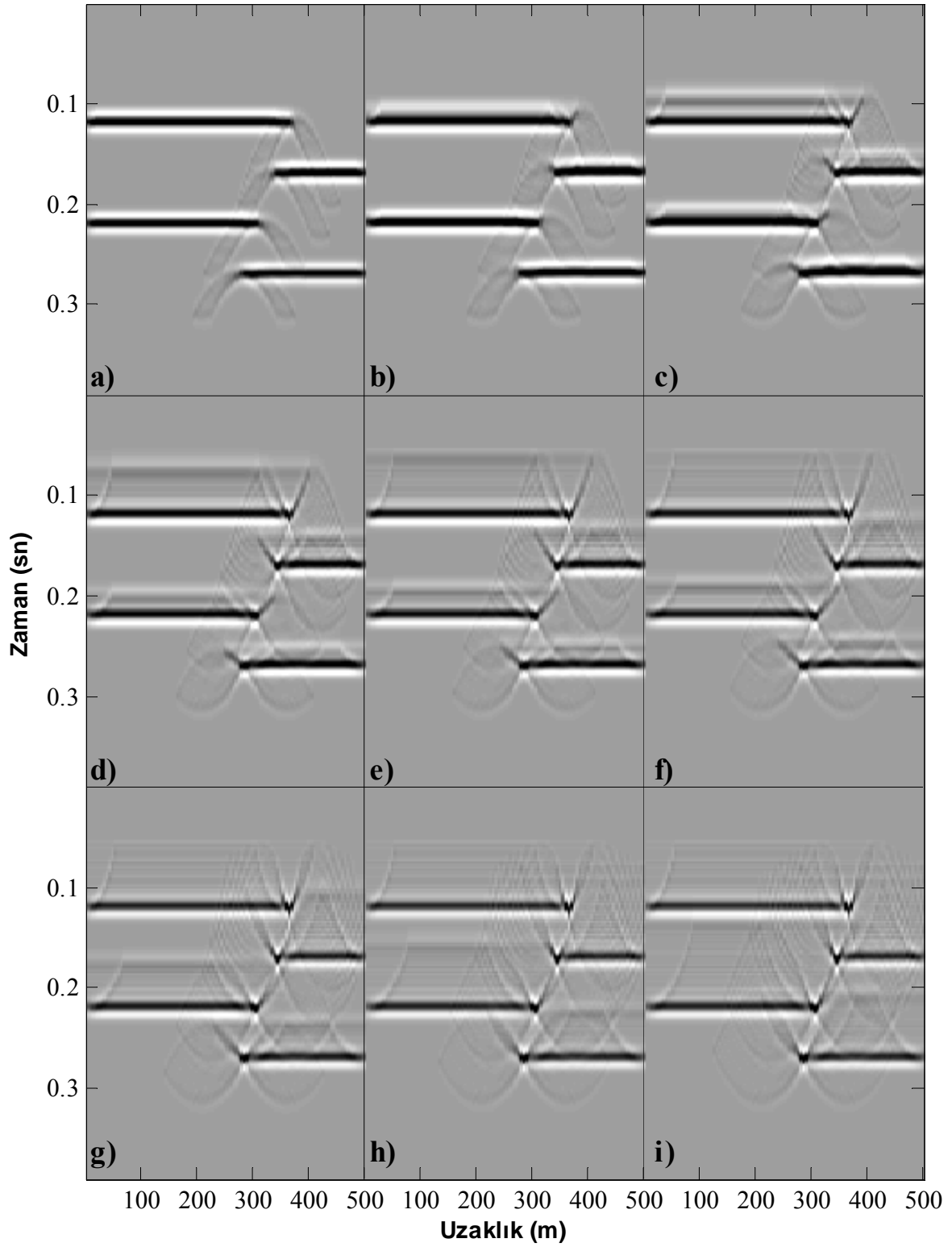


Şekil 3. 56. Fay modeli için hız testlerinde çözüm zamanları (yarı operatör boyu=50 m).

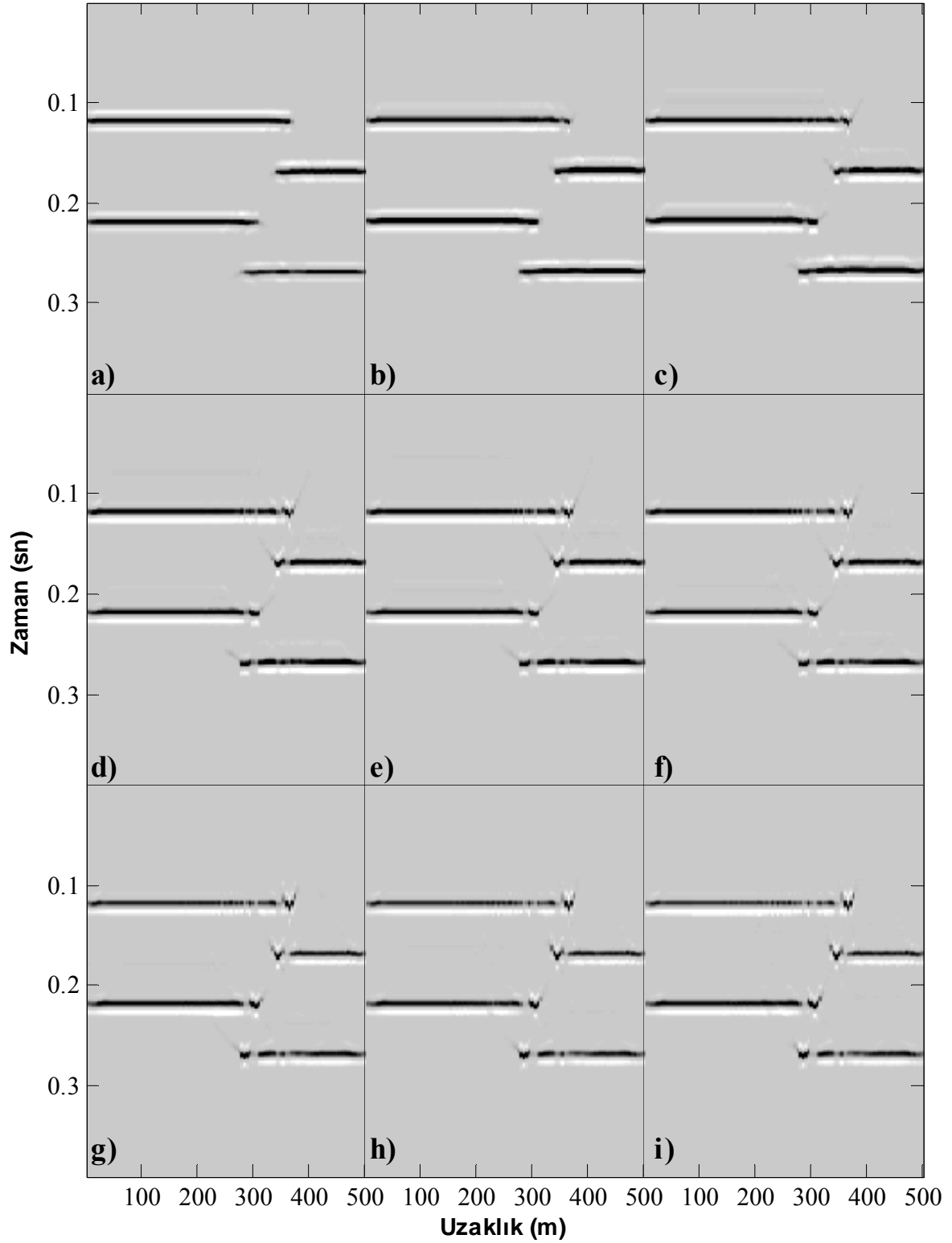


Şekil 3. 57. Fay modeli için hız testlerinde karekök ortalama hata oranları (yarı operatör boyu=50 m).

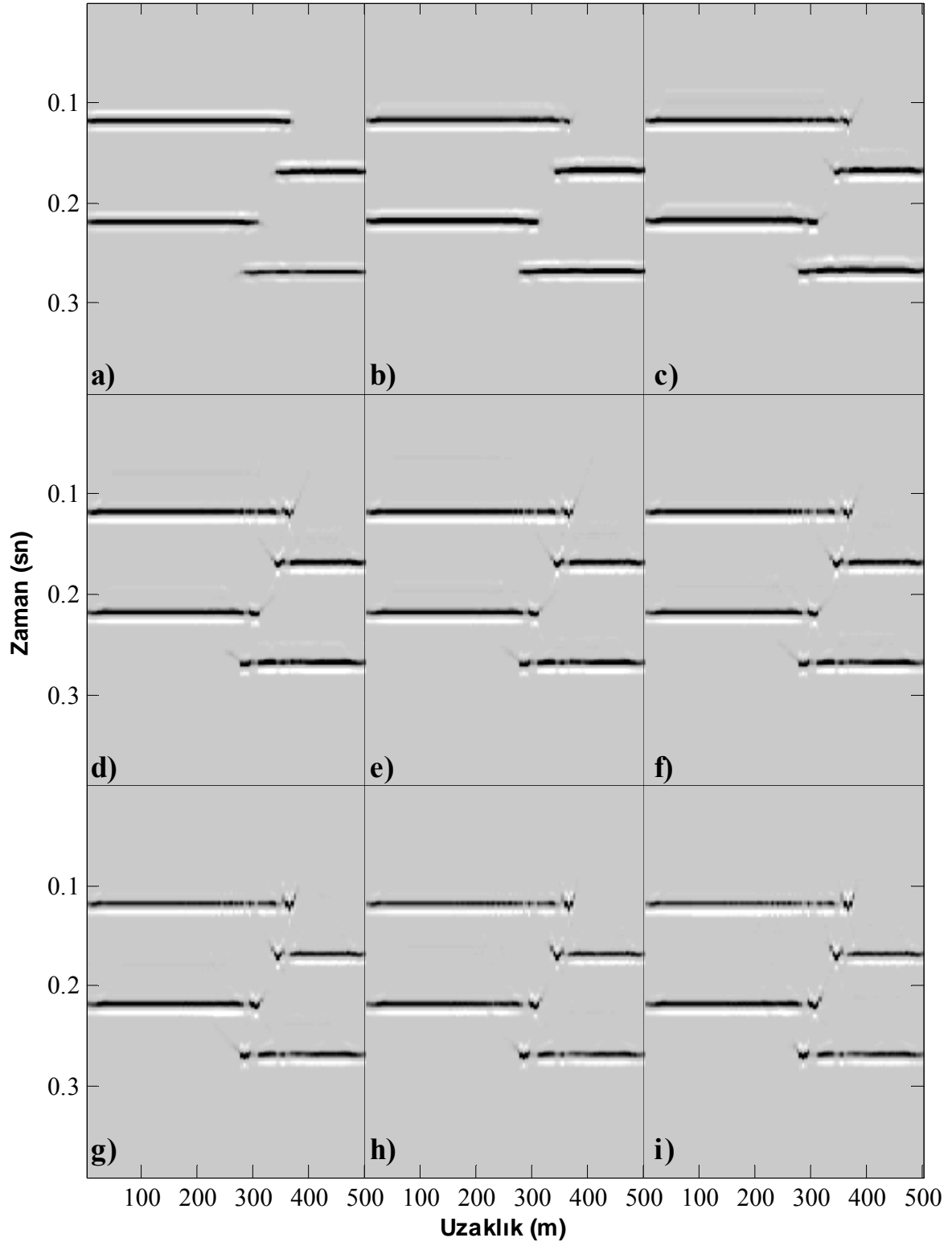
Fay modeli için farklı yarı operatör boyları (15 m, 25 m, 35 m, 45 m, 50 m, 55 m, 65 m, 75 m ve 85 m) ile Kirchhoff göçü, EKKG ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile GD ve EKKGD işlemleri uygulanmıştır. Şekil 3.58 farklı yarı operatör boyları ile gerçekleştirilen Kirchhoff göç çıkışlarını göstermektedir. Yarı operatör boyunun küçük (15 m, 25 m, 35 m, 45 m) seçilmesi (Şekil 3.58a-d) süreksizlikteki saçılmaları saçıcı noktaya toplayamamıştır. Yarı operatör boyunun büyük (55 m, 65 m, 75 m ve 85 m) seçilmesi (Şekil 3.58g-i) saçıcı noktaya toplamayı başarmış fakat hata oranını arttırmıştır. Bunun sebebi hız testlerinde olduğu gibi yatay olaylardır. Şekil 3.59 yine aynı operatör yarı boyları ile uygulanan GD çıkışlarını göstermektedir. GD işleminin hata oranını azalttığı görülmektedir. Bunun yanında EKKG işleminin de hata oranını GD işlemi kadar azalttığı ve görüntü kalitesini arttırdığı görülmektedir (Şekil 3.60). EKKGD işlemi yarı operatör boyu testleri için en küçük hata oranını sağlayarak çözünürlüğü en iyi göç çıkışlarını üretmiştir (Şekil 3.61). EKKGD en iyi çözünürlüğü sağlamasına rağmen en maliyetli yöntem olmuştur (Şekil 3.62). EKKG ile GD işlemlerinin maliyetleri yaklaşık olarak eşittir. Ayrıca yarı operatör boyu arttıkça tüm yöntemler için çözüm zamanı artmaktadır. Hata oranlarına bakıldığında (Şekil 3.63) en düşük hata oranını EKKGD işlemi vermiştir. GD ve EKKG işlemleri aynı hata oranlarını üretmiştir. Kirchhoff göçü çözüm zamanı açısından en iyi yöntem olmasına rağmen en fazla göç hatasını üreten yöntem olmuştur. Şekil 3.63 hata oranlarını göstermektedir. Özellikle GD, EKKG ve EKKGD eğrilerine bakıldığında 15 m den 50 m yarı operatör boyuna kadar hata oranlarında bir artış gözlenirken, 50 m yarı operatör boyundan sonra sabit kalmıştır. Çözüm zamanlarına ve hata oranlarına bakıldığında yarı operatör boyunun olması gereken değerden (50 m) küçük olması hata oranını düşürürken, büyük olması hata oranını çok fazla yükseltmese de çözüm zamanını oldukça arttırmaktadır. Operatör boyunun küçük olması ile süreksizlik sınırlarındaki saçılmalar tam olarak bir noktaya toplanamasa da, yatay olaylardan daha az gürültü bileşeni oluşturarak düşük hata oranlarında çıkışlar üretmişlerdir. Kısacası hız testlerinde olduğu gibi, fay modeli için hata oranlarından yorum yapmak yanıltıcı olabilir.



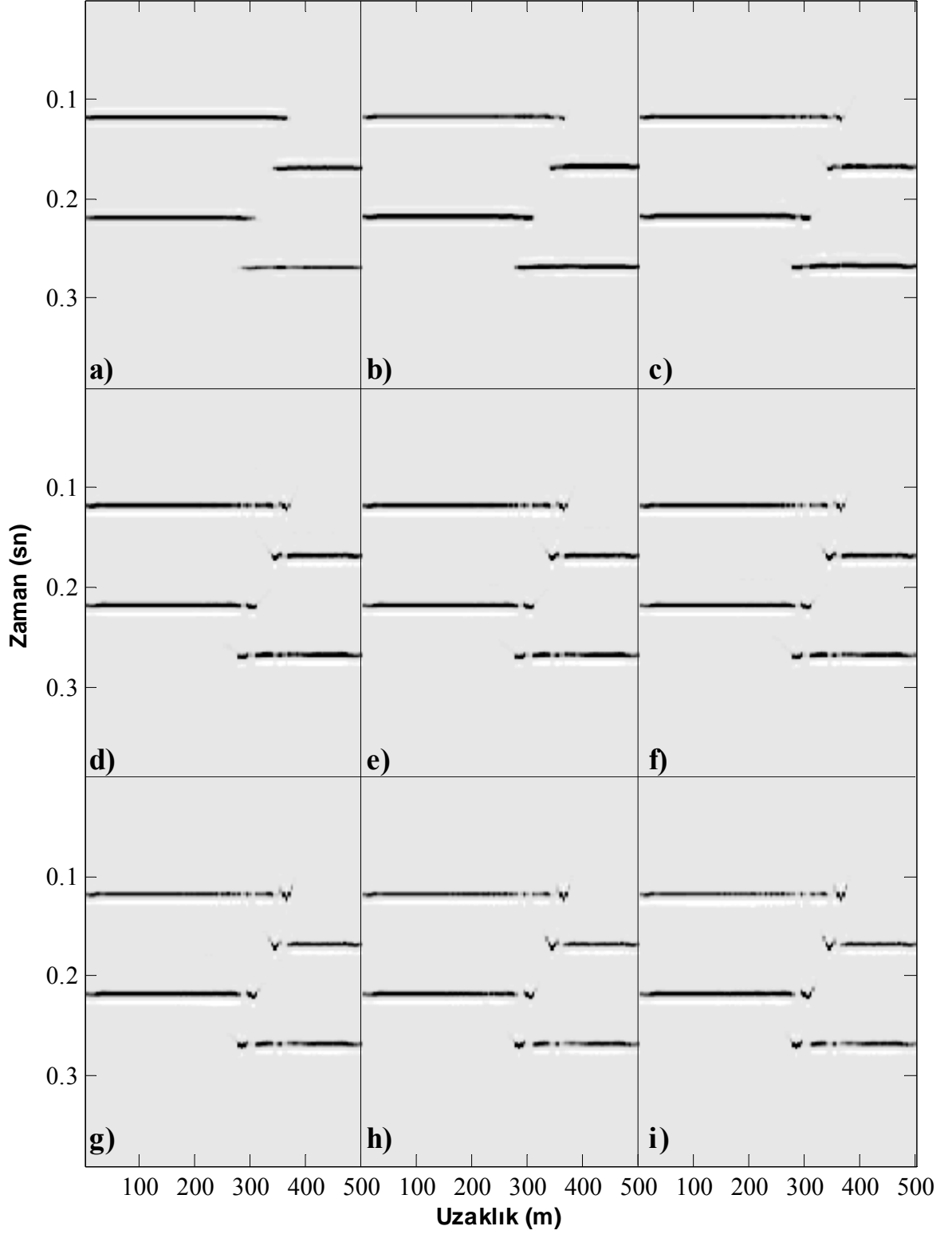
Şekil 3. 58. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ile uygulanan Kirchhoff göç çıkışları (hız=1000 m/sn).



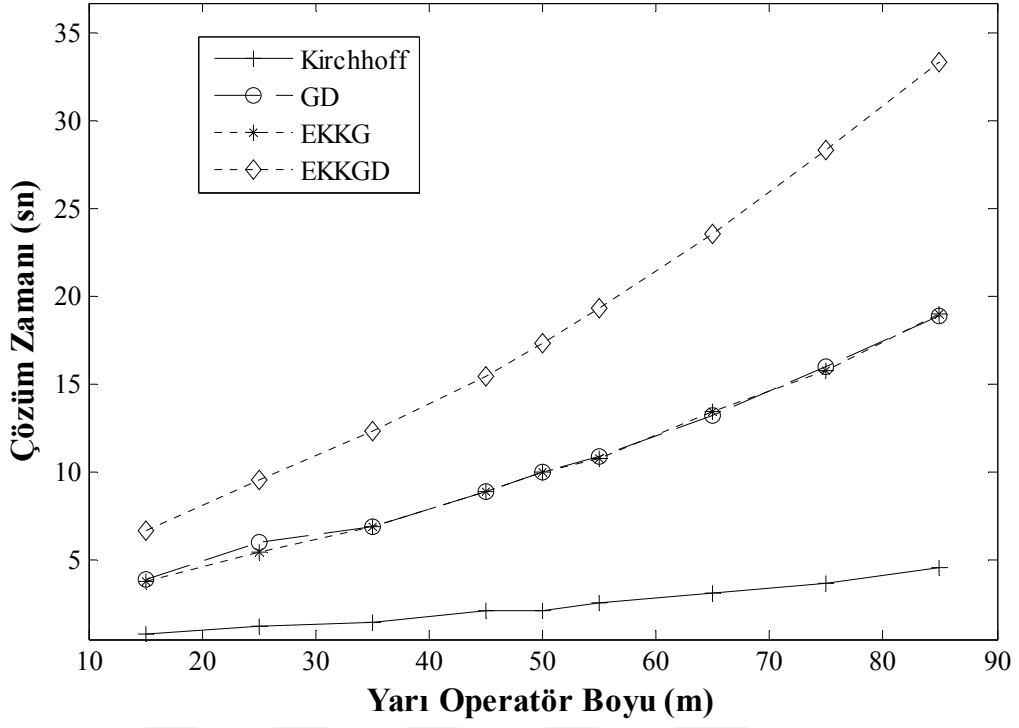
Şekil 3. 59. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan GD çıkışları (hız=1000 m/sn).



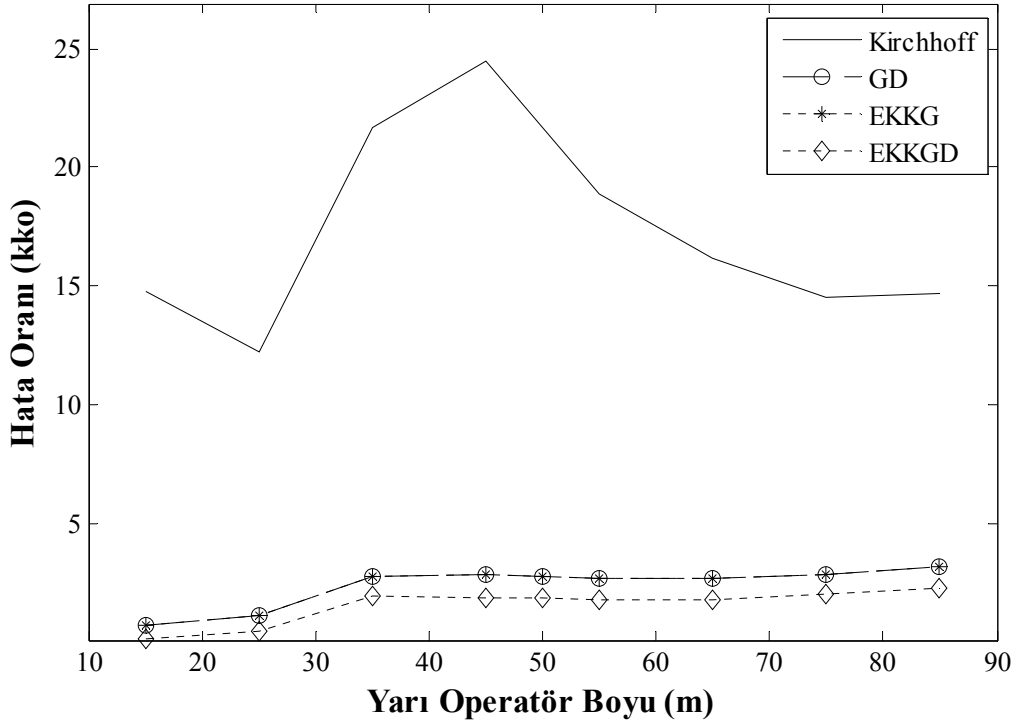
Şekil 3. 60. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ile uygulanan EKKG çıkışları (hız=1000 m/sn).



Şekil 3. 61. Fay modeli için a) 15 m, b) 25 m, c) 35 m, d) 45 m, e) 50 m, f) 55 m, g) 65 m, h) 70 m ve i) 75 m operatör yarı boyları ve geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile uygulanan EKKGD çıkışları (hız=1000 m/sn).



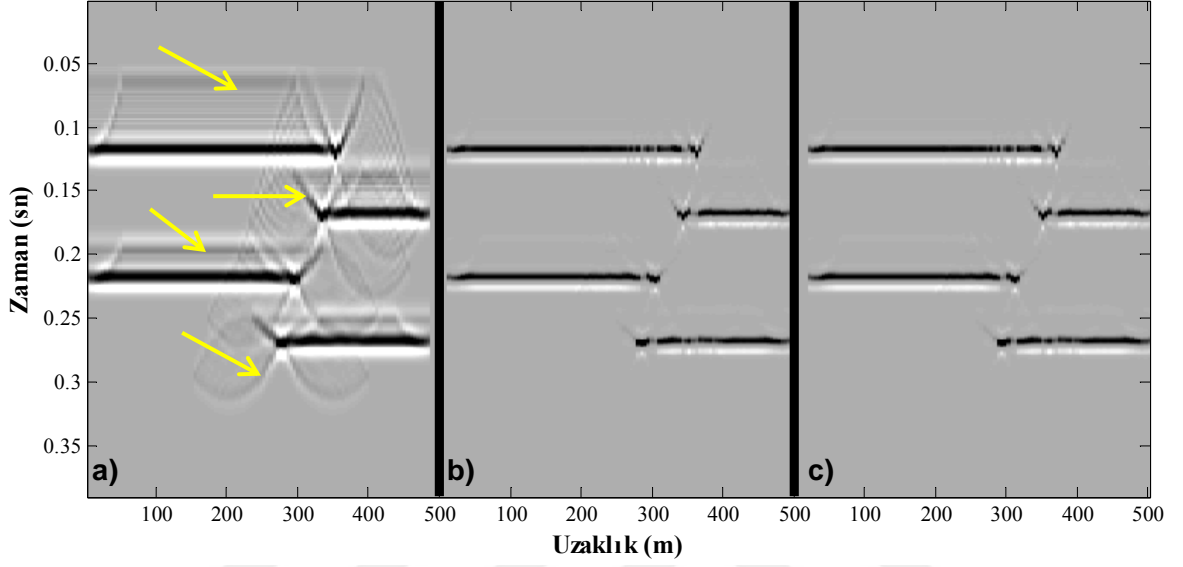
Şekil 3. 62. Fay modeli için yarı operatör boyu testlerinde çözüm zamanları (hız=1000 m/sn).



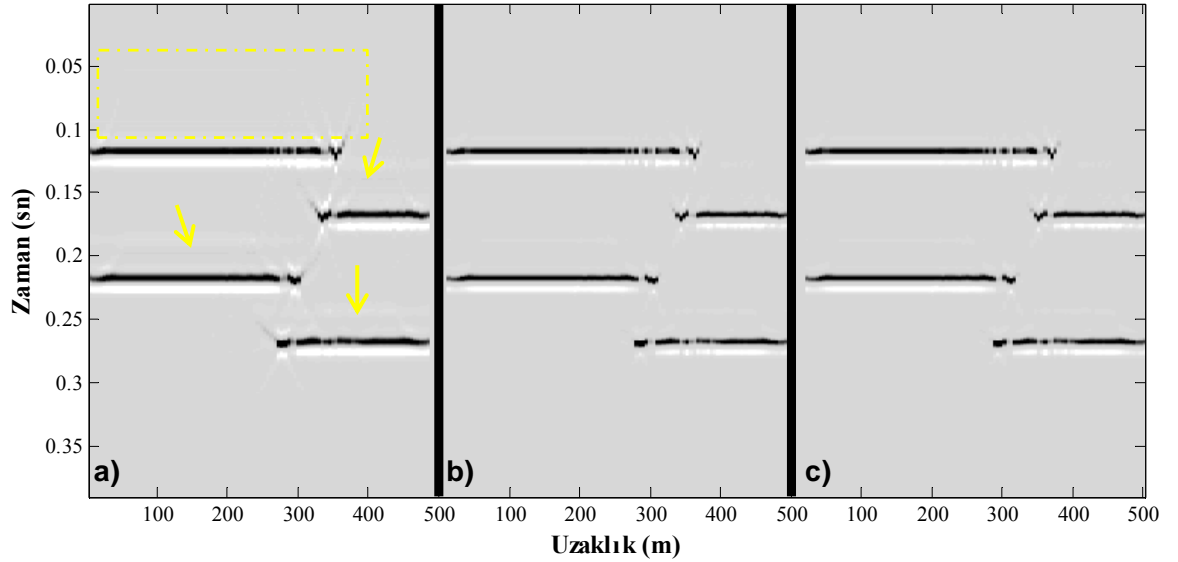
Şekil 3. 63. Fay modeli için yarı operatör boyu testlerinde karekök ortalama hata oranları (hız=1000 m/sn).

Fay modeli için yarı operatör boyu 100 m kullanılarak 27 farklı göç çıkışı ve beklenen göç çıkışı (Şekil 3.51b) ile olan farklarının karelerinin toplamları (kko) hesaplanmıştır. Tablo 3.5 fay modeli için tüm göç çıkışlarının çözüm zamanlarını, Tablo 3.6 ise fay modeli için tüm göç çıkışlarının hata oranlarını göstermektedir. Fay modeli için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışı Şekil 3.64' de gösterilmektedir. Şekil üzerinde oklar ile işaret edilen alanlara bakıldığında EKKG yöntemleri (Şekil 3.64b, c) Kirchhoff göçüne (Şekil 3.64a) göre daha kaliteli, hata oranı daha düşük (Tablo 3.6) çıkışlar ürettiği görülmektedir. Şekil 3.65 ile Şekil 3.72 arasında sırasıyla fay modeli için $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. Şekil 3.65' de kesikli çizgiler ve oklar ile işaret edilen alanlar incelendiğinde $(L^T L)^{-1}$ netleştirme fonksiyonu ile elde edilen EKKGD çıkışları (Şekil 3.65b,c) GD çıkışına (Şekil 3.65a) göre daha düşük hata oranına sahip oldukları görülebilir (Tablo 3.6). Buradan hareketle netleştirme fonksiyonunun hata oranını düşürdüğü EKKG üzerinde uygulandığında hata oranının daha çok düşürülebildiği (Tablo 3.6) görülmektedir. Şekil 3.66a' da kesikli çizgiler ve oklar ile işaret edilen alanlar göz önüne alındığında $(L^T L)$ netleştirme fonksiyonu ile elde edilen GD çıkışı Kirchhoff göçüne göre hata oranını yükseltmiştir (Tablo 3.6). Şekil 3.66a' da kesikli çizgiler ve oklar ile işaret edilen alanlara bakarak $cov(L^T L)$ netleştirme fonksiyonunun $(L^T L)$ netleştirme fonksiyonu gibi yüksek hatalara neden olduğu (Tablo 3.6) görülebilir. Şekil 3.68 hata oranını neredeyse sıfırlayan $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile elde edilen göç çıkışlarını göstermektedir. Şekil 3.69a' da kesikli çizgiler ve oklara bakıldığında $(dd^T)^{-1}$ netleştirme fonksiyonunun geleneksel netleştirme fonksiyonu ile aynı kalitedeki (Tablo 3.6) göç çıkışlarını yaklaşık %60 maliyetiyle (Tablo 3.5) üretmiştir. (dd^T) netleştirme fonksiyonu çok düşük hata oranlarına sahip (Tablo 3.6) fay modeli için beklenen çıkış verisine (Şekil 3.70b,c) çok yakın sonuçlar (Şekil 3.31) üretmiştir. $cov(dd^T)$ netleştirme fonksiyonu hata oranını yükseltmesine (Tablo 3.4) rağmen (dd^T) netleştirme fonksiyonu gibi beklenen göç çıkışına oldukça yakın sonuçlar üretmiştir (Şekil 3.71). $cov((dd^T)^{-1})$ netleştirme fonksiyonu (Şekil 3.72) en düşük hata oranına düşük maliyet ile ulaşmıştır. Nokta saçılma modeli ve papyon-kravat modelinde olduğu gibi $(dd^T)^{-1}$ ile $(L^T L)^{-1}$ netleştirme fonksiyonları ile elde edilen göç çıkışları ve $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile elde edilen göç çıkışları aynı hata oranlarına ve aynı görüntü kalitesine

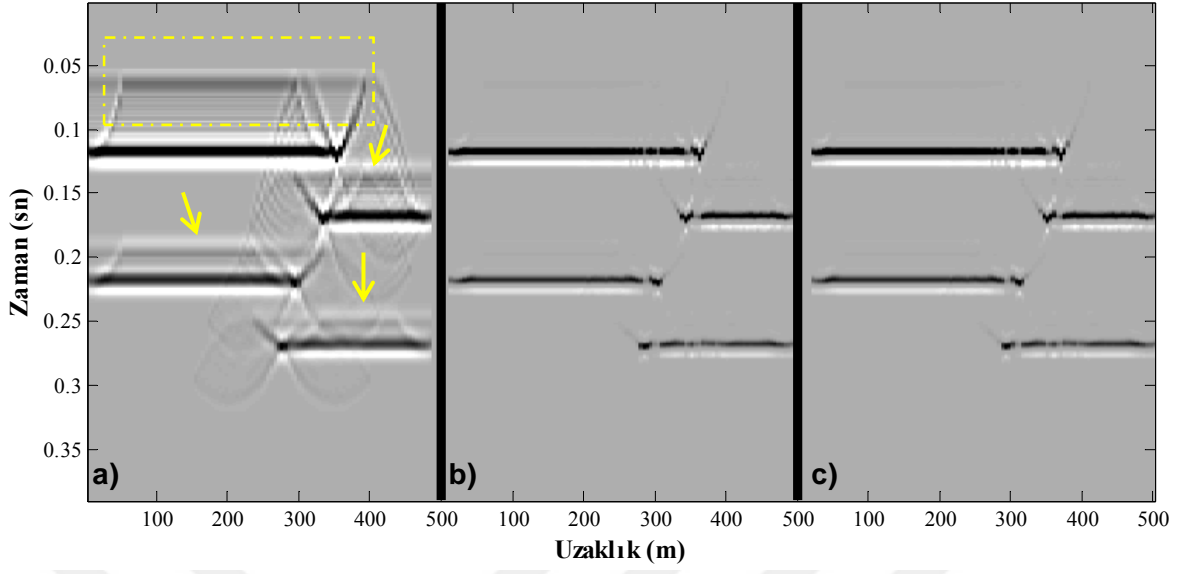
sahiptir. Bunun yanında $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonlarının göç hatalarını daha iyi sönümlediği hem şekiller üzerinden hem de Tablo 3.6 üzerinden açıkça görülmektedir. Ayrıca EKKGD çıkışları tüm netleştirme fonksiyonları için daha düşük hata oranlarına sahiptirler.



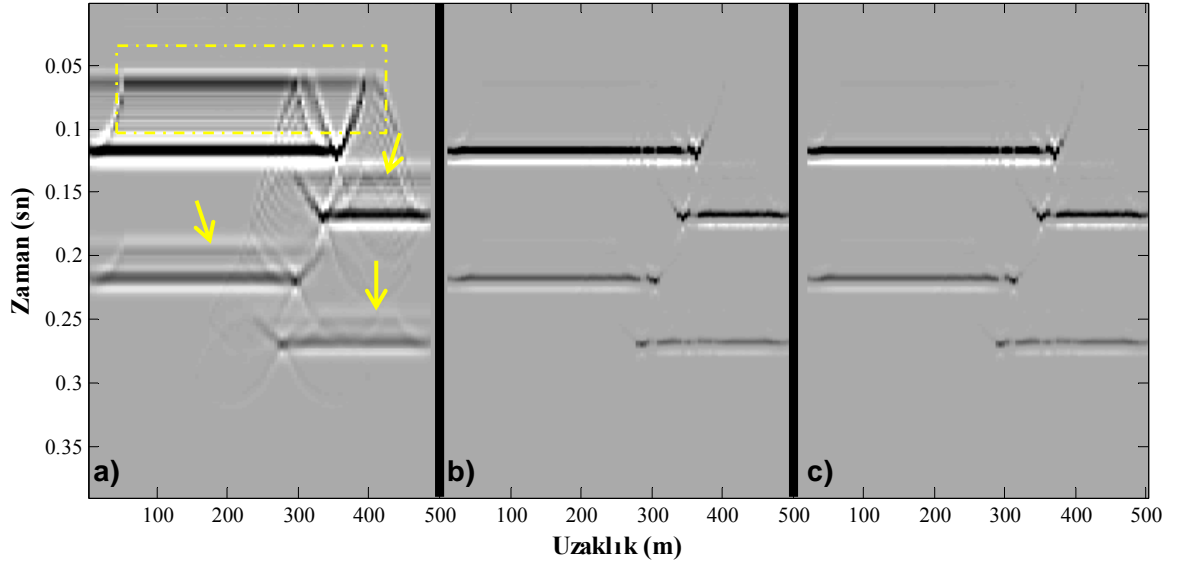
Şekil 3. 64. Fay modeli için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıkışları.



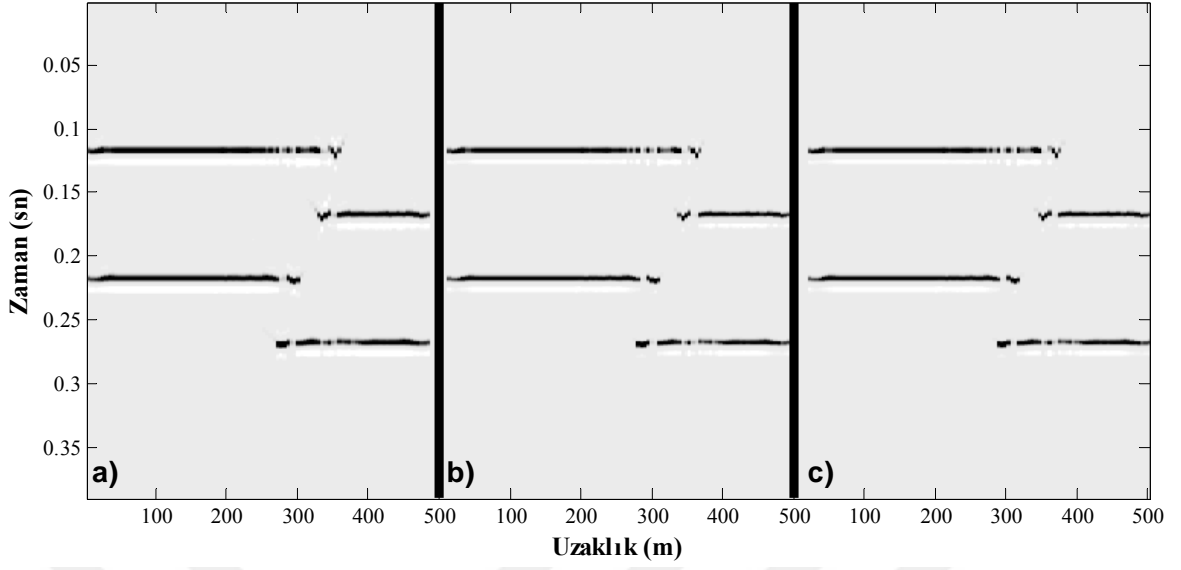
Şekil 3. 65. Fay modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.



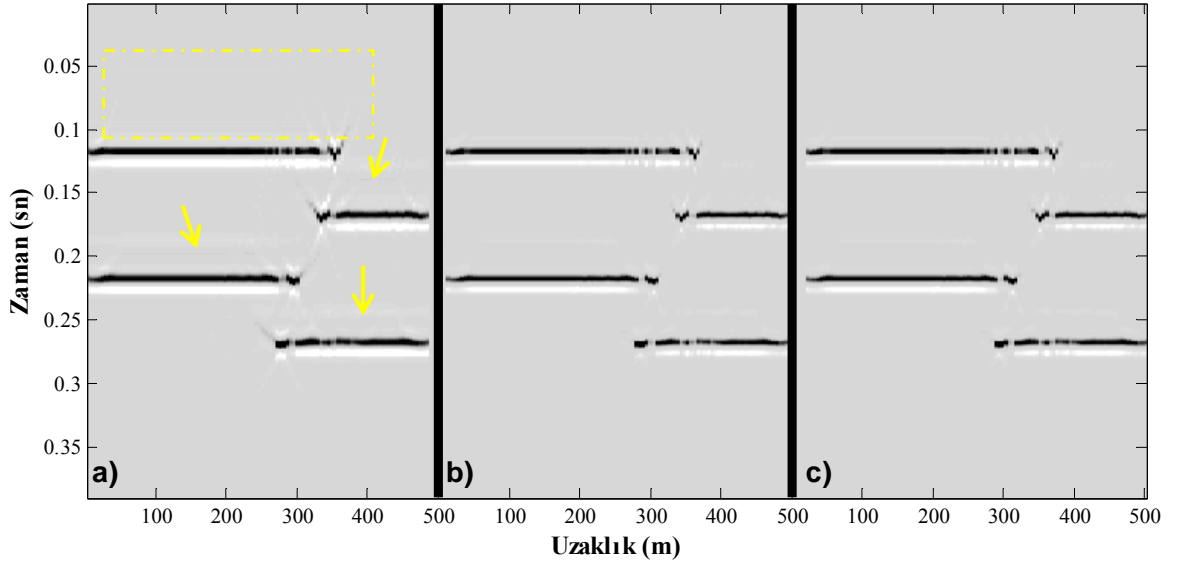
Şekil 3. 66. Fay modeli için $(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



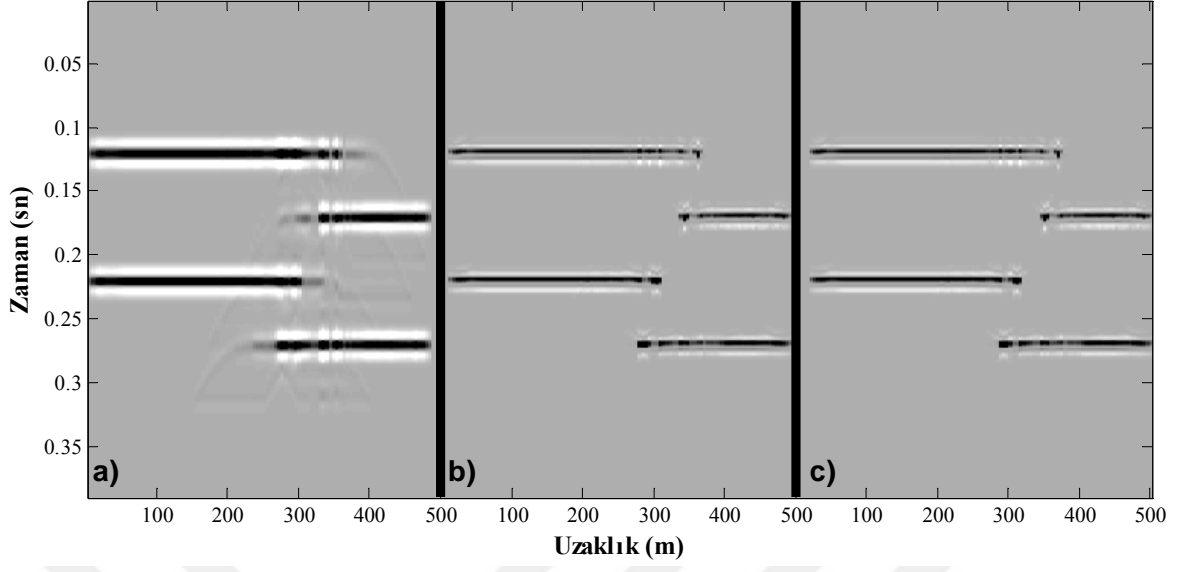
Şekil 3. 67. Fay modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



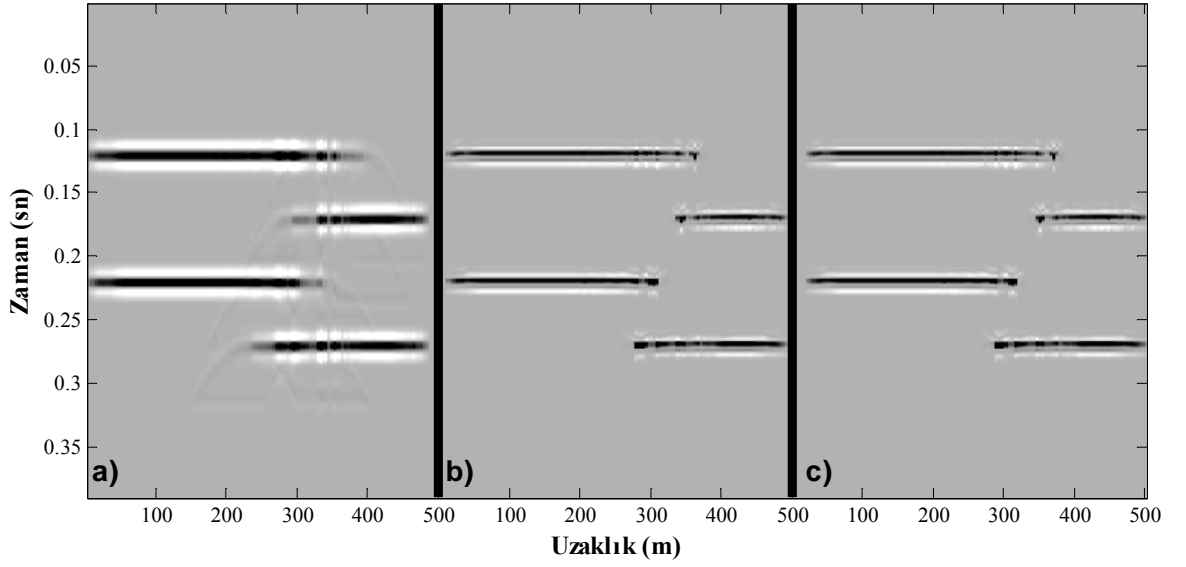
Şekil 3. 68. Fay modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) Gauss-seidel yöntemi ile EKKGD çıktıları.



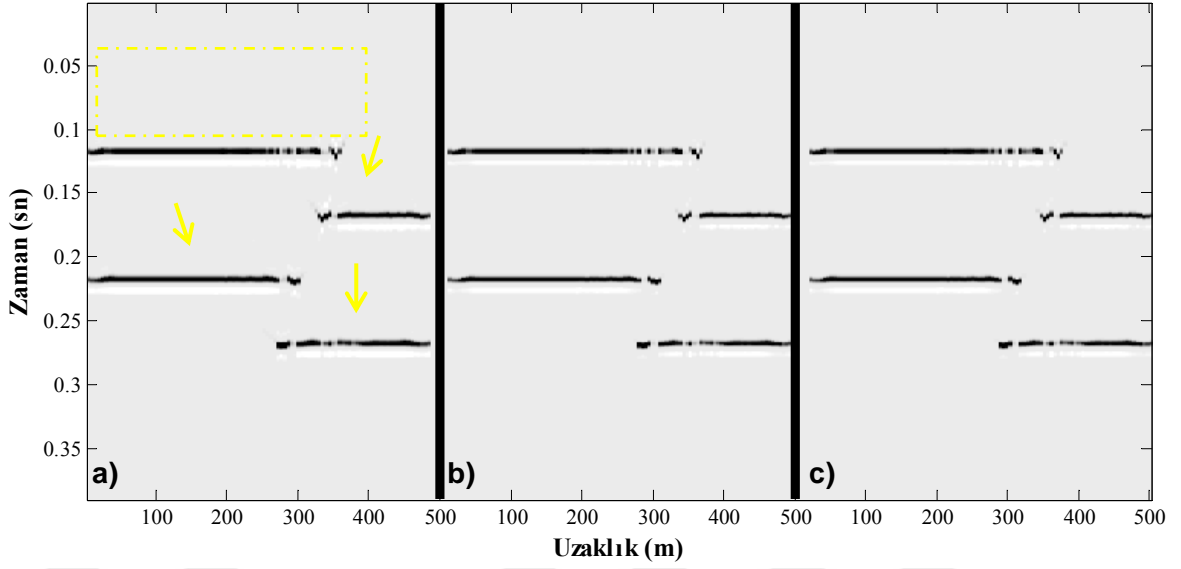
Şekil 3. 69. Fay modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 70. Fay modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 71. Fay modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 72. Fay modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

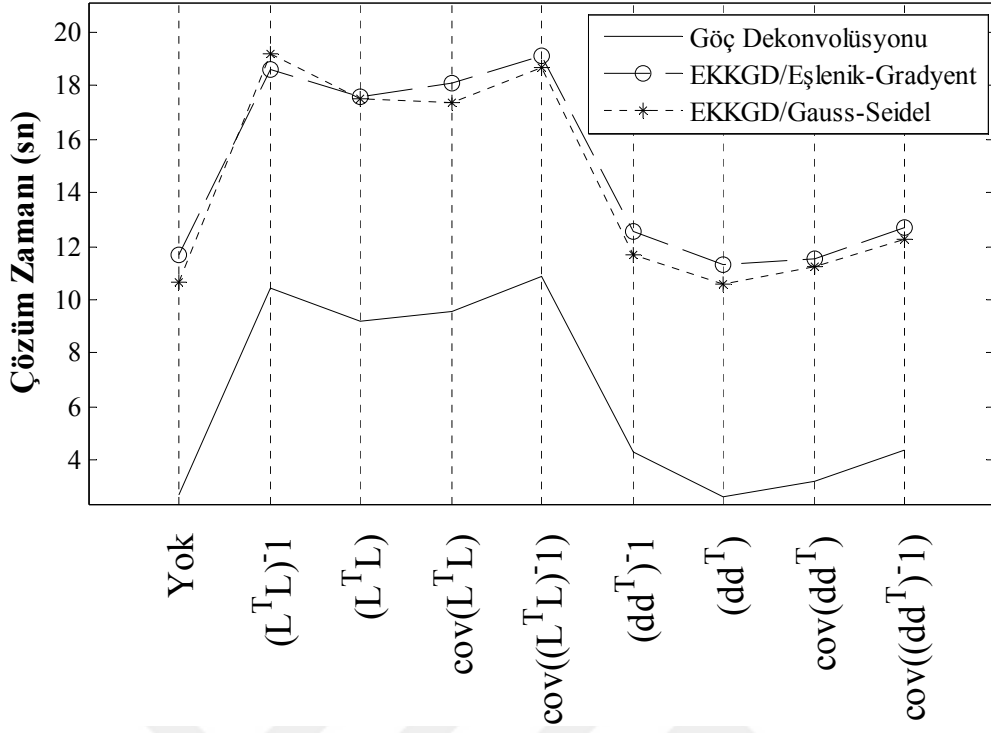
Tablo 3. 5. Fay modeli için netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 100, zaman örnek sayısı: 390, yarı operatör boyu: 100 m, 10iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD(sn)	EKKGD Eşlenik- Gradyent(sn)	EKKGD Gauss- Seidel(sn)	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	2,732	11,654	10,627	En Düşük Maliyet
$(L^T L)^{-1}$	10,454	18,634	19,201	Yüksek Maliyet
$(L^T L)$	9,217	17,558	17,476	Yüksek Maliyet
$cov(L^T L)$	9,558	18,065	17,343	Yüksek Maliyet
$cov((L^T L)^{-1})$	10,890	19,120	18,71	Yüksek Maliyet
$(dd^T)^{-1}$	4,308	12,549	11,693	Düşük Maliyet
(dd^T)	2,644	11,293	10,557	Düşük Maliyet
$cov(dd^T)$	3,237	11,500	11,246	Düşük Maliyet
$cov((dd^T)^{-1})$	4,392	12,678	12,264	Düşük Maliyet

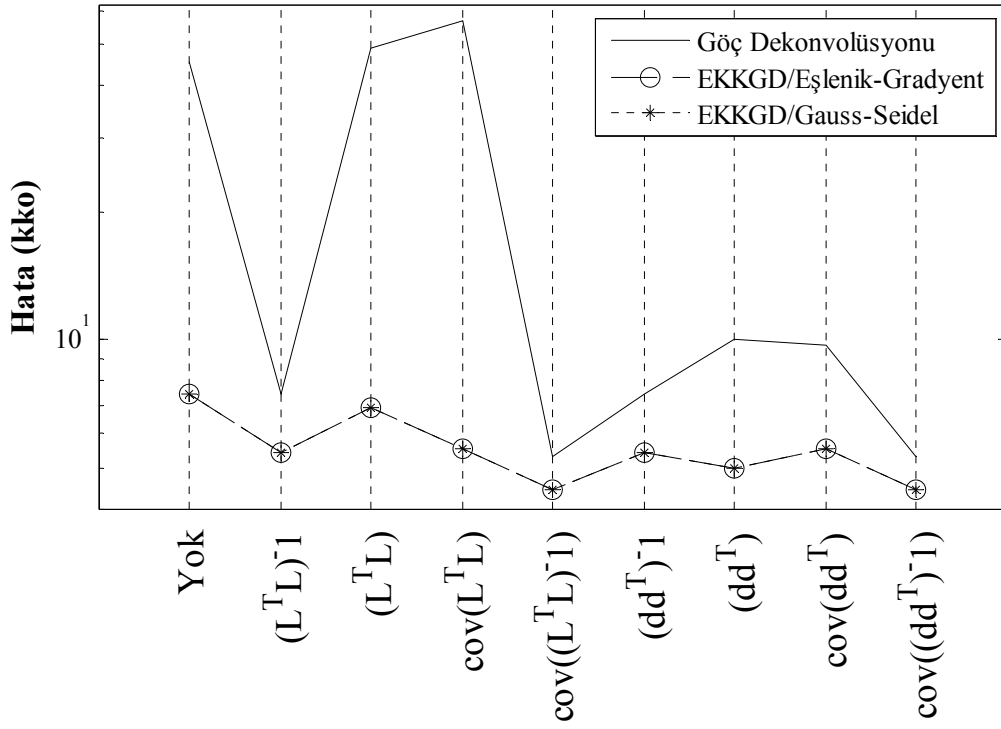
Tablo 3. 6. Fay modeli için netleştirme fonksiyonlarına göre hata oranları. Göç çıkışı hatalarının (artifacts) karelerinin toplamı.

Hata Oranı (kko)/ Netleştirme Fonksiyonu	GD	EKKGD Eşlenik- Gradyent	EKKGD Gauss- Seidel	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	45,42	7,41	7,41	Orta
$(L^T L)^{-1}$	7,41	5,39	5,39	Düşük
$(L^T L)$	49,13	6,93	6,93	Yüksek
$cov(L^T L)$	56,59	5,52	5,52	Düşük
$cov((L^T L)^{-1})$	5,32	4,45	4,45	Düşük
$(dd^T)^{-1}$	7,41	5,39	5,39	Düşük
(dd^T)	10,08	4,99	4,99	Çok Düşük
$cov(dd^T)$	9,75	5,26	5,26	Düşük
$cov((dd^T)^{-1})$	5,32	4,45	4,45	Çok Düşük

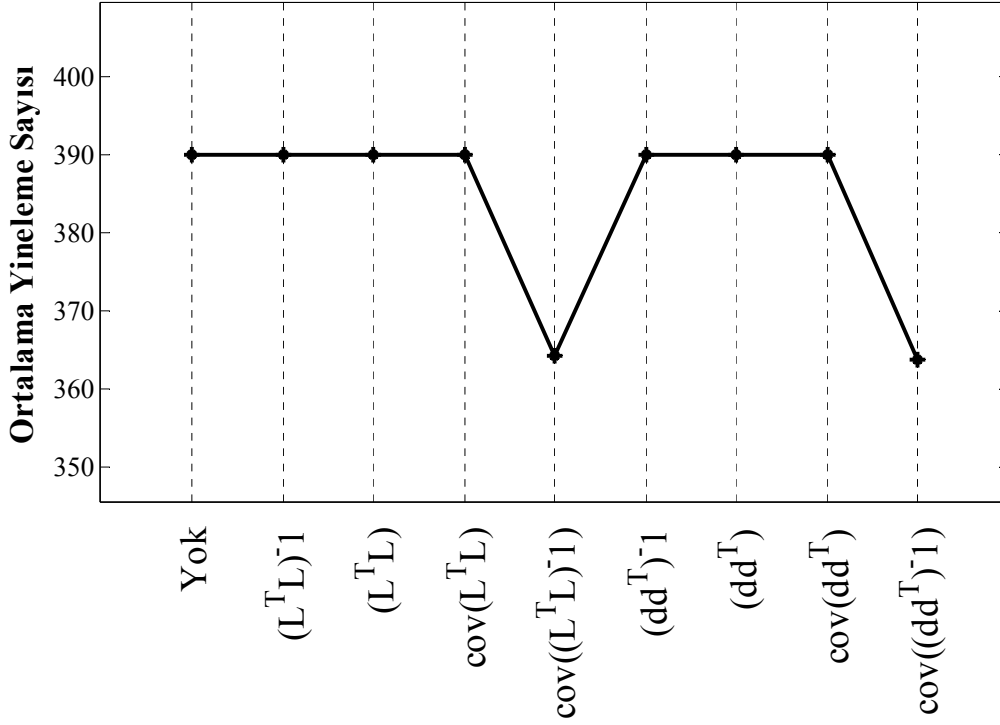
Şekil 3.73 fay modeli için tüm yöntemlerin çözüm zamanlarını grafik üzerinde göstermektedir. Giriş verisi (d) kullanılarak elde edilen göç çıkışları, GD yaklaşık %64 daha az zamanda, EKKGD için yaklaşık %35 daha az zamanda çözüme ulaşmışlardır. Şekil 3.74 göç çıkışlarının toplam karekök hatalarını göstermektedir. Buna göre, $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları kullanılarak gerçekleştirilen göç işlemleri istenen çıkışa en fazla yaklaşan yöntemlerdir. Şekil 3.75 eşlenik-gradyent yöntemi için ortalama yineleme sayılarını göstermektedir. Buna göre $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları yineleme sayılarını azaltabilen netleştirme fonksiyonları olduğu görülmektedir.



Şekil 3. 73. Fay modeli için farklı netleştirme fonksiyonları ile uygulanan göç yöntemlerinin çözüm zamanları.



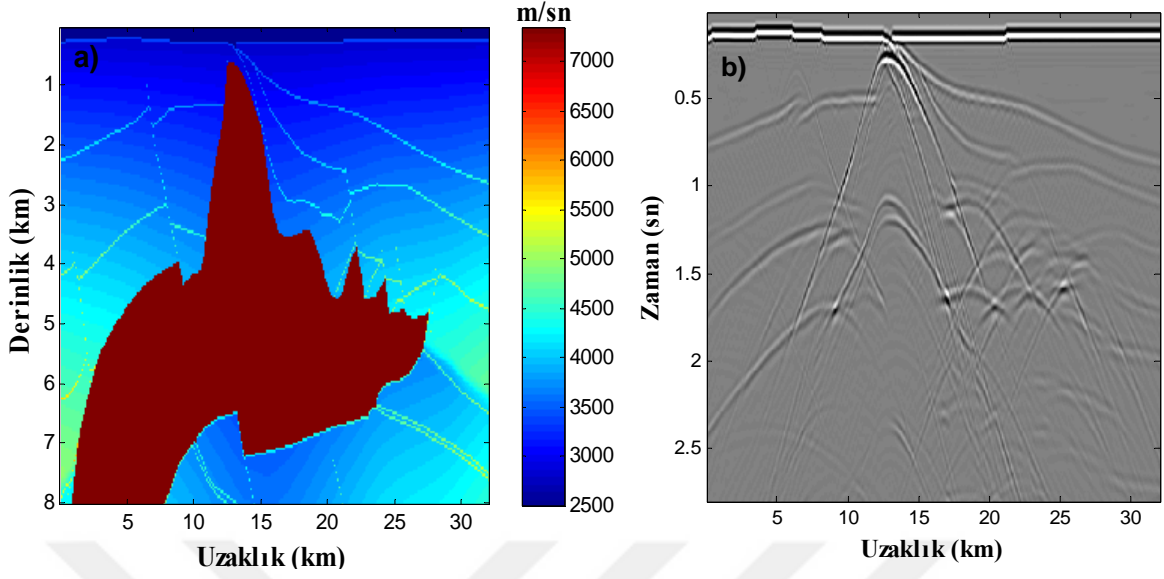
Şekil 3. 74. Fay modeli için farklı netleştirme fonksiyonları uygulanan göç yöntemlerinin kko hata miktarları.



Şekil 3. 75. Fay modeli için farklı netleştirme fonksiyonları ile uygulanan eşlenik-gradyent (EKKGD) yöntemi için ortalama yineleme sayıları.

3.1.4. SEG/EAGE Tuz Modeli Üzerinde Göç Uygulamaları

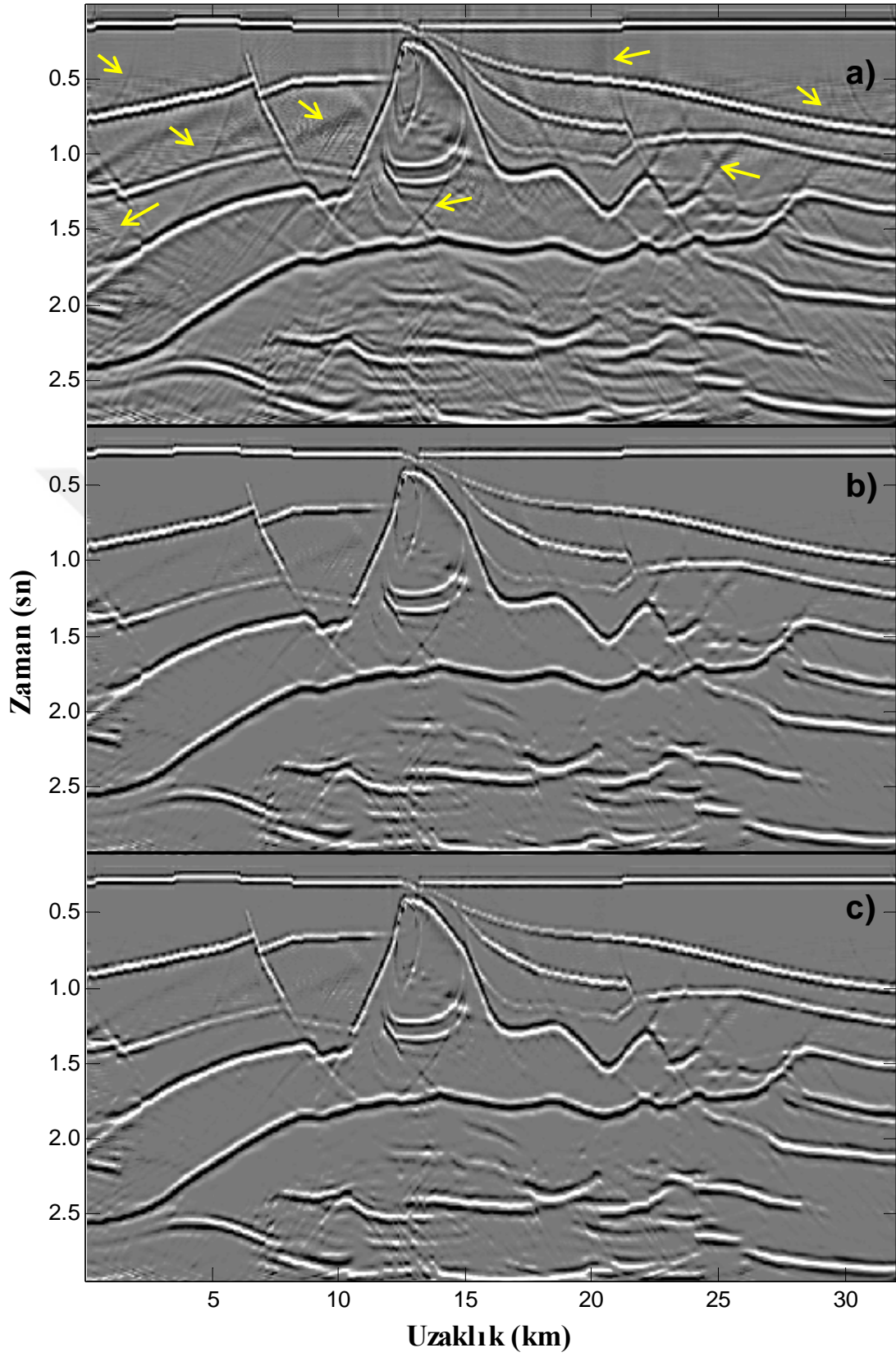
Tuz modelleri karmaşık jeolojisi ve çok fazla kırıklı (faylı) yapı karakteri ile göç işlemleri için çözülmesi zor modellerden birisi olarak kabul edilebilir. Göç algoritmalarının doğruluk ve güvenilirlik testleri çoğunlukla bu modeller üzerinde gerçekleştirilir. Bu çalışmada kullanılan tuz modeli Şekil 3.76a' da gösterilmektedir. Model oldukça değişken jeolojiye, çok sayıda faya, saçıcı tepe noktalarına ve topoğrafik değişkenliğe sahiptir. Model tepkisinde (Şekil 3.76b) faylardan ve saçıcı tepe noktalarından çok fazla saçılma hiperbolü ve farklı boyutlardaki senklinalik yapılardan dolayı da papyon-kravat etkileri görülmektedir. Model için derinlik örnekleme aralığı ($dz = 40m$), uzaklık örnekleme aralığı ($dx = 40m$), zaman örnekleme aralığı ($dt = 0.008sn$) kayıt süresi ($T = 2.8sn$) hat uzunluğu 32km dir. Tuz modeli için yarı operatör boyu 3km kullanılarak 27 farklı göç çıkışı üretilmiştir.



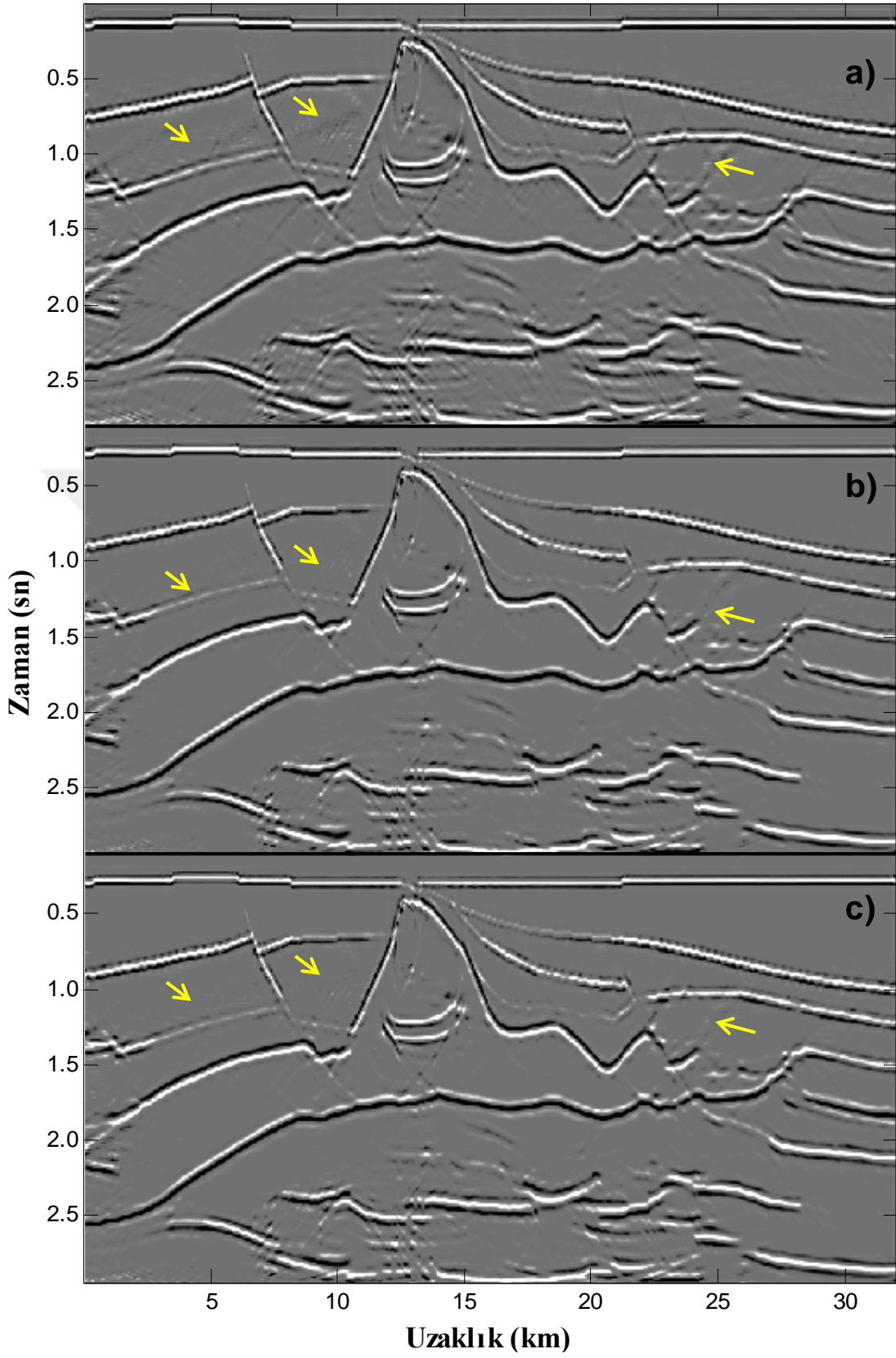
Şekil 3. 76. a) SEG/EAGE tuz modeli, b) sıfır ofset kesiti.

Tablo 3.7 tuz modeli için tüm göç yöntemlerinin çözüm zamanlarını göstermektedir. Tuz modeli için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışı Şekil 3.77' de gösterilmektedir. EKKG yöntemleri (Şekil 3.77b, c) önceki model verilerinde olduğu gibi Kirchhoff göçüne (Şekil 3.77a) göre göç hatalarını en küçüklediği açıkça görülmektedir. Ok işaretleri ile gösterilen göç hataları EKKG yöntemi ile sönmülenebilmiştir. Şekil 3.78 ile Şekil 3.85 arasında sırasıyla tuz modeli için $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$, $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. Şekil 3.78' de oklar ile gösterilen alanlara bakıldığında $(L^T L)^{-1}$ netleştirme fonksiyonu ile elde edilen EKKGD çıkışları (Şekil 3.78b,c) GD çıkışına (Şekil 3.78a) göre daha kaliteli göç çıkışları üretmiştir. Şekil 3.79' da $(L^T L)$ netleştirme fonksiyonu ile elde edilen GD çıkışı özellikle oklar ile gösterilen alanlara bakıldığında Kirchhoff göçüne oranla daha fazla göç hatası ürettiği görülebilmektedir. Şekil 3.80' e bakıldığında $cov(L^T L)$ netleştirme fonksiyonu DG çıkışında bulanık görüntünün yanında tüm göç çıkışları için 10 ila 15 km arasında bir gölge etkisi oluşmuştur. Şekil 3.81 ise $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen göç çıkışlarını göstermektedir. EKKGD ile elde edilen göç çıkışları, şekil üzerinde oklar ile gösterilen göç hatalarını sönmülemiştir, ancak 10 ila 15 km arasındaki gölge etkisini giderememiştir. Şekil 3.82' deki oklar ile gösterilen alanlara bakıldığında $(dd^T)^{-1}$

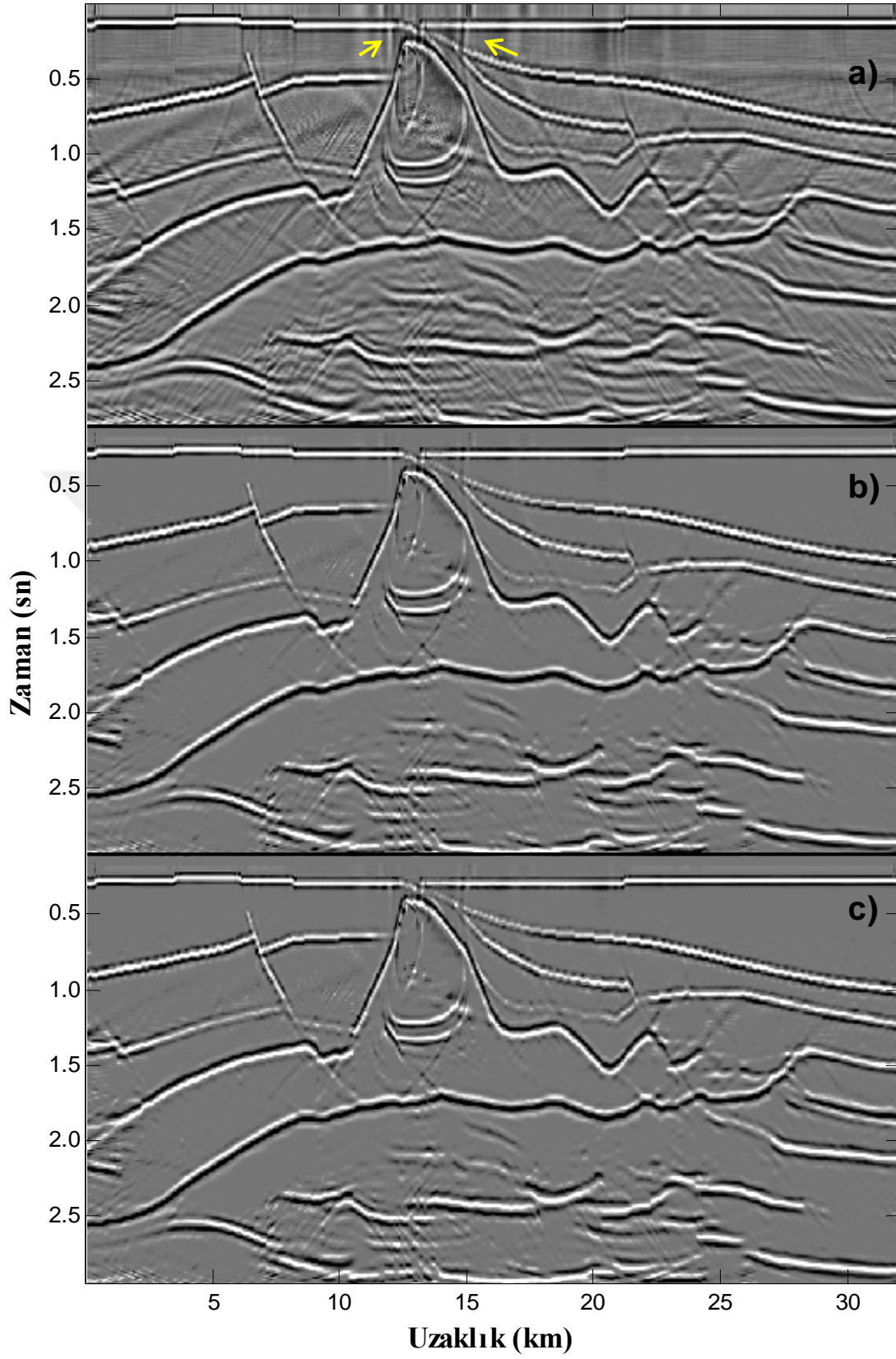
netleştirme fonksiyonu, GD çıkışını yaklaşık %70, EKKGD çıkışlarını yaklaşık %40 daha az zamanlarda geleneksel netleştirme fonksiyonu ile aynı kalitede göç çıkışlarını üretmiştir. İlk üç modelde hata oranları ve görüntü kaliteleri bakımından aynı sonuçları veren $(L^T L)^{-1}$ ve $(dd^T)^{-1}$ fonksiyonları Tuz modeli için de benzer kalitede sonuçlar üretmiştir. (dd^T) netleştirme fonksiyonu yalnızca yatay olayların genliklerine odaklayarak kusurlu çıkışlar (Şekil 3.83) üretmiştir. $cov(dd^T)$ netleştirme fonksiyonu (dd^T) netleştirme fonksiyonu gibi yatay olayların genliklerine odaklayarak kusurlu sonuçlar üretmiştir (Şekil 3.84). Şekil 3.85' de oklar ile gösterilen alanlara ve 10-15 km arasındaki gölge etkisine bakıldığında $cov((dd^T)^{-1})$ netleştirme fonksiyonu, $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile aynı kalitede ve aynı hatada çıkışlar üretmiştir. Önceki veri setlerinde olduğu gibi $(dd^T)^{-1}$ ile $(L^T L)^{-1}$ netleştirme fonksiyonları ile elde edilen göç çıkışları (Şekil 3.78, 3.82) ve $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile elde edilen göç çıkışları (Şekil 3.81, 3.85) aynı kalitede ve benzer hatalarda göç çıkışları üretmişlerdir. Bunun yanında ilk kez $cov((L^T L)^{-1})$ ile $cov((dd^T)^{-1})$ netleştirme fonksiyonları veri setinin bir kısmı için hata oluşturmuştur. Bunun nedeni derindeki yüksek genlikli yansımalar olabilir.



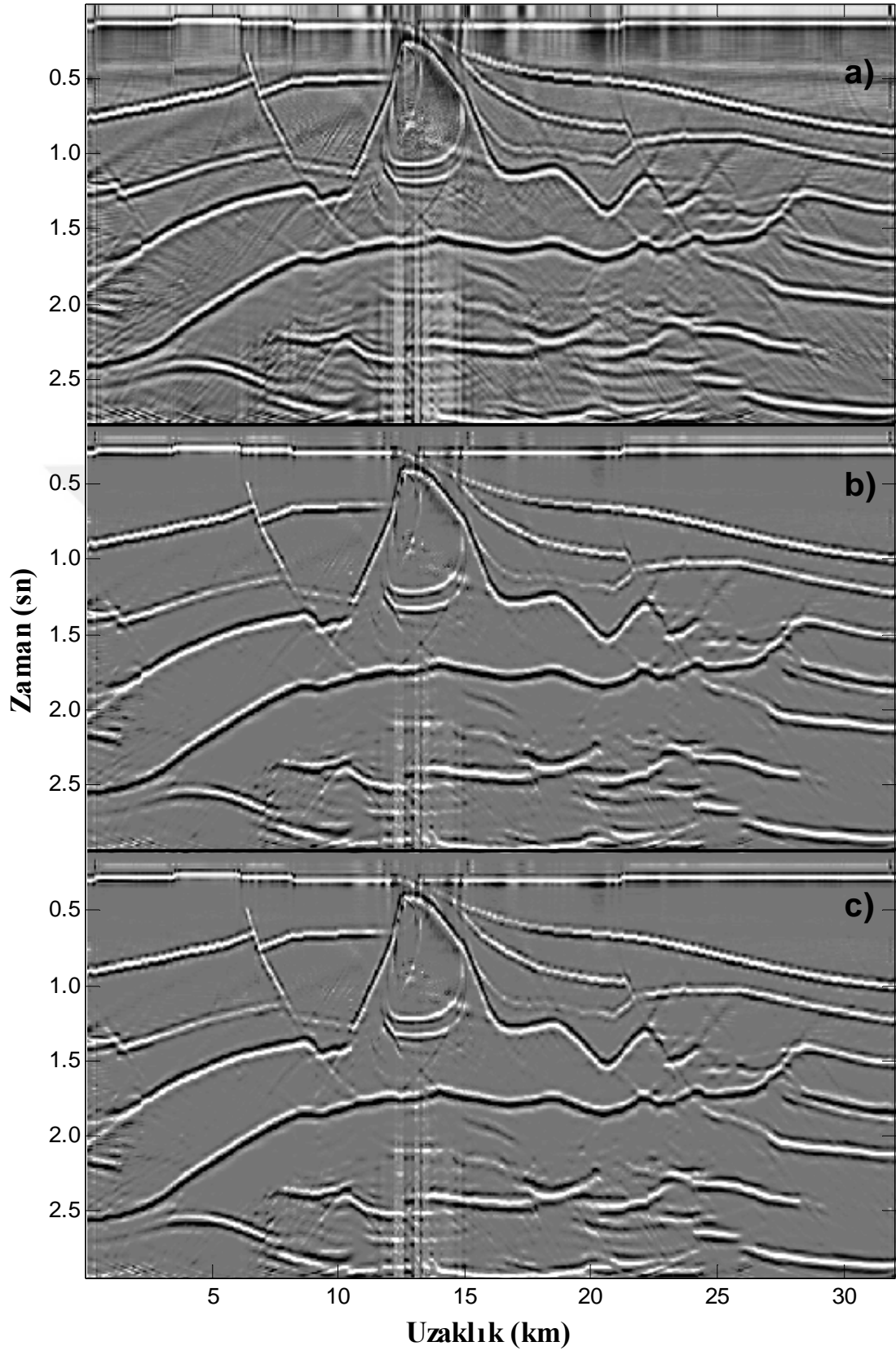
Şekil 3. 77. Tuz modeli için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıktıları.



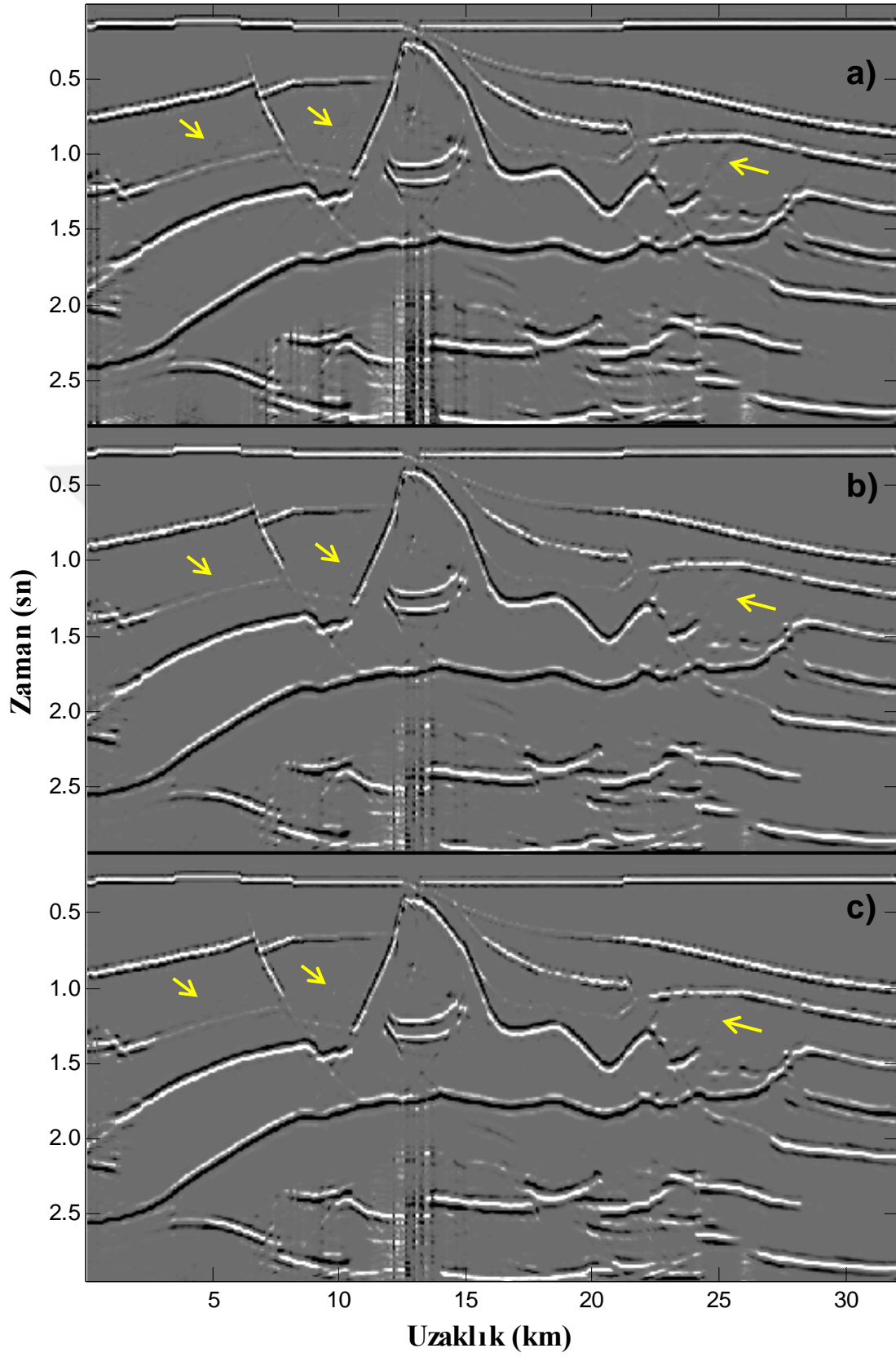
Şekil 3. 78. Tuz modeli için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD , b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



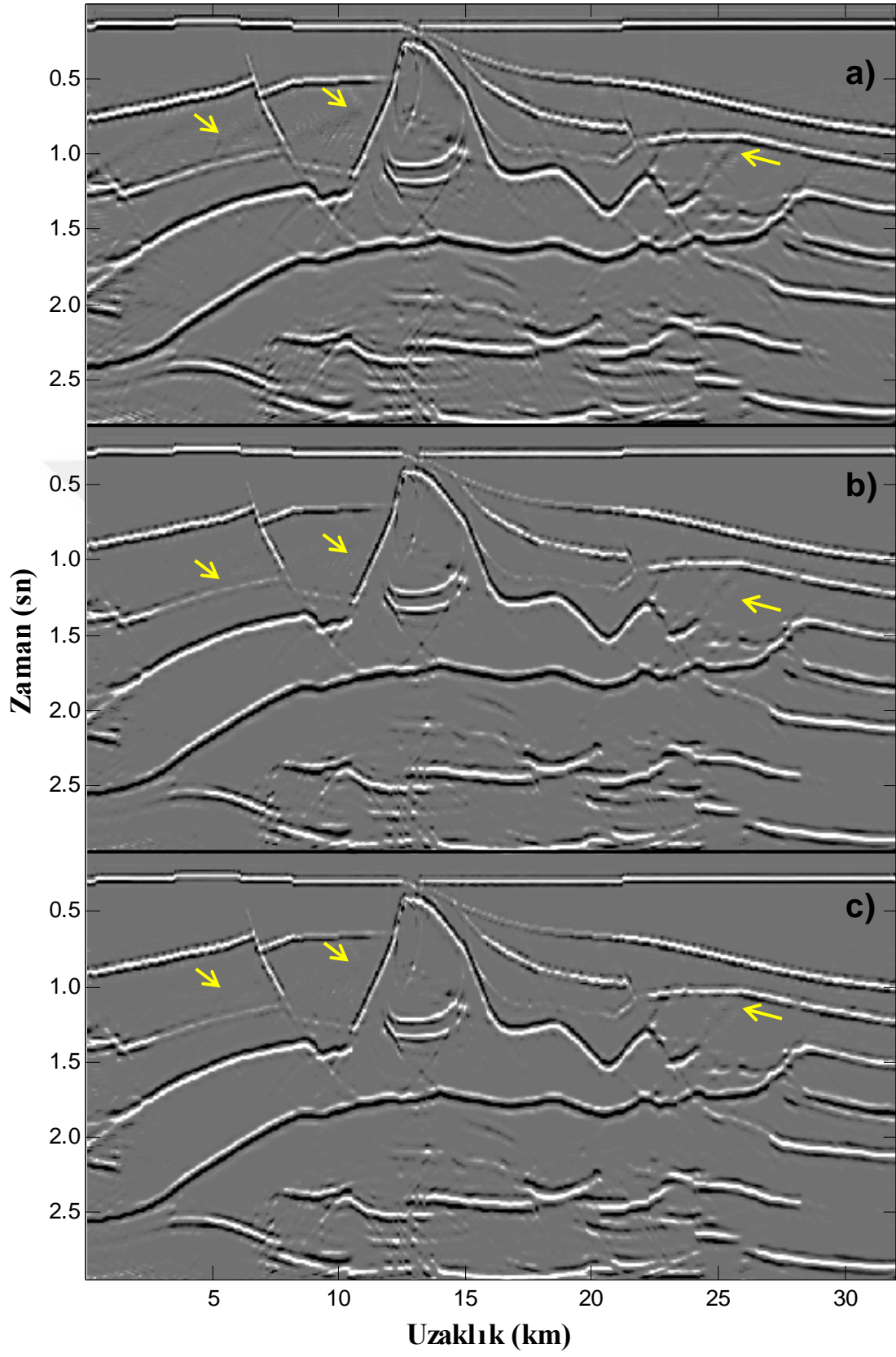
Şekil 3. 79. Tuz modeli için ($L^T L$) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



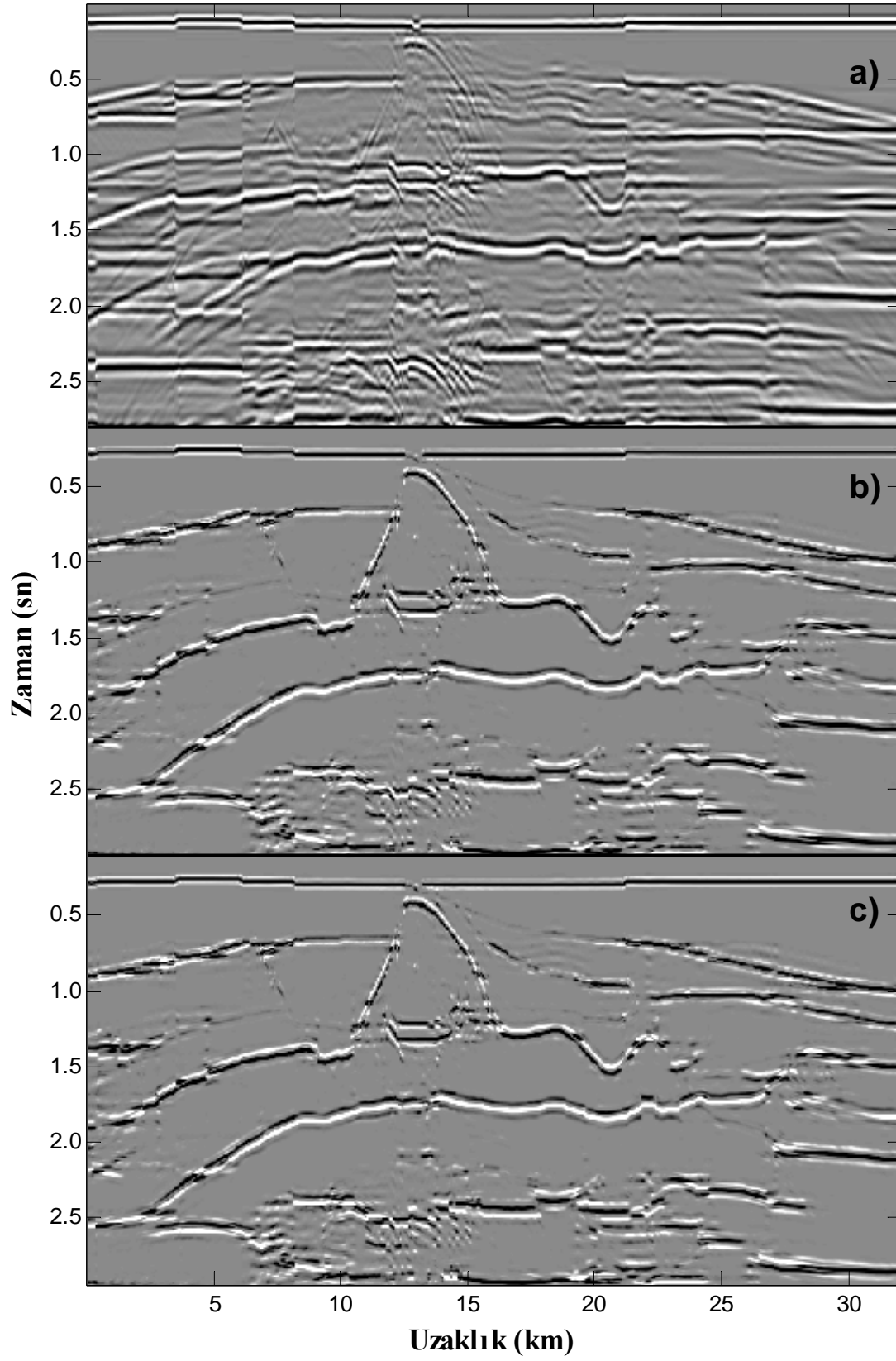
Şekil 3. 80. Tuz modeli için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



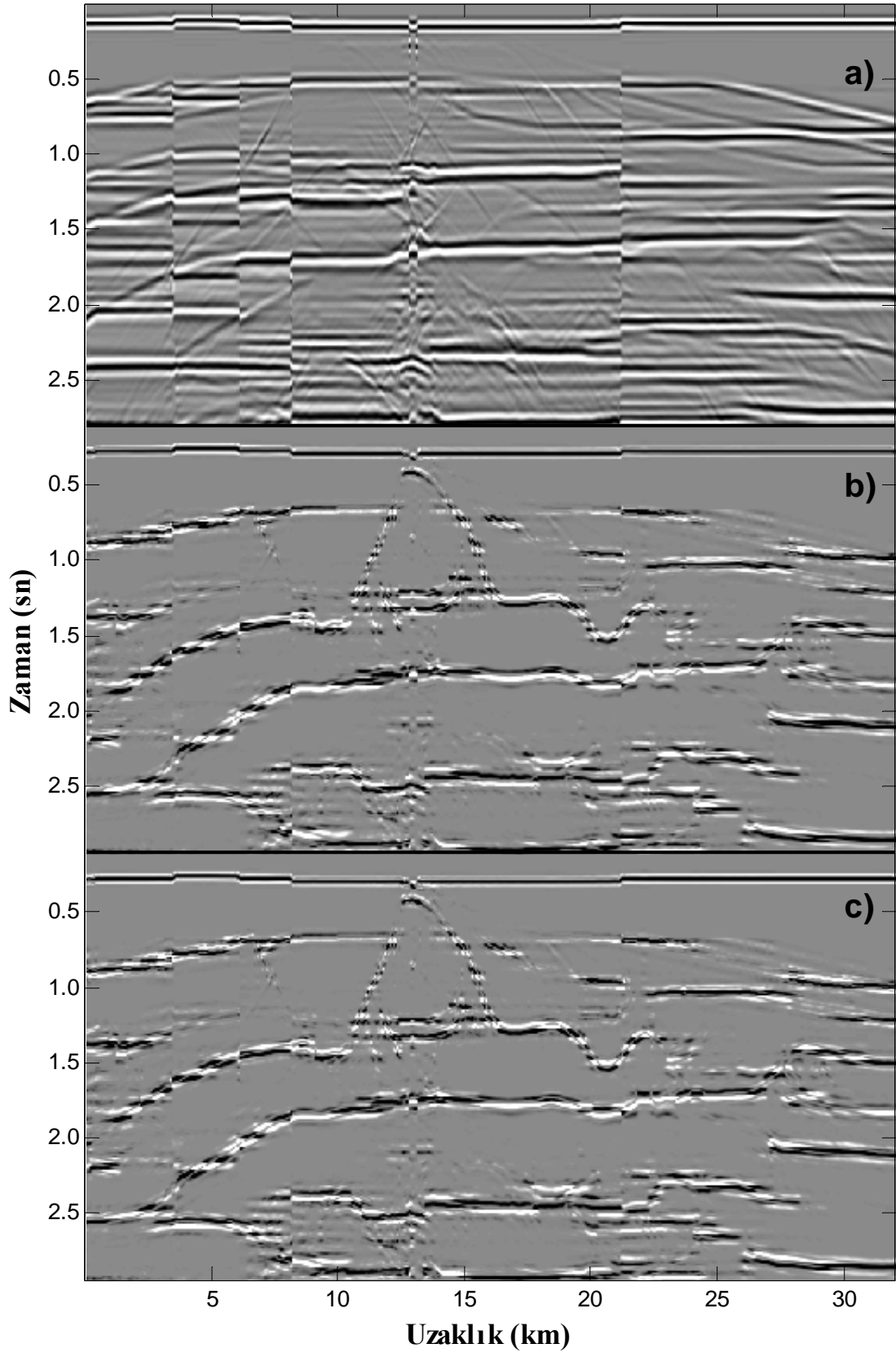
Şekil 3. 81. Tuz modeli için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



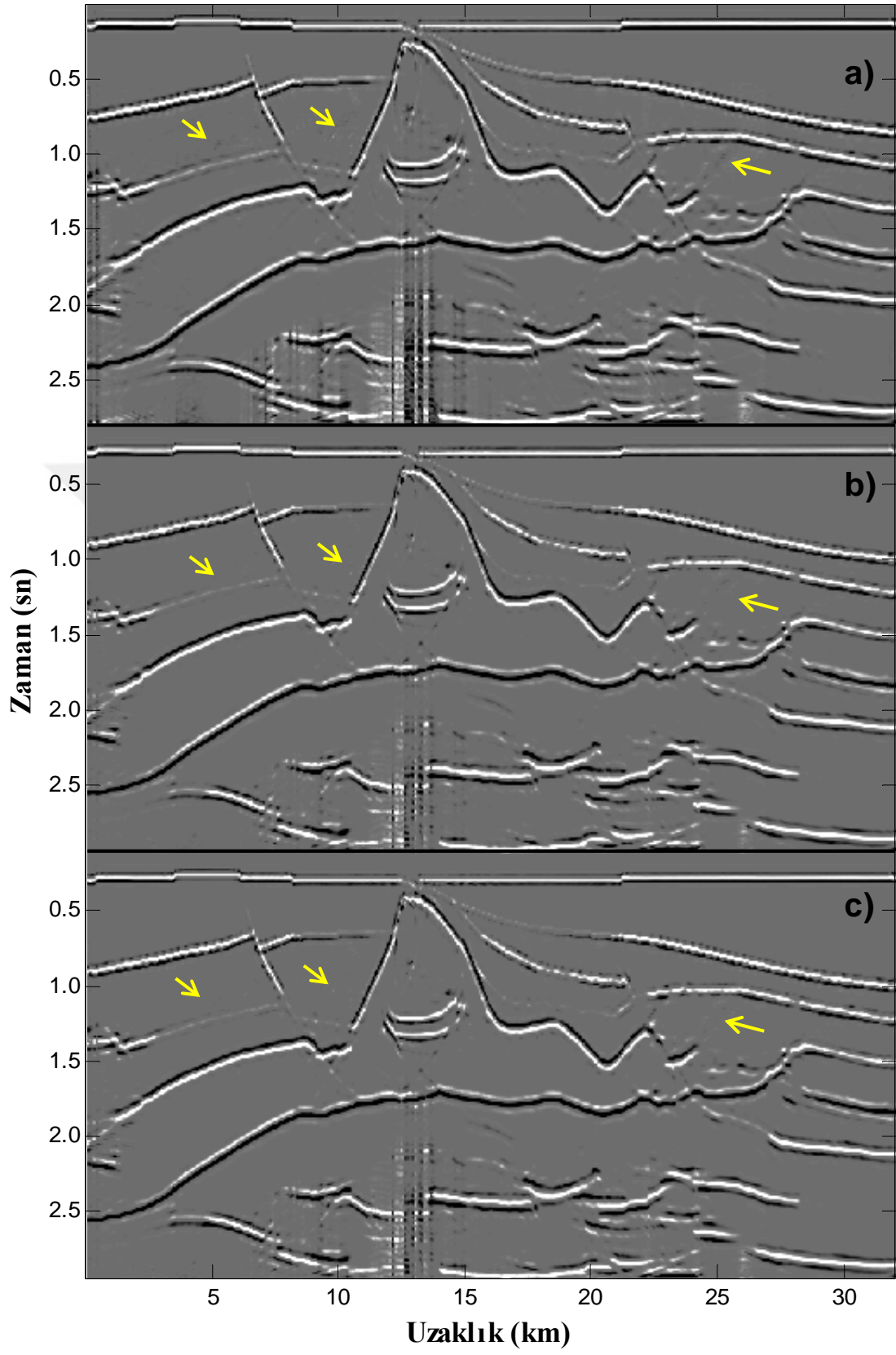
Şekil 3. 82. Tuz modeli için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 83. Tuz modeli için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 84. Tuz modeli için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.

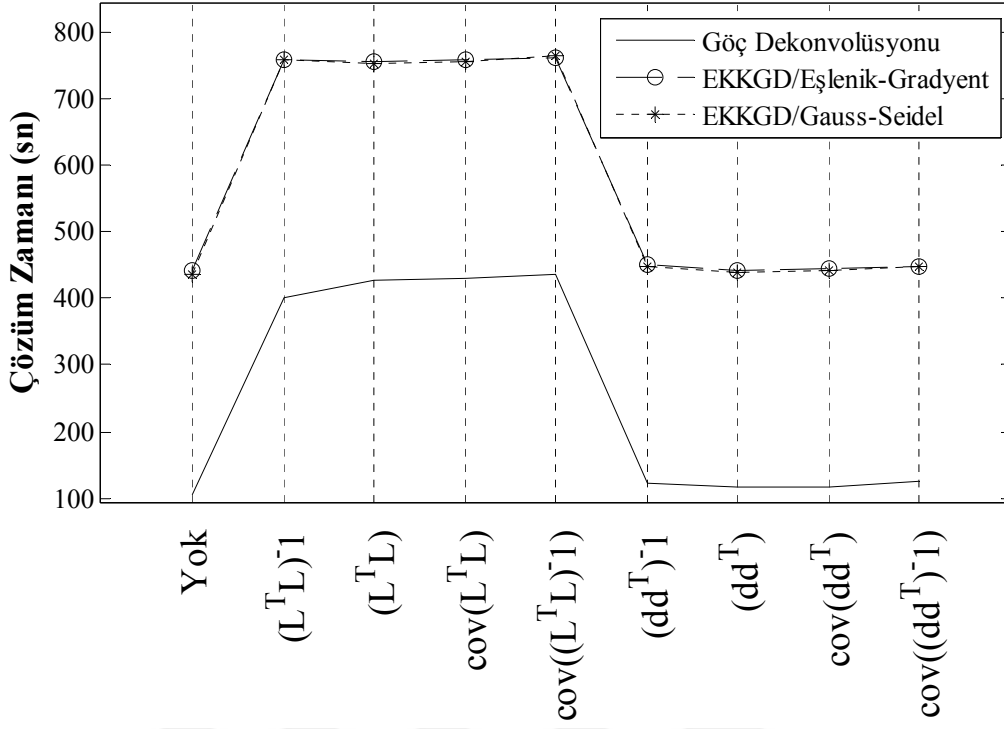


Şekil 3. 85. Tuz modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıkışları.

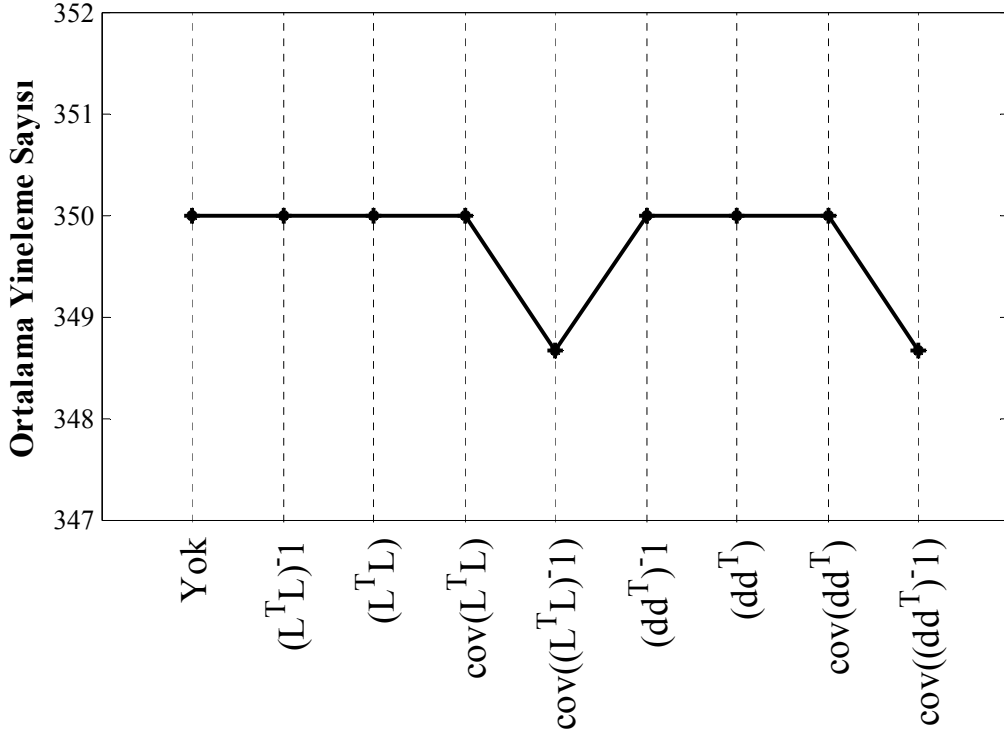
Tablo 3. 7. Tuz modeli için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 801, zaman örnek sayısı: 350, yarı operatör boyu: 3000 m, 75 iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD(sn)	EKKGD Eşlenik- Gradyent (sn)	EKKGD Gauss- Seidel (sn)	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	104,99	441,90	437,29	En Düşük Maliyet Orta Kalite
$(L^T L)^{-1}$	401,33	758,25	760,67	Yüksek Maliyet Yüksek Kalite
$(L^T L)$	427,86	755,25	754,58	Yüksek Maliyet Düşük Kalite
$cov(L^T L)$	429,35	760,42	755,07	Düşük Maliyet Kusurlu
$cov((L^T L)^{-1})$	435,29	763,41	766,31	Düşük Maliyet Yüksek Kalite
$(dd^T)^{-1}$	123,11	450,73	447,09	Düşük Maliyet Kusurlu
(dd^T)	115,67	442,24	439,18	Düşük Maliyet Yüksek Kalite Kısmi Kusur
$cov(dd^T)$	117,33	444,31	442,25	Yüksek Maliyet Düşük Kalite
$cov((dd^T)^{-1})$	124,21	448,89	448,32	Yüksek Maliyet Yüksek Kalite Kısmi Kusur

Şekil 3.86 tuz modeli için tüm yöntemlerin çözüm zamanlarını grafik üzerinde göstermektedir. Giriş verisi (d) kullanılarak elde edilen göç çıkışları, GD yaklaşık %72 daha az zamanda, EKKGD için yaklaşık %42 daha az zamanda çözüme ulaşmışlardır. Buna göre göç operatörünün (L) kullanıldığı bütün yöntemler daha fazla zaman almaktadır. Şekil 3.87 eşlenik-gradyent yöntemi için ortalama yineleme sayılarını göstermektedir. Buna göre $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları yineleme sayılarını azaltabilen netleştirme fonksiyonları olduğu görülmektedir.



Şekil 3. 86. Tuz modeli için uygulanan göç yöntemlerinin çözüm zamanları.



Şekil 3. 87. Tuz modeli için uygulanan eşlenik-gradyent yöntemi ile EKKGD işlemi için ortalama yineleme sayıları.

3.2. Arazi Verileri Üzerinde Göç Uygulamaları

Sismik göç uygulamalarının temel amacı yorumlanabilir kesitler üretmektir. Bu kapsamda 4 farklı model verisinden sonra gerçek veriler üzerinde testler yapılmıştır. İki farklı arazi verisi (Dokuz Eylül Üniversitesi Deniz Bilimleri ve Teknolojileri Enstitüsü tarafından toplanmıştır) çok sayıda fay ve yeterince karmaşık yapıya sahip deniz verileridir. Veri toplama parametreleri Tablo 3.8’ de ve yığma kesitleri hazırlanırken kullanılan veri işlem adımları Tablo 3.9’ da gösterilmektedir.

Tablo 3. 8. Arazi verilerinin veri toplama parametreleri.

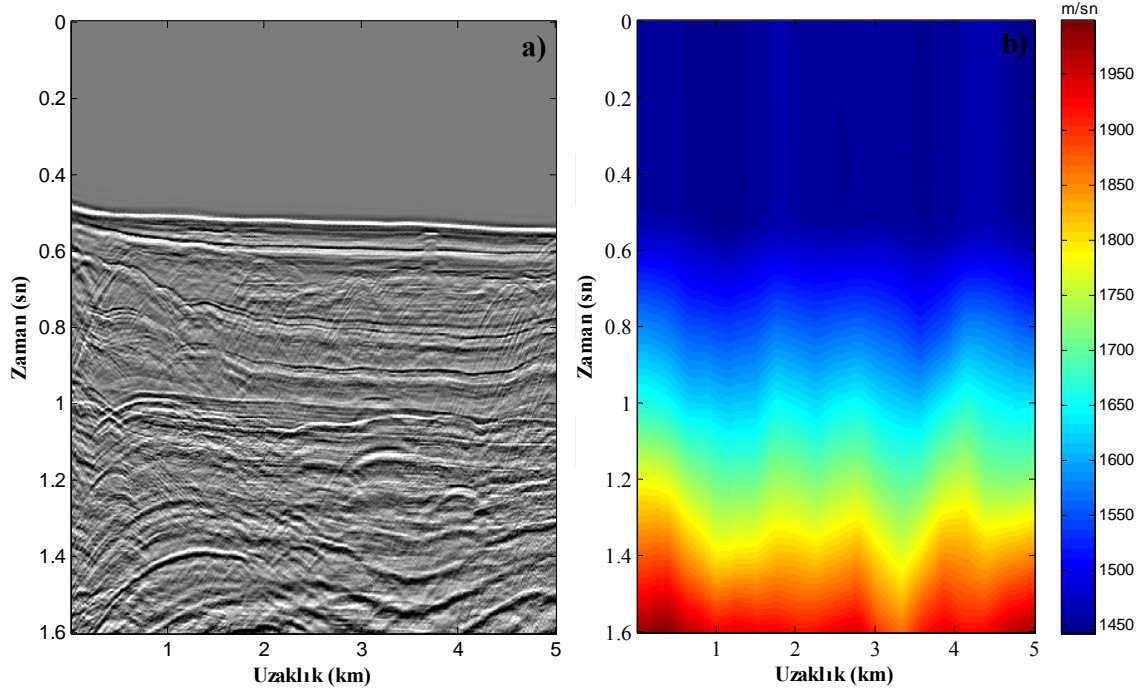
Parametre\Veri Seti	Arazi verisi-1	Arazi verisi-2
Uzaklık Örnekleme Aralığı (dx)	6.25 m	12.5 m
Zaman Örnekleme Aralığı (dt)	0.002 sn	0.002 sn
Kayıt Süresi (T)	1.6 sn	1.4 sn
Hat Uzunluğu	5006.25 m	12325 m
Veri Boyutu (Satır, Sütun)	801*801	701*986
Göç Yarı operatör boyu	468.75 m (75 iz)	625 m (50 iz)

Tablo 3. 9. Arazi verilerine uygulanan veri-işlem adımları.

Arazi Verisi-1 ve Arazi verisi-2 için Uygulanan Veri İşlem Adımları
İstenmeyen İzlerin Atılması (kill)
İstenmeyen Üst Alanların Atılması (top mute)
Frekans-Dalgasayısı Süzgeci (F-K)
Ortak Orta Nokta (OON, CMP) İzlerinin Ayıklanması
Hız Analizi
Normal Kayma Zamanı Düzeltmesi (NKZ, NMO)
Yığma
Otomatik Genlik Kazancı (AGC, yalnızca görüntülemek için)

3.2.1. Arazi Verisi-1 Üzerinde Göç Uygulamaları

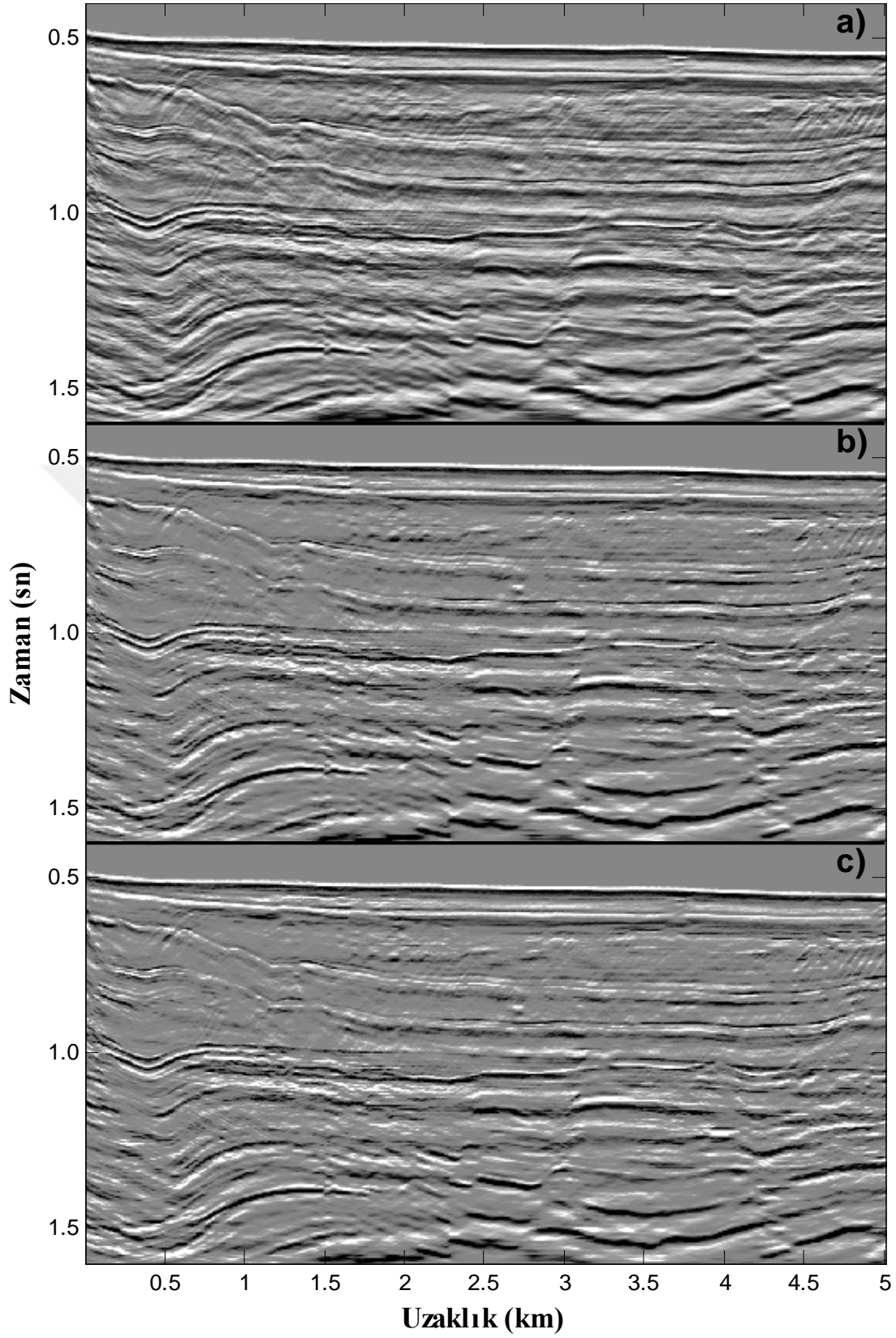
Şekil 3.88 arazi verisi-1 ve veri işlem sürecinde oluşturulan hız kesitini göstermektedir. Yapay modeller için kullanılan tüm göç algoritmaları, yarı operatör boyu 468.75 m kullanılarak bu veriye de uygulanmış ve toplam 27 farklı göç çıkışı elde edilmiştir. Göç çıkışlarına istenmeyen üst alanların atılması (top mute) ve görüntüleme amaçları için otomatik genlik kontrolü (Automatic Gain Control, AGC, 0.4 sn) işlemleri uygulanmıştır.



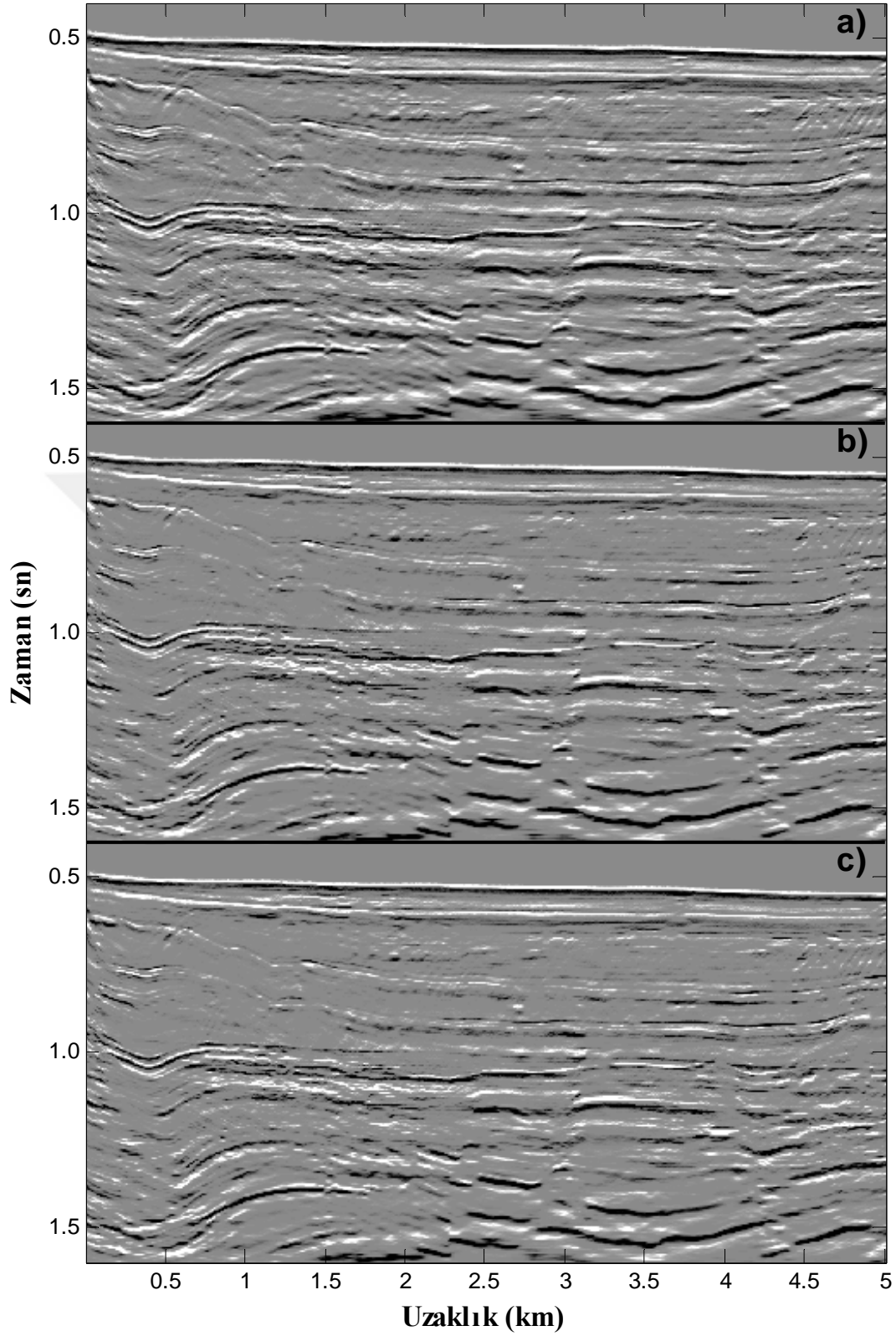
Şekil 3. 88. a) Arazi verisi-1 sıfır ofset kesiti, b) karekök ortalama hız (rms) hız kesiti.

Şekil 3.89 arazi verisi-1 için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışları gösterilmektedir. EKKG yöntemleri (Şekil 3.89b, c) önceki tüm veri setlerinde olduğu gibi Kirchhoff göçüne (Şekil 3.89a) göre daha net sonuçlar ürettiği tüm kesit üzerinden gözlenebilir. Şekil 3.90 ile Şekil 3.97 arasında sırasıyla arazi verisi-1 için $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$, $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. Geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile elde edilen EKKGD çıkışları, GD çıkışına göre daha da temizlenmiş görülmektedir (Şekil 3.90). EKKG ile Kirchhoff Göçü arasındaki

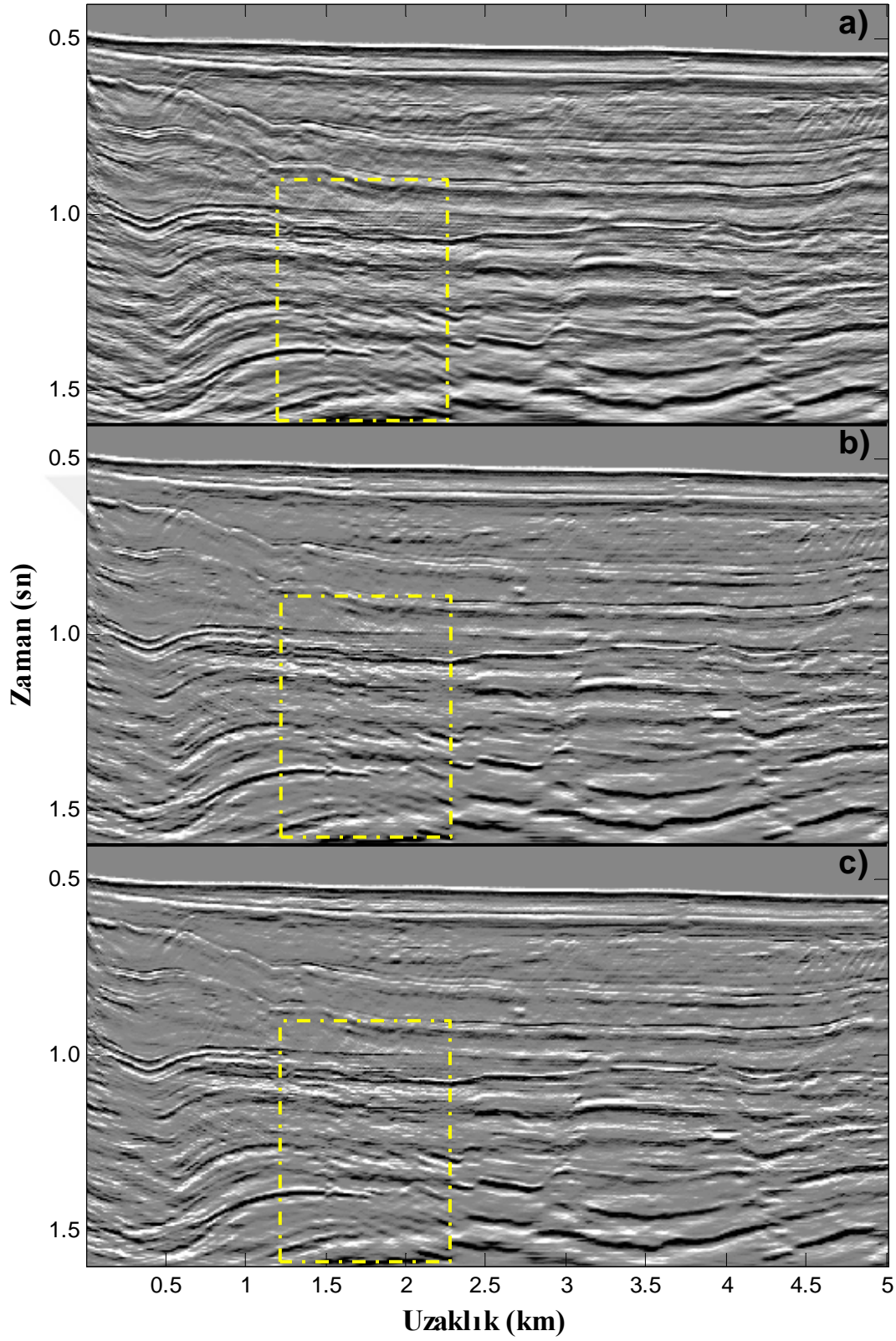
kıyaslama gibi tüm kesit üzerinden EKKGD işleminin göç hatalarını daha iyi sönümlediği tüm kesit üzerinden açıkça görülebilir. Şekil 3.91' deki kesikli çizgiler ile ayrılan alana bakıldığında $(L^T L)$ netleştirme fonksiyonunun kullanılmasının göç hatalarını daha da arttırdığı görülebilir. Şekil 3.92' ye bakıldığında $cov(L^T L)$ netleştirme fonksiyonu göç hatalarını $(L^T L)$ netleştirme fonksiyonuna göre daha da arttırdığı tüm kesitler üzerinden açıkça görülebilir. Şekil 3.93' e bakıldığında $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile elde edilen göç çıkışlarında ayrımlılığın belirgin olarak arttığı fakat 0.5-1 sn arasındaki genliklerin de zayıfladığı görülebilir. Şekil 3.94 $(dd^T)^{-1}$ netleştirme fonksiyonu ile elde edilen göç çıkışlarını göstermektedir. Önceki model verilerde olduğu gibi $(dd^T)^{-1}$ ve $(L^T L)^{-1}$ netleştirme fonksiyonları aynı kalitede göç çıkışları üretmiştir. (dd^T) netleştirme fonksiyonu önceki veri setlerinde olduğu gibi yalnızca yatay olayların genliklerine odaklayarak kusurlu çıkışlar (Şekil 3.95) üretmiştir. $cov(dd^T)$ netleştirme fonksiyonu (dd^T) netleştirme fonksiyonu gibi yatay olayların genliklerine odaklayarak arazi verisi-1 için de kusurlu sonuçlar üretmiştir (Şekil 3.96). $cov((dd^T)^{-1})$ netleştirme fonksiyonu (Şekil 3.97) diğer veri setlerinde olduğu gibi göç hatalarını en küçükleyerek yüksek kalitede göç çıkışları üretmiş, fakat $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile elde edilen göç çıkışları gibi 0.5-1 sn arasında genlik zayıflamalarına neden olmuştur.



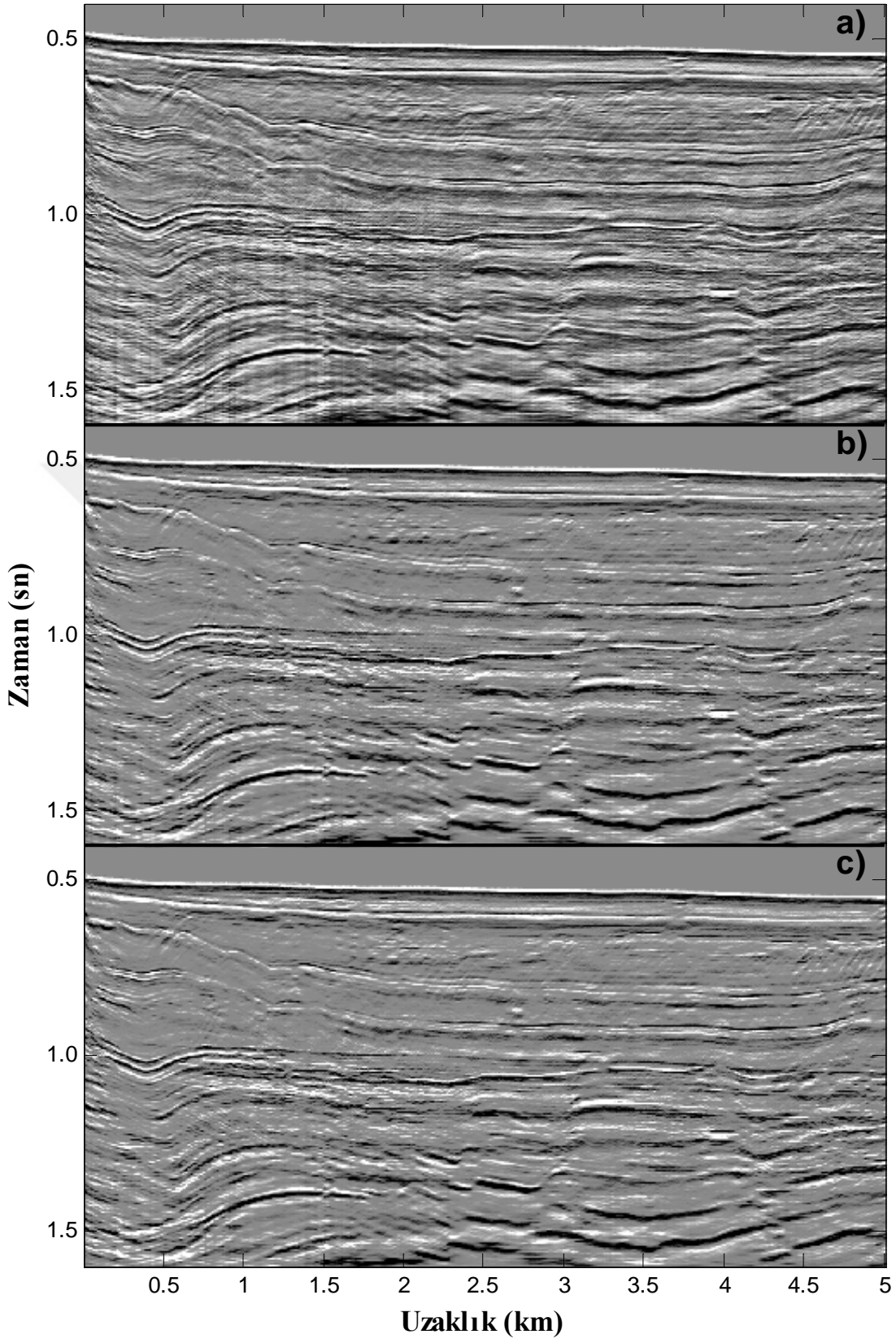
Şekil 3. 89. Arazi verisi-1 için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıktıları.



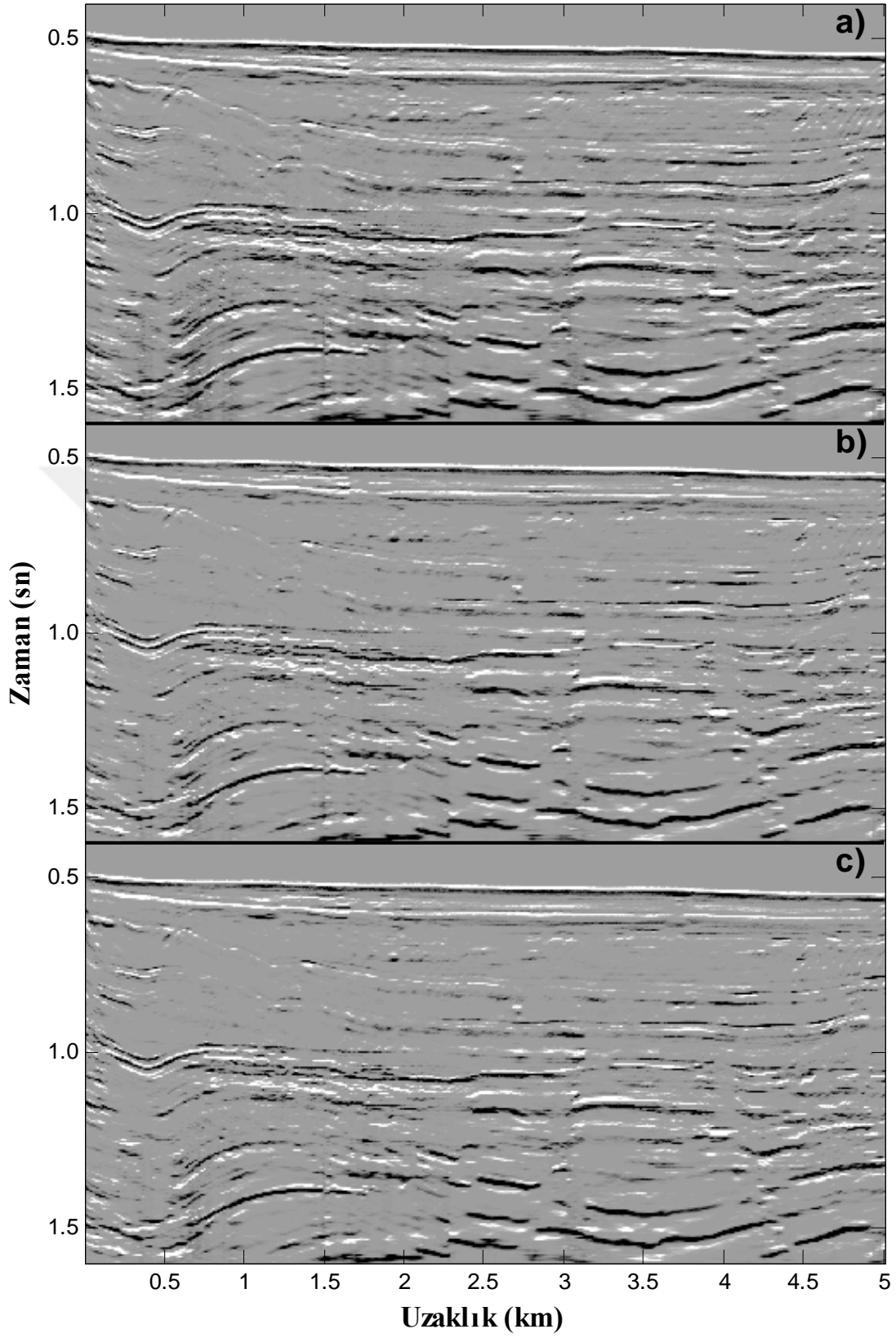
Şekil 3. 90. Arazi verisi-1 için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



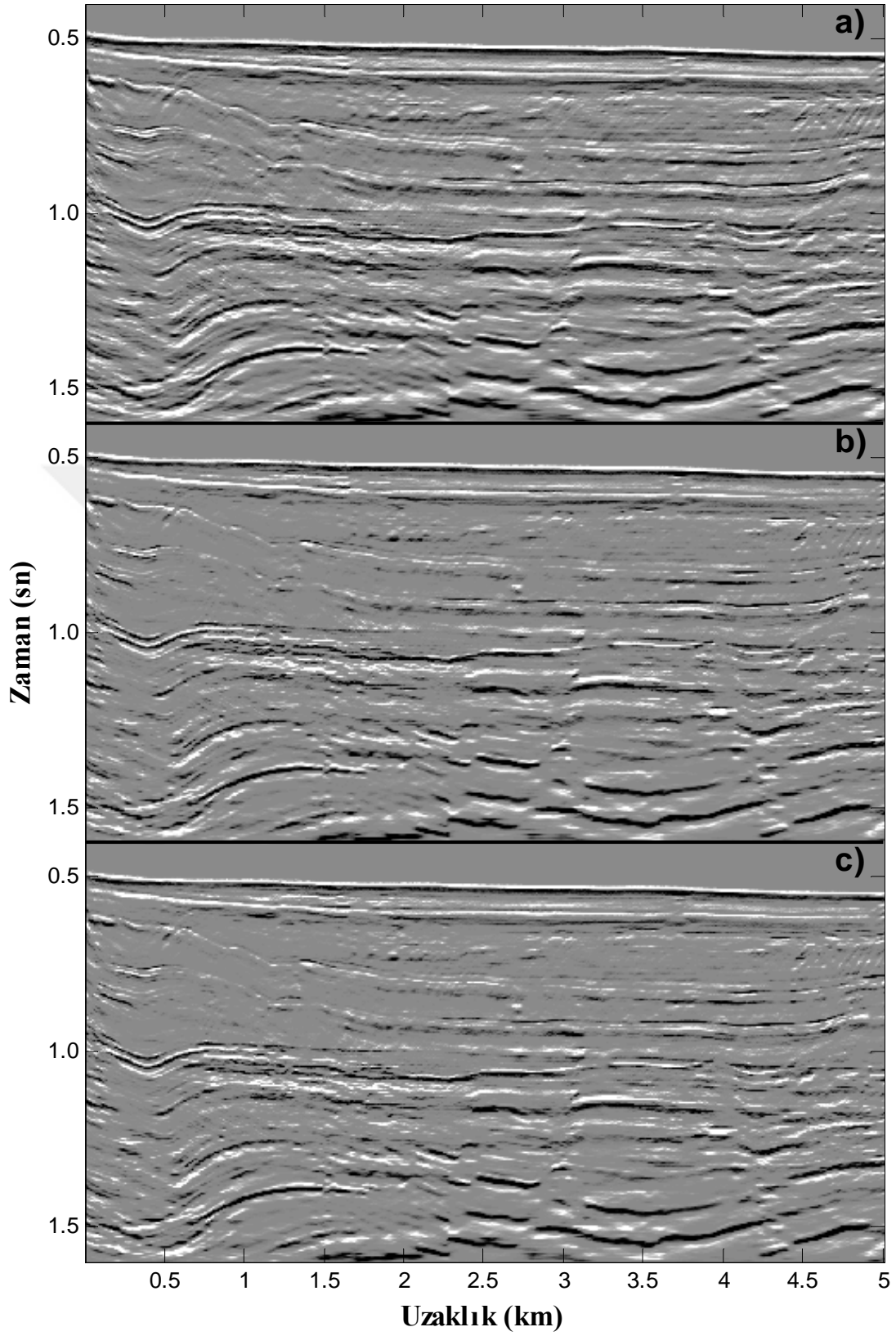
Şekil 3. 91. Arazi verisi-1 için ($L^T L$) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



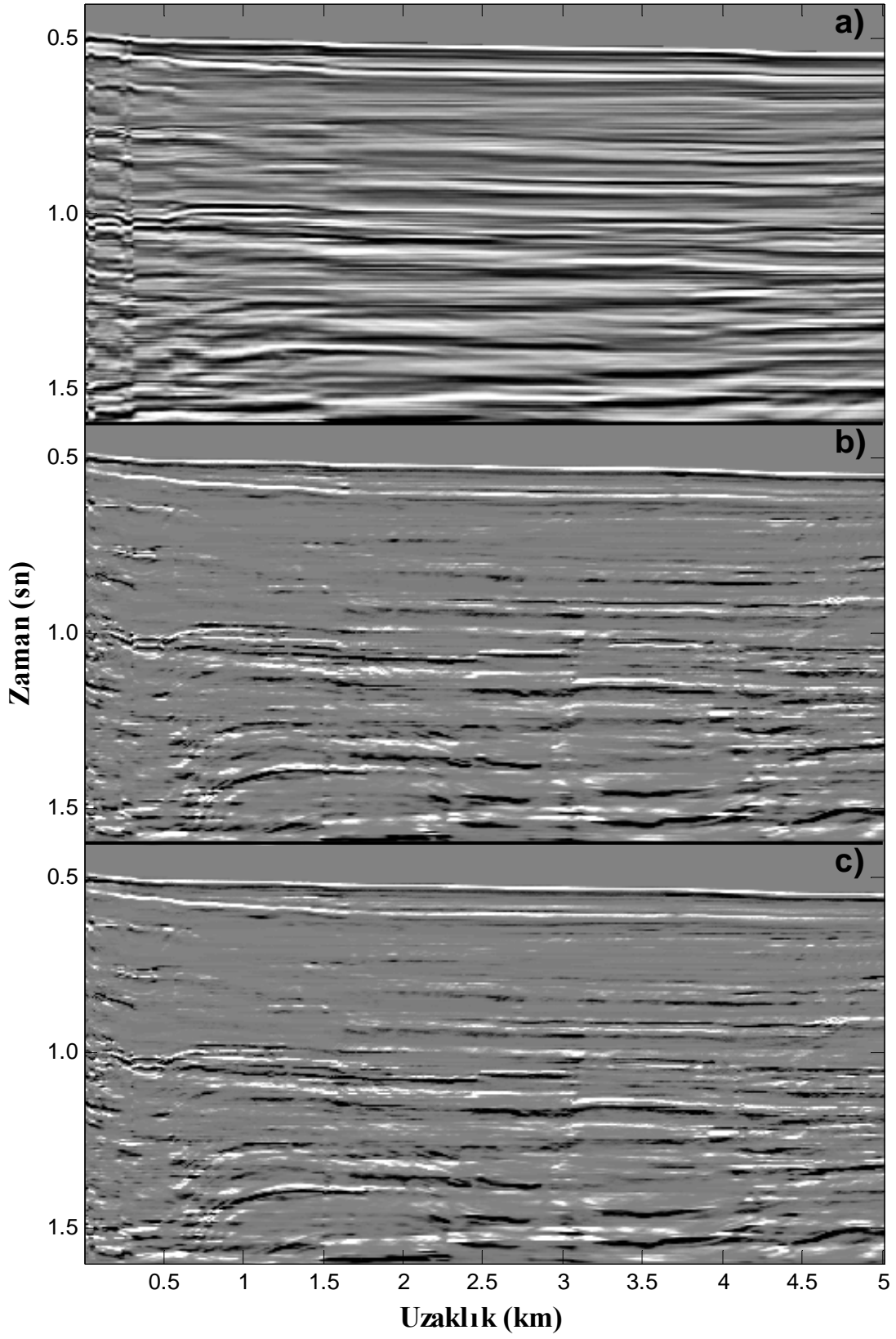
Şekil 3. 92. Arazi verisi-1 için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



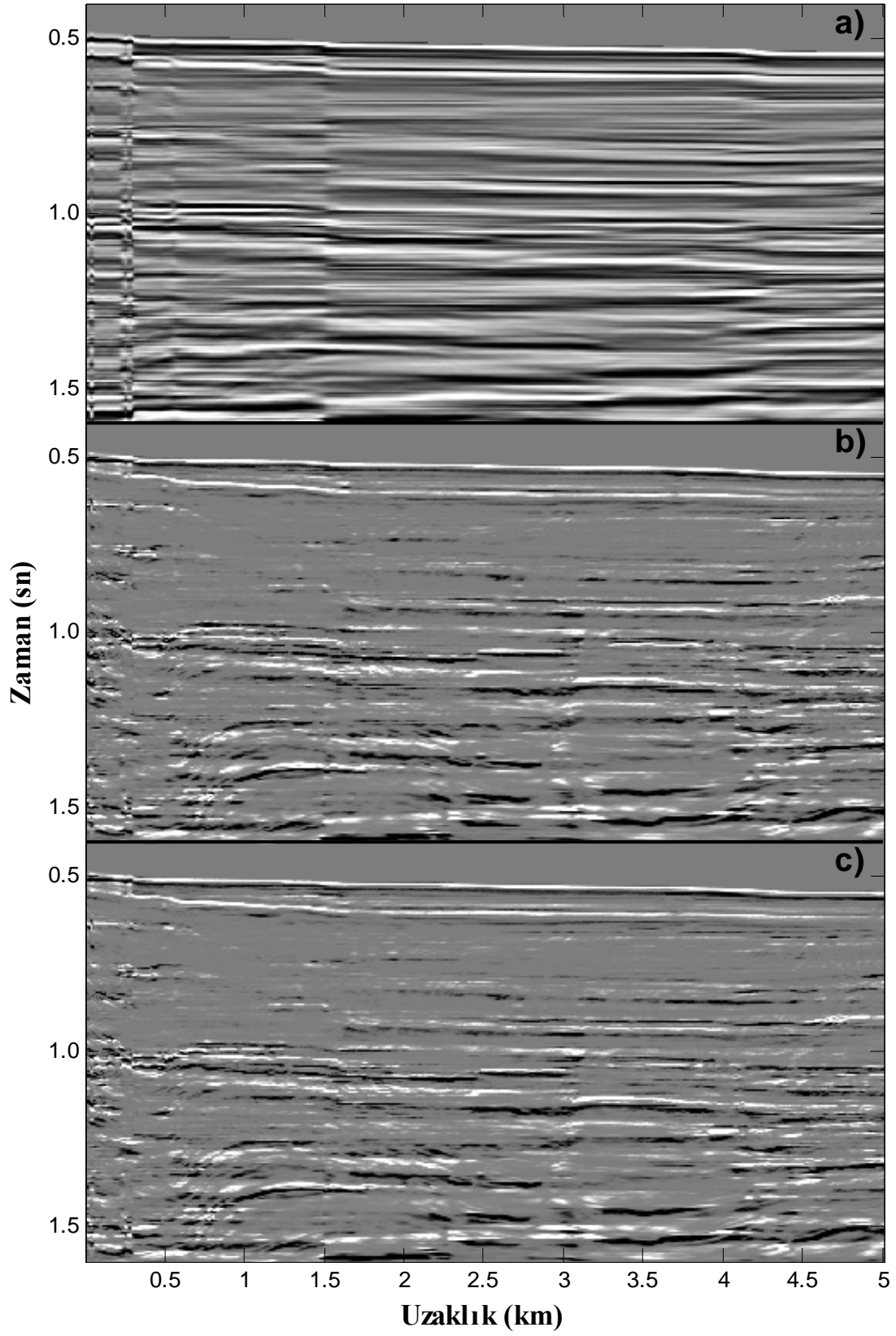
Şekil 3. 93. Arazi verisi-1 için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



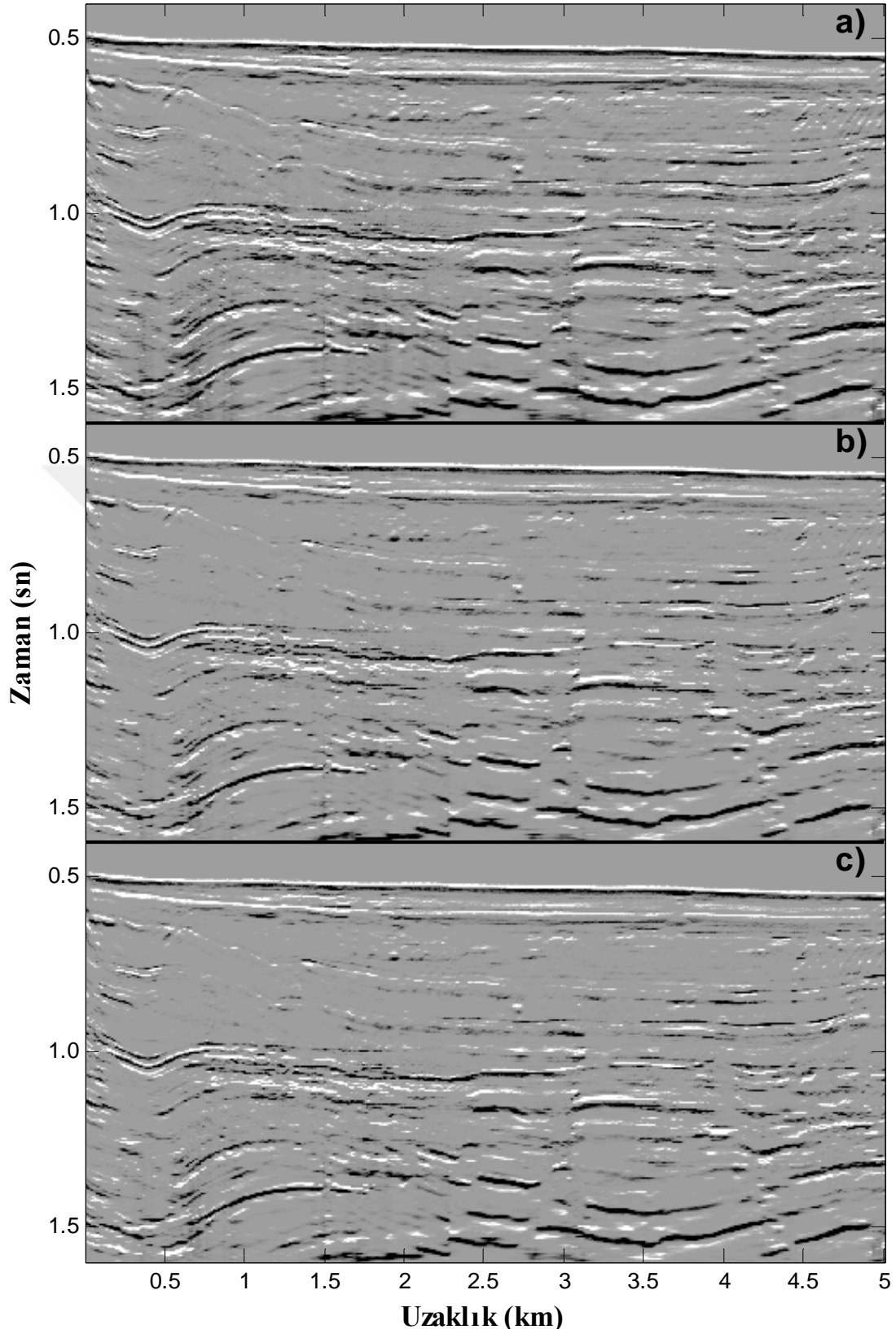
Şekil 3. 94. Arazi verisi-1 için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 95. Arazi verisi-1 için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 96. Arazi verisi-1 için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

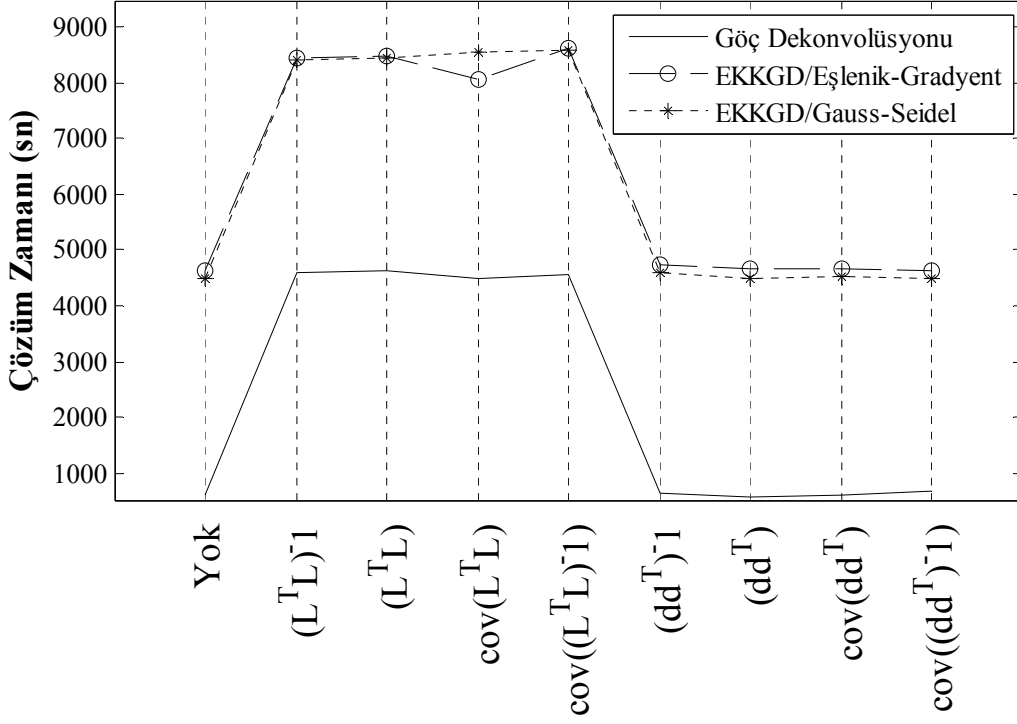


Şekil 3. 97. Arazi verisi-1 için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

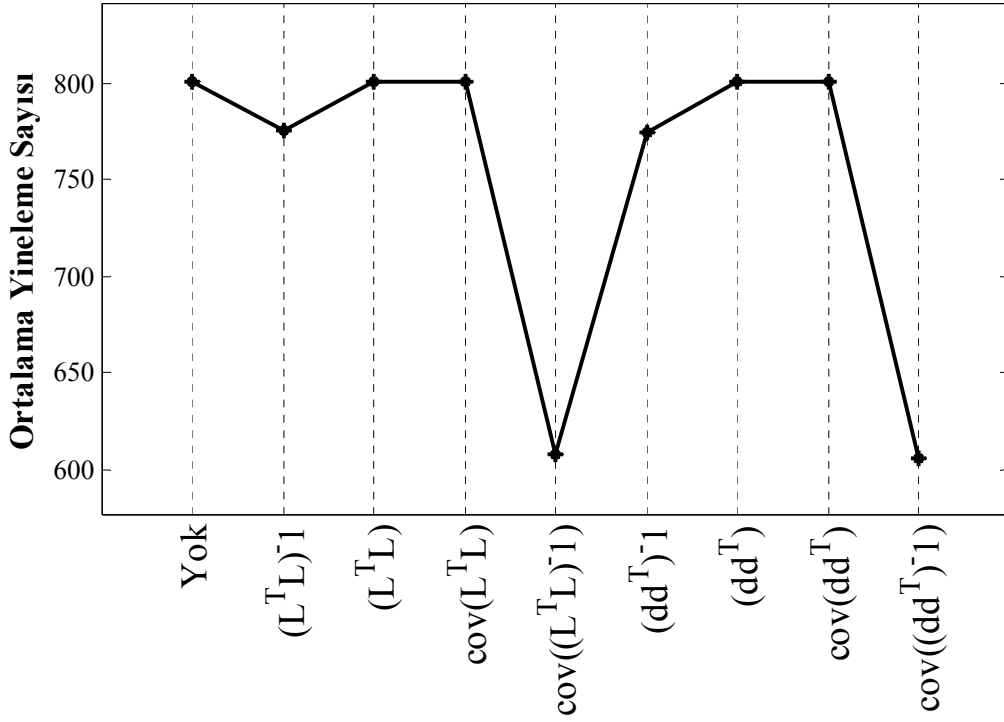
Şekil 3.98 Arazi verisi-1 için tüm yöntemlerin çözüm zamanlarını grafik üzerinde göstermektedir. Giriş verisi (d) kullanılarak elde edilen göç çıkışları, GD yaklaşık %87 daha az zamanda, EKKGD için yaklaşık %47 daha az zamanda çözüme ulaşmışlardır. Şekil 3.99 de Arazi verisi-1 için eşlenik-gradyent yönteminin ortalama yineleme sayılarını göstermektedir. Grafikte görüldüğü üzere $(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları yineleme sayısında bir azalma sağlayabilmişlerdir. Fakat $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları bu konuda en başarılı fonksiyonlar olduğu görülmektedir.

Tablo 3. 10. Arazi verisi-1 için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 801, zaman örnek sayısı: 801, yarı operatör boyu: 468.75 m, 75 iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD (sn)	EKKGD Eşlenik- Gradyent (sn)	EKKGD Gauss- Seidel (sn)	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	603,17	4620,71	4499,95	En Düşük Maliyet Orta Kalite
$(L^T L)^{-1}$	4605,88	8446,44	8418,49	Yüksek Maliyet Yüksek Kalite
$(L^T L)$	4617,30	8488,80	8464,23	Yüksek Maliyet Düşük Kalite
$cov(L^T L)$	4489,77	8064,57	8548,47	Yüksek Maliyet Düşük Kalite
$cov((L^T L)^{-1})$	4573,24	8619,56	8587,56	Yüksek Maliyet Yüksek Kalite
$(dd^T)^{-1}$	649,83	4720,79	4606,90	Hızlı Yüksek Kalite
(dd^T)	574,20	4659,87	4502,75	Hızlı Kusurlu
$cov(dd^T)$	592,70	4673,55	4522,11	Hızlı Kusurlu
$cov((dd^T)^{-1})$	663,51	4643,88	4491,06	Hızlı Yüksek Kalite



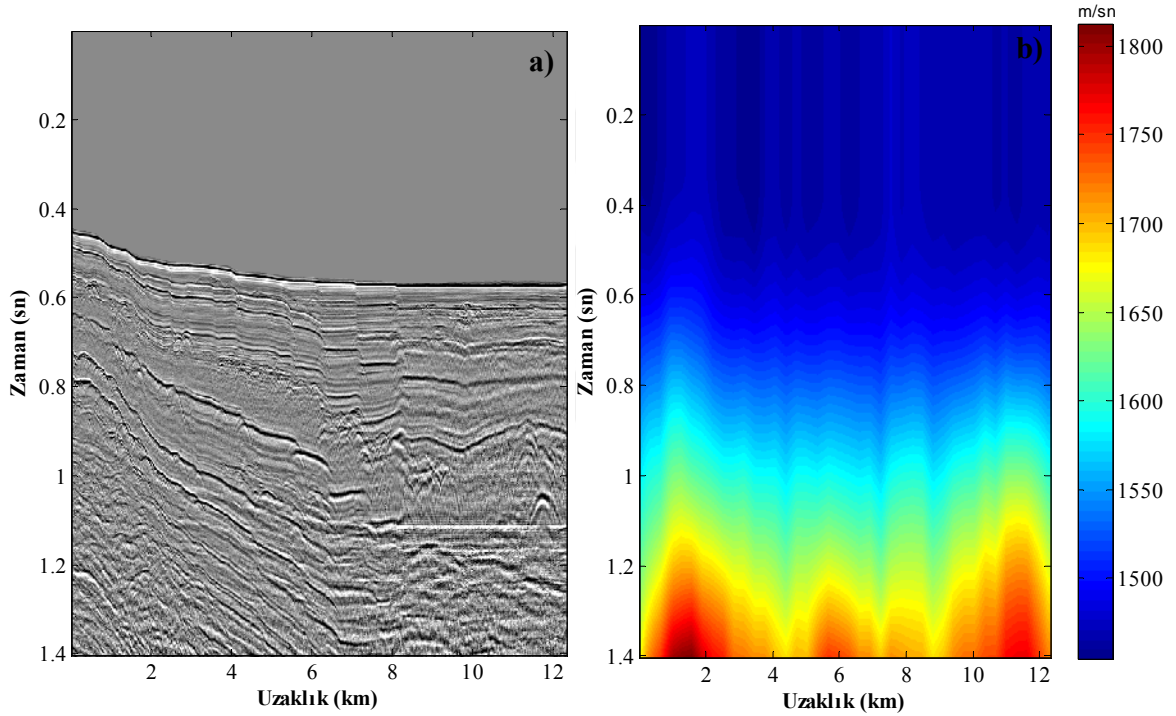
Şekil 3. 98. Arazi verisi-1 için uygulanan göç yöntemlerinin çözüm zamanları.



Şekil 3. 99. Arazi verisi-1 için uygulanan eşlenik-gradyent yönteminin ortalama yineleme sayıları.

3.2.2. Arazi Verisi-2 üzerinde Göç Uygulamaları

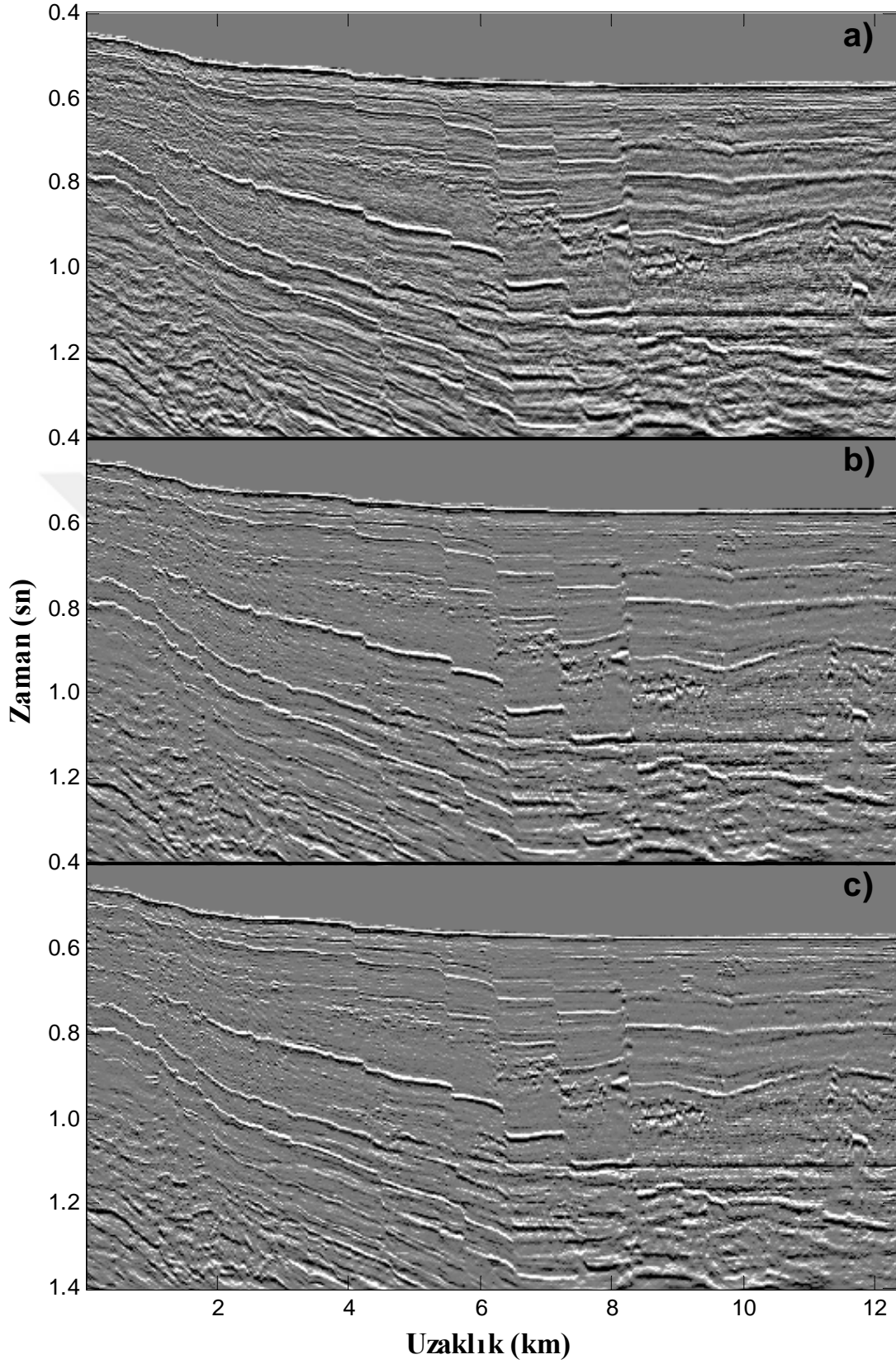
Arazi verisi-2 (Dokuz Eylül Üniversitesi Deniz Bilimleri ve Teknolojileri Enstitüsü tarafından toplanmıştır) çok sayıda fay ve saçılma hiperbollerine sahip bir deniz verisidir (Şekil 3.100a). Hız analizi sonucu elde edilen hız kesiti Şekil 3.100b' de gösterilmektedir. Tablo-1 de gösterilen tüm göç algoritmaları yarı operatör boyu 625 m kullanılarak uygulanmış ve toplam 27 farklı göç çıkışı elde edilmiştir. Göç çıkışlarına istenmeyen üst alanların atılması (top mute) ve otomatik genlik kontrolü (Automatic Gain Control, AGC, 0.35 sn) işlemleri uygulanmıştır.



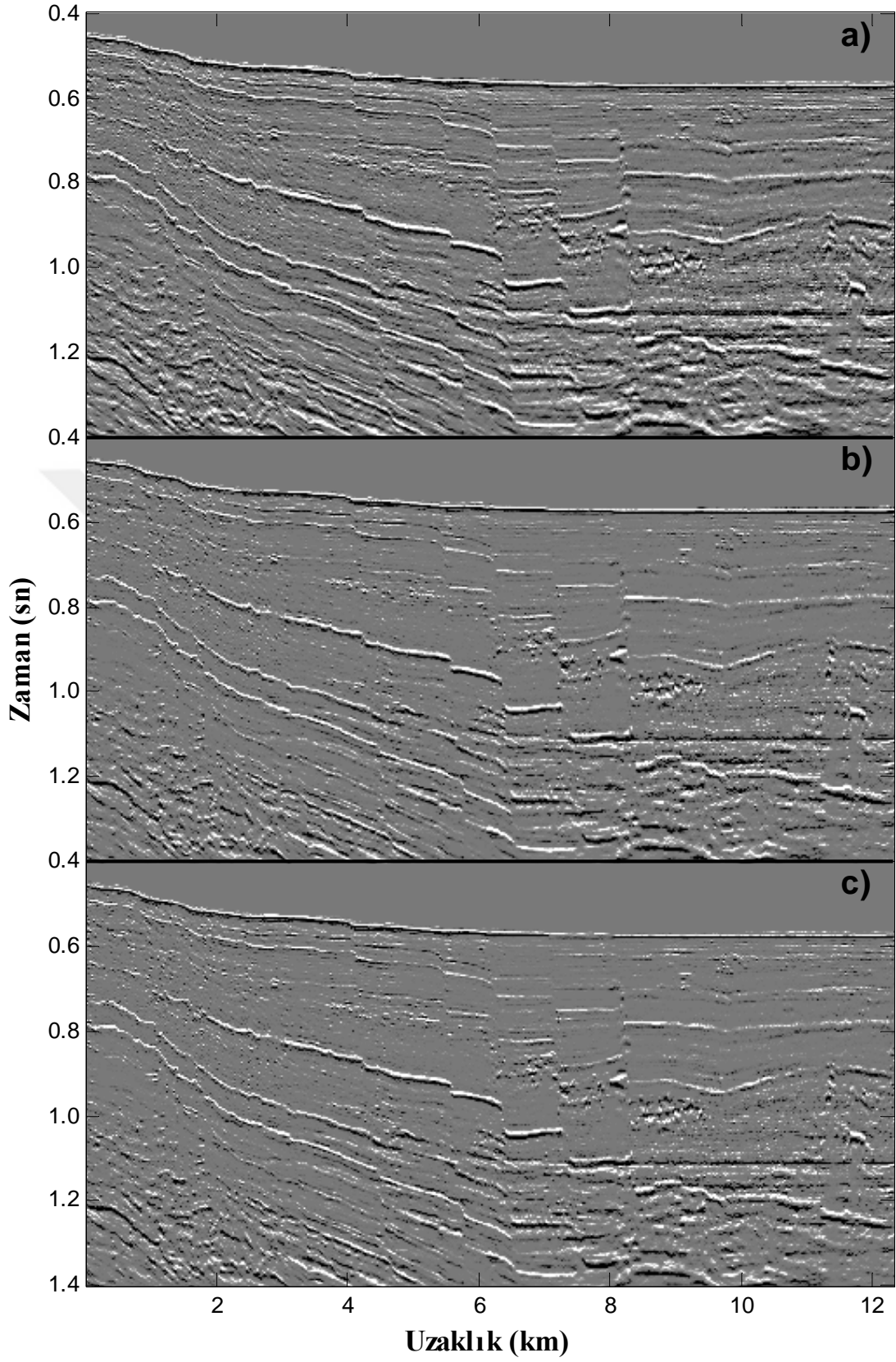
Şekil 3. 100. a) Arazi verisi-2 sıfır ofset kesiti, b) karekök ortalama hız (rms) hız kesiti.

Tablo 3.11 Arazi verisi-2 için tüm göç yöntemlerinin çözüm zamanlarını göstermektedir. Şekil 3.101' de Arazi verisi-1 için Kirchhoff göç çıkışı, eşlenik-gradyent yöntemi ile EKKG çıkışı ve gauss-seidel yöntemi ile EKKG çıkışları gösterilmektedir. EKKG yönteminin Kirchhoff göçüne göre üstünlüğü bu veri seti için de açıkça görülmektedir. Şekil 3.102 ile Şekil 3.109 arasında sırasıyla Arazi verisi-2 için $(L^T L)^{-1}$, $(L^T L)$, $cov(L^T L)$, $cov((L^T L)^{-1})$, $(dd^T)^{-1}$, (dd^T) , $cov(dd^T)$, $cov((dd^T)^{-1})$ netleştirme fonksiyonları ile GD, eşlenik-gradyent yöntemi ile EKKGD ve gauss-seidel yöntemi ile EKKGD çıkışları görülmektedir. $(L^T L)^{-1}$ ile elde edilen EKKGD çıkışları (Şekil 3.102b,c)

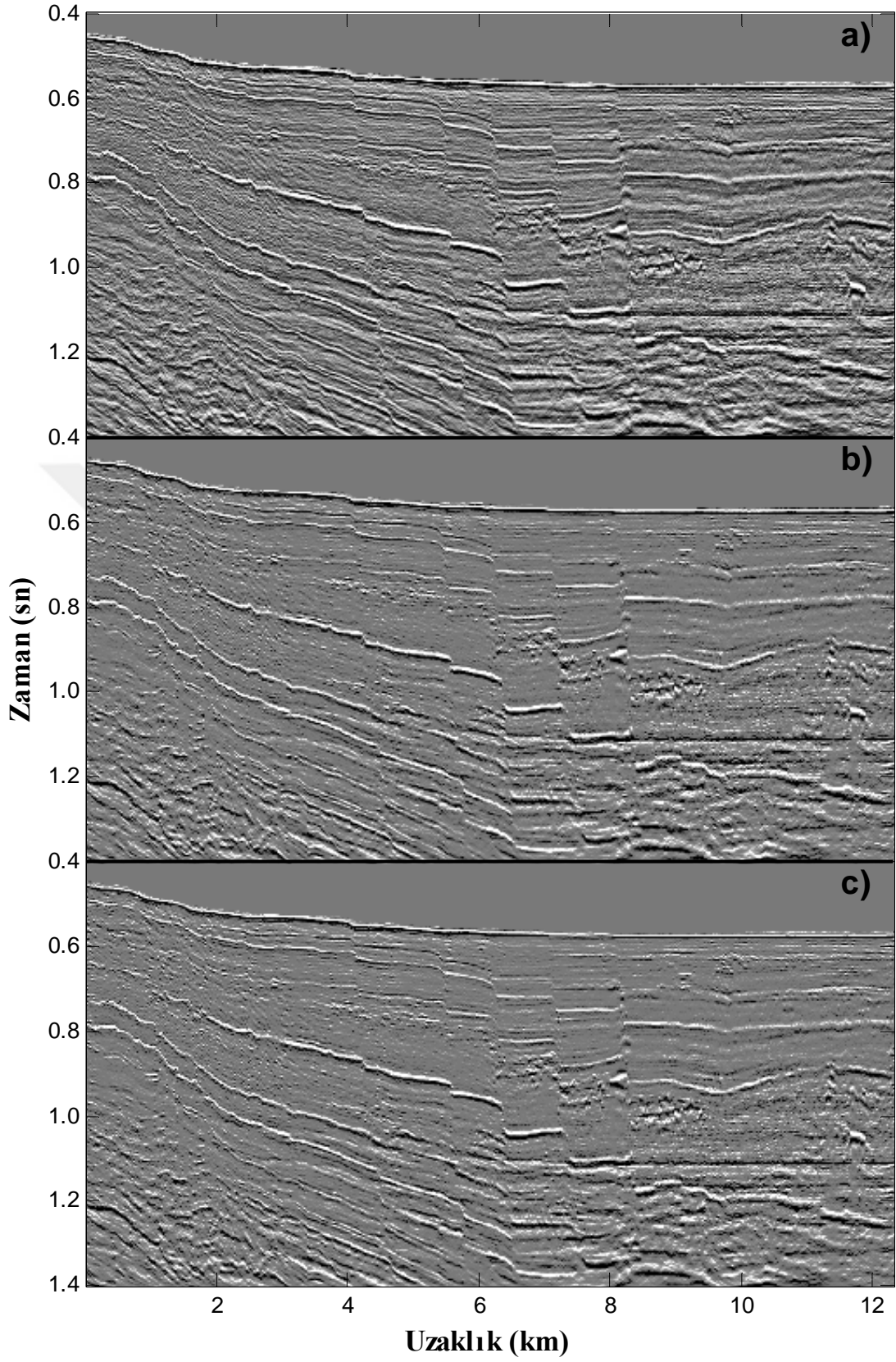
GD çıkışına (Şekil 3.66a) göre daha temiz sonuçlar olduğu açıkça görülmektedir. Şekil 3.103' e bakıldığında $(L^T L)$ netleştirme fonksiyonu ile elde edilen EKKGD çıkışları kötü kalitede olmamasına rağmen GD çıkışı Kirchhoff göç çıkışına oranla bir iyileşme sağlayamadığı açıkça görülmektedir. Şekil 3.104' de $cov(L^T L)$ netleştirme fonksiyonu kullanılarak elde edilen tüm göç çıkışlarında bloklaşma (veya gölge etkisi) açıkça görülmektedir. Bu gölge etkisi kısmen $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile elde edilen göç çıkışlarında da görülmektedir (Şekil 3.105). Fakat, $cov((L^T L)^{-1})$ netleştirme fonksiyonu çıkış kesitinin ayrımlılığını arttırmıştır. Şekil 3.106 $(dd^T)^{-1}$ netleştirme fonksiyonu ile elde edilen göç çıkışlarını göstermektedir. Diğer tüm veri setlerinde, geleneksel netleştirme fonksiyonu ile yaklaşık aynı kalitede göç çıkışları üreten $(dd^T)^{-1}$ netleştirme fonksiyonu arazi verisi-2 için de yaklaşık aynı kalitede sonuçlar ürettiği görülebilmektedir. (dd^T) netleştirme fonksiyonu önceki veri setlerinde olduğu gibi yalnızca yatay olayların genliklerine odaklayarak kusurlu çıkışlar (Şekil 3.107) üretmiştir. $cov(dd^T)$ netleştirme fonksiyonu (dd^T) netleştirme fonksiyonu gibi yatay olayların genliklerine odaklayarak kusurlu sonuçlar üretmiştir (Şekil 3.108). Şekil 3.109 da oklar ve kesikli çizgiler ile gösterilen alanlara bakıldığında $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile benzer göç çıkışlarının üretildiği açıkça görülebilir.



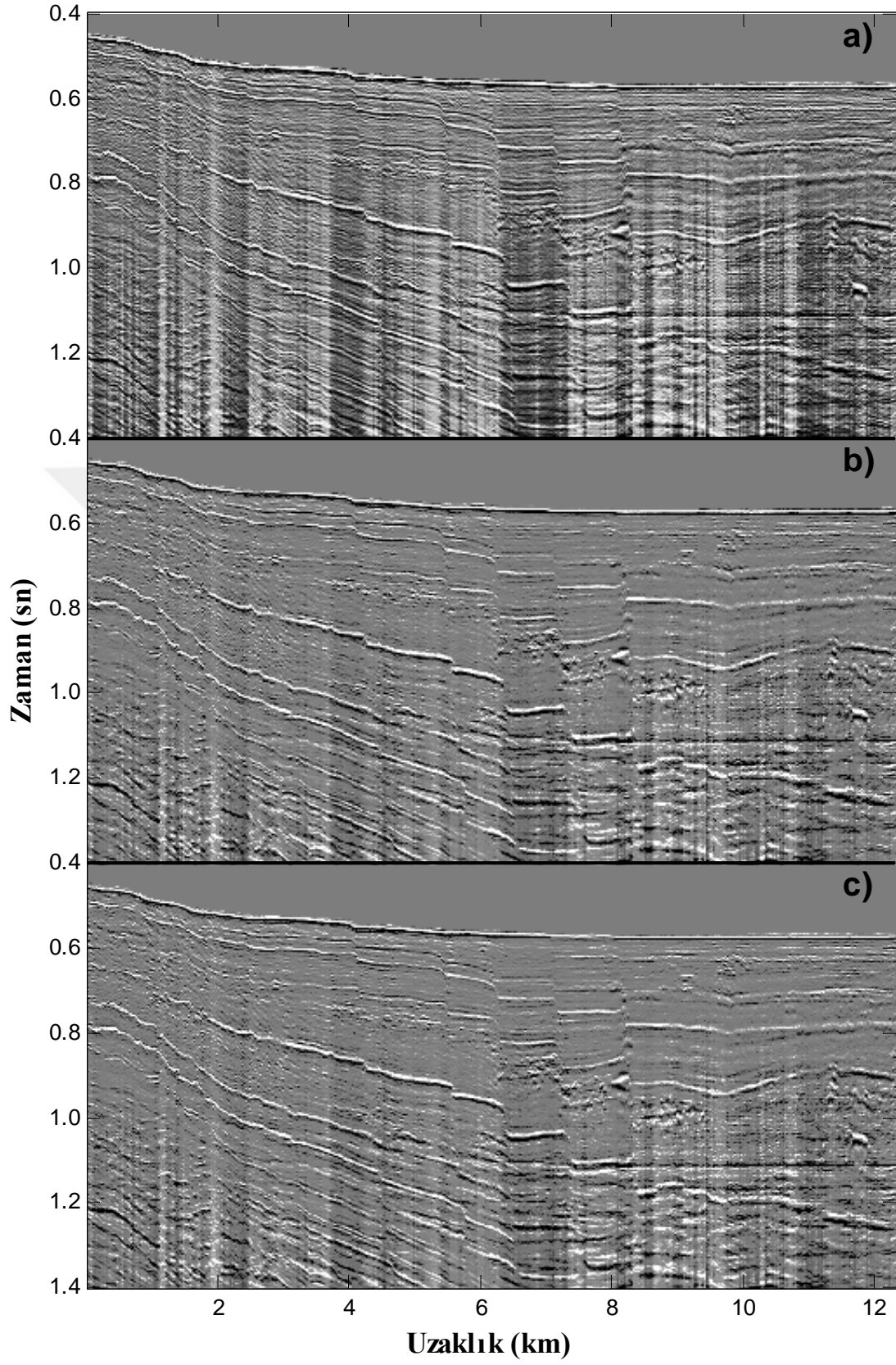
Şekil 3. 101. Arazi verisi-2 için a) Kirchhoff göç, b) eşlenik-gradyent yöntemi ile EKKG, c) gauss-seidel yöntemi ile EKKG çıktıları.



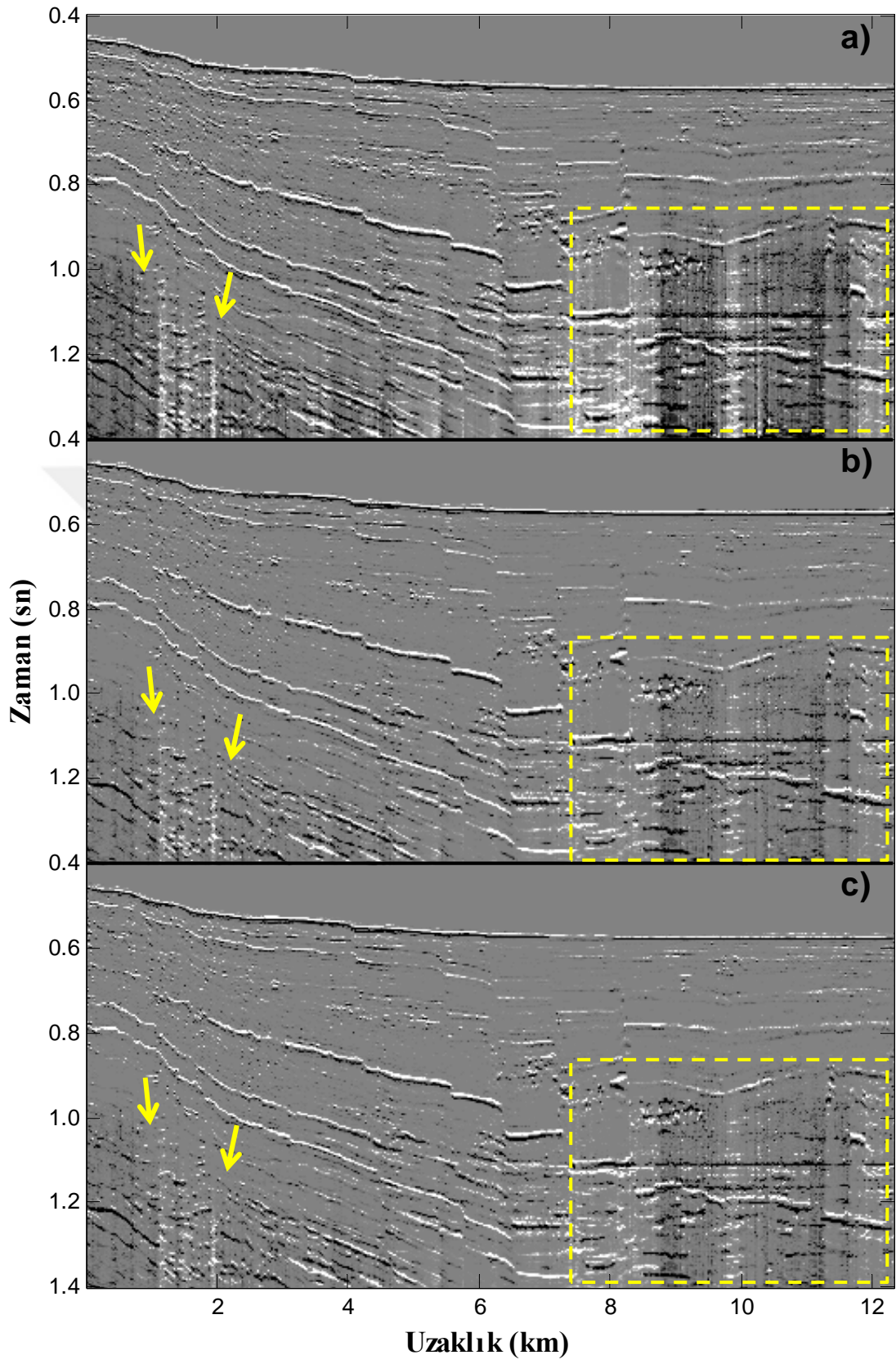
Şekil 3. 102. Arazi verisi-2 için $(L^T L)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



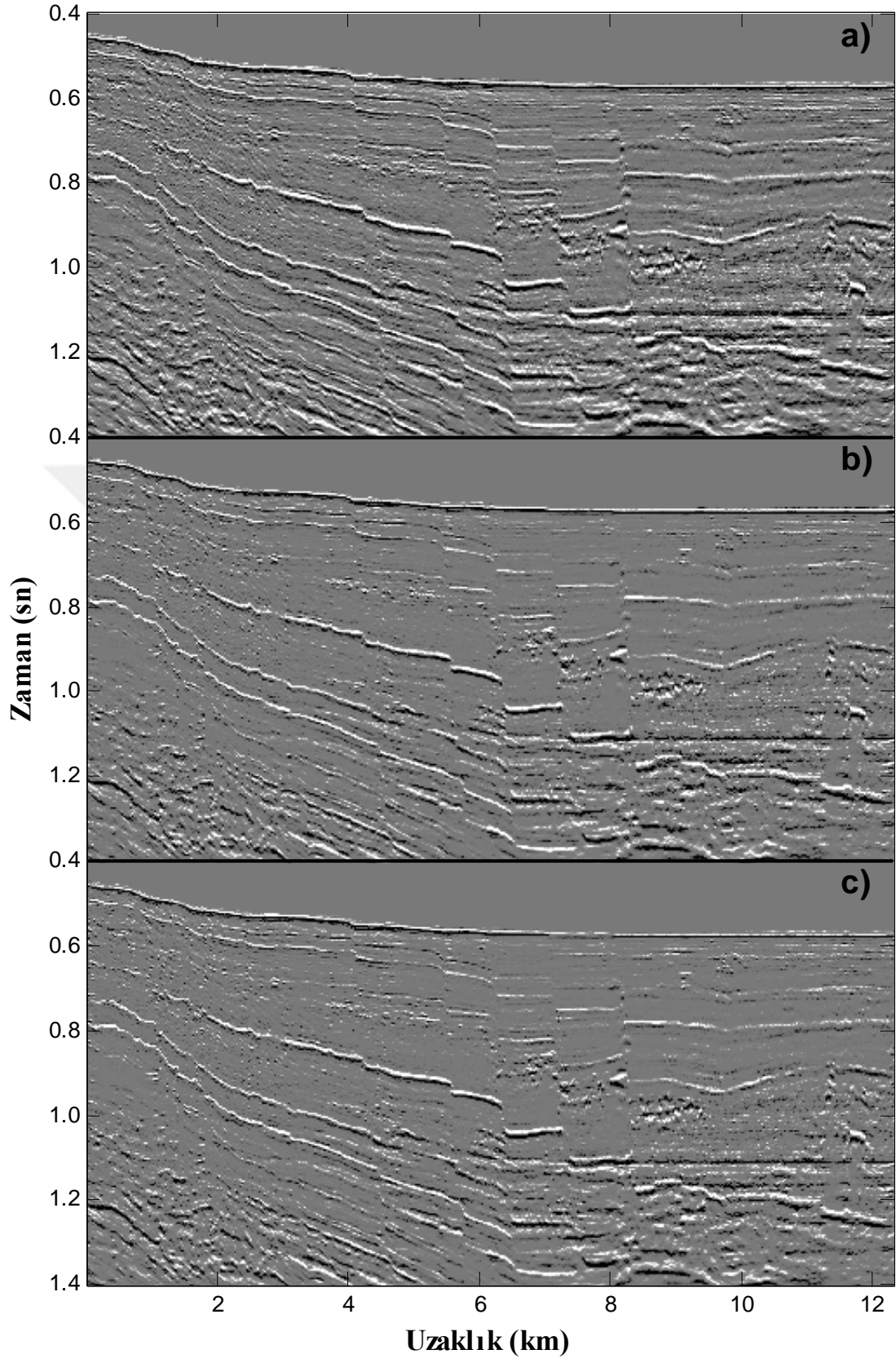
Şekil 3. 103. Arazi verisi-2 için ($L^T L$) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



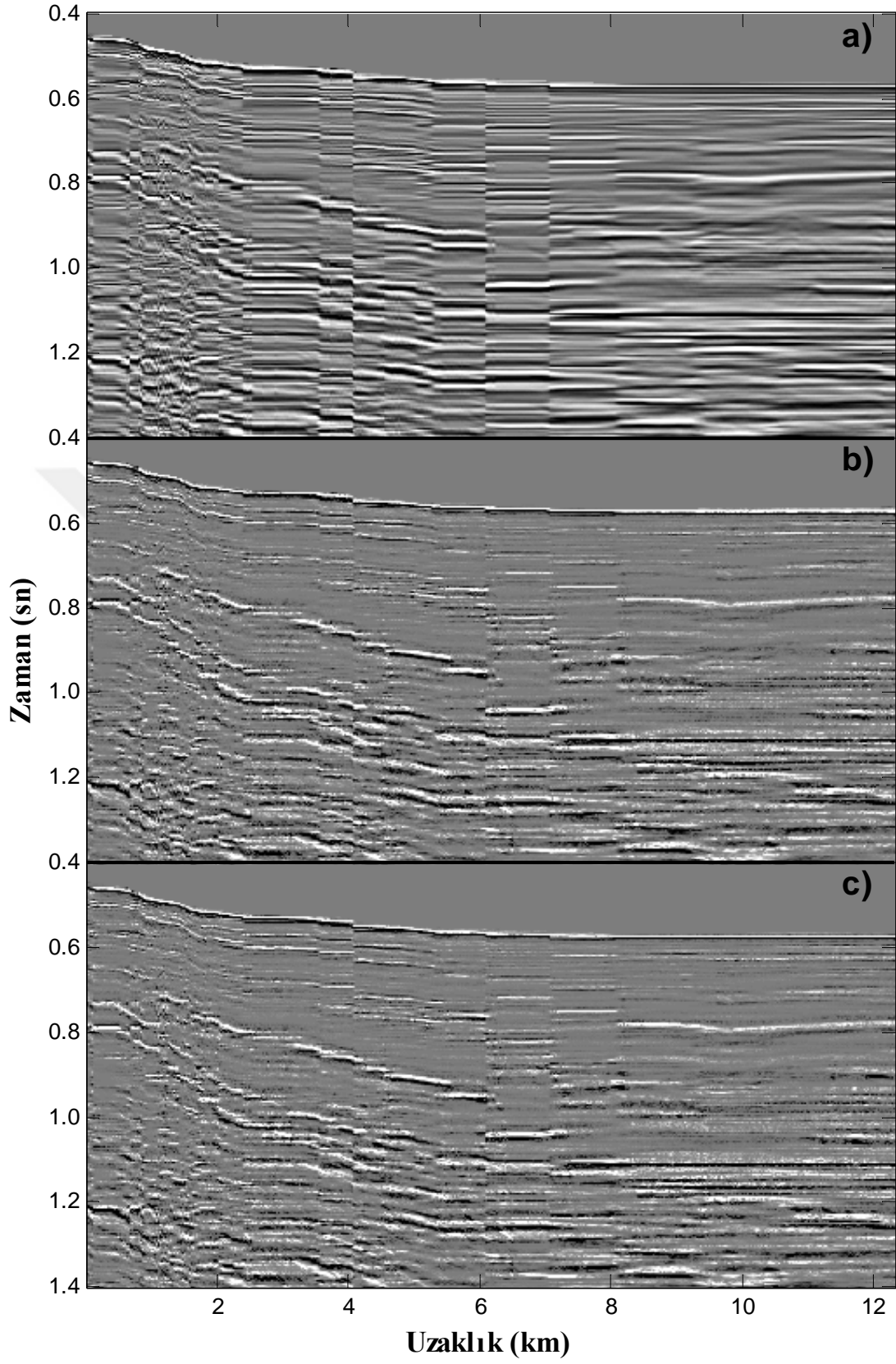
Şekil 3. 104. Arazi verisi-2 için $cov(L^T L)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



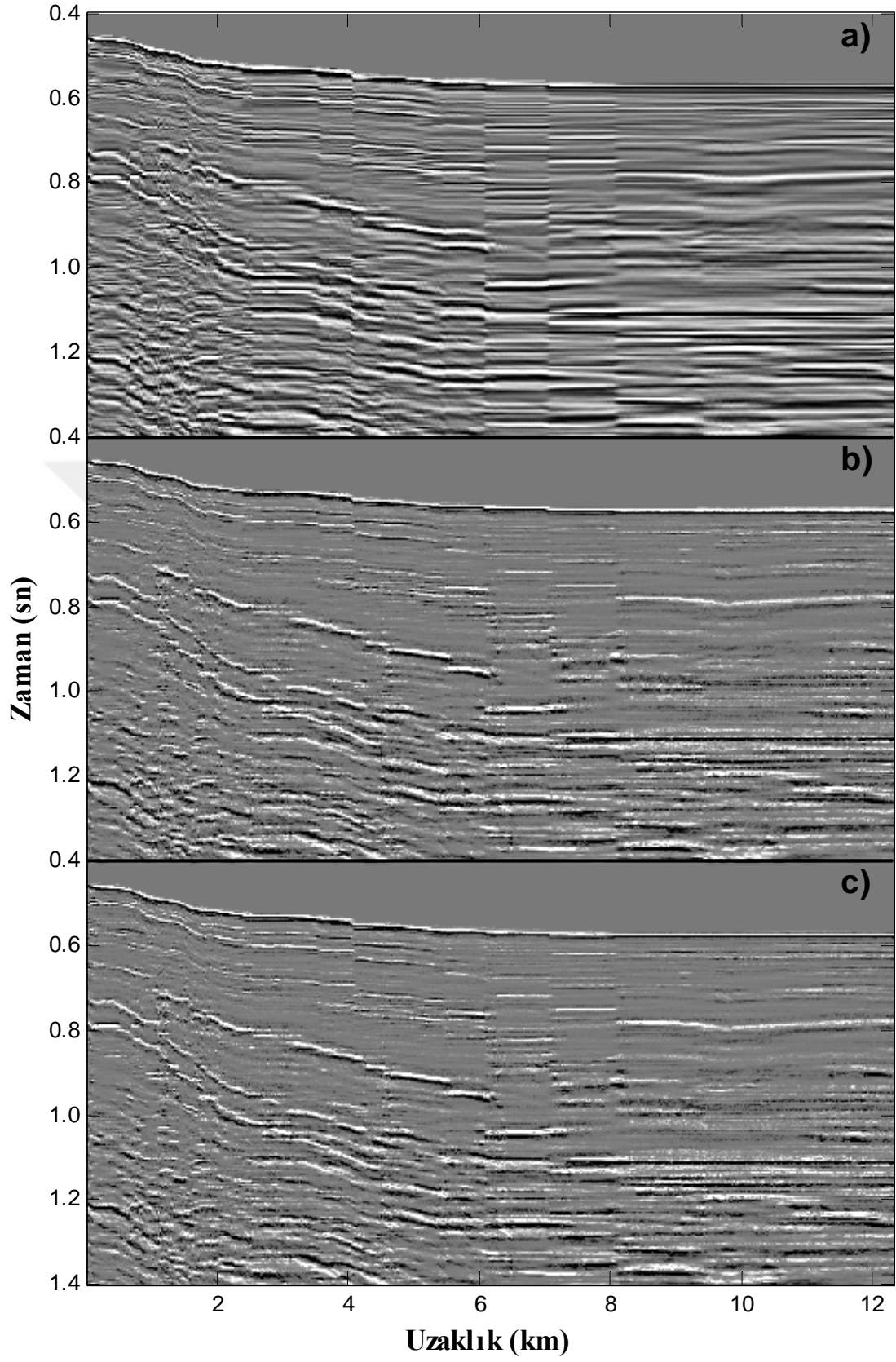
Şekil 3. 105. Arazi verisi-2 için $cov((L^T L)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



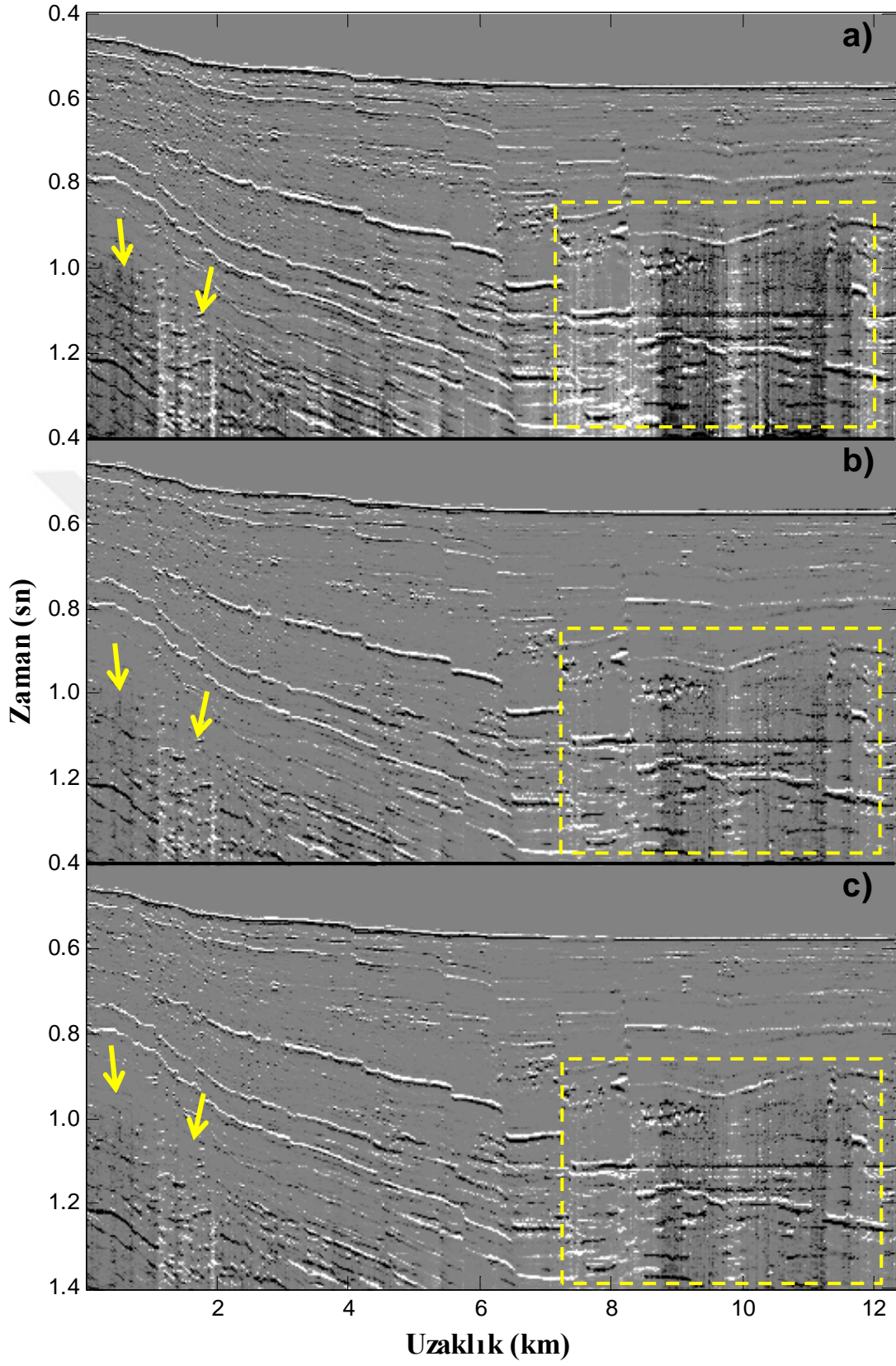
Şekil 3. 106. Arazi verisi-2 için $(dd^T)^{-1}$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 107. Arazi verisi-2 için (dd^T) netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.



Şekil 3. 108. Arazi verisi-2 için $cov(dd^T)$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

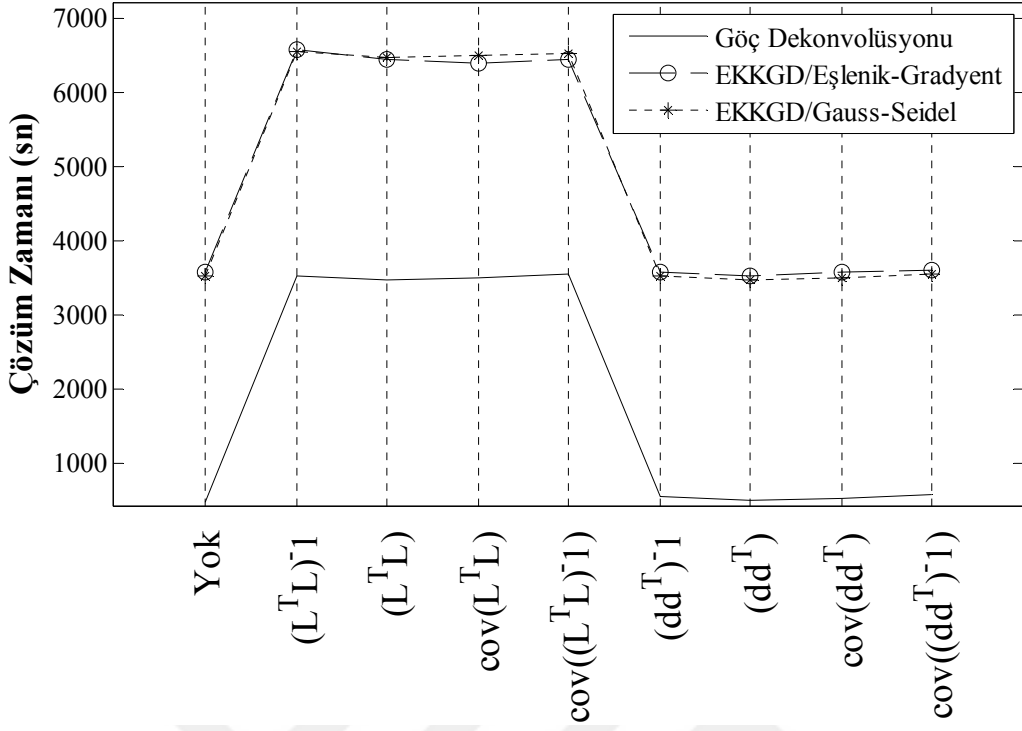


Şekil 3. 109. Arazi verisi-2 için $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile a) GD, b) eşlenik-gradyent yöntemi ile EKKGD, c) gauss-seidel yöntemi ile EKKGD çıktıları.

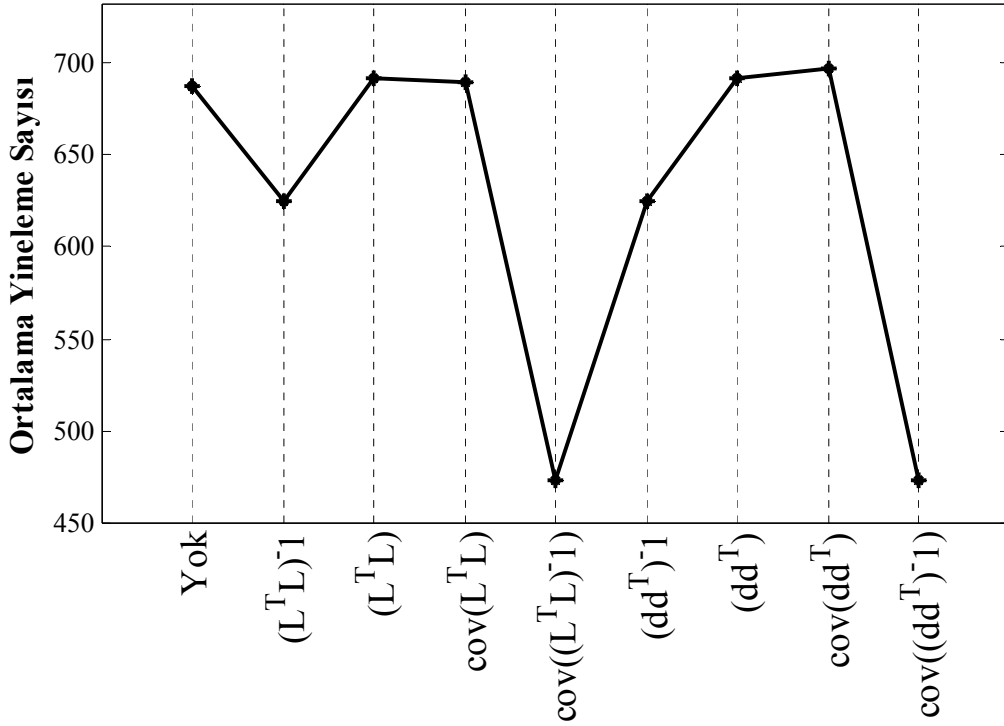
Tablo 3. 11. Arazi verisi-2 için Netleştirme fonksiyonlarına göre çözüm zamanları. (İz sayısı: 986, zaman örnek sayısı: 701, yarı operatör boyu: 625 m, 50iz)

Çözüm Zamanı t_s (sn)/ Netleştirme Fonksiyonu	GD (sn)	EKKGD Eşlenik- Gradyent (sn)	EKKGD Gauss- Seidel (sn)	Netleştirme Fonk. Değerlendirme
Yok Kirchhoff göçü/EKKG	467,23	3570,06	3534,60	En Düşük Maliyet Orta Kalite
$(L^T L)^{-1}$	3519,72	6578,06	6568,43	Yüksek Maliyet Yüksek Kalite
$(L^T L)$	3461,80	6449,01	6495,28	Yüksek Maliyet Düşük Kalite
$cov(L^T L)$	3486,12	6413,93	6506,35	Yüksek Maliyet Düşük Kalite
$cov((L^T L)^{-1})$	3549,65	6497,59	6536,19	Yüksek Maliyet Yüksek Kalite
$(dd^T)^{-1}$	552,47	3578,52	3533,66	Hızlı Yüksek Kalite
(dd^T)	488,69	3519,36	3462,62	Hızlı Kusurlu
$cov(dd^T)$	511,92	3582,69	3497,24	Hızlı Kusurlu
$cov((dd^T)^{-1})$	574,54	3616,30	3561,27	Hızlı Yüksek Kalite

Şekil 3.110 arazi verisi-2 için tüm yöntemlerin çözüm zamanlarını göstermektedir. Giriş verisi (d) kullanılarak elde edilen göç çıkışları, GD yaklaşık %85 daha az zamanda, EKKGD için yaklaşık %45 daha az zamanda çözüme ulaşmışlardır. Şekil 3.111 de arazi verisi-2 için eşlenik-gradyent yönteminin ortalama yineleme sayılarını göstermektedir. Grafikte görüldüğü üzere $(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov(L^T L)^{-1}$ netleştirme fonksiyonları arazi verisi-1 de olduğu gibi yineleme sayısında bir azalma sağlayabilmişlerdir. Fakat $cov((dd^T)^{-1})$ ve $cov(L^T L)^{-1}$ netleştirme fonksiyonları bu konuda tüm veri setleri için en başarılı fonksiyon olduğu görülmektedir.



Şekil 3. 110. Arazi verisi-2 için uygulanan göç yöntemlerinin çözüm zamanları.



Şekil 3. 111. Arazi verisi-2 için uygulanan eşlenik-gradyent yönteminin ortalama yineleme sayıları.

4. BULGULAR

4.1. Hız Fonksiyonunun Göç Algoritmalarına Etkisi

Göç işlemi için en değerli bilgi hız fonksiyonudur. Hız fonksiyonu doğru olmadığında, göç çıkışı yeraltı yapısını temsil edemez. Nokta saçıcı model için yapılan uygulamalarda hız fonksiyonunun doğru hız fonksiyonundan %5 den daha küçük veya büyük olması saçılma genliklerini saçıcı noktaya taşımakta zorluk çekmektedir. % 5 küçük hız fonksiyonu daha makul göç çıkışı üretirken, %5 büyük hız fonksiyonu daha hatalı göç çıkışı üretmiştir. Hata oranlarına bakıldığında, hız fonksiyonu doğru hıza (1000 m/sn) yaklaştıkça hata oranı düşmektedir. Yüksek hız fonksiyonları hata oranını, düşük hız fonksiyonlarına göre daha çok arttırmıştır. Benzer şekilde, papyon kravat modeli için hız fonksiyonunun doğru hız fonksiyonundan %5 den daha küçük olması durumunda papyon-kravat çözülememiştir. %5 den büyük olması durumunda ise papyon-kravat çözülmüş fakat senklinalin çukur noktası ovalleşmiştir. Benzer durumlar fay modeli için de geçerlidir, fakat fay modelinde yalnızca saçılmaların saçıcı noktaya taşınabildiği göç çıkışları üzerinden incelenerek yorum yapılabilmektedir.

4.2. Yarı Operatör Boyunun Göç Algoritmalarına Etkisi

Yarı operatör boyu, hız fonksiyonu ile hesaplanacak olan hiperbolün kanat uzunluğudur ve yığma kesitlerinin analizi ile belirlenmektedir. Saçılmaların, saçıcı noktaya taşınabilmesi için yığma kesiti üzerinde saçılmanın uzunluğu belirlenmesi gerekir. Nokta saçıcı model için operatör boyunun küçük seçilmesi, saçılma genliklerini saçıcı noktaya taşıyamamıştır. Hata oranlarına bakıldığında, operatör yarı boyunun olması gereken değerden (75 m) kısa olması durumunda hata oranları artış göstermektedir. Uzun olması durumunda ise hata oranları çok az artış göstermektedir. Bunun nedeni, operatör yarı boyunun olması gereken değerden daha uzun olduğunda hesaplanan operatörler farklı olayların genlikleri ile çakışmasıdır. Papyon kravat modeli için farklı yarı operatör boyları ile yapılan testler incelendiğinde; yarı boyunun kısa olması, eğimli olayların eğimlerini değiştirmez, fakat genlikleri tam anlamıyla gerçek yansıma noktasına taşımakta zorluk çektiği görülmektedir. Hata oranlarına bakıldığında operatör yarı boyunun papyon-kravat

modeli için senklinalin çukur noktasında oluşan hiperbole eşit olması durumunda en küçük hata değerine ulaştığını ve daha da büyük olması durumunda hata değerinin artmadığı görülmüştür. Bunun yanında tüm göç uygulamaları için operatör boyunun artması çözüm zamanını da arttırmıştır.

4.3. Beklenen Çıkışa En Fazla Yakınsamayı Sağlayan Göç Algoritmaları

Nokta saçıcı, papyon-kravat ve fay modelleri için hesaplanan hata oranları dikkate alındığında beklenen çıkışa en yakın sonuçları üreten netleştirme fonksiyonları $(L^T L)^{-1}$, (dd^T) , $(dd^T)^{-1}$, $cov(dd^T)$, $cov((dd^T)^{-1})$ ve $cov(L^T L)^{-1}$ dir. Bunun yanında geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$, giriş verisinden elde edilen diğer netleştirme $((dd^T)^{-1}$ ve $cov((dd^T)^{-1})$ fonksiyonlarına göre daha az hata üretmemiştir. Aynı zamanda $(dd^T)^{-1}$ netleştirme fonksiyonu geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile aynı hata oranlarına sahiptir, fakat çözüm zamanı olarak $(L^T L)^{-1}$ ' e göre GD için yaklaşık %65-75 oranında, EKKGD için %40-45 arasında bir iyileşme sağlamıştır. Buna göre geleneksel netleştirme fonksiyonu yerine $(dd^T)^{-1}$ netleştirme fonksiyonunu kullanmak işlem maliyetini düşürecektir. Tablo 4.1 faydalı sonuçlar ürettiği görülen netleştirme fonksiyonlarının nokta saçılma modeli, papyon-kravat modeli ve fay modeli için Kirchhoff göç çıkışına oranla hata paylarını göstermektedir. Bazı modeller için minimum hataları veren ancak kusurlu oldukları görülen (dd^T) ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları elendiğinde, aynı hata oranlarına sahip $cov((dd^T)^{-1})$ veya $cov((L^T L)^{-1})$ netleştirme fonksiyonları kullanılarak elde edilen EKKGD çıkışları en küçük hata oranlarını vermektedir (Tablo 4.1).

Tablo 4. 1. EKKG ve $(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonlarının kullanılması ile GD ve EKKGD çıkışı hatalarının Kirchhoff göç çıkışı hatalarına oranları. Kirchhoff göç çıkışı hatası 1 birim alınarak oranlar hesaplanmıştır.

Hata Oranı/Model	Nokta Saçılma	Papyon-Kravat	Fay
EKKG/Kirchhoff	0.53	0.10	0.16
EKKGD $(L^T L)^{-1}$ /Kirchhoff	0.33	0.04	0.12
GD $(L^T L)^{-1}$ /Kirchhoff	0.53	0.10	0.16
EKKGD $(dd^T)^{-1}$ /Kirchhoff	0.33	0.04	0.12
GD $(dd^T)^{-1}$ /Kirchhoff	0.53	0.10	0.16
EKKGD $cov((dd^T)^{-1})$ /Kirchhoff	0.28	0.03	0.10
GD $cov((dd^T)^{-1})$ /Kirchhoff	0.33	0.04	0.12
EKKGD $cov(L^T L)^{-1}$ /Kirchhoff	0.28	0.03	0.10
GD $cov(L^T L)^{-1}$ /Kirchhoff	0.33	0.04	0.12

Dikkat edilirse EKKG ve geleneksel netleştirme fonksiyonu ile uygulanan GD işlemi Kirchhoff göç çıkışına oranla aynı hata oranlarını (0.68, 0.41 ve 0.44) vermektedir (Tablo 4.1).

4.4. Netleştirme Fonksiyonlarının Yineleme Sayılarına Etkisi

Aoki ve Schuster (2009) EKKG işlemi için netleştirme fonksiyonu kullanarak yineleme sayılarında bir azalma sağlamıştır. Nokta saçıcı, papyon-kravat ve her iki arazi verisi (toplamda 4/6 veri seti) için uygulamalarda geleneksel netleştirme fonksiyonu yineleme sayılarında bir azalmaya neden olmuştur. Fakat gerek fay modeli, gerek tuz modeli için yineleme sayıları en yüksek yineleme sayılarına eşittir. Bunun yanında yineleme sayılarının azalması işlem zamanının kısalması beklentisini oluşturmaktadır. Fakat tüm model ve gerçek veriler için geleneksel netleştirme fonksiyonunun kullanılması işlem zamanını arttırmıştır. Doğal olarak fazladan hesaplanan netleştirme fonksiyonu yineleme sayısını azaltsa da işlem zamanını arttırdığı görülmektedir. Bununla birlikte tüm uygulamalar için yineleme sayılarını en fazla düşüren netleştirme fonksiyonları $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ dir.

4.5. Zaman Ekonomisi Sağlayan Fonksiyonlar

İşlem zamanı göç algoritmalarının önemli bir niteliğidir ve GD ile EKKG yöntemlerinin en önemli dezavantajıdır. İşlem zamanının ana etkenleri giriş verisi boyutu (iz sayısı ve zaman örnek sayısı) ve yarı operatör boyudur. Kirchhoff göçünün matris ortamında hesaplanması da fazladan maliyet gerektirir. Çünkü göç operatörü (L) matrisini hesaplamak çok zaman alıcı bir işlemdir. Tablo 4.2 tüm veri setleri için matris ortamında Kirchhoff göç ve Kirchhoff integral göçünün çözüm zamanları gösterilmektedir. Buna göre, Kirchhoff göçünün matris ortamında uygulanması en az 11 kat maliyetlidir. Bunun yanında, büyük boyutlu (L) matrisinin devriği ile çarpılması ve tersinin alınması işlemi maliyeti çok daha arttırmaktadır. Ayrıca, (L) matrisinin çokça sıfırdan oluşan bir matris olması çözüm zamanını arttırmaktadır. Bu çalışmada işlem zamanını kısaltmak için kullanılan farklı netleştirme fonksiyonları farklı işlem zamanlarında çözüme ulaşmışlardır. (L) matrisinin yerine, giriş verisinden (d) türetilen matrisin kullanılması çözüm zamanını düşürdüğü görülmektedir. Tablo 4.3 faydalı sonuçlar ürettiği görülen netleştirme

fonksiyonlarının tüm veri setleri için Kirchhoff göç çıkışına oranla çözüm zamanlarını (maliyetlerini) göstermektedir. Yapılan çalışmalarda EKKG işleminin GD işlemi ile yaklaşık aynı zamanlarda çözüme ulaştığı görülmektedir. Ayrıca geleneksel netleştirme fonksiyonu $((L^T L)^{-1})$ ve göç operatörü (L) kullanılarak üretilen diğer tüm netleştirme fonksiyonları çözüm maliyetini daha çok arttırmaktadır (Tablo 4.3).

Tablo 4. 2. Tüm veri setleri için Kirchhoff integral göçü, Matris ortamında Kirchhoff göçü ve maliyet oranları.

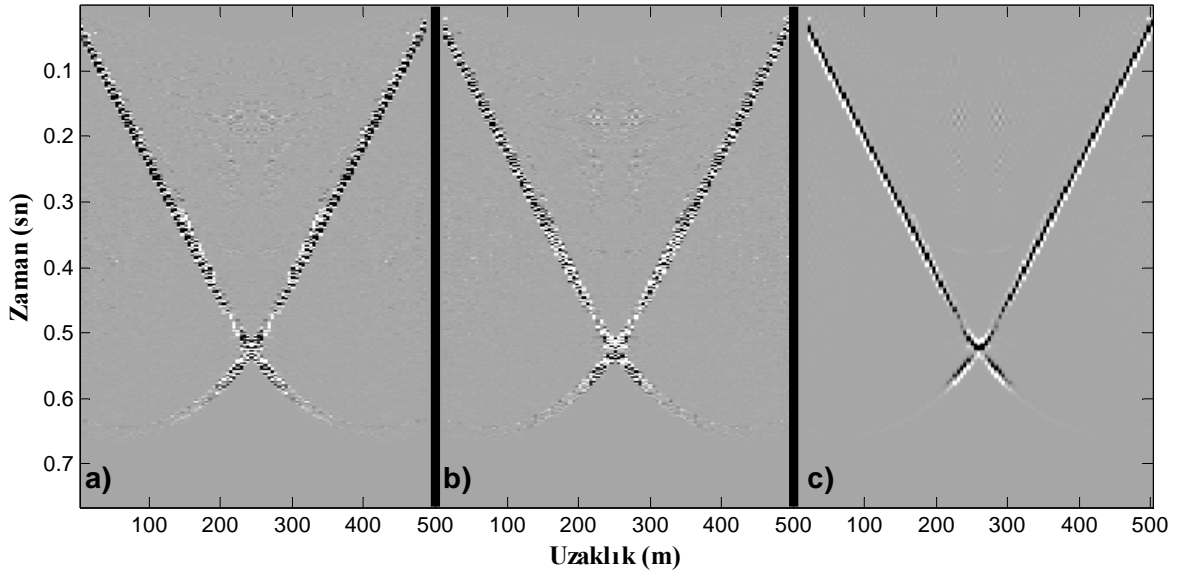
Model/Veri	Kirchhoff İntegral	Kirchhoff Matris $(L^T d)$	Kirchhoff Matris/ Kirchhoff İntegral
Nokta Saçıcı	0.098 sn	1.074 sn	10.9
Papyon-kravat	0.579 sn	38.599 sn	66.7
Fay	0.203 sn	2.732 sn	13.5
Tuz	2.409 sn	78.732 sn	32.7
Deniz verisi-1	5.443 sn	398.156 sn	73.2
Deniz verisi-2	7.082 sn	469.940 sn	66.6

Tablo 4. 3. EKKG ve $(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonlarının kullanılması ile GD ve EKKGD çıkışı çözüm zamanlarının Kirchhoff göç çıkışı çözüm zamanına oranları. Kirchhoff göç çıkışı toplam zamanı 1 birim alınarak diğer oranlar hesaplanmıştır.

Çözüm Zamanı Oranı/Model	Nokta Saçılma	Papyon-Kravat	Fay	Tuz	Arazi 1	Arazi 2	Ort
EKKG/Kirchhoff	3.68	7.69	3.89	4.16	7.46	7.56	5.74
EKKG/GD	0.99	0.99	1.01	1.08	0.97	1.00	1.00
EKKGD $(L^T L)^{-1}$ /Kirchhoff	6.35	11.29	7.02	7.24	13.96	14.06	8.82
GD $(L^T L)^{-1}$ /Kirchhoff	3.68	7.60	3.82	3.82	7.63	7.53	5.68
EKKGD $(dd^T)^{-1}$ /Kirchhoff	3.89	7.84	4.28	4.26	7.63	7.56	5.92
GD $(dd^T)^{-1}$ /Kirchhoff	1.30	1.17	1.58	1.15	1.07	1.18	1.07
EKKGD $cov((dd^T)^{-1})$ /Kirchhoff	4.10	7.77	4.49	4.26	7.44	7.62	5.95
GD $cov((dd^T)^{-1})$ /Kirchhoff	1.27	1.16	1.60	1.11	1.09	1.23	1.24
EKKGD $cov(L^T L)^{-1}$ /Kirchhoff	6.66	9.93	6.85	7.28	14.24	14.95	9.99
GD $cov(L^T L)^{-1}$ /Kirchhoff	3.80	7.74	3.98	4.14	7.58	7.60	5.80

4.6. EKKG ve GD Karşılaştırması

Yu and Schuster (2003) GD işleminin EKKG işlemine oranla daha az maliyetli olduğunu belirtmiştir. Her iki yöntemin algoritmalarına bakıldığında (Denklem 14, 19) ve netleştirme fonksiyonu $F = (L^T L)^{-1}$ olduğunda aynı parametreler ile aynı sonucu ürettikleri görülebilir. Ayrıca nokta saçıcı, papyon-kravat ve fay modellerinden elde edilen hata oranları (Tablo 4.1) incelendiğinde, bu iki yöntemin birebir aynı hata oranlarını yaklaşık aynı zamanlarda verdikleri görülür. EKKG yönteminde en fazla yineleme sayısı düşürülerek işlem zamanı %10-15 oranında azaltılabilir. Şekil 4.1 papyon-kravat modeli için farklı yineleme sayıları ile elde edilen göç çıkışlarını göstermektedir. Göç çıkışlarına bakıldığında en fazla yineleme sayısının küçük seçilmesi göç çıkışı kalitesini olumsuz etkilediği görülebilir.



Şekil 4. 1. Papyon-kravat modeli için eşlenik-gradyent yöntemi ile, a) en fazla yineleme sayısı 20, b) en fazla yineleme sayısı 30 ve c) en fazla yineleme sayısı 766, ortalama yineleme sayısı 540 olan EKKG çıkışları.

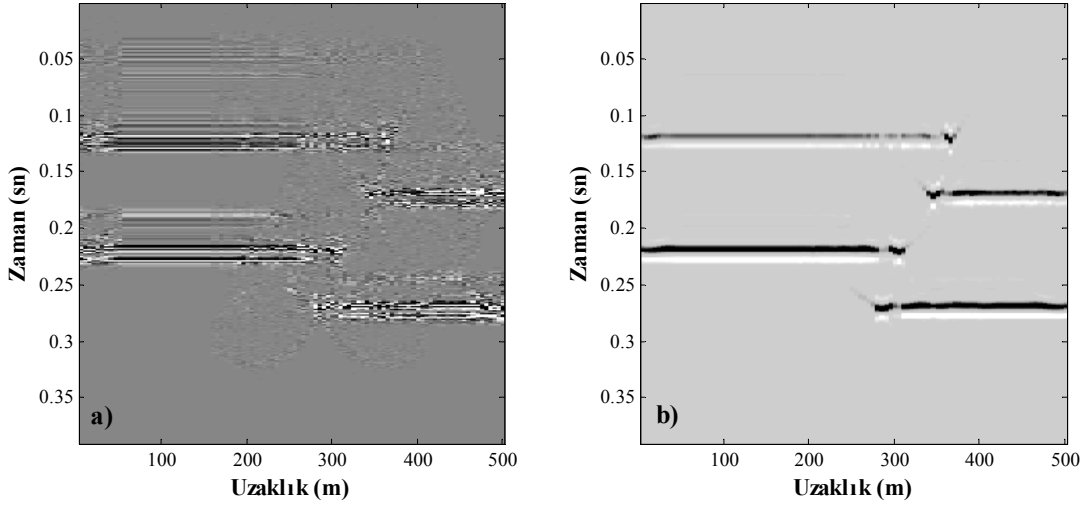
4.7. Netleştirme Fonksiyonlarını Kullanmanın Etkisi

EKKG yöntemini netleştirme fonksiyonları ile kullanmak hata oranını azalttığı gözlenmiştir (Tablo 4.1). Fakat bazı durumlarda hata oranlarına göre değerlendirmeler yanıltıcı olabilir. Örneğin (dd^T) ve $cov(dd^T)$ netleştirme fonksiyonları kullanılarak elde edilen EKKGD çıkışları çok düşük hata oranına sahiptir. Bunun asıl sebebi bu netleştirme

fonksiyonlarının göç hatalarını minimize etmesidir. Ancak bunun yanında çıkış kesitleri incelendiğinde yatay ve yüksek genlikli olaylara bir odaklanma olduğu görülebilir. Nokta saçılma modeli tek bir noktada odaklandığından ve fay modeli yatay yönde olaylar olduğundan dolayı bu modeller için en küçük hata değerlerine sahip netleştirme fonksiyonları olmuşlardır. Diğer yandan, $(L^T L)^{-1}$ ve $(dd^T)^{-1}$ netleştirme fonksiyonları hata oranını GD için yaklaşık %73 oranında, EKKGD için yaklaşık %83 oranında bir iyileşme sağlarken, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları ise hata oranını GD için yaklaşık olarak %83, EKKGD için %86 oranında düşürdüğü görülmektedir. GD için netleştirme fonksiyonları kullanımının EKKG' üne oranla daha etkili olduğu görülmektedir. Hata oranlarında, GD ile %73-83 bir iyileşme sağlanırken, EKKGD işlemi ile %83-86' lık bir iyileşme sağlanmıştır. $(L^T L)^{-1}$ veya $(dd^T)^{-1}$ netleştirme fonksiyonları kullanılarak elde edilen EKKGD çıkışları hata oranları bakımından $cov((dd^T)^{-1})$ veya $cov((L^T L)^{-1})$ netleştirme fonksiyonları ile elde edilen GD çıkışları ile benzerlik göstermektedirler.

4.8. EKKG ve EKKGD Yöntemleri için Yinelemeli (Eşlenik-Gradyent ve Gauss-Seidel) Çözüm Algoritmalarının Etkisi

Bu çalışmanın başlangıcında gauss-seidel matris formu eşlenik-gradyent yöntemine göre yaklaşık aynı zamanlarda daha kaliteli sonuçlar verdiği için her iki yöntemde çalışmanın sonuna kadar kullanılmıştır. Bunun sebebi, eşlenik-gradyent yöntemi EKKG yöntemleri için en çok kullanılan yöntem olmasıdır. Çalışmanın başlangıcında gauss-seidel yönteminin daha kaliteli sonuçlar vermesinin sebebi tamamen eşlenik-gradyent yönteminin sonlandırma kriterine (ϵ) bağlı olduğu fark edilmiştir. Çalışmanın başlangıcında sonlandırma kriteri epsilon değeri ($\epsilon = 2.2204e - 16$) olarak belirlenmiştir, fakat çok küçük bir değer olmasına rağmen gauss-seidel matris formu yöntemine göre kalitesiz çıkışlar üretmiştir. Şekil 4.2 fay modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonunun kullanıldığı ve eşlenik-gradyent yöntemi için sonlandırma parametresinin farklı kullanımı durumunda elde edilen çıkışları göstermektedir. Daha sonrasında eşlenik-gradyent sonlandırma kriteri ($\epsilon = 4.9304e - 32$) olarak kullanılmış ve çözümlerin kalitesi gauss-seidel yöntemi ile birbirine eşitlenmiştir. Fakat sonlandırma kriterinin çok daha küçük seçilmesi ($\epsilon = 4.9304e - 32$) eşlenik-gradyent yönteminin çözüm zamanını arttırmıştır. Böylece, gauss-seidel matris formu çözümü daha kısa zamanlarda çözüme ulaşmıştır.



Şekil 4. 2. Fay modeli için $cov((dd^T)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile (a) sonlandırma kriteri $\varepsilon = 2.2204e - 16$ için (b) sonlandırma kriteri $\varepsilon = 4.9304e - 32$ için elde edilen geç çıkışları.

4.9. Düzenleme (Regularization) Parametresinin Etkisi

Bilinmektedir ki, bir matrisin tersinin alınabilmesi için iyi koşullandırılmış olması gerekir. Eğer matris iyi koşullu değil ise sağlıklı bir tersi alınamaz. Bu durumun üstesinden gelmenin bir yolu matrisin köşegenine sönüm parametresi eklemektir. Sabit bir sönüm parametresi ile gerçekleştirilen EKK çözümleri Sönümlü (Damped) EKK olarak bilinmektedir. Bir diğer yöntem ise düzenleme (regularization) parametresidir. Düzenleme parametresi EKKG ve GD işlemlerinin en önemli avantajıdır. Öyle ki, düzenleme parametresi kullanılmaması durumunda EKKG ve GD işlemleri Kirchhoff göçüne neredeyse eşittir. Şekil 4.3 birkaç farklı sönüm ve düzenleme parametresi ile elde edilen EKKGD çıkışlarını göstermektedir. $(L^T L)$ fonksiyonu çok kötü koşullu bir matris olmasına rağmen çözüme ulaşırken (Şekil 4.3a) daha kötü koşullu (dd^T) fonksiyonu çözüme ulaşamamıştır (Şekil 4.3f). Sönüm parametresinin eklenmesi ile $(L^T L)^{-1}$ fonksiyonunun hesaplanması (Şekil 4.3b,c) çözüme bir katkı sağlamamıştır. Bunun yanında düzenleme parametresinin katsayısını kullanmadan ($\delta = 1$) da bir iyileşme söz konusu olmamıştır (Şekil 4.3d). Fakat düzenleme parametresinin katsayısı ($\delta = 10000$) olarak alındığında göç hatalarının azaldığı ve ayrımlılığın arttığı açıkça görülmektedir. Netleştirme fonksiyonu olarak $(dd^T)^{-1}$ fonksiyonunu kullanılması son derece dikkat isteyen bir işlemdir (Şekil 4.3f, g, i). Görüldüğü gibi (dd^T) fonksiyonu son derece kötü koşullu bir matristir. Bu nedenle büyük sönüm parametresi ($I * 1000$, Şekil 4.3h) veya düzenleme

parametresinin katsayısının büyük ($\delta = 10000$, Şekil 4.3j) olması gerekir. Bu iki durumda da $(L^T L)$ ile (dd^T) fonksiyonlarının bir farkı görülememektedir. Bu nedenle özellikle netleştirme fonksiyonu olarak giriş verisinin (d) kullanılması durumunda düzenleme parametresine ve katsayısına dikkat edilmesi gerekmektedir. Bunun yanında, Cauchy düzenleme parametresi tam bir kapanım sağlayamamıştır. Ancak L1 normu:

$$R(m) = \sum_i^N |m_i| \quad (21)$$

kullanılarak elde edilen L1 düzenleme (regularization) parametresi:

$$Q_{ii} = (|m_i| + \epsilon)^{-1} * \delta^2 \quad (22)$$

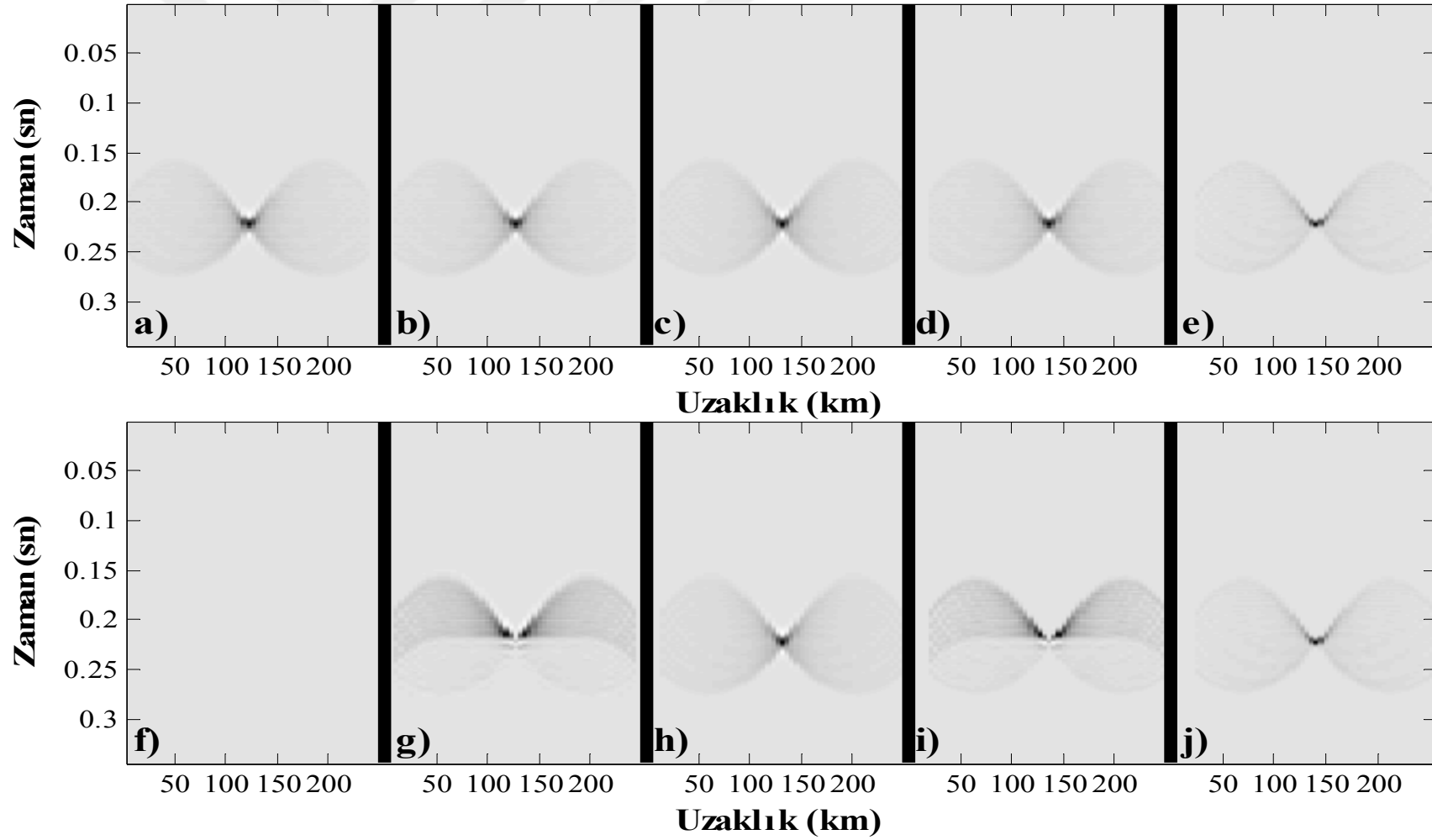
veya L2 normu:

$$R(m) = \sum_i^N m_i^2 \quad (23)$$

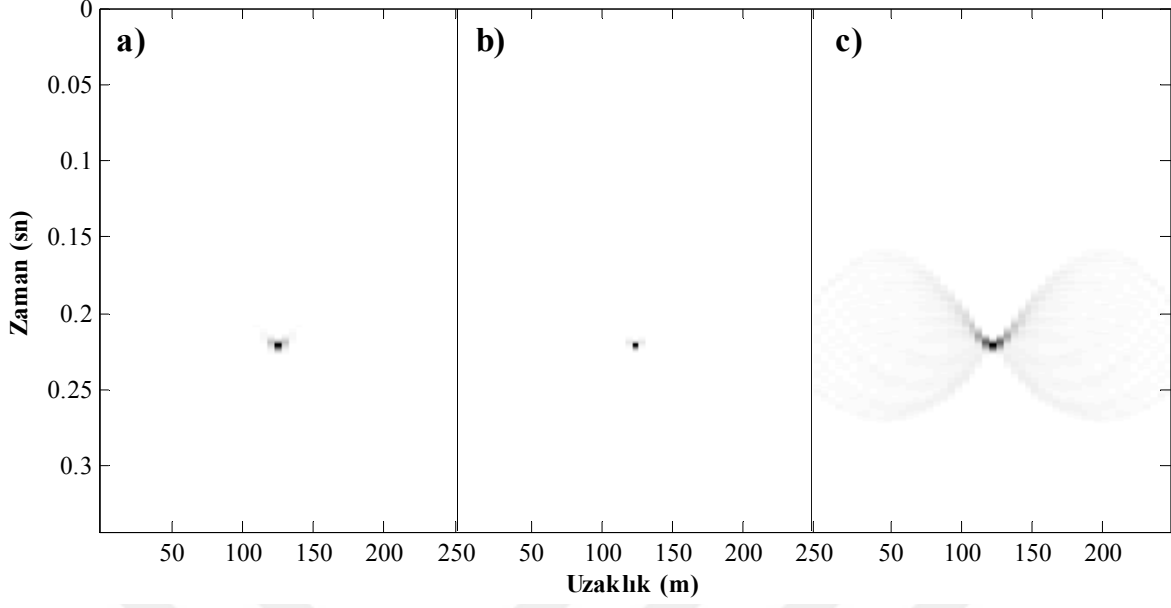
kullanılarak elde edilen L2 düzenleme parametresi:

$$Q_{ii} = (m_i^2 + \epsilon)^{-1} * \delta^2 \quad (24)$$

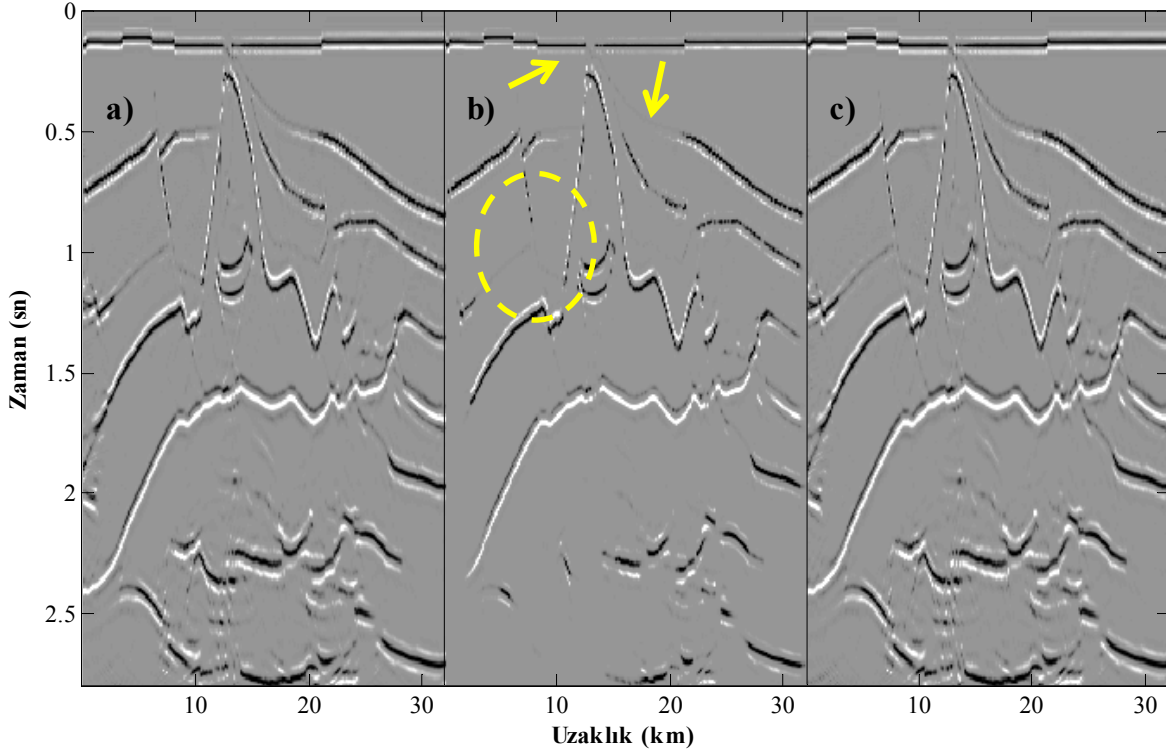
kullanıldığında tam bir kapanım sağlanabilir. Şekil 4.4 Nokta saçılma modeli için L1, L2 ve Cauchy normlarından elde edilen düzenleme parametreleri ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile gerçekleştirilen EKKGD çıkışlarını göstermektedir. Buradan açıkça görülmektedir ki L2 normu en iyi kapanımı sağlamıştır ve L1 normu da buna yaklaşmıştır fakat Cauchy normu bunların uzağında kalmıştır. Şekil 4.5 Tuz modeli için farklı düzenleme parametreleri ile elde edilen çıkışları göstermektedir. Tuz modeli için L2 normunun genlik koruma bakımından başarısız olduğu görülmektedir. Şekil 4.6 Arazi verisi-1 için farklı normlar ile elde edilen göç çıkışlarını göstermektedir. Arazi verisi-1 için de L1 normu genlik korumada başarısız olduğu açıkça görülmektedir. Bu üç farklı uygulamada da genlikleri koruyan Cauchy normu en güvenilir düzenleme parametresini üretmiştir. Bu nedenle, bu çalışmada hem model veriler hem de gerçek veriler için Cauchy düzenleme parametresi kullanılmıştır.



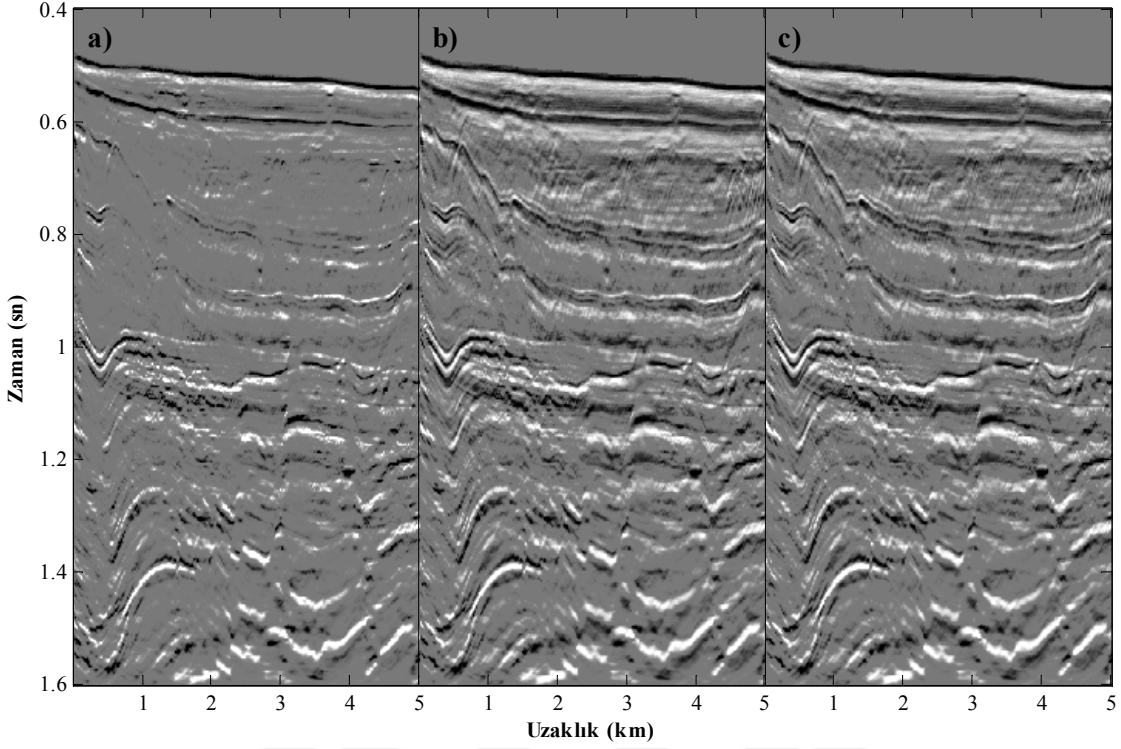
Şekil 4. 3. Düzenleme ve sönüm parametresi testleri. Nokta saçılma modeli için a) $(L^T L)^{-1}$, b) $(L^T L + I)^{-1}$, c) $(L^T L + I * 1000)^{-1}$, d) $(L^T L + Q)^{-1}$, e) $(L^T L + Q * 10000^2)^{-1}$, f) $(dd^T)^{-1}$, g) $(dd^T + I)^{-1}$, h) $(dd^T + I * 1000)^{-1}$, i) $(dd^T + Q)^{-1}$ ve j) $(dd^T + Q * 10000^2)^{-1}$ ile elde edilen EKKGD çıktıları.



Şekil 4. 4. Nokta saçılma modeli için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.



Şekil 4. 5. Tuz modeli için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.

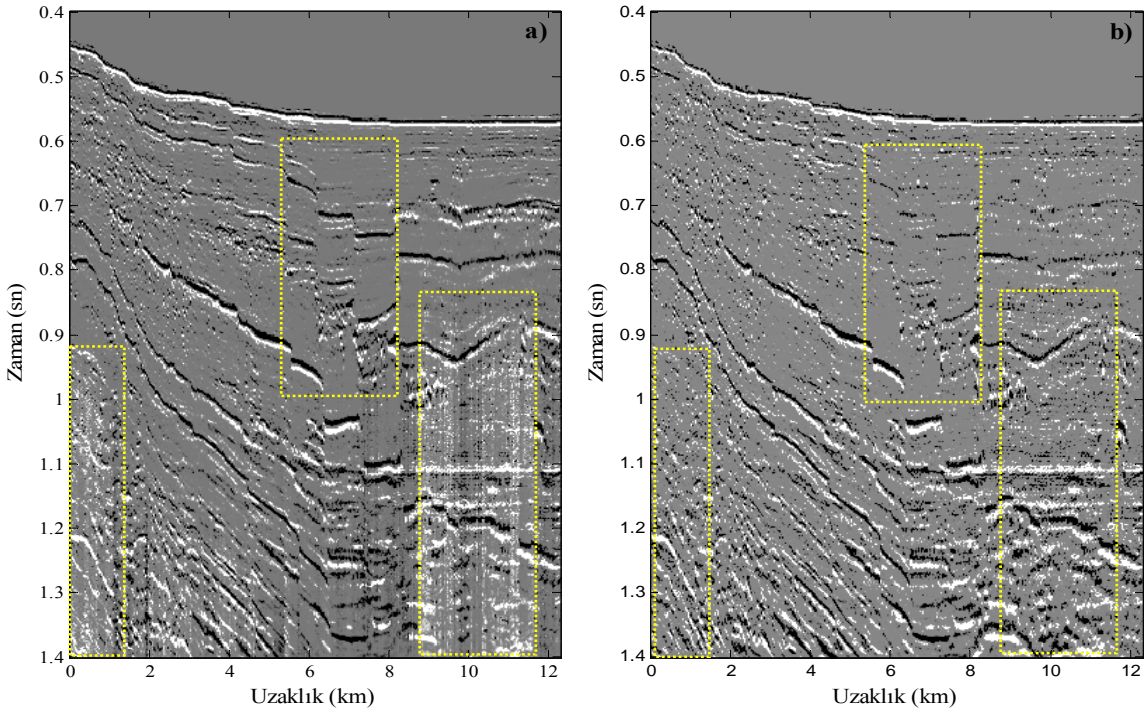


Şekil 4. 6. Arazi verisi-1 için a) L1 düzenleme parametresi, b) L2 düzenleme parametresi ve c) Cauchy düzenleme parametresi ve $(L^T L)^{-1}$ netleştirme fonksiyonu ile EKKGD çıkışları.

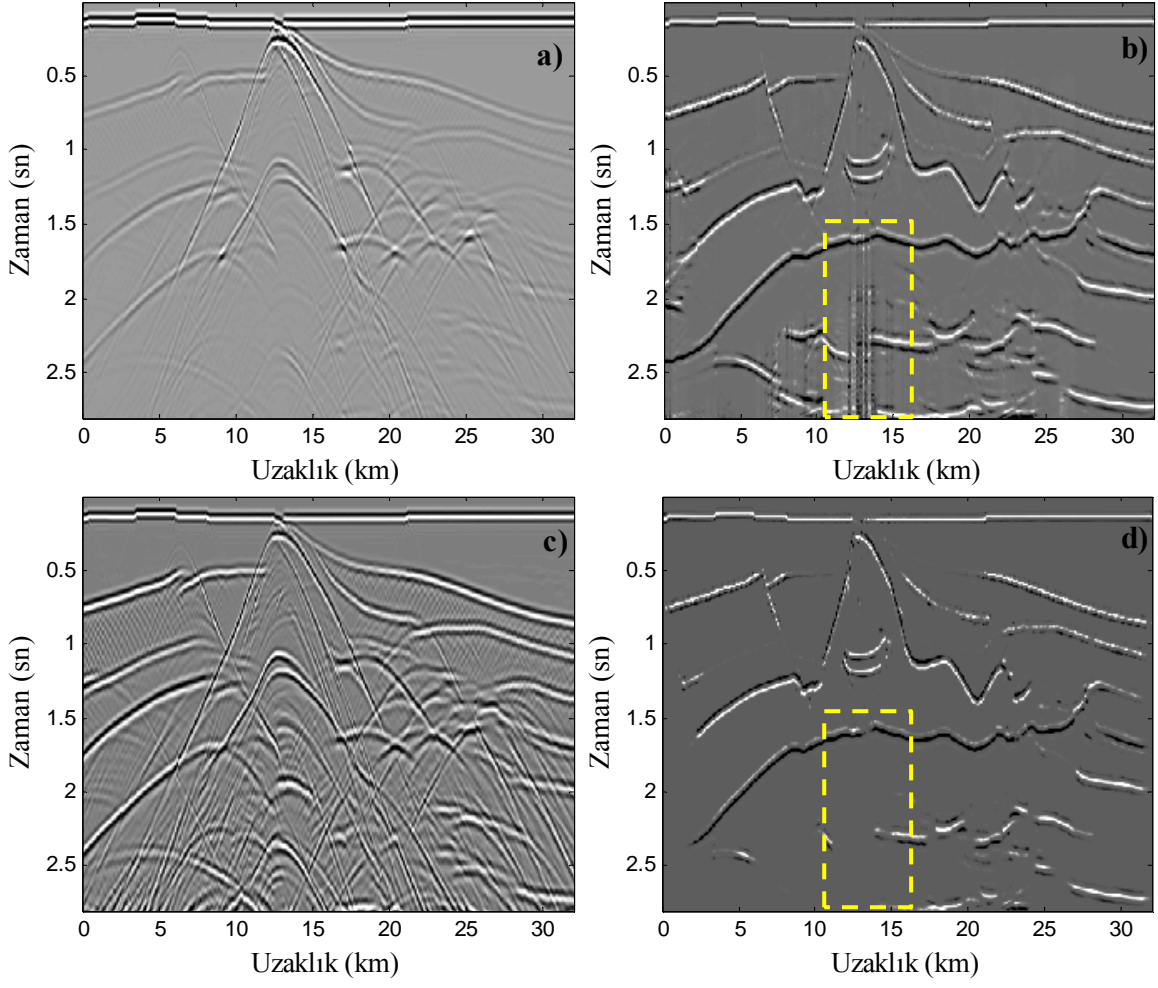
4.9. Maliyet ve Fayda açısından en uygun çıkışları üreten netleştirme fonksiyonları.

Yukarıda belirtildiği gibi, giriş verisinden, d , elde edilen netleştirme fonksiyonlarının daha az göç hatasını daha az maliyetle ürettiği görülmüştür. Fakat bunlardan (dd^T) ve $cov(dd^T)$ netleştirme fonksiyonları kusurlu göç çıkışları üretmiştir. Bunun yanında model verilerde en düşük hata payına sahip olan $cov((L^T L)^{-1})$ ve $cov((dd^T)^{-1})$ netleştirme fonksiyonları Tuz modeli ve Arazi verisi-2 için çözümlerde gölge etkisi oluşturmuştur. Diğer veri setleri için son derece faydalı sonuçlar üreten bu fonksiyonların asıl sorununun genlik dengesizliği olduğu öngörülmektedir. Bu nedenle kazanç işlemi gerçekleştirilerek Tuz modeli ve Arazi verisi-2 veri setleri için çözümler gerçekleştirilmiştir. Şekil 4.7 Arazi verisi-2 için kazanç işleminden önce ve sonra $cov((dd^T)^{-1})$ fonksiyonu ile gerçekleştirilen EKKGD işleminin sonuçlarını göstermektedir. Her iki yandaki kesikli çizgilerle sınırlı alanlardan gölge etkisinin giderildiği görülürken orta kısımdaki alanda ise üst bölgelerdeki genlikleri olumsuz etkilendiğini göstermektedir. Şekil 4.8 Tuz modeli için kazanç işleminden önce ve sonra

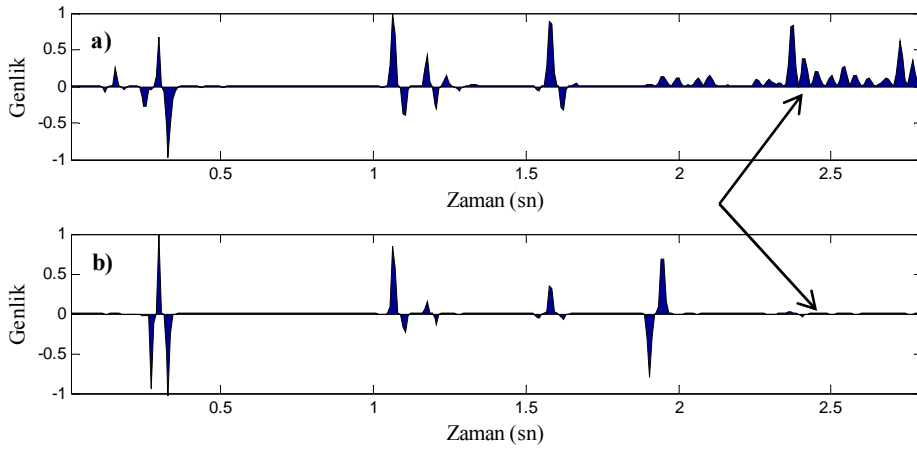
$cov((dd^T)^{-1})$ netleştirme fonksiyonu ile gerçekleştirilen EKKGD çıkışlarını göstermektedir. Kazanç işleminden sonra uygulanan EKKGD işleminin (Şekil 4.8d) gölge etkisinin kaybolduğu açıkça görülürken küçük genlikli olayların da soğurulduğu görülmektedir. Şekil 4.9 ise kazanç işleminden önce ve sonra gerçekleştirilen EKKGD çıkışlarının 12.5 km deki izlerini karşılaştırmaktadır. Burada gölge etkisinin nedeni pozitif bölgeye kayan genlikler olduğu görülmektedir ve eğilim analizi ile giderilememektedir. Bu durum, sonraki çalışmalarda araştırılması ve kullanıcıya bırakılması gereken bir durum olarak değerlendirilebilir. $(L^T L)$, (dd^T) , $cov(dd^T)$ ve $cov(L^T L)$ netleştirme fonksiyonlarının beklenen sonuçları veremedikleri düşünüldüğünde, farklı netleştirme fonksiyonları ($(L^T L)^{-1}$, $(dd^T)^{-1}$, $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$) ile elde edilen sonuçlar aynı şekillerde gösterilmiştir (Şekil 4.11 - 4.28). Yukarıda $(L^T L)^{-1}$ ve $(dd^T)^{-1}$ netleştirme fonksiyonları ile elde edilen göç çıkışları ile $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları ile elde edilen göç çıkışları aynı şekilde görülerek ancak ayırt edilebileceklerdir. Şekil 4.11 - 4.28 bakıldığında her iki çiftin aynı sonuçları ürettiği açıkça görülmektedir. Bunun yanında $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları ile elde edilen GD ve EKKGD çıkışları göç hatalarını daha çok bastırdığı da görülebilir.



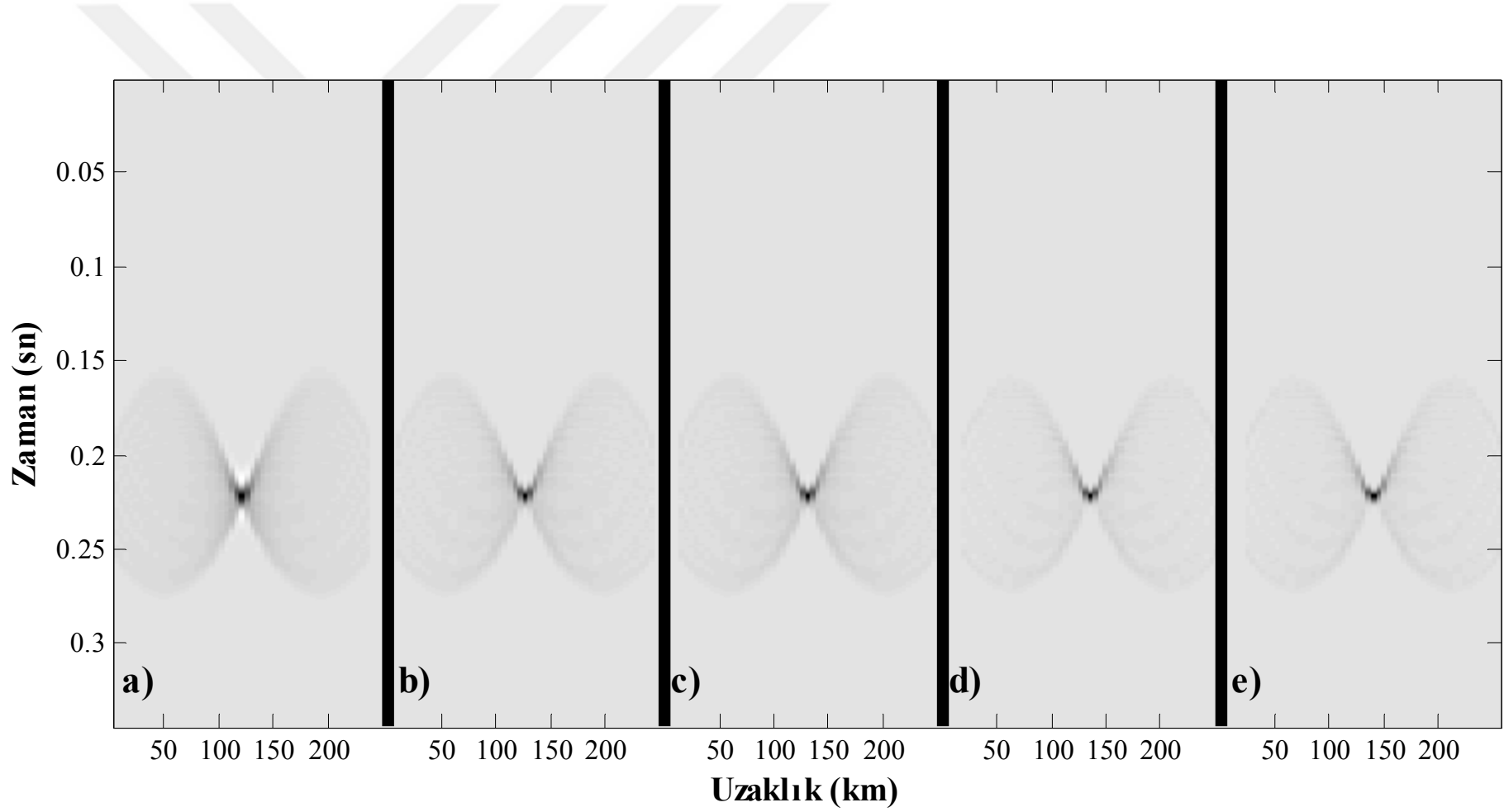
Şekil 4. 7. Arazi verisi-2 için a) kazanç uygulamadan önce $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı ve kazanç uygulaması, b) kazanç uygulamadan sonra $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı.



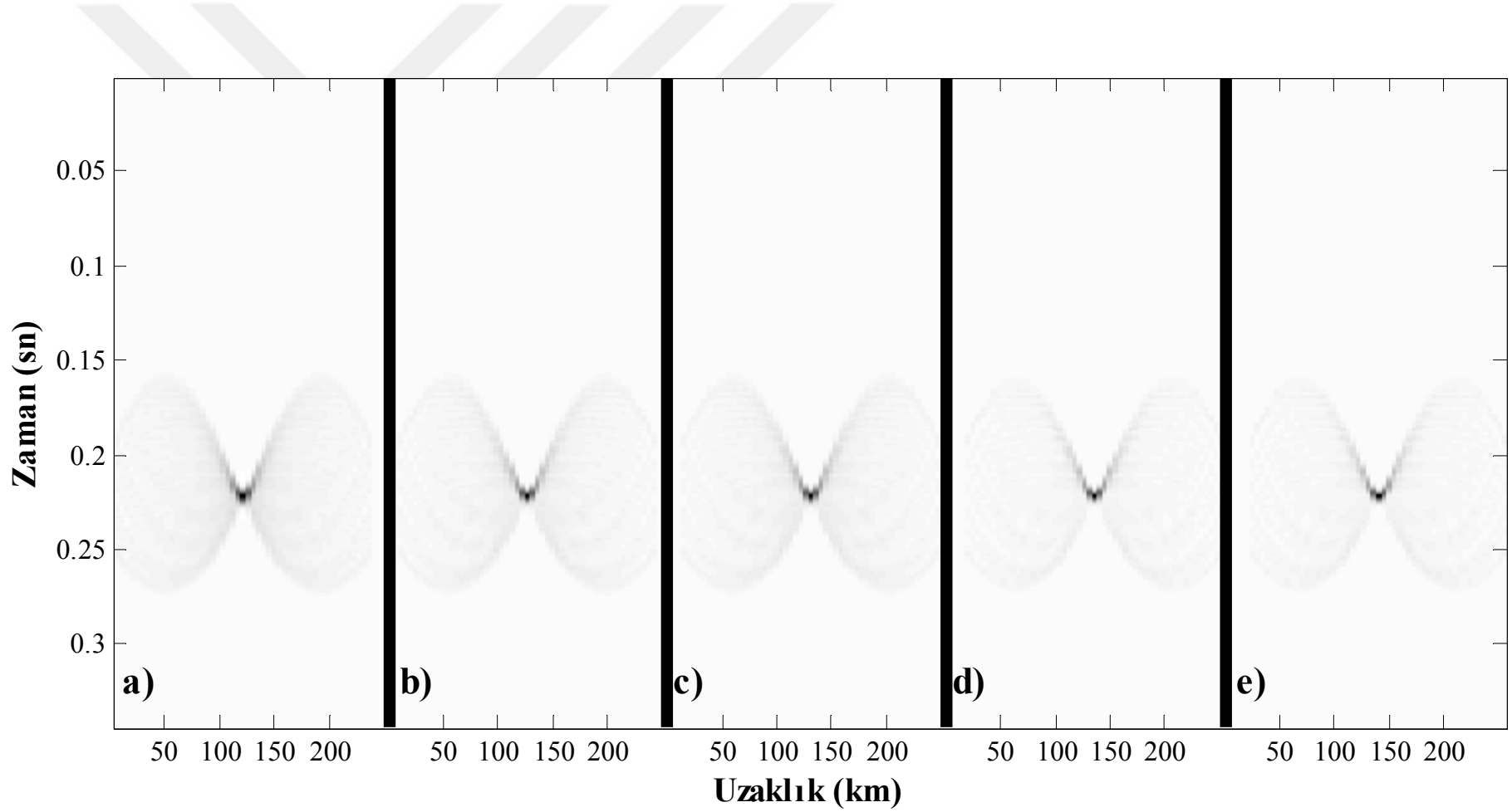
Şekil 4. 8. Tuz modeli için a) Giriş verisi, b) a' nın $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı ve kazanç uygulaması, c) kazanç uygulamasından sonra giriş verisi d) c' nin $cov((dd^T)^{-1})$ netleştirme fonksiyonu ile EKKGD çıkışı.



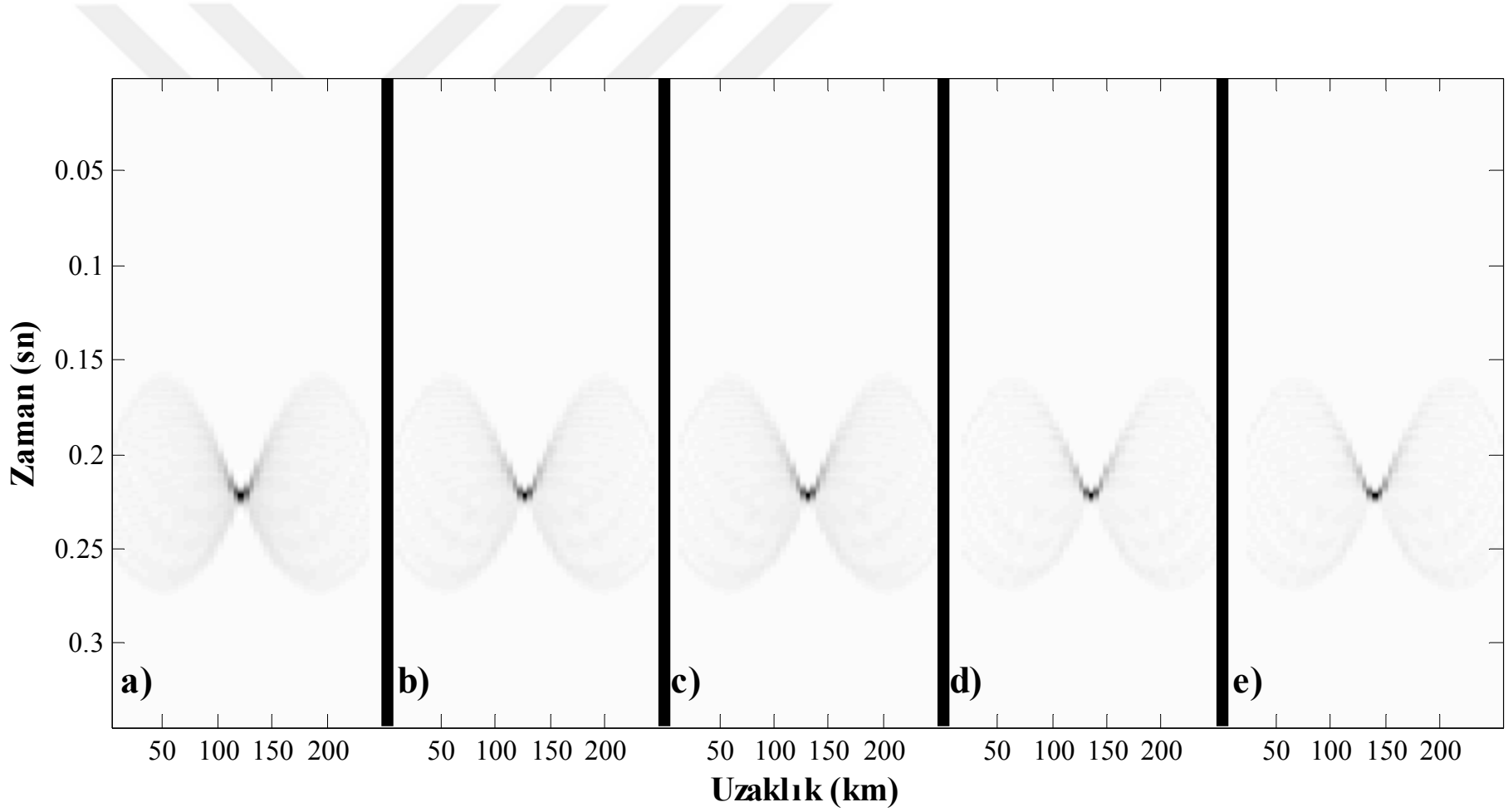
Şekil 4. 9. Tuz modeli için a) Kazanç uygulanmadan elde edilen göç çıkışının (Şekil 4.8b) 12.5 km deki izi, b) kazanç uygulandıktan sonra elde edilen göç çıkışının (Şekil 4.8d) 12.5 km deki izi.



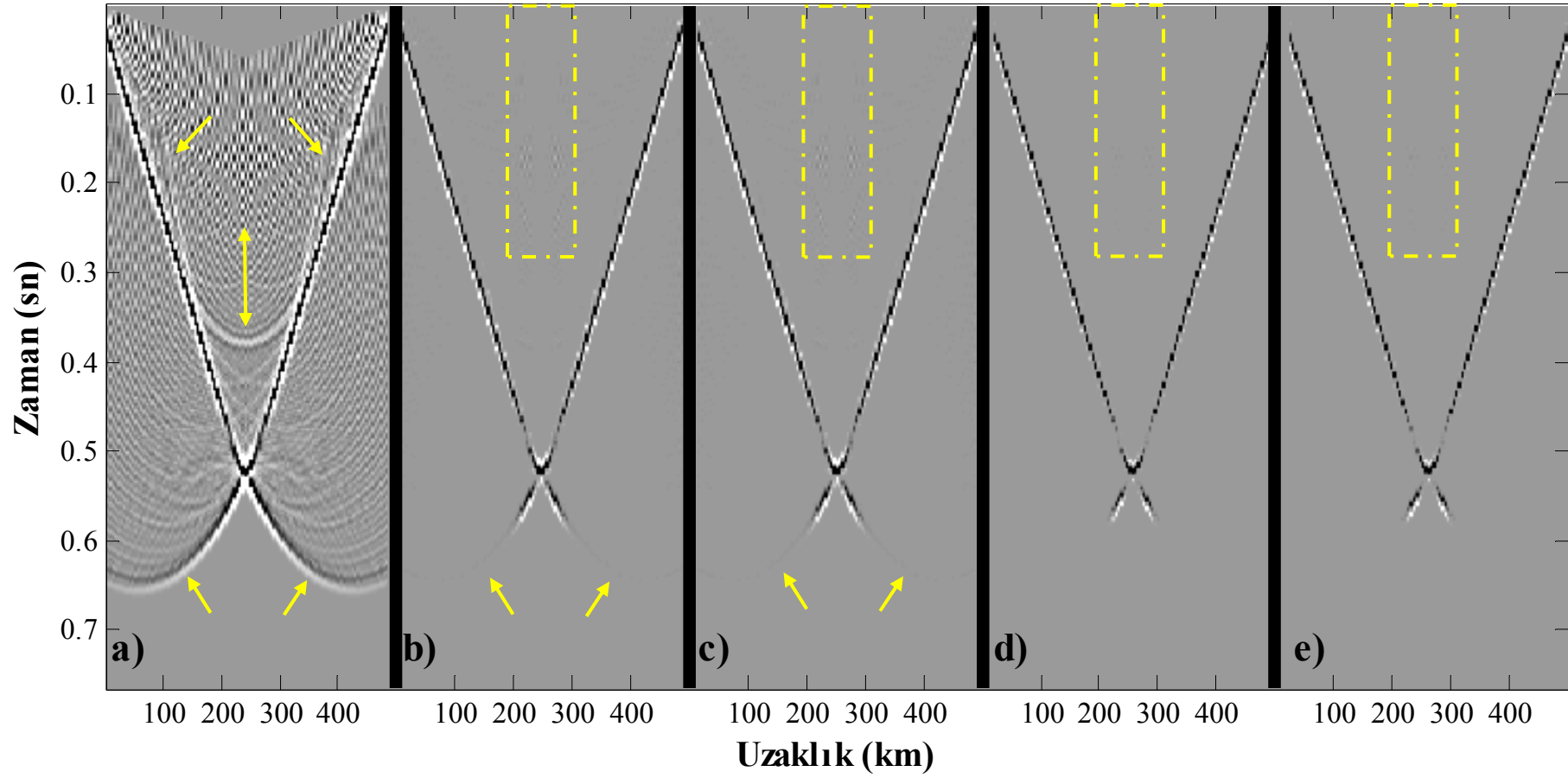
Şekil 4. 10. Nokta saçılma modeli için, a) Kirchhoff geç çıkışı, (b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve (e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



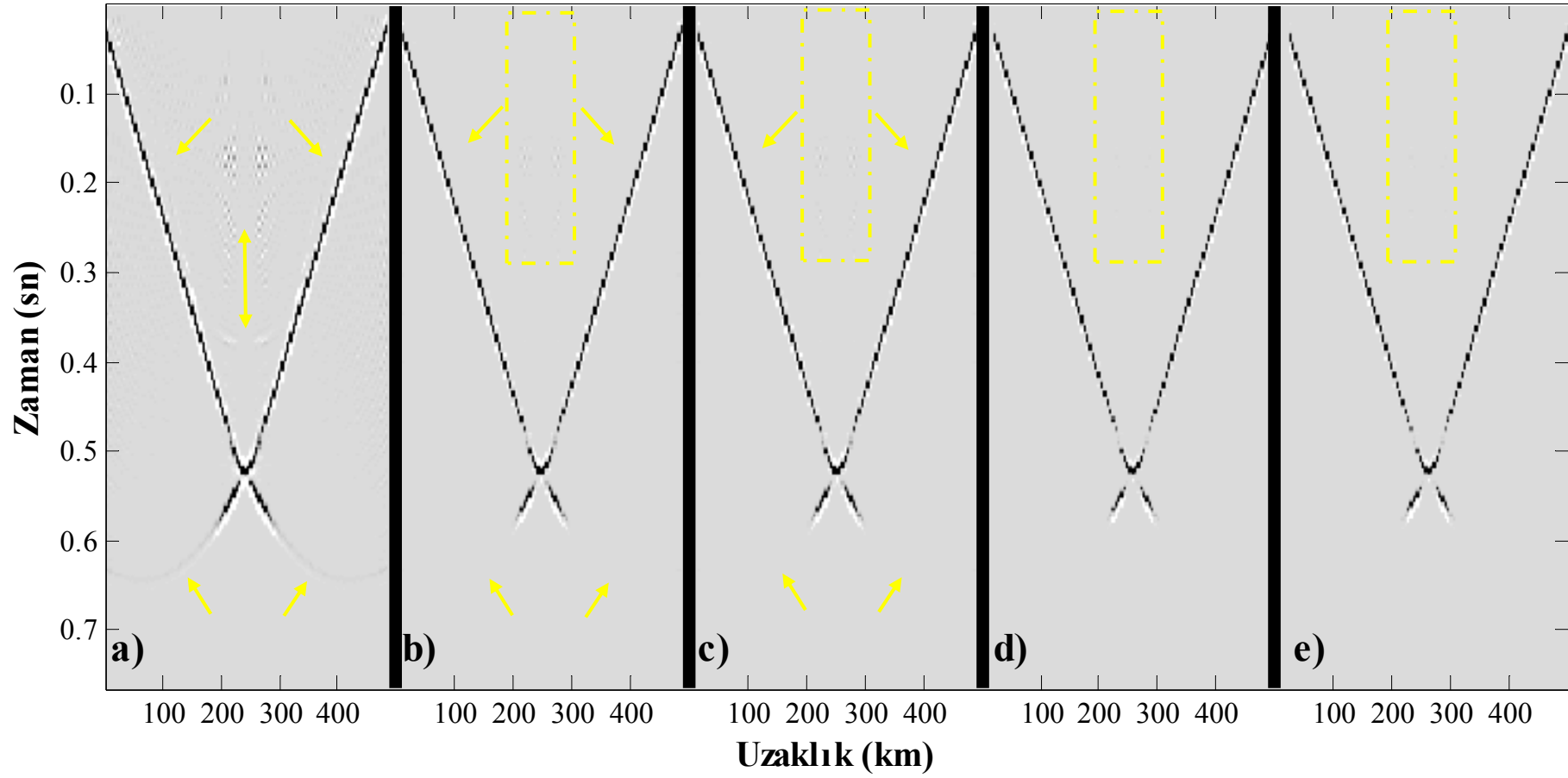
Şekil 4. 11. Nokta saçılma modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



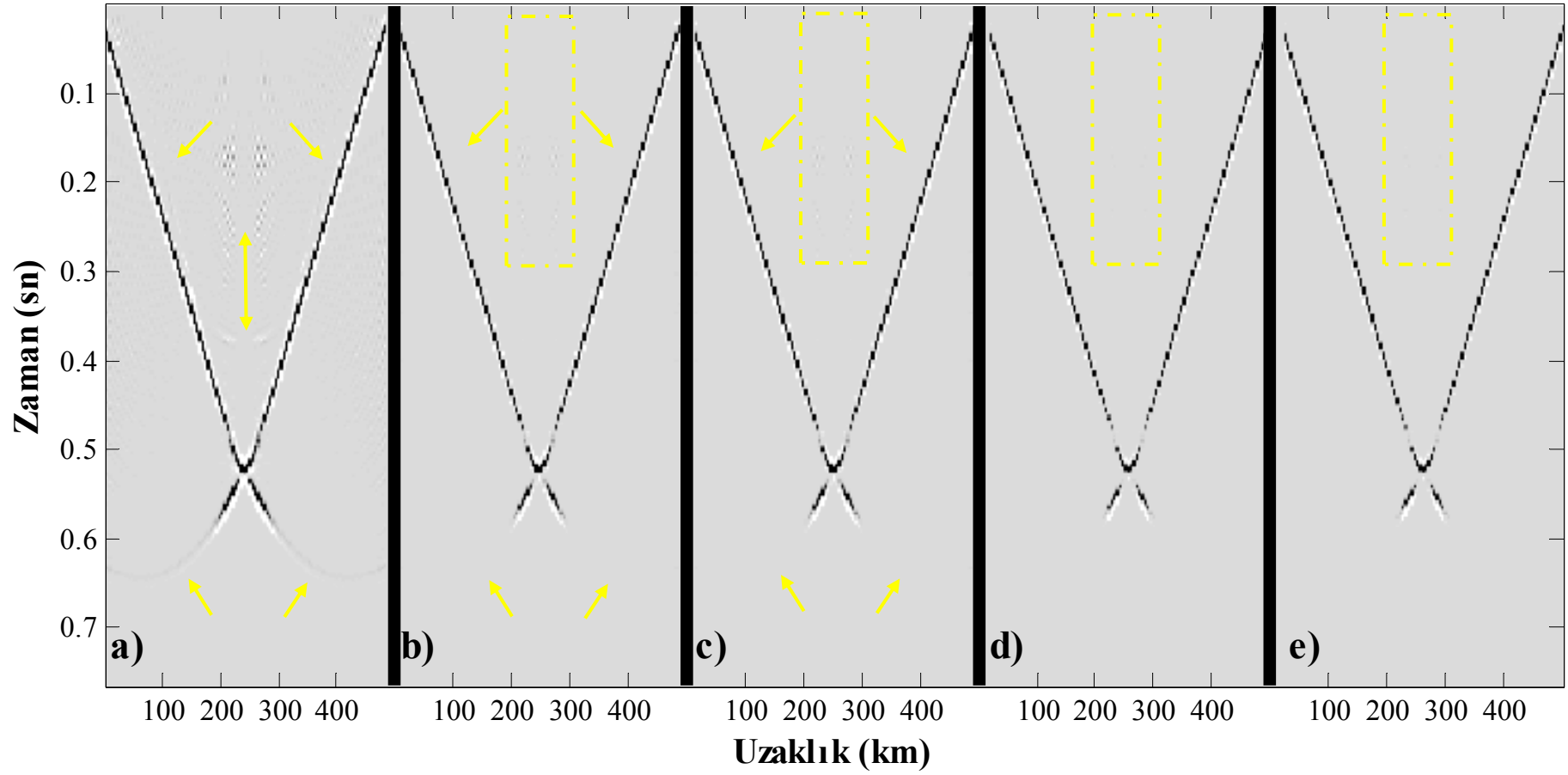
Şekil 4. 12. Nokta saçılma modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.



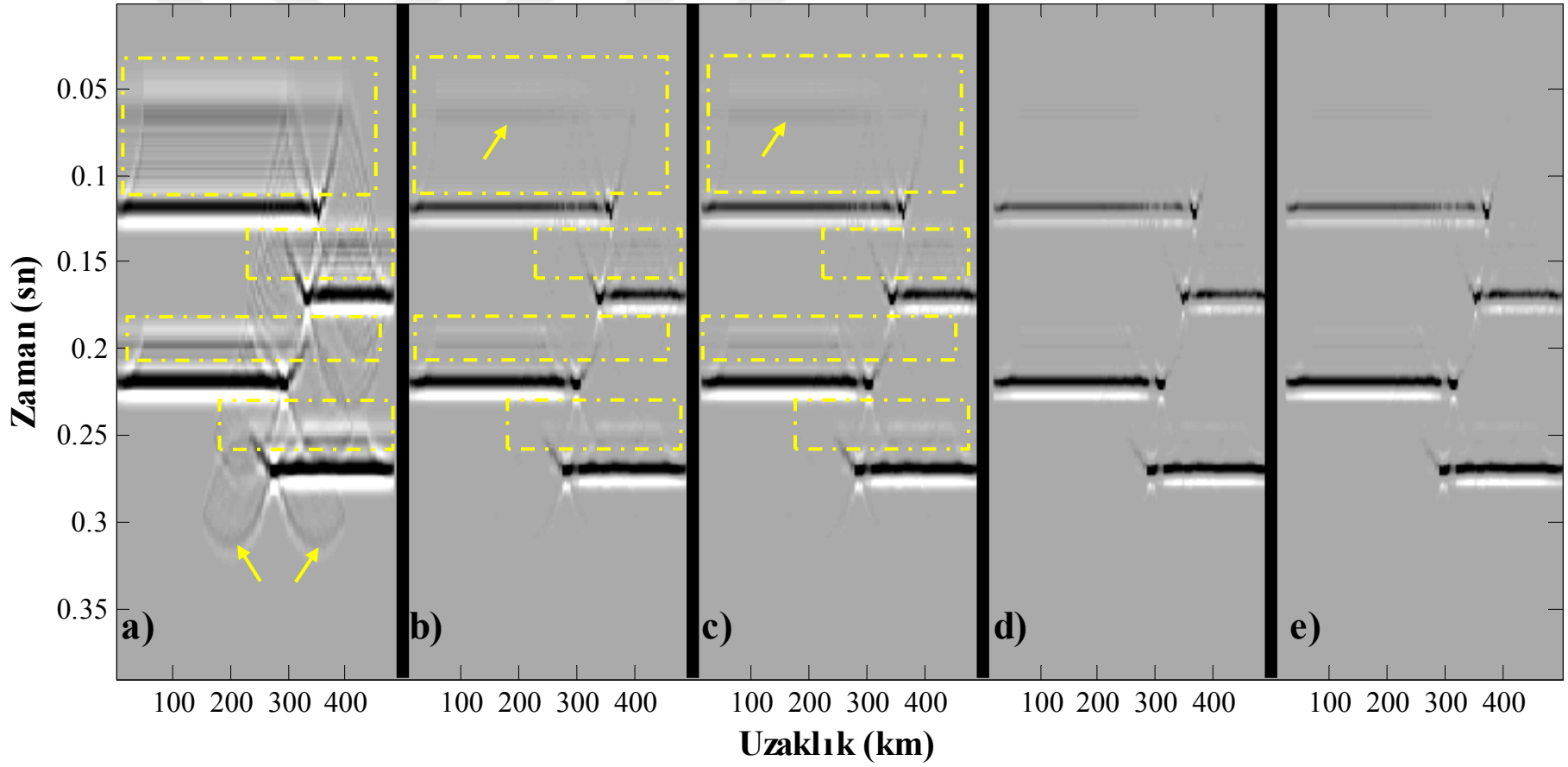
Şekil 4. 13. Papyon-kravat modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



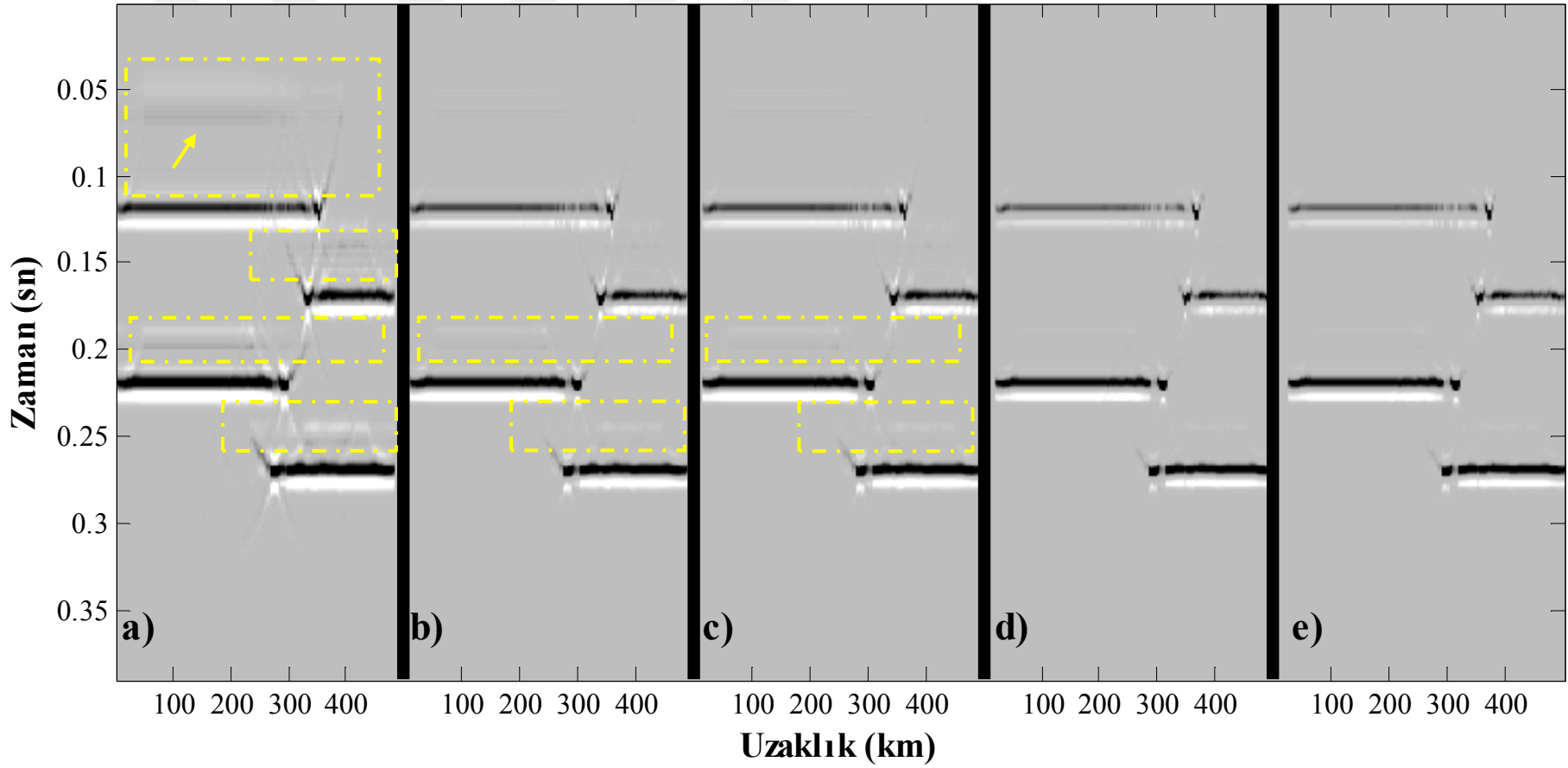
Şekil 4. 14. Papyon-kravat modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



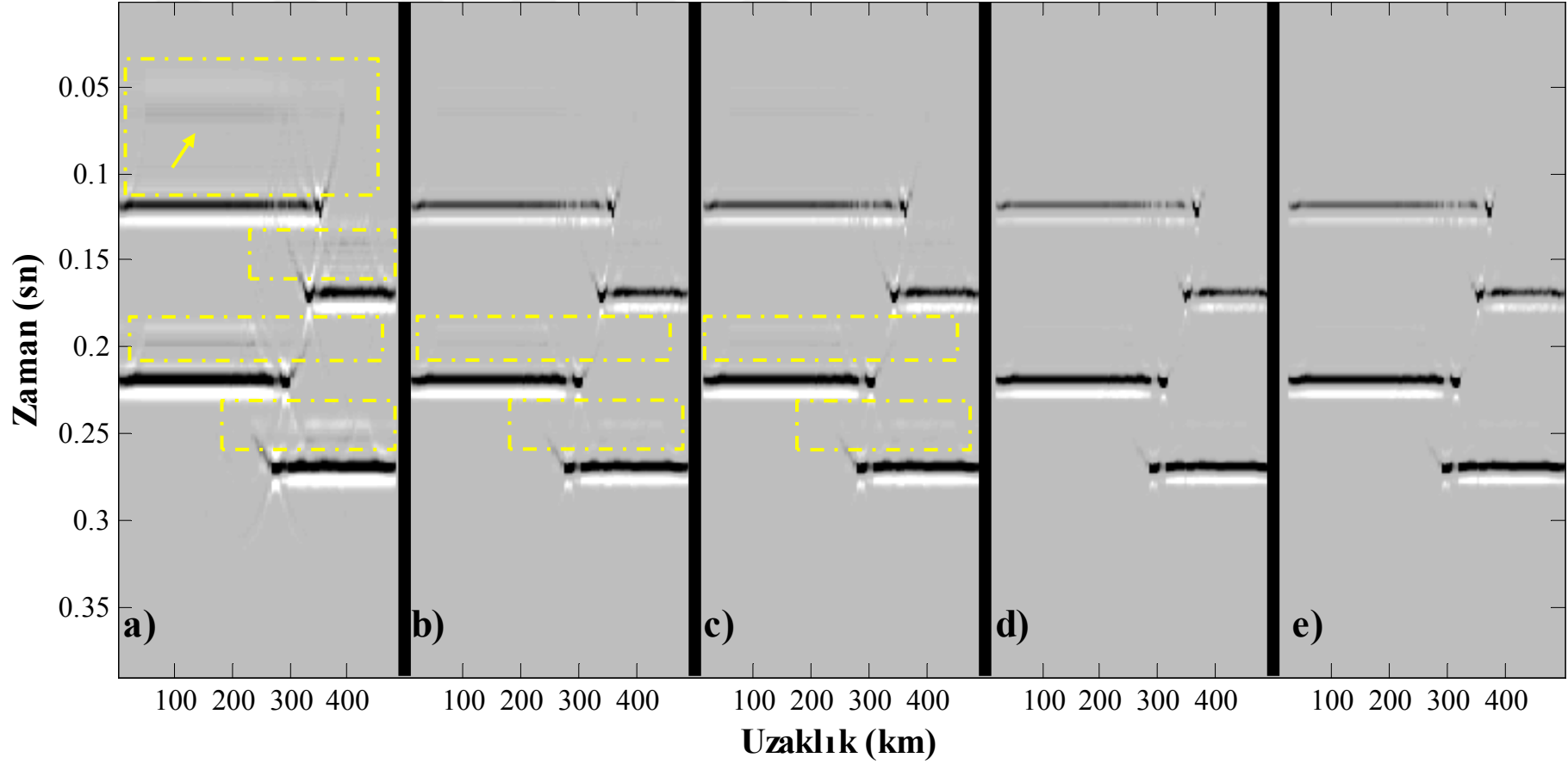
Şekil 4. 15. Papyon-kravat modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.



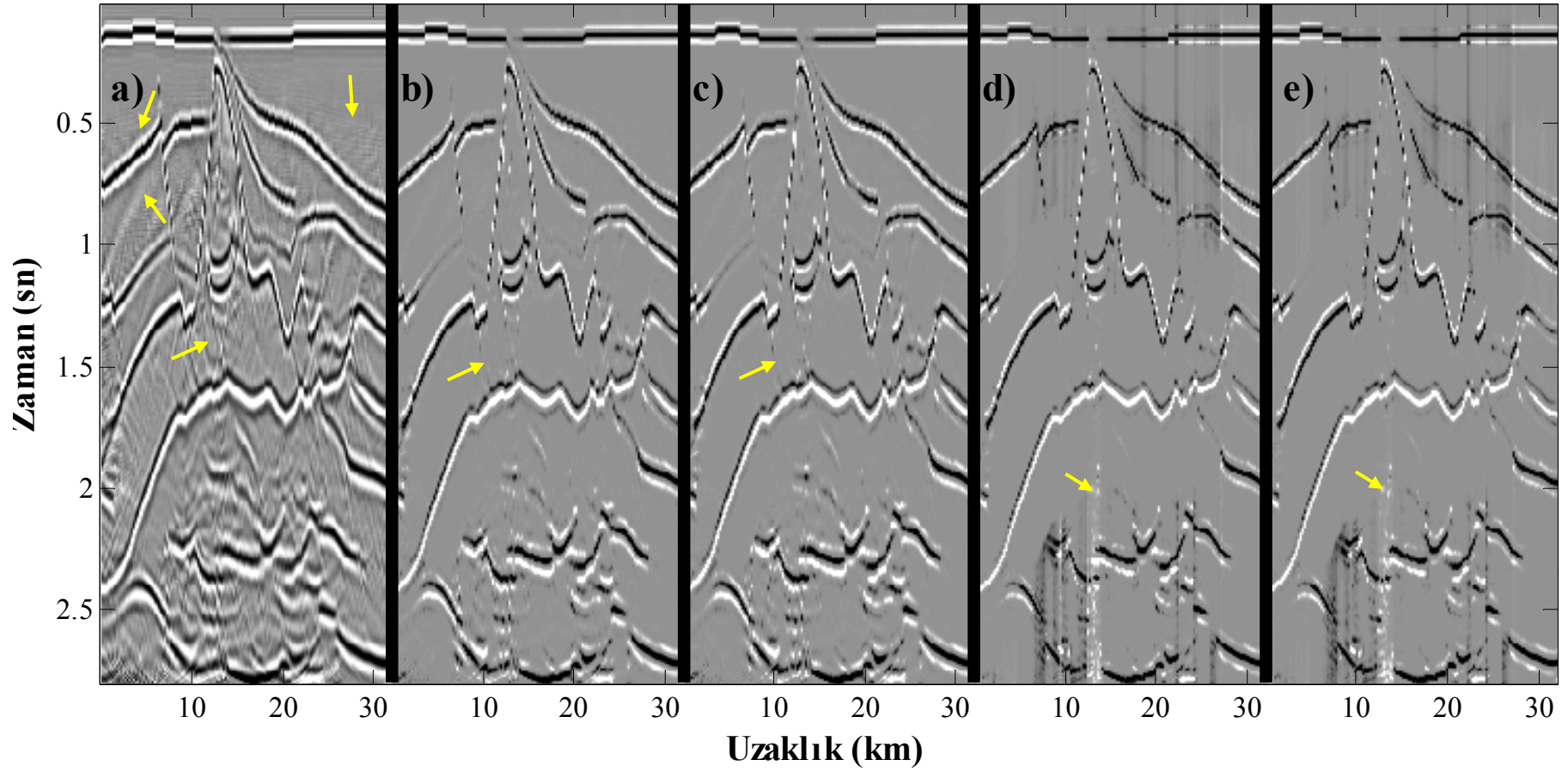
Şekil 4. 16. Fay modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



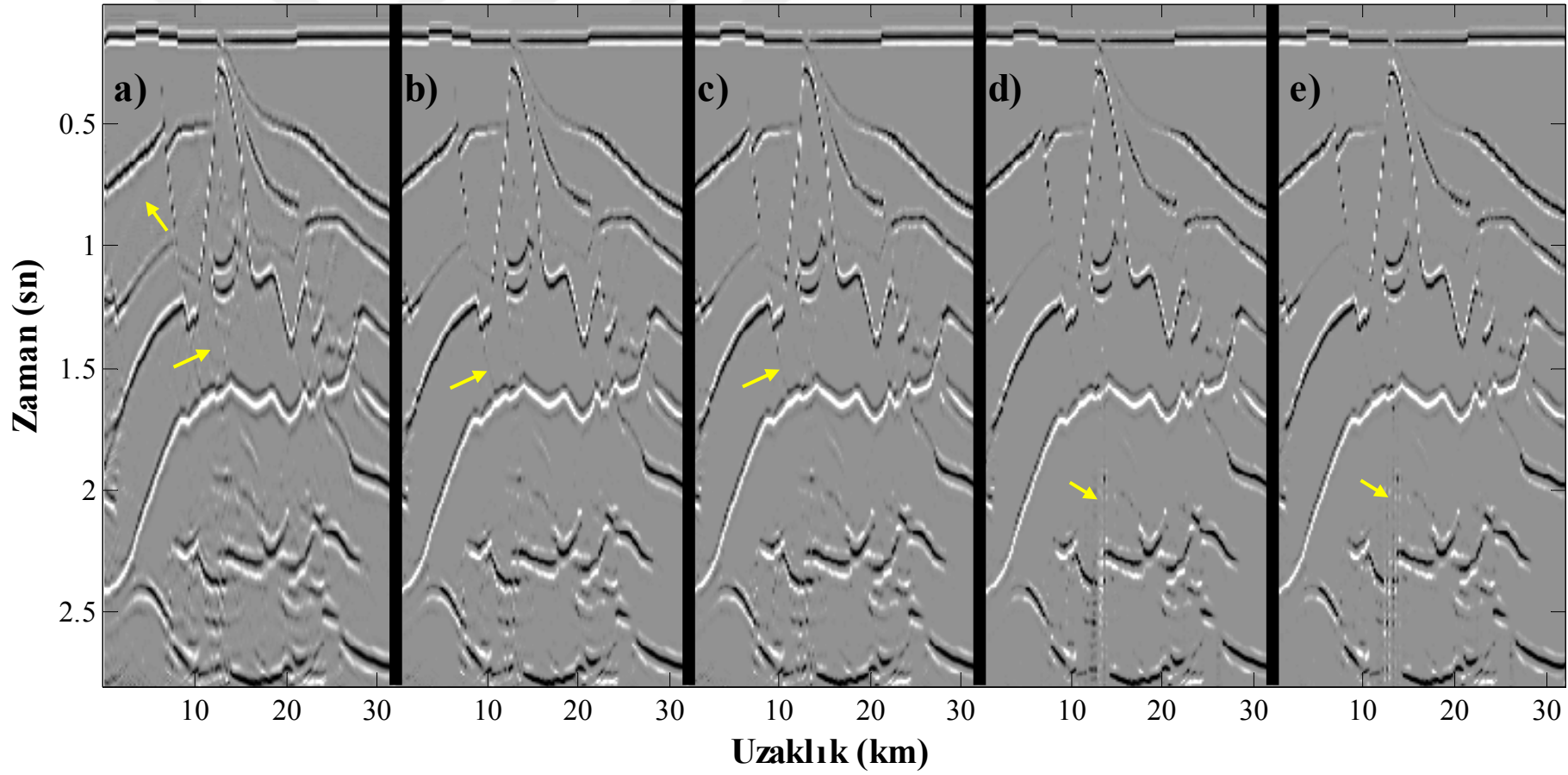
Şekil 4. 17. Fay modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



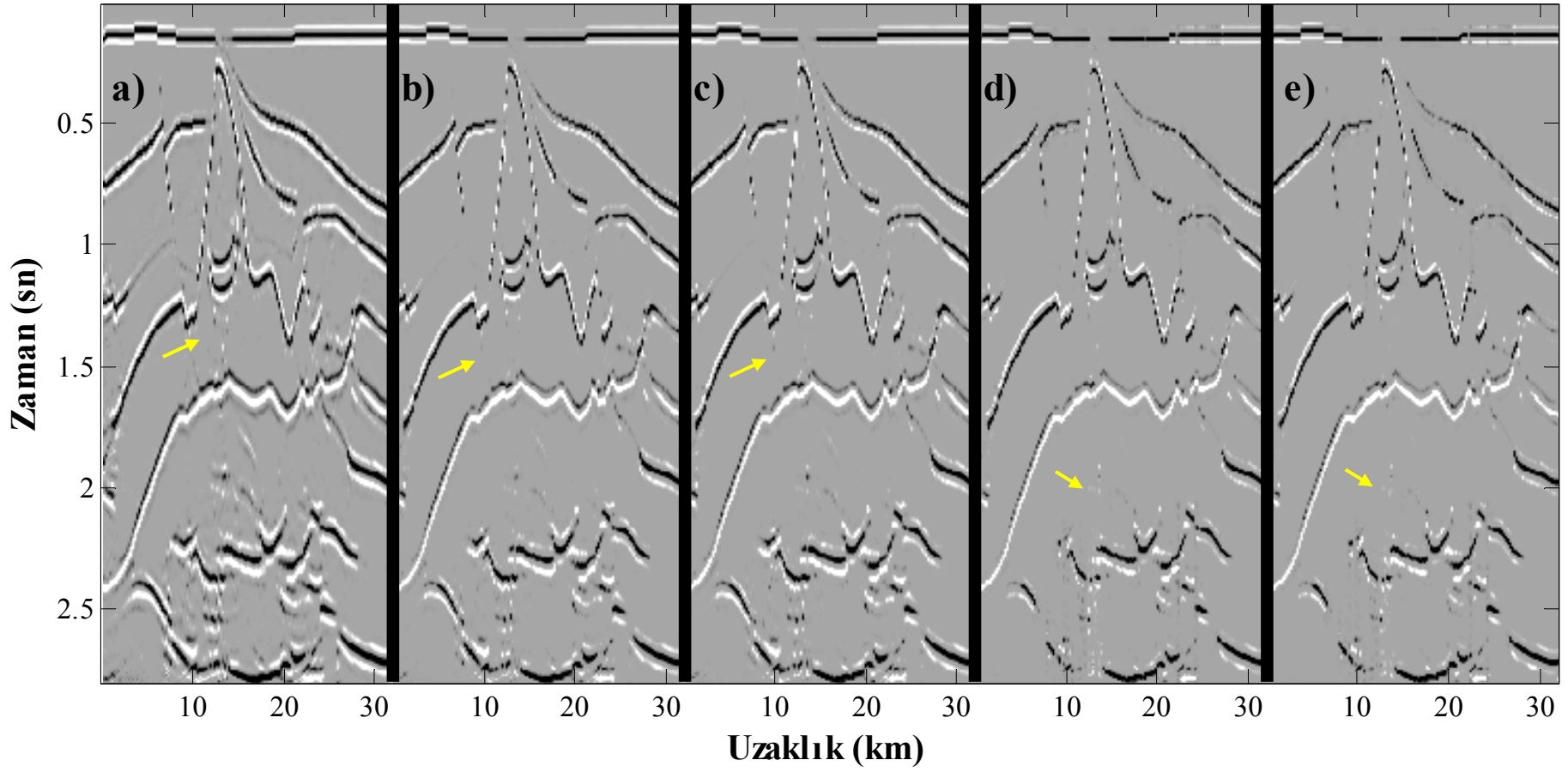
Şekil 4. 18. Fay modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.



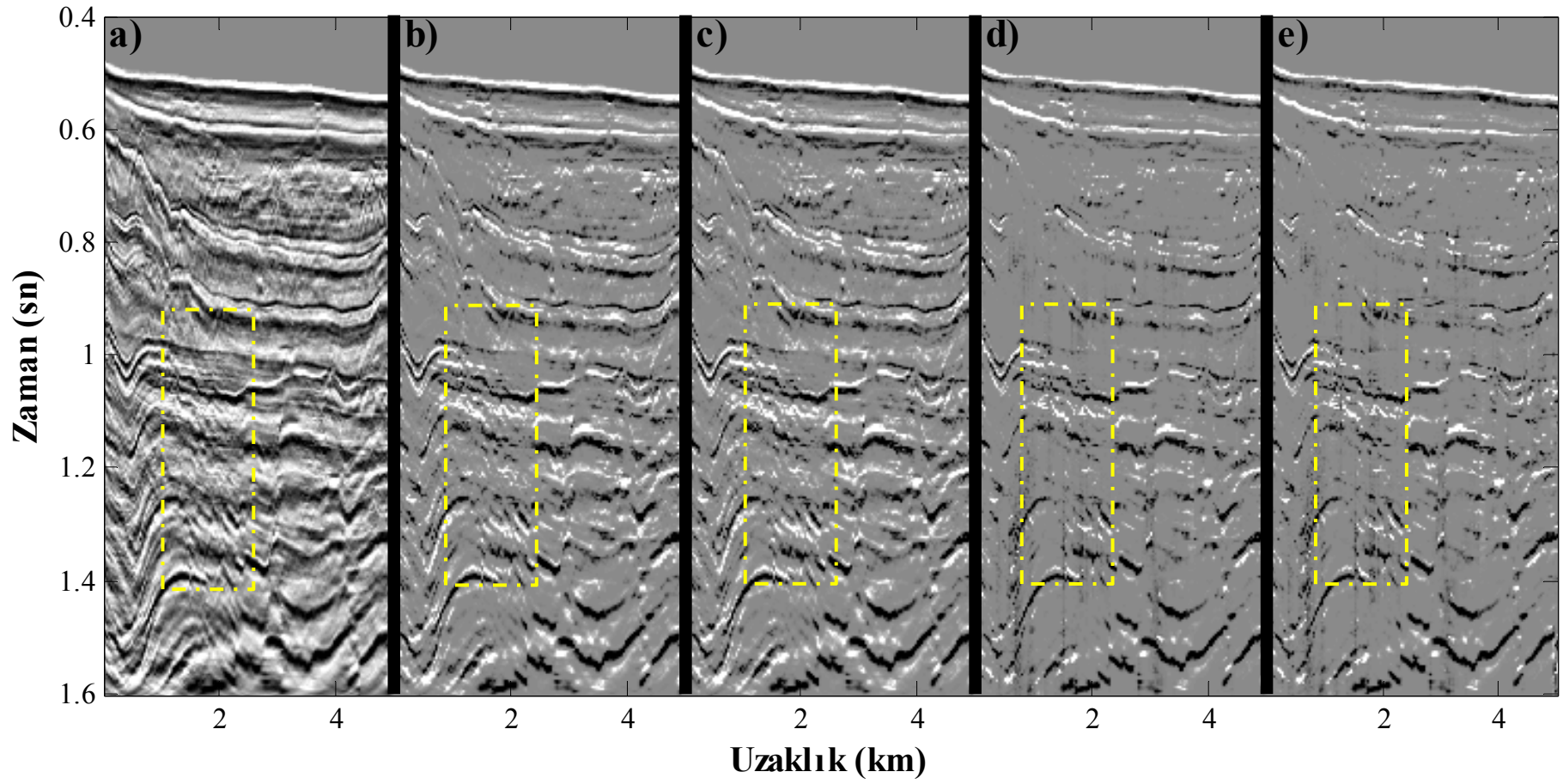
Şekil 4. 19. Tuz modeli için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



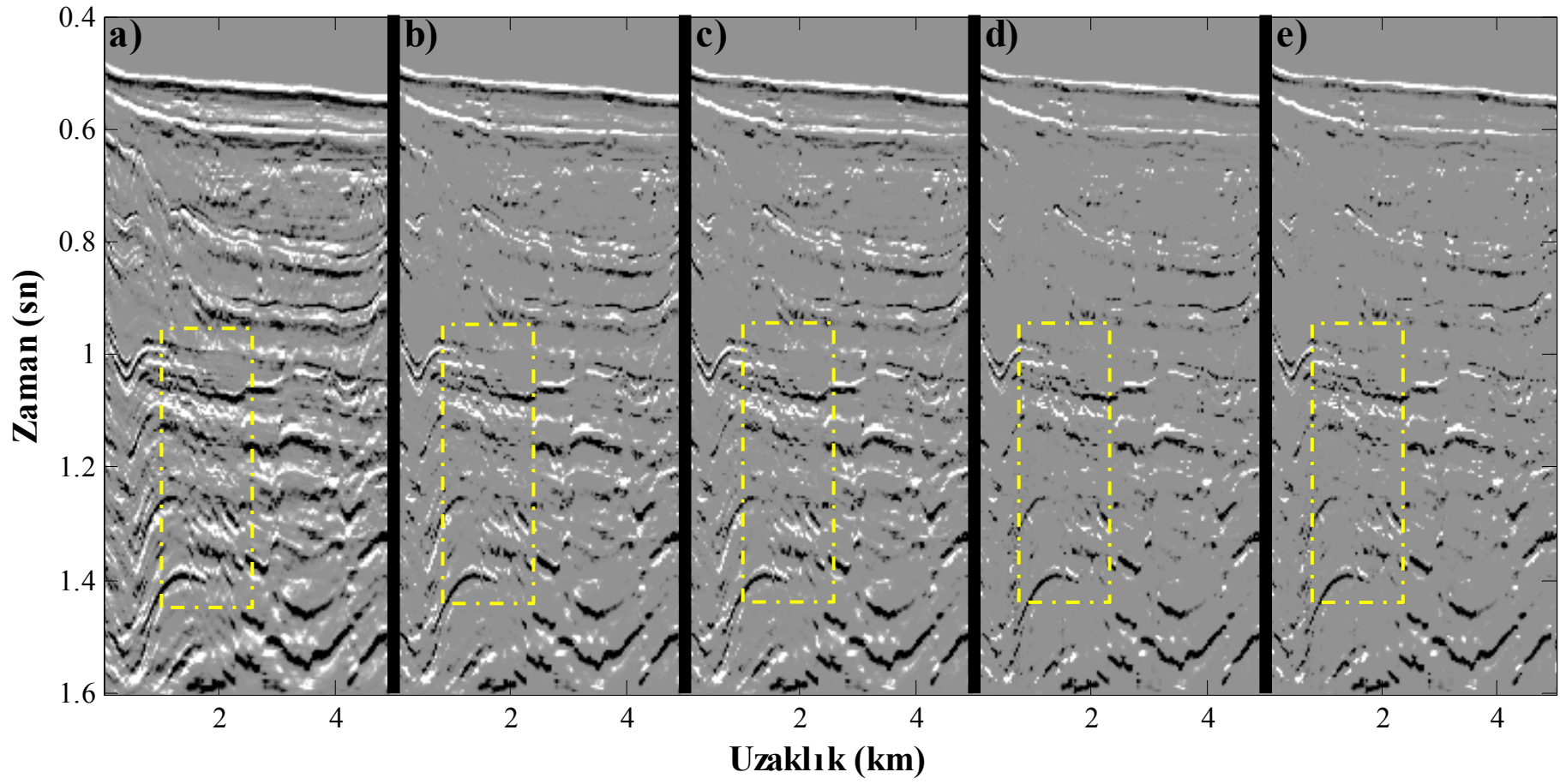
Şekil 4. 20. Tuz modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



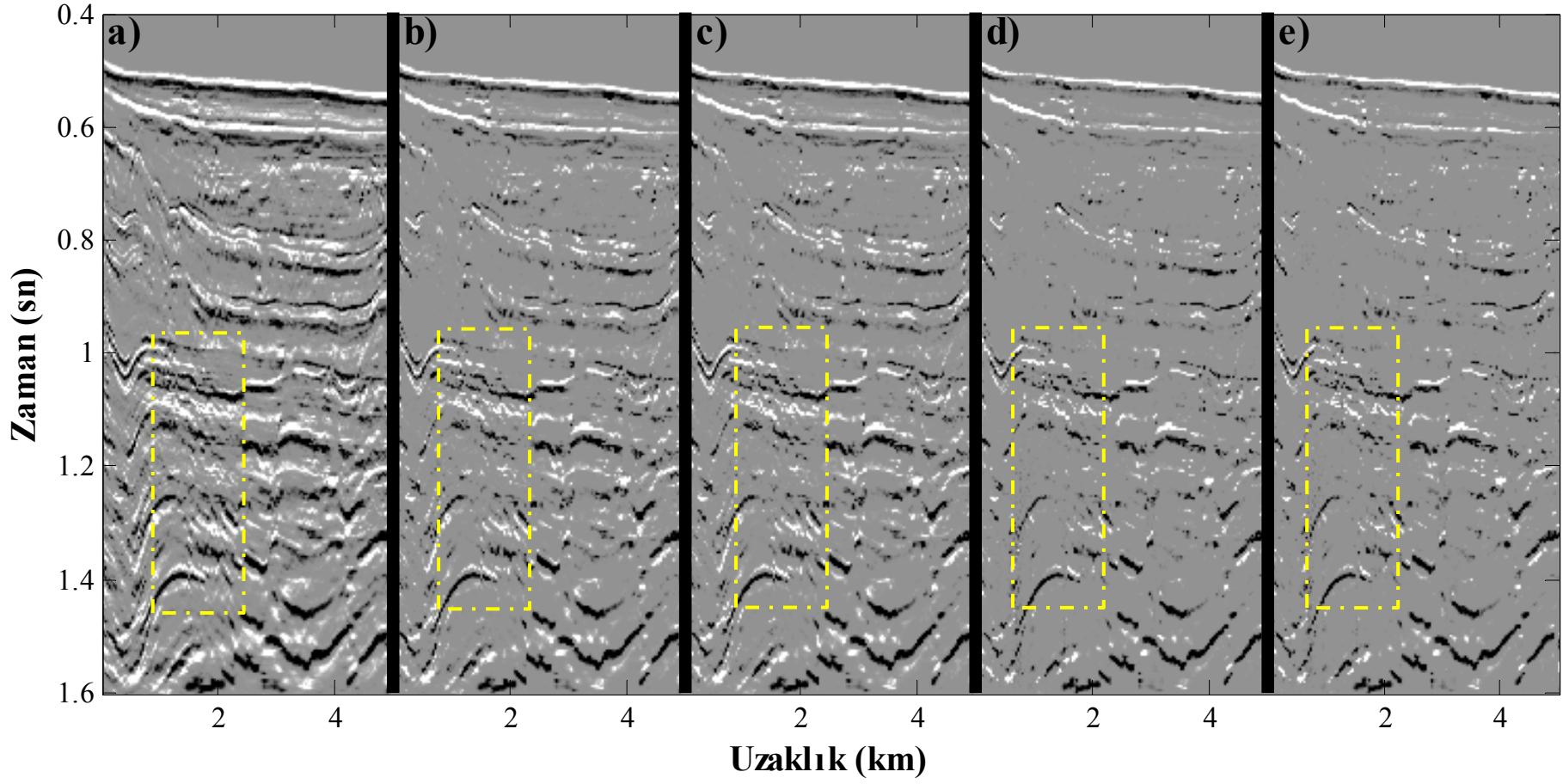
Şekil 4. 21. Tuz modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.



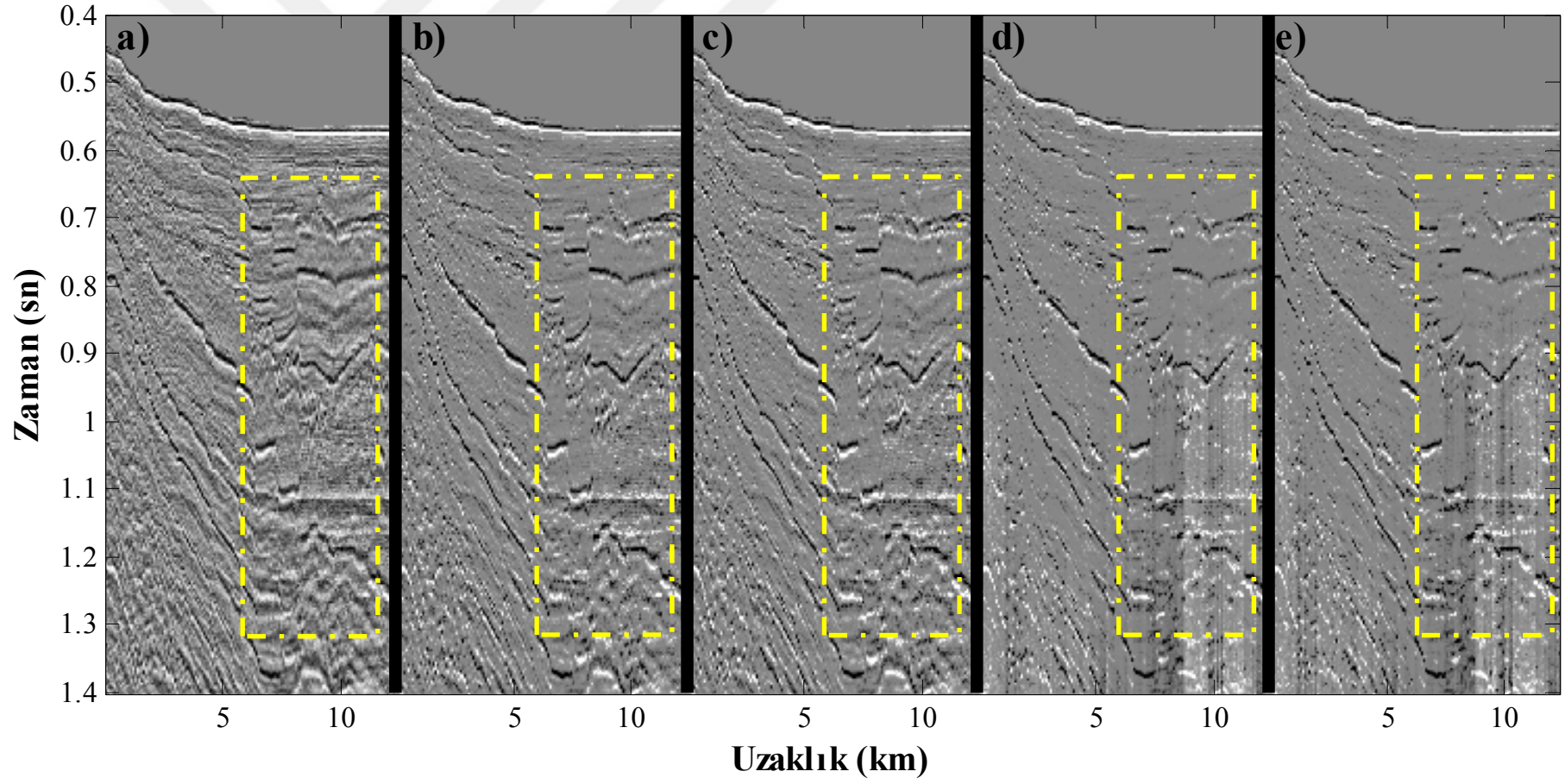
Şekil 4. 22. Arazi verisi-1 için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



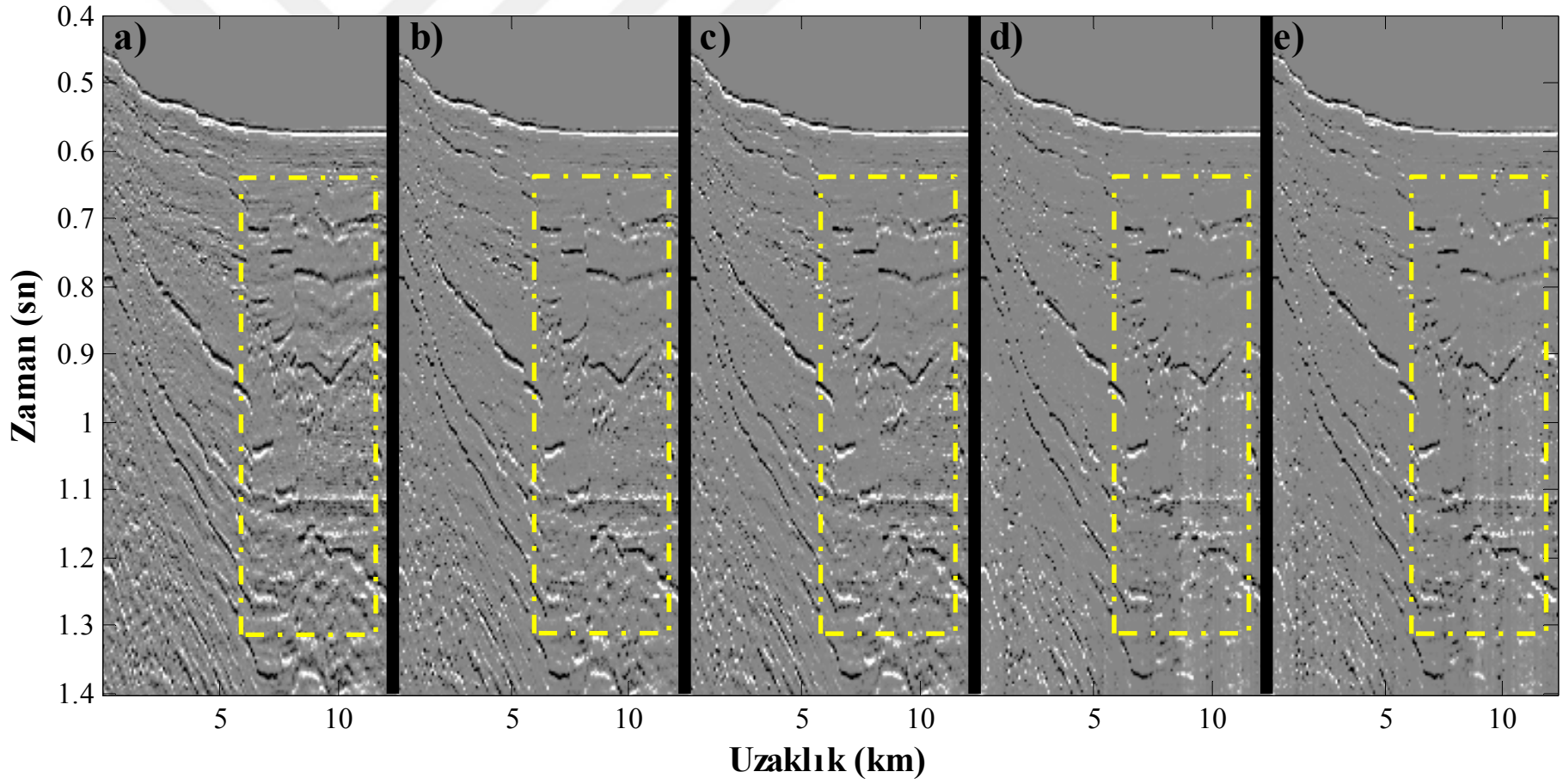
Şekil 4. 23. Arazi verisi-1 Modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



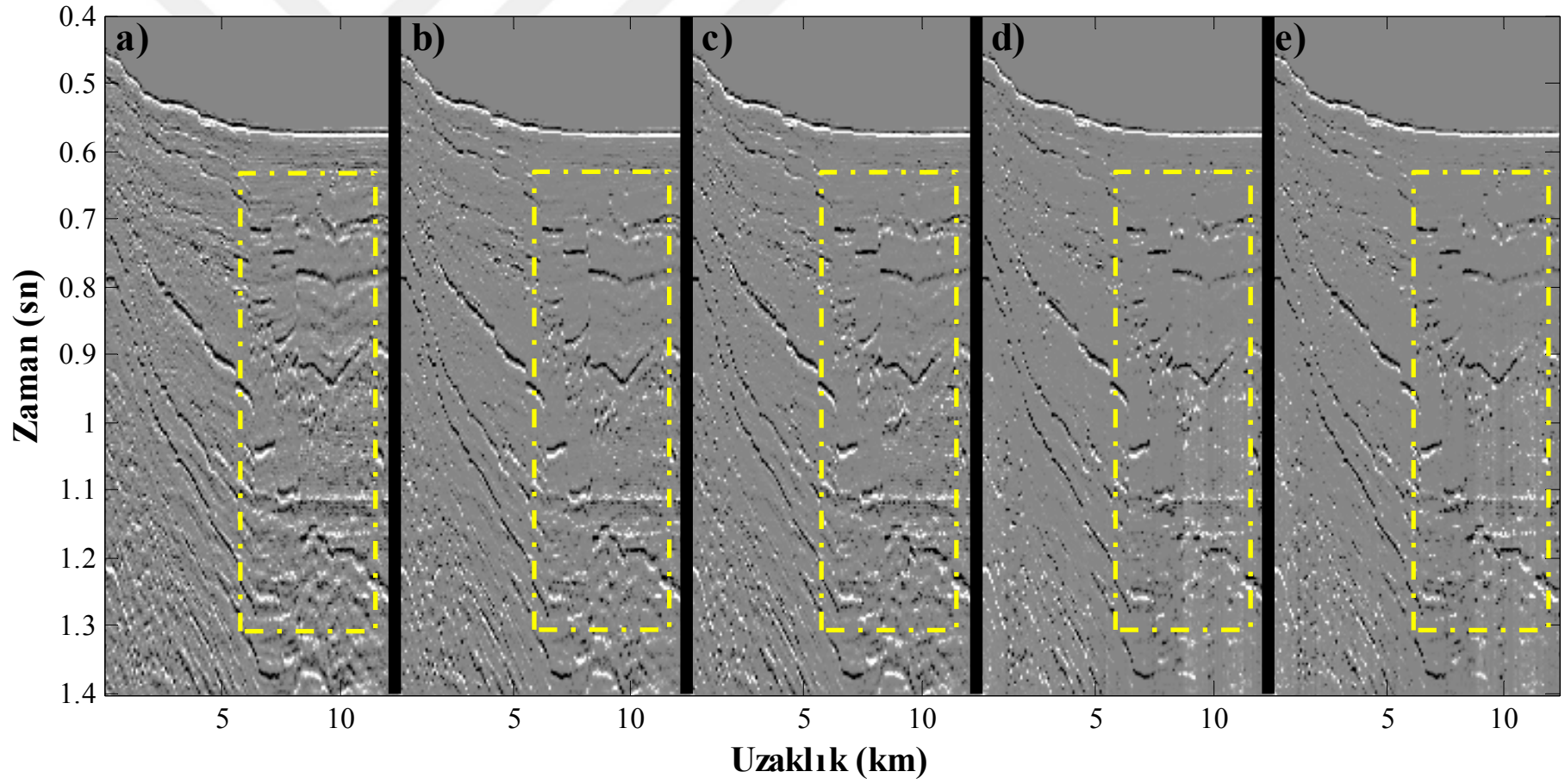
Şekil 4. 24. Arazi verisi-1 için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(dd^T)^{-1}$, d) $cov((dd^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.



Şekil 4. 25. Arazi verisi-2 için, a) Kirchhoff göç çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak elde edilen GD çıkışları.



Şekil 4. 26. Arazi verisi-2 Modeli için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak eşlenik-gradyent yöntemi ile elde edilen EKKGD çıkışları.



Şekil 4. 27. Arazi verisi-2 için, a) EKKG çıkışı, b) $(L^T L)^{-1}$, c) $(d d^T)^{-1}$, d) $cov((d d^T)^{-1})$ ve e) $cov((L^T L)^{-1})$ netleştirme fonksiyonu kullanılarak gauss-seidel yöntemi ile elde edilen EKKGD çıkışları.

5. TARTIŞMALAR VE SONUÇLAR

Sismik yansıma yönteminin önceliği yeraltının yorumlanabilir bir görüntüsünü elde etmektir. Bunun için, yorumlanacak olan sismik veriyi yeraltı jeolojisi ile uyumlu bir hale getirecek olan en önemli veri-işlem adımı göç işlemidir. Sismik göç işlemi veri toplama sırasındaki bozuklukları (acquisition footprints) gidererek yansıma genliklerini gerçek yeraltı konumlarına taşıırken, kendi doğası ile ilişkili gürültüler (artifacts) oluşturmaktadır. Bu gürültüleri gidermek veya bastırmak için farklı yöntemler mevcuttur. Bunlar Göç Dekonvolüsyonu (GD), En Küçük Kareler Göçü (EKKG) ve bu iki yöntemin birlikte kullanılması, yani En Küçük Kareler Göç Dekonvolüsyonu (EKKGD) işlemleridir. Ancak tüm bu yöntemlerin ortak dezavantajları yüksek maliyetli olmasıdır. Fakat fayda/maliyet bakımından en iyi yöntemin hangisi olduğu araştırılmaktadır.

Göç işleminin maliyetine etki eden etmenler sırasıyla veri boyutu, yarı operatör boyu ve hız fonksiyonudur. Yapılan hız ve yarı operatör boyu testlerinden yola çıkarak; yüksek hızlarda göç işlemi daha hızlı çalışırken, bunun tam aksine uzun operatör boylarında çok daha yavaş çalışmaktadır. Bunun yanında, Kuhl ve Sacchi (2003) hız modelinde küçük hatalar olduğu zamanda bile EKKG' nün yüksek kaliteli genlik bilgisini sağlayabildiğini göstermişlerdir. Uygulamalarda elde edilen göç çıkışlarına ve hata oranlarına bakıldığında EKKG, GD ve EKKGD işlemleri daha kaliteli sonuçlar ürettiği açıkça görülebilmektedir. Ayrıca, yüksek hız bilgisi göç çıkış kalitesini, düşük hız bilgisine göre daha çok etkilemektedir. Yarı operatör boyu sismik yığıma kesitlerindeki saçılma olaylarından veya eğimli yapılar göz önüne alınarak belirlenir. Yarı operatör boyunun artması direkt olarak işlem zamanına etki etmektedir. Yarı operatör boyunun büyük seçilmesi sonuç kesitlerinin kalitesi açısından herhangi bir sorun teşkil etmese de işlem zamanını oldukça arttırabilir. Bu nedenle sismik yığıma kesitleri iyi analiz edilerek optimum yarı operatör boyu belirlenmelidir.

GD ve EKKG aynı parametreleri kullanarak çözüme ulaşmaktadırlar. Basit modellerden hesaplanan hata oranlarına dikkat edildiğinde bu iki yöntemin aynı hata oranlarına sahip olduğu görülmüştür. Yu and Schuster (2003)' e göre GD işlemi EKKG işlemine oranla daha az maliyetli olduğunu belirtmişlerdir. GD ile EKKG yöntemi arasındaki temel fark $(L^T L)^{-1}$ in hesaplanmasıdır. Bunun için GD yöntemi doğrudan matrisin tersini alırken, EKKG yöntemi yinelemeli yöntemler kullanarak matrisin tersini

alabilmektedir. Bu şekilde EKKG yöntemi maliyet bakımından GD yöntemine oranla daha avantajlı olduğu söylenebilir. Yapılan uygulamalar göstermiştir ki, en fazla yineleme sayısının veri boyutuna (zaman örnek sayısı) eşit olduğunda GD ve EKKG işlemlerinin çözüm zamanları yaklaşık eşittir. GD işleminde matrisin tersini almak için farklı teknikler kullanıldığında veya EKKG işlemi için en fazla yineleme sayısı düşürüldüğünde bu oran değişebilir.

Aoki ve Schuster (2009) EKKG işleminin maliyetini azaltmak için netleştirme fonksiyonu kullanarak yineleme sayısını azaltmayı başarmıştır. Fakat burada dikkat edilmesi gereken husus fazladan hesaplanan $(L^T L)^{-1}$ fonksiyonunun iki kez art arda tersinin alınmasıdır. Bu çalışmada $(L^T L)^{-1}$ fonksiyonu EKKG için yinelemeli olarak çözülürken, netleştirme fonksiyonu için doğrudan çözülmüştür. Bu durumda, yapılan uygulamaların çoğunda (fay ve tuz modelleri hariç) EKKGD işleminin yineleme sayılarını azalttığı görülmüştür. Fakat yineleme sayısının azalması çözüm maliyetini düşürmemiş, aksine yaklaşık 1.75 kat arttırmıştır. Bununla birlikte EKKGD çıkışları EKKG çıkışlarına göre hata oranlarını nokta saçılma modeli için %20, papyon-kravat modeli için %6 ve fay modeli için %4 oranında azalttığı görülmüştür.

EKKG yönteminin çözümünde yinelemeli çözüm algoritmaları eşlenik-gradyent ve gauss-seidel sıklıkla kullanılmaktadır. Bu çalışma için gauss-seidel yönteminin matris formu kullanılmıştır. Gauss-seidel matris formunda çözüldüğünde matrisin satır veya sütun sayısına eşit olan sabit yineleme sayısına sahiptir. Ayrıca eşlenik-gradyent algoritması için en fazla yineleme sayısı matrisin satır veya sütun sayısına eşit olarak kullanılmıştır. Yapılan ilk uygulamalarda eşlenik-gradyent algoritması için sonlandırma kriteri olarak çok küçük bir değer olarak görülen $(\varepsilon = 2.2204e - 16)$ sayısı kullanılmıştır. Fakat bazı uygulamalar için gauss-seidel yönteminden farklı sonuçlar verdiği sentetik veriler üzerindeki testlerden tespit edilince sonlandırma kriteri daha da küçültülerek $(\varepsilon = 4.9304e - 32)$, bu çalışmada kullanılan katsayı) kullanılmıştır. Sonlandırma kriteri bu şekilde belirlendiğinde daha maliyetli olması beklenen gauss-seidel yöntemi, eşlenik-gradyent yöntemine göre çözüm zamanını (maliyeti) yaklaşık %3.2 düşürmüştür (papyon-kravat modeli hariç). Papyon-kravat modeli için ise eşlenik-gradyent yöntemi yaklaşık %9 daha az zamanda çözüme ulaşmıştır.

Düzenleme parametresi GD, EKKG ve EKKGD yöntemlerinin en önemli özelliğidir. Öyle ki, düzenleme parametresi kullanılmadan gerçekleştirilen bu yöntemler Kirchhoff göçü ile aynı çıkışları vermektedirler. Bu çalışmada kullanılan Cauchy düzenleme

parametresi hem model hem de gerçek veriler için makul sonuçlar üretmiştir. Bunun yanında, model veriler için mükemmele yakın sonuçlar üreten L1 düzenleme parametresi ise gerçek veriler için Cauchy düzenleme parametresine göre başarısız olmuştur. Bu nedenle çalışma boyunca Cauchy düzenleme parametresi kullanılmıştır.

Uygulamalarda $(L^T L)^{-1}$ ve $(dd^T)^{-1}$ netleştirme fonksiyonları aynı hata oranlarını vermelerinin yanında yineleme sayılarını da yaklaşık aynı oranda azaltmışlardır. Bu iki netleştirme fonksiyonu fay ve tuz modelleri hariç tüm veri setleri için yineleme sayılarını azaltmayı başarmışlardır. Bunun yanında $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ netleştirme fonksiyonları fay ve tuz modelleri dâhil tüm modeller için yineleme sayılarını en çok azaltan fonksiyonlar olmuştur.

Göç operatörü matrisinden (L) türetilen netleştirme fonksiyonları çok sayıda eleman içerdiğinden ve bu elemanların birçoğunun sıfır olmasından dolayı çözüm zamanını arttırmaktadır. Giriş verisinden (d) türetilen netleştirme fonksiyonları kullanıldığında GD için %80, EKKGD için ise yaklaşık %40 oranında bir zaman tasarrufu sağlanmıştır.

Uygulamalarda kullanılan bütün netleştirme fonksiyonları başarılı olamamıştır. Buna karşılık, geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ tüm uygulamalarda başarılı olmuştur. EKKGD ile birlikte kullanıldığında daha da başarılı olmuştur. $(L^T L)$ netleştirme fonksiyonu, göç hatası oranlarını arttırarak beklenen etkinin uzağında kalmıştır. $cov(L^T L)$ netleştirme fonksiyonu kusurlu sonuçlar vermemesine rağmen hata oranını beklenen derecede azaltamamıştır. (dd^T) ve $cov((dd^T)^{-1})$ fonksiyonları en küçük göç hatalarına sahip olsalar da kusurlu göç çıkışları ürettikleri görülmüştür. $(dd^T)^{-1}$ fonksiyonu ise GD için yaklaşık %20 maliyetle, EKKGD için yaklaşık %65 maliyet ile geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ile birebir aynı (eşit hata oranları) göç çıkışlarını ürettiği görülmüştür. En küçük hata oranlarına sahip olan netleştirme fonksiyonları $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ dir.

Sonuç kesitlerinden, hata oranlarından ve çözüm zamanlarından yola çıkarak en başarılı netleştirme fonksiyonları $cov((dd^T)^{-1})$ ve $cov((L^T L)^{-1})$ dir. Bu fonksiyonlardan sonra geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ ve $(dd^T)^{-1}$ netleştirme fonksiyonu gelmektedir. Bu iki çift netleştirme fonksiyonu sonuç kesitlerine bakıldığında benzer çıkışları, hata oranlarına bakıldığında aynı hata oranlarını vermiştir. Netleştirme fonksiyonlarını oluştururken giriş verisinden (d) yararlanmak GD için yaklaşık %80, EKKGD için yaklaşık %35 zaman tasarrufu sağlanabileceği ortaya çıkmıştır.

Sonuç olarak, GD veya EKKG işlemlerini kullanmak göç hatalarını Kirchhoff göç çıkışına oranla oldukça azalttığı görülmüştür. Geleneksel netleştirme fonksiyonu yerine aynı hata oranlarına sahip olan $(dd^T)^{-1}$ netleştirme fonksiyonunun kullanılması yaklaşık büyük oranda zaman kazancı sağlamaktadır. Geleneksel netleştirme fonksiyonu $(L^T L)^{-1}$ veya $(dd^T)^{-1}$ kullanmak yerine $cov((L^T L)^{-1})$ veya $cov((dd^T)^{-1})$ netleştirme fonksiyonları kullanılarak hata oranını yaklaşık %15 daha fazla düşürmüştür. Ancak $cov((L^T L)^{-1})$ yerine $cov((dd^T)^{-1})$ netleştirme fonksiyonunu kullanmak yaklaşık %45 zaman kazancı sağlayacaktır. Buna göre, Kirchhoff göç çıkışına oranla çok düşük hata oranlarından birini üreten ve yine Kirchhoff göç çıkışına göre yaklaşık 1.24 kat daha fazla zamanda çözüme ulaşan $cov((dd^T)^{-1})$ netleştirme fonksiyonu kullanılarak gerçekleştirilen GD işlemi göç hatalarını bastırmak için en iyi göç yöntemi olarak belirlenmiştir.

KAYNAKLAR

- Aoki, N., ve Schuster G., 2009. Fast least-squares migration with a deblurring filter, Geophysics, 74, 6.
- Baysal, E., Kosloff, D. ve Sherwood, J., W., C., 1983. Reverse time migration, Geophysics, 48, 1514-1524.
- Berkhout, A., J., 1980. Seismic Migration, Elsevier, New York.
- Berryhill, J., R., 1979. Wave-equation datuming, Geophysics, 44, 1329–1344.
- Claerbout, J., F., 1985. Imaging the Earth's Interior, Blackwell Scientific Publications.
- Claerbout, J., F., ve Doherty, S., M., 1972. Downward continuation of moveout-corrected seismograms, Geophysics, 37, 741–768.
- Clapp, M., L., 2005. Imaging under salt: Illumination compensation by regularized inversion, Doktora Tezi, Stanford University, U.S.A.
- Dai, W., Boonyasiriwat C., ve Schuster G. T., 2010. 3D multisource least-squares reverse time migration, SEG 80th Annual International Meeting, Expanded Abstracts, 29, 3120–3124.
- Dai, W., Wang, X. ve Schuster, G., T., 2011. Least-squares migration of multisource data with a deblurring filter, Geophysics, 76, 5, 135–146.
- Dondurur, D., 2009. Deniz Sismiğinde Veri İşlem, TMMOB Jeofizik Mühendisleri Odası, 294 sayfa, ISBN 978-9944-89-751-8.
- Drineas, P., Mahoney M., W., Muthukrishnan S. ve Sarlos T., 2011. Faster Least Squares Approximation, Numerische Mathematik, 117, 2, 219–249.
- Ganssle, G., 2015. Stabilized Least Squares Migration, Doktora Tezi, University of New Orleans, Theses and Dissertations.
- Gazdag, J., 1978. Wave equation migration with the phase-shift method, Geophysics, 43, 1342–1351.
- Gureli, O., 2007. İntegral (Kirchhoff) Göçünde Doğru Genlikli İşlecin Belirlenmesi, Doktora Tezi, Ankara Üniversitesi.
- Gray, S., H., Etgen, J., Delinger, J. ve Whitmore, D., 2001. Seismic migration problems and solutions, Geophysics, 66, 5, 1622-1640.
- Hu, J. ve Schuster G., T., 1998. Migration Deconvolution, Proceedings of SPIE - The International Society for Optical Engineering, 3453, 118-124.

- Hu, J., Schuster G., T. ve Valasek, A., 2001. Poststack Migration Deconvolution, Geophysics, 66, 3, 939-952.
- Huo, J., 2015. Accelerating Extended Least Squares Migration with Conjugate Gradient Iteration, SEG New Orleans Annual Meeting, 4243-4248.
- Kanbur, Z., 2002. Yığma Öncesi Migrasyon Tekniği ile Marmara Denizi Tekirdağ Havzası Yansıma Verilerinin Yorumlanması, Doktora Tezi, 121158, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Kaplan, S., T., 2010. Regularized wave equation migration for imaging and data reconstruction, Doktora Tezi, University of Alberta.
- Kuhl, H. ve Sacchi, M., D., 2003. Least-squares wave-equation migration for AVP/AVA inversion, Geophysics, 68, 262-273.
- Li, Z., H., Wang Y., F. ve Yang C., C., 2011. A fast global optimization algorithm for regularized migration imaging, Chinese Journal Geophysics, 54, 367-374.
- Luo, G., 2006. Migration Deconvolution Applied to Marine Seismic Data from Campeche Bay: Gulf of Mexico, SEG Annual Meeting, 2599- 2603.
- Luo, S. ve Hale, D., 2014. Least-squares migration in the presence of velocity errors. Geophysics, 79, 4, 153-161.
- Nemeth, T., Wu, C. ve Schuster, G., T., 1999. Least-squares migration of incomplete reflection data, Geophysics, 64, 208-221.
- Mayne, W., H., 1962. Common reflection point horizontal data stacking techniques, Geophysics, 27, 927-938.
- McMechan, G., A., 1983. Migration by extrapolation of time-dependent boundary values, Geophysical Prospecting, 31, 413-420.
- Oruç, B., 2006. Teori ve Örneklerle Jeofizikte Modelleme, Kocaeli Üniversitesi Yayınları.
- Rebollo, R., C. ve Sacchi, M., D., 2010. Time Domain Least Squares Prestack Migration, GeoCanada-Working with the Earth.
- Sacchi, M., D., 1997. Reweighting Strategies in Seismic Deconvolution, Geophysics 129, 651-656.
- Schmidt, M., 2005. Least Squares Optimization with L1-Norm Regularization, Report, The University of British Columbia.
- Schneider, W., A., 1971. Developments in seismic data processing and analysis (1968-1970), Geophysics, 36, 1043-1073.

- Schneider, W., 1978. Integral formulation for migration in two and three dimensions, Geophysics, 43, 49-76.
- Schuster, G., T., 1993, Least-squares crosswell migration, SEG 63rd Annual International Meeting, Expanded Abstracts, 12, 25–28.
- Schuster, G., T. ve Hu, J., 2000. Green's function for migration: Continuous recording geometry, Geophysics, 65, 1, 167-175.
- Schuster, G., T, 2010. Basic of seismic imaging, Cambridge University (ISBN: 9780521871242), U.K.
- Statton, A., 2013. Least squares Kirchhoff depth migration with preconditioning, Doktora Tezi, University of Alberta.
- Stolt, R., H., 1978. Migration by Fourier transform, Geophysics, 43, 23-48.
- Tang, Y., 2008. Wave-equation Hessian by phase encoding, SEG 78th Annual International Meeting, Expanded Abstracts, 27, 2201–2205.
- Tang, Y., ve Biondi, B., 2009. Least-squares migration/inversion of blended data, SEG 79th Annual International Meeting, Expanded Abstracts, 28, 2859–2863.
- Tarantola, A., 1984. Linearized inversion of seismic reflection data, Geophysical Prospecting, 32, 998-1015.
- Tibshirani, R., 1994. Regression shrinkage and selection via the lasso, Technical Report, University of Toronto.
- Topçu, A., 2014. Bilgisayar Destekli Numerik Analiz Ders Notları, Eskişehir Osmangazi Üniversitesi, İnşaat Mühendisliği Bölümü.
- Valenciano, A., A., Biondi, B. ve Guitton, A., 2006. Target-oriented wave-equation inversion, Geophysics, 71, 4, A35–A38.
- Valenciano, A., 2008. Imaging by Wave-equation Inversion, Doktora Tezi, Stanford University.
- Wang, J. ve Sacchi, M., D., 2007. High-resolution wave equation AVP imaging with sparseness constraints, Geophysics, 72, 1, 11-18.
- Wang, Y., F., Yang, C., C. ve Duan, Q., L., 2009. On iterative regularization methods for migration deconvolution and inversion in seismic imaging, Chinese Journal of Geophysics, 52, 704–715.
- Wang, Y. ve Yang, C., 2010. Accelerating migration deconvolution using a nonmonotone gradient method, Geophysics, 75, 131-S137.

- Wei, H. ve Zhou, H., W., 2014. Stochastic conjugate gradient method for least square seismic inversion problems. SEG Denver 2014 Annual Meeting. Expanded Abstracts, 4003-4007.
- Whitmore, N., D., 1983. Iterative depth migration by backward time propagation, 53rd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 382–385
- Wu, S., Ma Y., Wang, Y. ve Xu, C., 2016. Super-Resolution Least-Squares Kirchhoff Prestack Migration, SEG International Exposition and 86th Annual Meeting, 4273-4277.
- Yilmaz, O., 2001. Seismic Data Analysis, Society of Exploration Geophysics, USA.
- Yousefzadeh, A. ve Bancroft, J., C., 2012a, Kirchhoff imaging in a matrix form Part I: modeling, GeoConvention 2012, Expanded Abstract,
- Yousefzadeh, A. ve Bancroft, J., C., 2012b, Kirchhoff imaging in a matrix form Part II: least squares migration, GeoCanada 2012, Expanded Abstract,
- Yousefzadeh, A., 2013, High Resolution Seismic Imaging usin Least Squares Migration, University of Calgary, PhD Thesis.
- Yu, J. ve Schuster, G.,T., 2003. Migration Deconvolution vs. Least Squares Migration, 2003 SEG Annual Meeting.
- Yu, J., 2004. 3-D Migration Deconvolution in Depth Domain: Field Data Example, 74th SEG Annual Meeting.
- Yu, J., Hu, J., Schuster, G., T., ve Estill, R., 2006. Prestack Migration Deconvolution, Geophysics, 71.
- Zhan, G. ve Schuster, G., T., 2010, Skeletonized least-squares wave equation migration, 80th SEG Annual International Meeting, Expanded Abstracts, 29, 3380–3384.
- URL1, http://www.ahay.org/RSF/book/data/segsalt/paper_html/paper.html
- URL2, https://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method

EKLER

EK-A

EŞLENİK-GRADYENT (CONJUGATE-GRADIENT) YÖNTEMİ

Eşlenik-gradyent yöntemi jeofizikte ters çözüm için ideal bir yöntemdir. Yöntem, başlangıç noktası ne olursa olsun daha az sayıda daha etkili bir optimizasyonu hedefler (Oruç, 2006). Algoritmanın basit formu ardışık olarak gradyentin ve bir önceki adımın düzlemine bulmaktır (Clearbout, 1985). Eşlenik-gradyent yöntemi en küçük kareler çözümünde de kullanılabilir (Oruç, 2006). $d = Gm$ denklem sistemi düzenlenecek olursa. Rezidüel;

$$r = d - Gm \quad (A.1)$$

olur. Çözüme ardışık olarak ulaşmak için her adımda bulunan rezidüel (r_i), çözüme (m_i) eklenir.

$$m_{i+1} = m_i + \alpha r_i \quad (A.2)$$

Burada α , arama doğrultusunda adım büyüklüğüdür. Denklem A.2 α nın fonksiyonu olarak yazılırsa;

$$F(\alpha) = \Phi(m_i + \alpha r_i) \quad (A.3)$$

olmak üzere. Burada α katsayısını bulmak için denklem A.2'nin türevi alınıp sıfıra eşitlenir ve α denklemini türetilebilir.

$$\begin{aligned} \frac{\partial F}{\partial \alpha} &= \nabla \Phi(m_i + \alpha r_i) \cdot r_i \\ 0 &= [G(m_i + \alpha r_i) - d] \cdot r_i \\ 0 &= (Gm_i + \alpha Gr_i - d) \cdot r_i \\ 0 &= (-r_i + \alpha Gr_i) \cdot r_i \\ \alpha &= \frac{r_i \cdot r_i}{r_i \cdot Gr_i} \end{aligned} \quad (A.4)$$

Buraya kadar olan kısım En Dik İniş (Steepest Descent) yöntemine karşılık gelmektedir. Yani her bir yinelemede adım büyüklüğü (α) hesaplanarak, negatif gradyent yönünde, minimum değere ulaşıncaya kadar devam eden yinelemeler sonucunda çözüme ulaşılabilir. Fakat yineleme sayısını azaltmak için her çözümde bir önceki adımın düzlemi bulunarak çözüme daha az yineleme ile ulaşılabilir. Düzlemin bulunabilmesi, rezidüele (r) dik bir doğrultunun bulunması anlamına gelir. Fakat bunun için bir doğrultunun bulunmuş olması gerekir. Bu nedenle En Dik İniş yöntemi ile eşlenik-gradyent yöntemi ilk adımda birbirinin aynısıdır. Dolayısıyla ikinci doğrultu (p) başlangıçta rezidüele (r) eşit kabul edilir.

$$\begin{aligned} p_1 &= (m_n - m_1) \\ p_0 &= r_0 = d - Gm \end{aligned} \quad (\text{A.5})$$

p_0 ve r_1 birbirlerine dik ise iç çarpımlarının sıfır olması gerekir. Böylelikle;

$$\begin{aligned} 0 &= p_0^T r_1 \\ 0 &= p_0^T (d - Gm_1) \\ 0 &= p_0^T (Gm_n - Gm_1) \\ 0 &= G(m_n - m_1) \\ 0 &= p_0^T Gp_1 \end{aligned} \quad (\text{A.6})$$

Buradan p_1 ' in p_0 ' a göre G dikliğinde olduğu; yani p_0 ' a göre eşlenik doğrultuda olduğu anlaşılır (Oruç, 2006). Şimdi p_1 değişkeni, r_1 ve p_0 kullanılarak;

$$p_1 = r_1 - \beta_1 p_0 \quad (\text{A.7})$$

biçiminde yazılabilir. p_1 denklem A.6' da yerine yazılırsa;

$$\begin{aligned} 0 &= p_0^T G(r_1 - \beta_1 p_0) \\ \beta_1 &= \frac{p_0^T G r_1}{p_0^T G p_0}, \text{ yani, } \beta_{n+1} = \frac{p_n^T G r_{n+1}}{p_n^T G p_n} \end{aligned} \quad (\text{A.8})$$

denklemleri elde edilir. Eşlenik-gradyent yönteminin ardışık uygulanması için adım boyu (α) değişken dönüşümü yapılarak ($p_0 = r_0$) yeniden yazılırsa;

$$\alpha = \frac{r_i^T r_i}{p_i^T G p_i} \quad (\text{A.9})$$

olur. Bu durumda ardışık çözüm başlangıç tanımlamaları ($r = d - Gm = p$ ve $x_0 = 0$) yapıldıktan sonra;

$$1) \alpha_i = \frac{r_i^T r_i}{p_i^T G p_i}$$

$$2) m_i = m_{i-1} + \alpha_i p_i$$

$$3) r_{i+1} = r_i - \alpha_i G p_i$$

eğer $\sqrt{r_i^T r_i} < \varepsilon$ ise sonlandır

$$4) \beta_{i+1} = \frac{p_i^T G r_{i+1}}{p_i^T G p_i}$$

$$5) p_{i+1} = r_{i+1} - \beta_{i+1} p_i$$

olmak üzere çözüme ulaşana kadar devam eder.

EK-B GAUSS-SEİDEL YÖNTEMİ

Jacobi yönteminin düzenlenmiş formu olan bu yöntem çok fazla yineleme gerektirmeden sonuca ulaşabilmektedir. $Gm = d$ şeklindeki bir denklem sistemi;

$$\begin{aligned} g_{11}m_1 + g_{12}m_2 + \dots + g_{1n}m_n &= d_1 \\ g_{21}m_1 + g_{22}m_2 + \dots + g_{2n}m_n &= d_2 \\ \vdots & \\ g_{n1}m_1 + g_{n2}m_2 + \dots + g_{nn}m_n &= d_n \end{aligned} \quad (\text{B.1})$$

şeklinde yazılabilir. Burada g_{ij} , G matrisinin elemanlarıdır. Bu tanımlamalar ile m vektörünün elemanları herhangi bir başlangıç (genelde sıfırdır) ile çözüme götürülür.

$$\begin{aligned} m_1 &= \frac{1}{g_{11}}(d_1 - g_{12}m_2 - g_{13}m_3 - \dots - g_{1n}m_n) \\ m_2 &= \frac{1}{g_{22}}(d_2 - g_{21}m_1 - g_{23}m_3 - \dots - g_{2n}m_n) \\ &\vdots \\ m_n &= \frac{1}{g_{nn}}(d_n - g_{n2}m_2 - g_{n3}m_3 - \dots - g_{nn}m_n) \end{aligned} \quad (\text{B.2})$$

Tek bir denklem halinde yazılırsa (Jacobi Yöntemi):

$$m_i^{k+1} = \frac{1}{g_{ii}}(d_i - \sum_{i \neq j} g_{ij}m_j^k) \quad (\text{B.3})$$

Burada k yineleme numarasını belirtir. Her bir yenileme de çözüme yakınsar ve sonlandırma şartı ile sonlandırılır (Topçu, 2014). Gauss-seidel yöntemi ise her bir yineleme içerisinde, bulunan m vektörünün elemanını diğer elemanın hesaplanmasında kullanılır. Böylece yineleme sayısını azaltır. Gauss-seidel Yöntemi:

$$m_i^{k+1} = \frac{1}{g_{ii}}(d_i - \sum_{j < i} g_{ij}x_j^{k+1} - \sum_{j > i} g_{ij}x_j^k) \quad (\text{B.4})$$

şeklinde yazılabilir. Bunun yanında matris formu ile de yazmak mümkündür. Bu durumda G matrisi $G = A_* + U$ şeklinde ayrılabilir. Burada A_* ve U

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}, A_* = \begin{bmatrix} g_{11} & 0 & \cdots & 0 \\ g_{21} & g_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix} U = \begin{bmatrix} 0 & g_{12} & \cdots & g_{1n} \\ 0 & 0 & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

şeklinde ifade edilir. Doğrusal denklem sistemi (url2):

$$A_* m = d - Um \quad (\text{B.5})$$

şeklinde yazıldıktan sonra bir sonraki çözümü verecek denklem sistemi;

$$m^{k+1} = A_*^{-1}(d - Um^k) \quad (\text{B.6})$$

şeklinde yazılır.

EK-C

EN KÜÇÜK KARELER GÖÇ DEKONVOLÜSYONU MATLAB KODU

```

function [out,tused,it]=ekkgd(veri,v,dx,dt,l,opt,iton,opt2,aci,katsayi,kq,is,sc)
% Bu program Kirchhoff Zaman Göçü Bağınıtısı ile
% Kirchhoff zaman göçü opt=0, iton=0, opt2=0
% Kirchhoff zaman göçü dekonvolüsyonu, opt=1-8, iton=0
% En Küçük kareler Kirchhoff Zaman Göçü Dekonvolüsyonu opt=1-4
% opt, ağırlıklandırma (netleştirme) fonksiyonu
% opt ==0, EKKG
% opt ==1, (L'L)^-1
% opt ==2, (L'L)
% opt ==3, cov(L'L)
% opt ==4, cov((L'L)^-1)
% opt ==5, (dd')^-1
% opt ==6, (dd')
% opt ==7, cov(dd')
% opt ==8, cov((dd')^-1)
% opt2, düzenleme parametresi
% opt2 ==0, düzenleme parametresi yok
% opt2 ==1, L1 Norm
% opt2 ==2, L2 Norm
% opt2 ==3, Cauchy Norm
% iton, iteratif çözüm yöntemi
% iton ==1, Eşlenik-Gradyent
% iton ==2, Gauss-Seidel Matris Formu
% l      : operatör yarı boyu (aperture length) (1,1)
% dt     : zaman örnekleme aralığı (1,1)
% dx     : Uzaklık Örnekleme aralığı (1,1)
% v      : Hız Fonksiyonu (m,n)
% veri   : Veri (m,n)
% katsayi : Norm hesabında paydaya eklenen katsayı
% kq     : Düzenleme parametresinin katsayısı
% is     : Max. yineleme sayısı
% sc     : ölçekleme parametresi
[a,b]=size(veri);
if nargin < 13; sc=1;end
if nargin < 12; is=a;end
if nargin < 11; kq=10000;end
if nargin < 10; katsayi=0.001;end
if nargin < 9; aci=75;end
if nargin < 8; opt2=0;end
if nargin < 7; iton=0;end
if nargin < 6; opt=0;end
tstart=clock;
l=round(1/dx);
x=-l*dx:dx:l*dx;
x2=x.^2;
oen=2*l+1;
oboy=1+round(1*dx/min(min(v))/dt*5);
veril=zeros(a+oboy,b+2*1);
veril(1:a,l+1:b+l)=veri;
t=dt:dt:a*dt;
sinir=cos(aci*pi/180);
opr=zeros(a+oboy,oen);
el=0:size(opr,1):size(opr,1)*(oen-1);
iti=[];
for j=1:b
    d=veril(:,j:j+oen-1);
    L=zeros((a+oboy)*oen,a);
    for i=1:a
        % saçılma operatörü hesaplanıyor
        op=sqrt(t(i)^2+4*x2./v(i,j)^2);
        opl=round(op./dt)+el;
    end
end

```

```

        % Kirchhoff katsayısı (cos(theta))
        kan=t(i)./op;
        % sınır açısından büyük açılar sıfırlanıyor
        kan(kan<sinir)=0;
        % hesaplanan katsayılar L matrisinde yerine yazılıyor
        L(opl,i)=kan;
    end
% Kirchhoff zaman göçü
mmig=L'*d(:);
if opt==0 && iton==0
    out(:,j)=mmig;
    iti(j)=1;
else
    % düzenleme parametresi hesaplanıyor
    if opt2==1
        q=(abs(mmig)+katsayi).^-1;
    elseif opt2==2
        q=(mmig.^2+katsayi).^-1;
    elseif opt2==3
        q=log(1+abs(mmig).^2./max(abs(mmig))^2+katsayi).^-1;
    elseif opt2==0
        q=ones(1,a).*(1/katsayi);
    end
    Q=diag(q);
    if opt==0
        if iton==0
            % Kirchoff
            out(:,j)=mmig;
        elseif iton==1
            % EKKG Eşlenik Gradyent
            [out(:,j),iti(j)]= cg(L'*L+Q.*kq^2,mmig,is);
        elseif iton==2
            % EKKGD Gauss Seidel
            out(:,j)= gauss_seidel(L'*L+Q.*kq^2,mmig);
            iti(j)=std(mmig);
        end
    else
        % netleştirme fonksiyonu hesaplanıyor
        switch opt
            case 1
                kk=((L'*L+Q.*kq^2))^-1;
            case 2
                kk=((L'*L+Q./kq^2));
            case 3
                kk=cov((L'*L+Q./kq^2));
            case 4
                kk=cov((L'*L+Q.*kq^2))^-1;
            case 5
                kk=veril(1:a,j:j+oen-1);
                kk=kk(:)';
                kk=repmat(kk,a,1);
                kk=((kk*kk'+Q.*kq^2))^-1;
            case 6
                kk=veril(1:a,j:j+oen-1);
                kk=kk(:)';
                kk=repmat(kk,a,1);
                kk=((kk*kk'+Q./kq^2));
            case 7
                kk=veril(1:a,j:j+oen-1);
                kk=kk(:)';
                kk=repmat(kk,a,1);
                kk=cov((kk*kk'+Q./kq^2));
            case 8
                kk=veril(1:a,j:j+oen-1);
                kk=kk(:)';
                kk=repmat(kk,a,1);
                kk=cov((kk*kk'+Q.*kq^2))^-1;
        end
    end
end

```

```

        case 9
        kk=diag(1./(diag(Q).*kq^2));
        end
        if iton==0
            % GD
            out(:,j)=kk*mmig;
            iti(j)=1;
        elseif iton==1
            % EKKGD Eşlenik Gradyent
            [out(:,j),iti(j)]=cg((L'*L+Q).*kq^2),kk*mmig,is);
        elseif iton==2
            % EKKGD Gauss_Seidel
            out(:,j)=gauss_seidel((L'*L+Q).*kq^2),kk*mmig);
            iti(j)=1;
        end
    end
end
end
disp(['Tamamlanan İz' num2str(j),'/',num2str(b)])
% ölçekleme yapılıyor
if sc==1
    sc=v(:,j).*sqrt(pi.*t)';
    out(:,j)=out(:,j)./sc;
end
end
it=mean(iti);
tused=etime(clock,tstart);
disp(['Total elapsed time ' num2str(tused)])

```

EK-D

EN KÜÇÜK KARELER EŞLENİK-GRADYENT YÖNTEMİ MATLAB KODU

```

function [x,iter]=cg(A,b,mi,sk)
% Bu program Eşlenik-Gradyent yöntemi ile
% matris ters çözümünü gerçekleştirir.
% Kaynak : Jon F. Clearbout SEP-44
% 'Conjugate Gradients for Beginners'
% http://sep.stanford.edu/data/media/public/oldreports/sep44/44_14.pdf
[m,n]=size(A);
if nargin<4; sk=eps^2; end
if nargin<3; mi=m; end
x=zeros(n,1);
s1=zeros(m,1);
s=x;
res=b;
for iter=1:mi
    g=A'*res;
    gg=A*g;
    if iter==1
        alfa=(gg'*res)/(gg'*gg);
        beta=0;
    else
        gdg=gg'*gg;
        sds=s1'*s1;
        gds=gg'*s1;
        determ=gdg*sds-gds*gds+eps*100;
        gdr=gg'*res;
        sdr=s1'*res;
        alfa=(sds*gdr-gds*sdr)/determ;
        beta=(-gds*gdr+gdg*sdr)/determ;
    end
    s=alfa*g+beta*s;
    s1=alfa*gg+beta*s1;
    x=x+s;
    res=res-s1;
    if sum(res)<sk ,break, end
end

```

EK-E**EN KÜÇÜK KARELER EŞLENİK-GRADYENT YÖNTEMİ MATLAB KODU**

```
function [x]=gauss_seidel(a,b)
% Bu program Gauss-Seidel yöntemi ile
% matris ters çözümünü gerçekleştirir.
% Kaynak : Wikipedia
% https://en.wikipedia.org/wiki/Gauss%25E2%2580%2593Seidel\_method
a1=tril(a);
a2=a-a1;
x=zeros(length(a),1);
x=a1^-1*(b-a2*x);
```



ÖZGEÇMİŞ

1984 tarihinde İstanbul'da doğdu. İlk ve orta öğrenimini İstanbul'da tamamladıktan sonra 1998 yılında başladığı Beylikdüzü 75. Yıl Cumhuriyet Lisesinden (Yabancı Dil Ağırlıklı, YDA) 2002 yılında mezun oldu. 2003 yılında Karadeniz Teknik Üniversitesi Jeofizik Mühendisliği Bölümünde Lisans öğrenimine başladı ve 2007 yılında mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesinde Lisansüstü eğitimine başladı. 2009 yılında aynı Üniversiteye Araştırma Görevlisi olarak atandı. 2011 yılında Yüksek Lisans diplomasını aldı ve Doktora programına başladı. 2012 yılında 3 aylık süreyle Delft Technology of University' de ve 2014-2015 yıllarında 9 ay süreyle Miskolc University de misafir araştırmacı (öğrenci) olarak bulundu. 2016 yılında Maden Tetkik ve Arama Genel Müdürlüğü Deniz Araştırmaları Dairesine atandı ve halen burada görev yapmaktadır.

rguney@ktu.edu.tr, recep.guney@mta.gov.tr