İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİKSEL BOŞALMA YOLUNUN SAYISAL ANALİZİ

YÜKSEK LİSANS TEZİ

Elektrik Müh. Cenker ELBİRLER

Anabilim Dalı : ELEKTRİK MÜHENDİSLİĞİ

Programı : ELEKTRİK MÜHENDİSLİĞİ

KASIM 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİKSEL BOŞALMA YOLUNUN SAYISAL ANALİZİ

YÜKSEK LİSANS TEZİ
Elektrik Müh. Cenker ELBİRLER
(504051005)

Tezin Enstitüye Verildiği Tarih : 12 Kasım 2007
Tezin Savunulduğu Tarih : 29 Kasım 2007

Tez Danışmanı : Doç.Dr. Özcan KALENDERLİ
Diğer Jüri Üyeleri Prof.Dr. Kevork MARDİKYAN(İ.T.Ü.)
Doç.Dr. Cevdet IŞIK (İ.T.Ü.)

KASIM 2007

ÖNSÖZ

Yapmış olduğum bu çalışmada, benden yardımlarını hiçbir zaman esirgememiş, en yoğun anlarında dahi bana zaman ayırarak çalışmamı hızla yürütmemi sağlamış olan değerli ve sayın hocam Doç. Dr. Özcan Kalenderli'ye; çalışmamı tamamlayabilmem için bana gerekli zamanı sağlayan, maddi ve manevi anlamda desteklerini esirgemeyen sevgili aileme teşekkürü bir borç bilirim.

Ekim 2007 **Cenker Elbirler**

İÇİNDEKİLER

KISALTMALAR **v**
TABLO LİSTESİ **vi**
ŞEKİL LİSTESİ **vii**
ÖZET **ix**
SUMMARY **x**

1. GİRİŞ **1**
1.1. Amaç 1
1.2. Problem 1
1.3. Kapsam 2

2. YILDIRIM VE OLUŞUMU **3**
2.1. Yıldırımın Oluşumu Hakkında Genel Bilgi 3
2.2. Yıldırım Bulutunun Yüklenmesine İlişkin Teoriler 4
2.2.1. Simpson ve Lemonosow Teorisi 4
2.2.2. Elster ve Geitel Teorisi 4
2.2.3. J.I. Frenkel Teorisi 5
2.3. Yıldırımın Oluşumu 5
2.4. Yıldırım Türleri 7
2.4.1. Yukarıya çıkan yıldırım 7
2.4.2. Aşağıya inen yıldırım 7

3. YILDIRIMIN YOLUNUN ANALİZİ **8**
3.1. Yıldırım Yolunun Analizinde Varolan Çalışmaların Durumu 8
3.2. Analiz Sırasında Karşılaşılan Zorluklar 9

4. YILDIRIM YOLUNUN MODELLENMESİNDE KULLANILAN YÖNTEMLER **10**
4.1. Sızma Teorisi 11
4.2. Rasgele Orta Nokta Kaydırma Algoritması 13
4.3. Laplasyen Büyümesi 15

5. ÖNERİLEN MODELLER VE ÖZELLİKLERİ **22**
5.1. Modeller Hakkında Genel Bilgi ve İşleyiş Mantığı 22
5.2. Sonlu Elemanlar Yöntemi ile Potansiyel Dağılımı Hesabı 23
5.3. Dört Yönlü İlerleme Modeli 28
5.3.1. Benzetimin evreleri ve yapılan hesaplar 29
5.3.2. Yöntemin özeti 42
5.3.3. Sonuçlar 43
5.4. Sekiz Yönlü İlerleme Modeli 43

5.4.1. Benzetimin evreleri ve yapılan hesaplar 41
5.4.2. Yöntemin özeti 51
5.4.3. Sonuçlar 52
5.5. Dört ve Sekiz Yönlü Modelin Karşılaştırılması 52
5.5.1. Benzerlikler 53
5.5.2. Farklılıklar 53
5.6. Yöntemlerde Sonucu Etkileyen Faktörler 57

6. BENZETİM 62
6.1. Matlab Programı Hakkında Genel Bilgi 62
6.2. Modelin Matlab Programı Akış Diyagramı ve Çalışması 63
6.3. Sistem Gereksinimleri 65
6.4. Örnek Benzetim Sonuçları 66
6.4.1. Dört yönlü ilerleme örnekleri 66
6.4.2. Sekiz yönlü ilerleme örnekleri 67
6.5. Değişik Uygulamalar 68
6.5.1. Dört yönlü ilerleme modeli için apartman modeli 69
6.5.2. Sekiz yönlü ilerleme modeli için apartman modeli 70
6.5.3. Sonuçlar 71

7. SONUÇ VE ÖNERİLER 77

KAYNAKLAR 79

EKLER 82

ÖZGEÇMİŞ 100

KISALTMALAR

DBM	: Elektriksel Delinme Modeli (Dielectric Breakdown Model)
RMDA	: Rasgele Orta Nokta Kaydırma Algoritması (Random Midpoint Displacement Algorithm)
SFY	: Sonlu Farklar Yöntemi
MSFY	: Merkezi Sonlu Farklar Yöntemi
2D	: İki boyutlu (2 Dimensional)
3D	: Üç boyutlu (3 Dimensional)
XP	: Experience
GHZ	: Gigahertz
AMD	: İleri Mikro Aygıtlar (Advanced Micro Devices)
MB	: Megabyte

TABLO LİSTESİ

		Sayfa No
Tablo 5.1	Mevcut Sistemde Benzetim Süreleri……..…….	54
Tablo 5.2	Farklı Sistemlerde Benzetim Süreleri…….	55
Tablo 6.1	Dört Yönlü İlerleme Modeli İçin Apartman Denemeleri ……..….	73
Tablo 6.2	Sekiz Yönlü İlerleme Modeli İçin Apartman Denemeleri ……..	74

ŞEKİL LİSTESİ

Sayfa No

Şekil 4.1 : Kablo Yalıtkanında ve Yalıtkan Maddede Ağaçlanma…… 10
Şekil 4.2 : İlk 4 Adım Boyunca 2D Sızma Fraktalına İlişkin Büyüme……. 11
Şekil 4.3 : RMDA'nın Yıldırım Modeline Uydurulmuş Şekli……... 14
Şekil 4.4 : Benzetim İçin Değişik Yük Biçimleri…….……. 16
Şekil 4.5 : Değişik Yük Biçimlerine Göre Benzetim Sonuçları……..... 17
Şekil 5.1 : Neumann Sınır Koşulu Örneği……. 25
Şekil 5.2 : İki Farklı Noktanın Potansiyel Hesabı…….……. 26
Şekil 5.3 : Neumann Sınır Koşulu'nun SFY Üzerinde Uygulanışı…… 27
Şekil 5.4 : 5 İterasyon Sonucu Hesaplanan Değerler……. 28
Şekil 5.5 : 200 İterasyon Sonucu Hesaplanan Değerler……. 28
Şekil 5.6 : 15x15 Matriste Potansiyel Dağılımı…….…… 30
Şekil 5.7 : Bir Sonraki Adımda Seçilebilecek Noktalar…… 31
Şekil 5.8 : Başka Bir Durum İçin Komşu Noktaların Belirlenmesi..……. 33
Şekil 5.9 : Komşu Noktanın Seçimi……..……. 35
Şekil 5.10 : İlk Adıma İlişkin Potansiyel ve Olasılık Değerleri……...… 36
Şekil 5.11 : 4 Adım Sonunda Potansiyel ve Olasılık Değerleri…...…… 37
Şekil 5.12 : η Faktörünün Olasılık Değerlerine Etkisi ……..… 38
Şekil 5.13 : 5x15'lik Bir Bölgede Üretilen 0–1 Arası Rasgele Sayılar …… 40
Şekil 5.14 : Son Olasılık Durumu …… 41
Şekil 5.15 : Dört ve Sekiz Yönlü İlerlemede Olası Noktalar ……..… 45
Şekil 5.16 : Aday Noktaların Belirlenmesi…….. 47
Şekil 5.17 : Sekiz Yönlü İlerleme Modeli Örneği…….……..…. 48
Şekil 5.18 : Alt ve Çapraz Uçlarda Potansiyel Dağılımı ……..…..……..… 56
Şekil 5.19 : 7x7 ve 5x5 Boyutlarda Potansiyel Dağılımı……. 58
Şekil 5.20 : 7x7 ve 5x5 Boyutlarda Komşu Nokta Olasılıkları …...…… 58
Şekil 6.1 : Modelin Matlab Program Kodu Akış Diyagramı ……..… 64
Şekil 6.2 : Dört Yönlü İlerleme Modeli İçin 15x15 Rasgele İki Örnek…… 66
Şekil 6.3 : Dört Yönlü İlerleme Modeli İçin 64x64 Rasgele İki Örnek ….. 67
Şekil 6.4 : Sekiz Yönlü İlerleme Modeli İçin 15x15 Rasgele İki Örnek…... 67
Şekil 6.5 : Sekiz Yönlü İlerleme Modeli İçin 64x64 Rasgele İki Örnek…... 68
Şekil 6.6 : Yıldırım Biçimi Üzerinde Apartman Modeli……..…. 69
Şekil 6.7 : Dört Yönlü 1 ve 2 Katlı Apartman Uygulaması……..……. 70
Şekil 6.8 : Dört Yönlü 2 ve 2-4 Katlı Apartman Uygulaması.....…. 70
Şekil 6.9 : Sekiz Yönlü 1 ve 2 Katlı Apartman Uygulaması.....…. 71
Şekil 6.10 : Sekiz Yönlü 3 ve 4 Katlı Apartman Uygulaması.....…... 71
Şekil 6.11 : Apartman Modelinde Değişkenlerin Tanımlanması …...……. 72
Şekil B.1 : Şekil 6.2'de Soldaki Sonuca İlişkin Matlab Çıktısı…...…… 94
Şekil B.2 : Şekil 6.2'de Sağdaki Sonuca İlişkin Matlab Çıktısı…….. 94
Şekil B.3 : Şekil 6.3'teki Soldaki Sonuca İlişkin Matlab Çıktısı……. 94
Şekil B.4 : Şekil 6.3'teki Sağdaki Sonuca İlişkin Matlab Çıktısı…….94
Şekil B.5 : Şekil 6.7'teki Soldaki Sonuca İlişkin Matlab Çıktısı……...……. 95

Şekil B.6 : Şekil 6.7'de Sağdaki Sonuca İlişkin Matlab Çıktısı…….. 95
Şekil B.7 : Şekil 6.9'da Soldaki Sonuca İlişkin Matlab Çıktısı……..…… 95
Şekil B.8 : Şekil 6.9'da Sağdaki Sonuca İlişkin Matlab Çıktısı……..…… 95
Şekil B.9 : Şekil 6.10'da Soldaki Sonuca İlişkin Matlab Çıktısı……. 96
Şekil B.10 : Şekil 6.10'da Sağdaki Sonuca İlişkin Matlab Çıktısı……. 96
Şekil B.11 : Şekil 6.11'de Soldaki Sonuca İlişkin Matlab Çıktısı……..96
Şekil B.12 : Şekil 6.11'de Sağdaki Sonuca İlişkin Matlab Çıktısı……. 96
Şekil C.1 : Soldan 3 Konumunda Rasgele 1 Kat Sonucu …….…….. 97
Şekil C.2 : Soldan 3 Konumunda Rasgele 2 Kat Sonucu ……..… 97
Şekil C.3 : Soldan 3 Konumunda Rasgele 3 Kat Sonucu..……. 97
Şekil C.4 : Soldan 3 Konumunda Rasgele 4 Kat Sonucu …….. 97
Şekil C.5 : Soldan 1 Konumunda Rasgele 2 Kat Sonucu …….. 98
Şekil C.6 : Soldan 2 Konumunda Rasgele 2 Kat Sonucu …….……. 98
Şekil C.7 : Soldan 3 Konumunda Rasgele 2 Kat Sonucu …….. 98
Şekil C.8 : Soldan 4 Konumunda Rasgele 2 Kat Sonucu …….……. 98
Şekil C.9 : Soldan 5 Konumunda Rasgele 2 Kat Sonucu …….. 99

ELEKTRİKSEL BOŞALMA YOLUNUN SAYILSAL ANALİZİ

ÖZET

Bu çalışmada, elektriksel boşalma yollarından biri olan yıldırımın izlediği yol modellenmiştir. Modelleme yapılırken, potansiyel dağılımı hesabından, olasılık bağıntılarından, rasgele sayı üretme algoritmalarından ve matris işlemlerinden yararlanılmıştır.

Önerilen modelde, yıldırımın doğasına yakın sonuçlar elde edebilmek için, kullanılan ve türetilen algoritmalarda, süperpozisyon tekniği uygulanmıştır. Öncelikle ortamın elektriksel özellikleri hesaplanmış, daha sonra da ortamın fiziksel özellikleri yaratılmaya çalışılarak hesaba etki etmesi sağlanmış, bu iki özelliğin birbiri üzerine bindirilmesi sonucunda yıldırımın ilerleyeceği yol ortaya konulmuştur.

Çalışmada, dört yönlü ve sekiz yönlü olmak üzere, iki tip ilerleme modeli önerilmiştir. İki modelin de çalışma sistemi ve gerçekleştirdiği adımlar aynıdır. Fark program kodu sırasında yapılan sorgu sayısında ve tekniklerinde ortaya çıkmaktadır. Benzetimler Matlab ortamında genel ve bazı özel koşullar altında gerçekleştirilmiştir. Modellerin bel kemiği sayılabilecek olan potansiyel dağılımı hesapları ise Sonlu Farklar Yöntemi kullanılarak hesaplanmıştır. Potansiyel dağılımına bağlı olarak işleyen modelde, fiziksel koşulları temsil etmesi açısından ise, rasgele sayı üretme algoritmaları ve olasılık hesapları modele katılarak gerçekle daha paralel sonuçların elde edilmesi sağlanmıştır.

Yıldırım, taşıdığı yüksek akım ve gerilim değerleri nedeniyle, başta insanların yaşamı olmak üzere, tüm canlılara karşı tehlike arz etmektedir, ek olarak sistemlere de zarar vererek maddi hasarlara yol açmaktadır. Bu çalışmanın amacı, yıldırımın doğasını daha iyi anlamak, bu sayede de yıldırımdan korunma tekniklerinin daha da gelişmesine olanak sağlayarak bu kayıpların en aza indirilmesini sağlamaktır.

NUMERICAL ANALYSIS OF ELECTRICAL DISCHARGE PATH

SUMMARY

In this study, the path of lightning which is a type of an electrical discharge is modelled. While modeling is being established, the potantial distribution calculation, probability correlations, random number generation algorithms and matrix calculations are used.

In the proposed model, the super-position method is applied in the used and derieved algorithms in order to obtain more realistic results to the nature of lightning. At first step, the electrical properties of the environment are calculated, then the physical specialties is tried to be constituted and accommodated to affect the calculations. Finally, by imbricating these two features, the path of the lightning stroke is produced.
In the study, two types of expansion model is proposed; 4 ways and 8 ways. The running principles and the executed steps are the same way in both models. The differences come out in the number and techiques of enquiries executed.

Simulations are executed in Matlab program under general and some special conditions. The potential distribution that can be accepted as the main and the most inportant part of the model is calculated by using the Finite Difference Method. In the field-dependent executing model, the random number producing algorithms and the probability calculations are adapted to represent the physical conditions, so more overlapping results are obtained.

Carrying high values of current and voltage, at the first place, lightning is hazard to the human life, then to all surrounding livings, additionally it causes finantial costs by damaging the operating systems. The purpose of this study is, to get a better knowledge about the nature of the lightning and so, by giving the opportunity to develop lightning protection techiques and minimize the losses.

1. GİRİŞ

1.1 Amaç

Bu tezin amacı elektriksel boşalmalardan birisi olan yıldırımın, atmosfer içersinde izlediği yolun modellemesini gerçekleştirebilmektir. Tezde, elektriksel delinme modeli (Di-electric Breakdown Model, (DBM)) olarak adlandırılan yöntem kullanılmış, yıldırım biçimine uydurulmuş ve bir dizi algoritma ile sonuca gidilmiştir.

1.2 Problem

Yıldırım, yeryüzünde meydana gelen atmosferik olaylardan en tanınmış olanıdır. Dünyanın var oluşu kadar eski ve doğal bir olay olan yıldırım, doğada meydana gelen tüm olaylarda olduğu gibi bir "denge" oluşturmak için meydana gelir. Yeryüzü ile atmosfer arasındaki elektriksel yük ve alan dağılımının çok büyük farklar oluşturması durumunda meydana gelen yıldırım, bir elektriksel boşalma gerçekleştirerek toprağa doğru yükleri akıtır, iki ortam arasındaki "elektriksel gerginliği" rahatlatır.

Yıldırımın, gök gürültülü fırtınalar tarafından oluşturulduğu bilinmektedir. Doğada bir denge unsuru olan yıldırım ne yazık ki oluşması sırasında ve ardında büyük zararlara neden olabilmektedir. Düştüğü yere ve büyüklüğüne göre değişiklik gösterecek olan bu zararlar, kimi zaman bir veya daha fazla canlının yaşamına mal olabilir, kimi zaman çiftçinin ürününe, kimi zaman bir yapıya zarar verebilir, kullanılmaz hale getirebilir. Benzer şekilde elektrik hatlarına da düşebilen yıldırım, hatların aşırı yüklenmesine, kopmasına, bozulmasına veya şebekeye aşırı yük bindirerek kullandığımız aletlerin bozulmasına, yanmasına neden olabilmektedir.

Yıldırımın izleyeceği yolun, bir başka deyişle yıldırımın meydana geldiği zaman izleyeceği yolların gerçeğe yakın olarak modellenebilmesi, tahmin edilebilmesi ve önceden belirlenebilmesi büyük yarar sağlayacaktır. Bu sayede modellemeye bağlı

olarak geliştirilebilecek “erken uyarı sistemleri”nin kullanımı ya da alınabilecek bazı önlemler ile yıldırımın verebileceği zararlar en aza indirilebilecektir.

Yıldırımın hangi koşullarda, ne şekilde, nerelere veya düşebileceğine dair gerçekçi olaslılık hesaplarını yapabilmek; bilime ve insanlığa büyük katkı sağlayacaktır.

1.3 Kapsam

Tezde birinci bölüm, tezin konusunu, amacını ve hangi sonuçlara hizmet etmesinin beklendiği hakkında bilgileri içermektedir.
Tezin ikinci bölümünde, yıldırımın oluşumuna ilişkin teorilere, yıldırım türleri gibi konulara yer verilmiş, yıldırım ve doğası hakkında daha detaylı bilgilerin verilmesi amaçlanmıştır.

Tezin üçüncü bölümünde, yıldırım modellemesi yapılırken yararlanılanacak parametrelerden, güdülen amaçtan ve yola çıkış noktaları hakkında bazı genel bilgilerden ve yıldırım yolu modellemesinde karşılaşılan zorluklardan söz edilmiştir.

Tezin dördüncü bölümünde, yıldırımın izlediği yol üzerine türetilen ve kullanılan güncel modellerden söz edilmiş, bu modellerin, kendi aralarında ve tez çalışmasında önerilen modellerle karşılaştırılması yapılmıştır.

Tezin beşinci bölümünde, tez çalışmasının asıl içeriği olan, önerilen modeller hakkında bilgi verilmiştir. Modellerin işleyiş mantığı, adım adım ve ayrıntılı olarak anlatılmış, şekiller ile açıklanmıştır. Önerilen iki model için özellikler, üstünlüklükler ve zayıf taraflar irdelenmiş, kendi aralarında, diğer modeller ile ve yıldırımın gerçek davranışı ile karşılaştırılmıştır.

Tezin altıncı bölümünde, beşinci bölümde önerilen modellerin benzetimi için yazılan Matlab kodu ve işleyişi hakkında ayrıntılı bilgi verilmiş, değişik durumlara ilişkin sonuçlar gösterilmiştir.

Tezin yedinci ve son bölümünde, sonuç kısmında, günümüzde yıldırım yolunun modellemesinin getireceği yararlardan söz edilmiş, modelin gerçekle olan tutarlığı irdelenmiş, modeli geliştirebilecek bazı önerilerde bulunulmuştur.

2. YILDIRIM VE OLUŞUMU

Bu tez çalışmasında, yıldırımın izlediği yolun belirlenebilmesi için Elektriksel Delinme Modeli (Di-electric Breakdown Model, (DBM)) üzerinden yola çıkılmıştır. Tezin olgunlaşma aşamasında, bu modelin, yıldırım biçimi için yeniden yapılandırılmasının ardından bir dizi matematiksel bağıntılar da eklenerek ve önerilen iki ayrı modelle harmanlanarak yıldırımın izleyebileceği olası yolları ortaya koyulmaya çalışılmıştır.

Bu bölümde, yıldırımın meydana gelmesine olanak tanıyan bulutların oluşumu, bulut türleri, bulutların yüklenmesi ilişkin teoriler hakkında genel bilgiler verilmiştir. Daha sonra da yıldırımın oluşması, özellikleri, karakteristikleri ve türleri hakkında bilgiler, bağıntılar ve istatistiksel bilgiler ortaya koyulacaktır.

2.1 YILDIRIMIN OLUŞUMU HAKKINDA GENEL BİLGİ

Yıldırım boşalmasının meydana gelmesinde temel olarak iki neden vardır; Bunlardan birincisi atmosferde yüksek miktarda nem bulunması, ikincisi ise sıcak hava akımları yardımıyla yüklü bulutların oluşmasıdır. Özetle, yıldırım oluşumu için neme ve yüklü bulutlara gereksinim vardır. Hava akımları, yere yakın hava tabakalarının iyice ısınması ile oluşur. Çok yükseklerden aşağı inen soğuk hava ile bu sıcak hava tabakası yer değiştirerek hava akımlarını meydana getirir. Nem ise yüksek sıcaklıkta buharlaşma ile meydana gelir. Hava, yukarı çıkışı sırasında soğur ve belirli bir yükseklikte su buharına doyacağı bir sıcaklığa erişir. Daha fazla yükselmesi yoğuşmaya sebep olur ve bulut oluşur. Yıldırım bulutunun oluşumunda üç aşama söz konusudur: Gençlik, Olgunluk ve Yaşlılık.

Gençlik aşamasında aşağıdan yukarı doğru ve kenarlardan ortaya doğru hava akımları artar. Bu durum yaklaşık 10 – 15 dakika sürer**.** Olgunluk aşamasında yağmurlar oluşur. Sıfıra yakın sıcaklık derecelerinde iyice azalan bulut kaldırma kuvveti, şiddetli yağmurlara sebep olur. Bu sırada yukarıdan aşağıya hareket eden soğuk rüzgarlar görülür. Bu rüzgarlar yere ulaştıklarında kısa süreli, şiddetli

fırtınalara sebep olurlar. Bu aşama yaklaşık 15 – 30 dakika sürer. Yaşlılık aşamasında ise hava akımları artık son bulmuştur. Bu aşama da yaklaşık 30 dakika sürer.

2.2 Yıldırım Bulutunun Yüklenmesine Dair Teoriler

Günümüzdeki çalışmalar ve bilgiler yıldırım bulutunun oluşumuna açıklık getirilebilse dahi bir bulutun elektriksel olarak nasıl yüklendiğine ilişkin kesin bilgi yoktur. Yine de havanın iyi bir iletken olmaması ve yük oluşumuna izin verebilecek yapıda olması, yüklenmenin ana sebepleri olarak düşünülmektedir [1, 2]. Bu durumu açıklayabilecek bazı teoriler geliştirilmiştir. Aşağıda bu teorilerden bazıları kısaca açıklanmıştır:

2.2.1 Simpson ve Lomonosow Teorisi

Bu iki araştırmacıya göre bulutlardaki yükler hava akımı yardımıyla oluşmaktadır. Sıcak ve soğuk havanın yer değiştirmesi sonucunda oluşan hava akımı bulutlardaki su damlacıklarını harekete geçirir. Hareket halindeki su damlacıkları, birbirleriyle sürtünerek, elektriksel olarak yüklenirler. Bulutlardaki hava akımları, su damlacıklarının dağılmasına ve tekrar birleşmesine sebep olur. Yapılan laboratuar çalışmalarında, dağılan su damlacıklarından küçük damlacıkların negatif, büyük damlacıkların ise pozitif olarak yüklendiği gözlenmiştir. Bu bilgilere göre büyük su damlacıkları yani pozitif yüklü damlacıklar bulutun alt kademelerinde ve rüzgar hızının büyük olduğu bölümlerde olmalıdırlar. Küçük, yani negatif yüklü su damlacıkları ise rüzgar tarafından itilmeli ve bulutun daha yukarı kısımlarında dağılmalıdırlar. Yıldırım bulutundaki yüklerin bu şekilde meydana geldiği kabul edilecek olursa bulutun alt kısımları pozitif yüklü olacağından yıldırım boşalması da pozitif kutbiyette olacaktır. Yapılan gözlemlerde pozitif kutbiyetteki yıldırım boşalmalarının %5–20 civarında olduğu, boşalmaların yaklaşık %70-95'inin negatif kutbiyette olduğu ortaya konmuştur. Bu nedenle Simpson ve Lomonosow teorisi yıldırım bulutlarındaki elektrik yüklerinin meydana gelişini tam olarak açıklayamamaktadır [3].

2.2.2 Elster ve Geitel Teorisi

Bu teoriye göre bulutların yüklenmesi, tesirle elektriklenme ile açıklanmaktadır. Dünya yüzeyindeki elektrik yükü $-5x10^5$ C kabul edilirse, bulutun içinde bulunan su damlacıklarının alt uçları pozitif ve üst uçları negatif olmak üzere kutuplanırlar. Yerçekimi etkisiyle aşağıya doğru düşen büyük su damlacıkları havanın oldukça yavaş hareket eden iyonlarına yaklaşırlar ve bu sırada su damlacığının pozitif alt ucu havanın negatif iyonunu tutarken pozitif iyonu da iter. Böylece ağır su damlacıkları negatif elektrikli parçacıklar haline gelir. Aynı şekilde kutuplanan küçük su damlacıkları yukarıya doğru hareket ederken havanın pozitif iyonlarını çekerler ve negatif iyonları iterler. Böylece hafif su damlacıkları da pozitif elektrikli parçacıklar haline gelirler. Bu teoriye göre bulutun alt kısımlarında negatif yükler bulunmaktadır. Teori negatif kutuplu yıldırım boşalmalarını açıklayabilmektedir gibi gözükse de aslında eksik yanları bulunmaktadır. Bir yıldırım bulutunun su damlacıklarından çok, buz kristalleri ve kar parçacıklarından oluştuğu göz önüne alınırsa, bu buz kristalleri ve kar parçacıklarının dünyanın elektrik alanı ile kutuplanma olasılıkları oldukça düşüktür [3].

2.2.3 J. I. Frenkel Teorisi

Frenkel'e göre havada her iki işaretli iyonlar var olduğundan, dünyanın negatif elektrik yükleri kaçmaya ve atmosferin pozitif elektrik yükleri ile birleşmeye yatkındır. Dolayısıyla dünyanın azalan elektrik yükünü sürekli olarak besleyecek bir olayın meydana gelmesi gerekmektedir. Buna göre dünyanın elektrik yükünün sabit kalmasında en önemli rolü negatif yıldırım boşalmaları sağlayacaktır. Bu teoriye göre her iki işaretli iyonlardan oluşan hava ile küçük su damlacıkları veya buz kristallerinden meydana gelen bir ortam göz önüne alınır ve havanın negatif iyonlarının daha küçük su damlacıklarına veya buz kristallerine konduğu var sayılır. Buna göre bulut, negatif elektrikli su damlacıkları ve pozitif iyonlu havadan oluşur (negatif iyonlar su damlacıkları tarafından tutulmuştur) [3].

2.3 Yıldırımın Oluşumu

Yıldırımın oluşması için öncelikle elektriksel olarak yüklenmiş yıldırım bulutunun oluşması gerekir. Yıldırım bulutunun yüklenmenin kesin olarak nasıl meydana geldiği hakkında kesin bilgilere sahip olmamamıza rağmen bu konuda geliştirilmiş

bazı teorilerden yukarıda söz etmiştik. Yıldırım bulutu oluştuktan sonra yük birikimleri oluştukça, zamanla, bulutla yeryüzü arasındaki potansiyel farkı artar. Bu artış nedeniyle yeryüzü ile yıldırım bulutu arasında kalan havanın da delinmesi kolaylaşır ve belli bir değerden sonra havanın delinmesiyle oluşan iletken kanal boyunca buluttan toprağa veya topraktan buluta doğru bir elektriksel boşalma başlar. Elektrik alan şiddetinin 25 kV/cm değerine ulaştığı andan itibaren artık yıldırım gözlenebilmektedir.

Buluttaki elektrik alan şiddeti yeterince arttığında elektriksel boşalmalar başlar. Bu boşalmalar birkaç şekilde meydana gelebilir. Bunlardan iki bulut arasında meydana gelenlere bulut – bulut (inter-cloud) türü, bulutun kendi içinde meydana gelen türüne bulut içi (intra-cloud) tür, bulut ve yeryüzü arasında meydana gelen türüne de bulut – yeryüzü (cloud-to-ground) türü boşalmalar adı verilir. Bu boşalmalardan toprak ile bağlantılı olanlara yıldırım, bulut içinde veya bulutlar arası olan türlerine ise şimşek adı verilir.

Yeryüzündeki elektrik alanı çeşitli sebeplerden ötürü eğer (yüksek kuleler, gökdelenler, v.b.) bozulmuşsa bu durumda bulut – yeryüzü boşalması görülebilmektedir. Bulut – yeryüzü boşalması, bulutun pozitif veya negatif yüklü bölgelerinden yere veya yeryüzündeki pozitif veya negatif yüklü sivri uçlarından buluta başlayabildiği için, dört şekilde olabilir.

Fiziksel nedenlerden ötürü, bulutun yüklenmesi sırasında yere yakın olan kısmında %70-%90 olasılıkla negatif elektrik yükleri yer alır [4, 5]. Bu durumda yerin de bulutun negatif yüklerine bakan bölümünde pozitif yükler toplanır. Bazı koşullarda bunun tersi yüklenme de olabilmektedir (%10-%30 olasılıkla). Fırtınanın, hava akımlarının artmasıyla buluttaki negatif yük oranı ve buna bağlı olarak da yerdeki pozitif yük toplanması hızlanarak devam eder. Aradaki bu fark belli bir eşik değerine ulaştığında kararsız bir ortam oluşur ve yıldırım görülür.

Yıldırım gerçekte birkaç ani oluşan darbeden meydana gelmektedir. İlk darbe "basamaklı öncü" (stepped leader) adını alırken, ardışık darbelere ise "fırlatma öncüleri" (dart leaders) adı verilir. Bu ikincil öncüler yol alırlarken bir önceki öncülerin izlediği yollardan gitmeye meyil göstermektedirler ve basamaklı öncüler kadar dallanma sergilememektedirler [4]. Her bir ikincil öncü, yıldırımı bir önceki ikincil öncünün getirdiği noktadan yaklaşık olarak 30 – 50 m daha öteye götürür.

2.4 Yıldırım Türleri

Yıldırımı yönüne göre iki gruba ayırabiliriz: 1) Yukarıya çıkan yıldırım 2) Aşağıya inen yıldırım.

2.4.1 Yukarıya Çıkan Yıldırım

Bu tür yıldırımlar genelde yerin pozitif yüklü sivri bölgelerinden, bulutun negatif yüklü bölgesine başlayan ön boşalmalar şeklinde görülür. Boşalmalar genelde düzgün araziler üzerindeki çok yüksek yapılardan (GSM kuleleri) veya yeryüzünün yüksek dağlık kesimlerinden başlar. Bu yüksek kesimlerin sivri uçlarından buluta doğru ön boşalmalar görülür. Bu sırada 1 – 10 kA arasında değişen akımlar oluşur. Boşalma tam olgunlaştığında ise akım değeri 10 kA'yı bulur.

2.4.2 Aşağıya İnen Yıldırım

Bir bulutun alt kısmındaki elektrik alan şiddeti yeterli düzeye geldiğinde toprağa doğru bir elektron demeti harekete geçer. Birinci demet 10 – 50 metrelik mesafeyi 50.000 – 60.000 km/s arasındaki hızla geçer. 30 – 100 μs süren bir aradan sonra ikinci bir boşalma birinci boşalmanın yolunu izler ve birinciden 30 – 50 m arası daha ileri gider. Daha sonra üçüncü boşalma ve ardından dördüncü boşalma meydana gelir. Her bir boşalma öncekinden 30 – 50 m daha ileri giderek öncü boşalmanın ucunun yeryüzüne yaklaşmasını sağlar. Öncü boşalma yere yaklaştıkça elektrik alan şiddeti havanın delinme dayanımı üzerine çıkacak kadar artar. Böylece yeryüzünün sivri bir noktasından bir boşalma yukarıya doğru ilerleyerek öncü boşalma ile birleşir. Yaklaşık 50.000 km/s'lik bir hızla aşağıdan yukarıya doğru iyonizasyonlu ve kanalda depo edilen yükü toprağa boşaltır. Bu boşalma sırasında 100 MV'luk bir gerilimle 200 kA'e kadar çıkan akım toprağa akar.

3. YILDIRIMIN YOLUNUN ANALİZİ

Her ne kadar yıldırım hakkında çok kısıtlı bir bilgi birikimine sahip olsak da, günümüzde yıldırımın karakteristiği ve modellenmesi çalışmaları son zamanlarda ilgi odağı haline gelmiş, yapılan çalışmaların sayısı artmış durumdadır. Yıldırım yolunun analizi için ilk olarak yıldırım görüntülerinin kayıt edilmesi gerekmektedir [6].

3.1 Yıldırım Yolunun Analizinde Varolan Çalışmaların Durumu

Yıldırım modellenmesi için ana başlama noktası, nasıl bir görünüme sahip olduğunun belirlenmesidir. Nasıl göründüğü, nasıl ve ne derecelerde dallanma gösterdiği, bu dalların uzunluğu ve açısı, kırıkları ve kavisleri yöntemin en önemli bileşenlerini oluşturacaktır.

Analiz edilecek verilerin ve buna bağlı olarak bir veritabanının oluşturulabilmesi için ilk olarak çeşitli durumlar altında yıldırımın görüntüleri kaydedilmelidir. Kayıtta, düşük (30 fps) ve yüksek (500 fps) hızlı kameralar kullanılabilmektedir. Fotoğraf makinelerinin de yardımı dokunabilmektedir. Her cihazın kendine göre avantajları ve dezavantajları bulunmaktadır. Bunlar uzaysal tanımlama için gerekli olan piksel çözünürlüğü ve dinamik analizler için gerekli olan zaman çözünürlüğüdür.

Yıldırım modellemesi için uygulanılması gerekli olan metodolojiler şunlardır:

- Kanal tanımlaması bilgisine dayanan inceleme,
- Görsel fraktal analizler,
- Darbe / yıldırım geometrik istatistikleri,
- Boşalma özellikli zaman analizi.

İncelenmesi gereken özellikler ve davranışlar ise,

- Elektriksel bozulma ve atmosferik iletkenlik ile bağlantılı olan dallanma,
- Atmosferik iletkenlik ve uzay yükleri ile bağlantılı olan titreşimler ve bükülmeler,

- Yük birikim süreçleri olan, atmosferik iletkenlik ve termo-elektrodinamik ile bağlantılı olan akım yoğunluğu tepe değeri, çeşitliliği, gecikmesi ve şimşek kümeleri değerleri [7].

3.2 Analiz Sırasında Karşılaşılan Zorluklar

Günümüzde yıldırımın kendisi ve yıldırımın karakteristiği hakkında hala kısıtlı bilgi bulunmaktadır. Bunun en önemli nedenlerinden birisi, yıldırım olayının çok büyük bir hacimde ve çok değişken koşullar altında gerçekleşiyor olmasıdır. Yıldırımın oluşumuna ilişkin kesin ve ayrıntılı bilgilere sahip olunmasına rağmen, yıldırımın izlediği yollar hakkında etkili bir modelleme gerekleştirilememiştir. Yıldırımın izlediği yolun, gaz ortamdaki, yani atmosferdeki iyonların dağılma durumlarına, türlerine ve yoğunluklarına göre değişiklik gösterdiği bilinmektedir. Kaldı ki, bu dağılımın nasıl bir yapıda olduğunu, yıldırım meydana geldiği esnada belirleyebilecek bir teknoloji var olmadığından, yıldırım oluştuğu ve yol aldığı sırada hangi koşullardan, nasıl etkilendiği konusunda yargılara varmakta güçlük çekilmektedir.

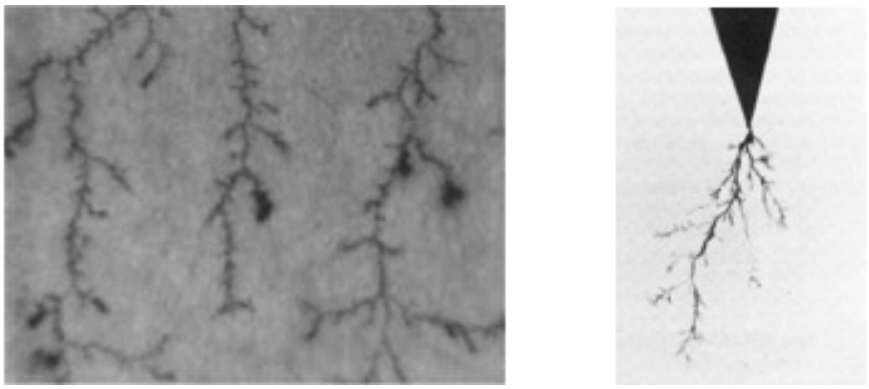
Aynı nedenden dolayı, laboratuar ortamının yaratılamaması da büyük bir sorun oluşturmaktadır. Laboratuarlarda yüksek gerilim uygulanarak atlamalar yapılması mümkündür fakat yine yolu etkileyen faktörlerin belirlenmesi oldukça güçtür.

Karşılaşılan diğer güçlüklerden biri de elektriksel boşalmaların ne katı ne sıvı ne de gaz halinde olmasıdır. Elektriksel boşalmalar maddenin dördüncü hali olarak nitelendirilebilecek olan “plazma” halindedirler. Dolayısıyla çok parlaktırlar ve çözümlenebilir bir yüzeyleri de yoktur. Bu nedenle geleneksel yüzey modellemeleri doğrudan uygulanabilir değildir [4].

Yıldırımın modellenmesi konusunda günümüzde oldukça fazla sayıda araştırma yapılmaktadır. Yapılan çalışmalar genel olarak, yıldırımın şeklinin fotoğraf veya görüntü bazında incelendikten sonra, olasılık kuramlarına, fraktal analiz ve diğer analiz yöntemlerine göre benzer modellerini türetme ilkesine dayalı olarak sürdürülmektedir. Yıldırım analizinin en gerçekçi olarak modellenebilmesi için bu görüntülerin yanı sıra, çevresindeki iyon ve gaz dağılımının da eşdeğer zamanlı olarak hatta yıldırım öncesi durumdaki dağılımın da bilinmesi gerekmektedir.

4. YILDIRIMIN YOLUNUN MODELLENMESİNDE KULLANILAN YÖNTEMLER

Yıldırımın doğası ve yapısı nedeniyle modellemede yaşanılan zorluklardan önceki kısımda söz edilmiştir. Bu zorlayıcı nedenlerden ötürü yıldırımla ilgili çok kısıtlı verilere sahibiz. Yine de özellikle son zamanlarda, yıldırımın etkili ve gerçeğe yakın bir modelinin ortaya konulabilmesi için bir takım çalışmalar yürütülmektedir. Bu çalışmaların günümüzde başını çeken iki grup vardır. Bunlardan ilki, Elektriksel Bozulma Yöntemini (Di-electric Breakdown Model, (DBM)) ilk defa yıldırım modeli için revize eden ve yıldırım modellemesi konusunda önemli adımlar atan L. Niemeyer'dir. Yıldırım modellemesi için uzun süredir çalışma yapan bir başka grup ise C. Ming'in önderliğindeki gruptur. İki araştırmacı da çıkış noktası olarak DBM'yi esas almış ve kablodaki delinmenin, yıldırım yolu ile benzerliğinden yararlanıp, model üzerinde değişiklikler yaparak yıldırım modelllerini geliştirmiştir. Bu çalışmalardan ileriki kısımlarda söz edilecektir. Aşağıda sırasıyla bir kablonun yalıtkan kısmında ve yüksek gerilim uygulanan bir yalıtkan maddede zorlanma sonucu meydana gelen ağaçlanmalara ilişkin resimler verilmiştir [8]:



Şekil 4.1: Kablo Yalıtkanında ve Yalıtkan Maddede Ağaçlanma

Yıldırımın modellenmesinde en yaygın olarak kullanılan yöntemlerden bazıları şunlardır:

1) Sızma Teorisi (Percolation Theory)

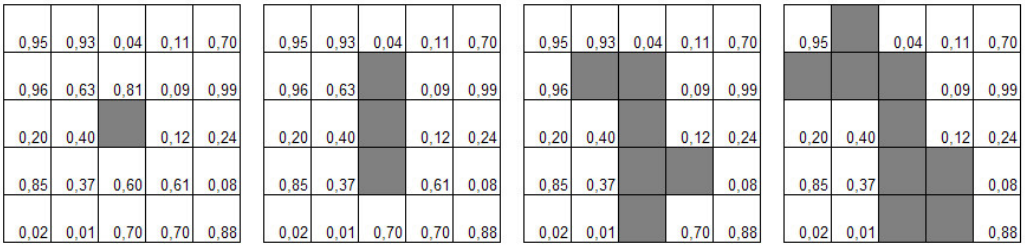
2) Rasgele Orta Nokta Kaydırma Algoritması (Random Midpoint Displacement Algorithm, (RMDA))

3) Laplasyen Büyümesi (Laplacian Growth)

Yıldırımın izlediği yolun modellenmesinde kullanılan yöntemlerden en basiti, Sızma Teorisi kullanılarak gerçekleştirilenidir. Modellemede teorinin, "alan sızması" (site percolation) olarak adlandırılan bir "alt modeli" uygulanarak sonuca gidilmektedir.

4.1 Sızma Teorisi

Sızma teorisi, dağınık sistemleri kolayca canlandırabilecek çoklu-fraktal analiz yöntemlerinden biridir. Sızma teorisinin en basit biçimi "alan sızması" şeklinde olanıdır. Bu yöntemde izlenen yol ve gelişim aşağıdaki şekilde belirtilmiştir.



Şekil 4.2 : İlk 4 Adım Boyunca 2D Sızma Fraktalına İlişkin Büyüme.

Burada gösterilen dört ızgarada, yayılma olasılığı katsayısı, $\rho = 0{,}5$ değeri için alan sızması tekniği uygulanmıştır. İlk adımda 5×5 gözlü ızgarada merkezdeki nokta dolu olarak seçilmiş, daha sonra her bir kare için 0 ile 1 arasında rasgele sayılar üretilmiş ve dolu olan noktanın komşu noktaları (2D olduğu için en yakın dört nokta olan sağ, sol, alt ve üst komşular) incelemeye alınmıştır. Bu noktalardan rasgele sayısı $\rho = 0{,}5$ değerinden büyük olduğu görülen üst ve alt komşular "dolu" olarak belirtilmiş, kalan sağ ve sol komşular da bir daha kullanılmamak üzere "ölü" olarak belirlenmiştir.

İçi karelerden oluşan bir kafes düşünüldüğünde, yöntemde bu bölgedeki kareler ya boş ya dolu ya da ölü olarak varsayılır. Bu kafes içersinde, bazı kareler belirli kurallara göre veya gelişi güzel olarak yerleştirilerek dolu hale getirilir. Zaman temelli olan bu yöntemde her bir zaman diliminde dolu olarak işaretlenen karelere komşu olan tüm kareler göz önüne alınır. Bu kareler iki boyutlu (2D) bölgede en yakın dört kare (sağ, sol, yukarı ve aşağı), üç boyutlu (3D) bölgede ise en yakındaki

altı kare (sağ, sol, yukarı, aşağı, ön ve arka)'dir. Her bir komşu kare için 0 ile 1 arasında rasgele sayılar üretilir. Bu 0 ile 1 arasında üretilen rasgele sayılardan, daha önceden belirlenmiş, yayılma olasılık değeri, ρ'den daha büyük ise, içinde bulunduğu kare "dolu" olarak işaretlenir. Diğer durumda ise, yani komşu karelerdeki rasgele sayılardan ρ değerinden küçük olan kareler "ölü" kareler olarak işaretlenir ve benzetimin geri kalanındaki "komşu" listesine katılmazlar. Daha sonra bu işlem bir sonraki zaman diliminde tekrarlanır [9,10].

Sistem bu şekilde işletilerek, belirli olasılık değerleri ve yukarısı için seçilen noktaların birleştirilmesi ile rasgele bir yol belirlenmiş olur ve bu rasgele patika yıldırım yoluna benzer görüntüler sergilemektedir.

Sızma teorisi, ρ'nin küçük değerleri için oldukça hassas sonuçlar üretmektedir ve bu değerler için fraktal yapılar ortaya çıkarmaktadır. Ne zaman ki ρ değeri bu küçük değerler aralığından çok fazla sapma gösterdiğinde, kafesin çoğunluğu ya boş ya da dolu olacağından sızma genellikle çekiciliğini kaybetmektedir.

Sızma teorisinin yaygın olarak kullanıldığı bir gerçek yaşam örneği verecek olursak, bir topluluk içindeki dokunma yoluyla bulaşan bir hastalığın yayılmasını gösterebiliriz. Bu incelemede, dolu kareler hastalığın bulaşmış olduğu bireyleri simgelemektedir. Hastalık kapmış olan kişilerin etrafındaki (en yakın komşu kareler) kişilere de hastalığı bulaştırma olasılığı vardır. Eğer bulaşma gerçekleşirse, bir sonraki zaman adımında hastalık kapan bu kişi tarafından diğer komşulara hastalığın sızma olasılığı olacaktır. Komşu olup da hastalık kapmayan bireyler için "hastalığa karşı bağışıklığı var" tespiti yapılabilir.

Sızma teorisi kullanılarak ortaya çıkan sonuçlar çoklu-fraktallar olarak bilinirler ve bu nedenle karakteristiklerinin belirlenebilmesi için bu sonuçlara özel analiz tekniklerinin uygulanması gerekmektedir [11].

Aşağıdaki bölümde yöntemin üstünlükleri ile eksik yanları genel hatları ile belirtilmiştir.

Yöntemin Üstünlükleri:

- Kurgusu Kolaydır: istenilen boyutta bir bölgede 0 ile 1 arasında rasgele (random) sayıların üretilmesi oldukça kolaydır. Microsoft Excel de dahil olmak üzere tüm hesaplama ve mühendislik programları (Matlab, Mathcad… vs) ile istenilen aralıklarda rasgele sayılar kolayca üretilebilmektedir. Ayrıca

C, Pascal gibi eski ve bilinen programlama dilleriyle bile belirli bir değer üzerindeki noktaların seçilmesi işlemi kolayca tanımlanabilir. Sonuç olarak yöntem kolayca programlanabilir bir yapıya sahiptir.

- Hızlıdır: Yapısı gereği az sayıda işlem ve basit programlamaya gereksinim duyduğundan dolayı, ilerleme çok hızlı olarak gerçekleşebilmekte ve model kısa sürede tamamlanabilmektedir.

Yöntemin Eksik Yanları:

- Tamamen Rastlantısal Olmasıdır: Yöntem, yapısı gereği seçilecek noktaları tamamen rasgele olarak yerleştirilmiş sayılar arasından, tamamen rasgele olarak seçtiğinden dolayı ilerleme sonucu elde edilen "yol" tamamen rastlantısaldır, herhangi bir yıldırım karakteristiği veya herhangi bir elektriksel bağıntı ile ilgisi yoktur.
- Benzer şekilde ilerleme her yönlü ve tamamen rastlantısal biçimde olduğu için görsel olarak da yıldırıma benzediği anlar olduğu gibi, tamamen farklı yapılar da ortaya çıkarabilmektedir.

4.2 Rasgele Orta Nokta Kaydırma Algoritması (Random Midpoint Displacement Algorithm)

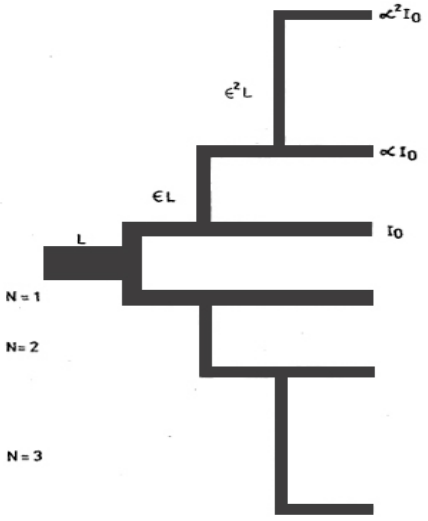
Yıldırımın modellenmesinde kullanılan algoritmalardan birisi de Rasgele Orta Nokta Kaydırma Algoritması (Random Midpoint Displacement Algorithm (RMDA))'dır.

Yöntemin ilkesi, aşağıda görüldüğü üzere, dallanma ve ayrılma esasına dayanır. Modelin özelliği her bir kolun ikiye ayrılmasıdır.

Yıldırım modeli için, en soldaki kısa ve kalın çizgi, başlangıç noktası olarak yani yıldırımın öncü ucu olarak kabul edilir. Daha sonra bu noktadan itibaren sürekli olarak ikiye ayrılan kollar, yıldırımın dalları olarak düşünüldüğünde elde edilen görüntü yıldırımla benzerlik göstermektedir. Dallanma noktaları, kolların orta noktasından, daha önceden rasgele olarak türetilen sayılar sonucu ulaşılan bir standart sapma değeri kadar kaydırılır yani dallanmalar, dalların orta noktalarına yakın kısımlarda meydana gelir. Dallar gittikçe incelirken (merkezden uzaklaştıkça akımın azalmasını temsil eder) boyları ise giderek artar (bu da toprağa yaklaştıkça

artan potansiyel farkından dolayı ilerlemenin artmasını ve delinmenin kolaylaşmasını temsil eder) ve böylece yıldırıma benzer bir görüntü oluşmuş olur.

Bir fraktal modelleme türü olan RMDA için basit ve temel bir yaklaşım, dönüş darbesinin yıldırım kanalının N tane düz ikincil daldan oluştuğu kabulüne dayanır. Rasgele Orta Nokta Kaydırma Algoritması (Random Midpoint Displacement Algorithm) kullanılarak kanalların parça parça (piece-wise) doğrusal görünmesi sağlanabilmektedir.



Şekil 4.3 : RMDA'nın Yıldırım Modeline Uydurulmuş Şekli

Yönteme "L" uzunluğunda düz bir başlangıç çubuğu ile başlanır. Ortalaması sıfır ve sapması σ olan rasgele sayılar üretilerek çubuğun orta noktası bu değer kadar kaydırılır. Bu işlem her bir segment için tekrarlanır. Bu yöntem benzer şekilde elektriksel dallanmaya uyarlanabilir. Bunun için toplam akım her bir kola belirli şekilde paylaştırılır. Burada dalların mesafesi bir ε katsayısı ve kuvvetleri ile doğru orantılı olarak artarken, akım ise α katsayısı ve kuvvetleri ile zayıflamaktadır. Yöntem küçük ölçekli örneklemeler için başarılı sonuçlar vermektedir [12,13].

Aşağıdaki bölümde, önceki yöntem için de gerçekleştirildiği gibi, bu yöntemin de üstün ve eksik yanları belirtilmiştir.

Yöntemin Üstünlükleri**:**

- Kurgusu Kolaydır: Sızma teorisine benzer şekilde, sistemin altyapısını oluşturmak kolaydır. Orta noktadan sapmaları belirleyecek rasgele sayılar

üretilir, bu noktalar için daha önceden atanmış olan ε ve α katsayıları ile yıldırımın daldan ne kadar uzaklaşacağı ve boyunun ne kadar olacağı programlama dilleriyle veya basit benzetim programları ile dahi kolayca gerçekleştirilebilir.

- Hızlıdır: Yine, Sızma Teorisi'ne benzer olarak, az sayıda işlem, basit eşitlikler ve karmaşıklıktan uzak programlama ile yöntemde sonuca hızla gidilebilmekte ve model kısa sürede oluşturulabilmektedir.

Yöntemin Eksik Yanları:

- Yıldırım karakteristiği ile tam örtüşmemesi: Görsel olarak RMDA, Sızma Teorisi'ne göre yıldırım modellemesinde görsel olarak daha başarılı ve gerçekçi sonuçlar ortaya koymasına rağmen gerçek bir yıldırım karakteristiğini yansıtamamaktadır. Bunun en önemli nedeni, aynı Sızma Teorisi'nde de olduğu gibi önceden belirlenmiş katsayılardan ve rasgele sayılardan yola çıkılarak modellemenin gerçekleştirilmesidir. Modelde kullanılan katsayılar, yapılan denemeler sonucu amprik olarak başarılı sonuçlara varacak olsa dahi, modelde elektriksel hiçbir bağıntının kullanılmamış olmasından dolayı yıldırımın karakteristiğini genel olarak gösterebilecek bir yapıya sahip olamayacaktır.
- Yöntemin zayıf kaldığı noktalardan birisi de yapısının dallanmasından ve kolların sürekli olarak ikiye ayrılmasından kaynaklanmaktadır. Yıldırımın dallanmalar gösterdiği bilinmektedir fakat bu denli dallanmadığını ve sürekli olarak ikiye bölüne bölüne toprağa ulaşmadığını biliyoruz. Kaldı ki yıldırım ana koldan ikiye olduğu gibi, üçe dörde hatta daha fazla kola da ayrılabilmektedir.

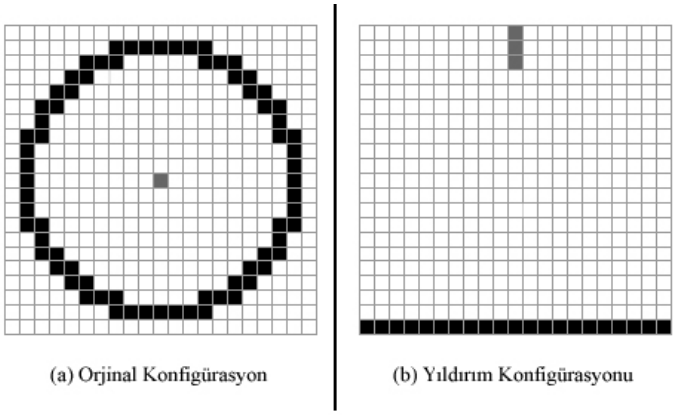
4.3 Laplasyen Büyümesi (Laplacian Growth)

Laplasyen Büyümesi Modeli'nde genel olarak üç farklı teknik uygulanabilmektedir:

1) Sınırlı Bölgede Yayılma (Diffusion Limited Aggregation)

2) Elektriksel Delinme Modeli (Dielectric Breakdown Model (DBM))

3) Hastings-Levitov Konform Dönüşümü (Conformal Mapping)



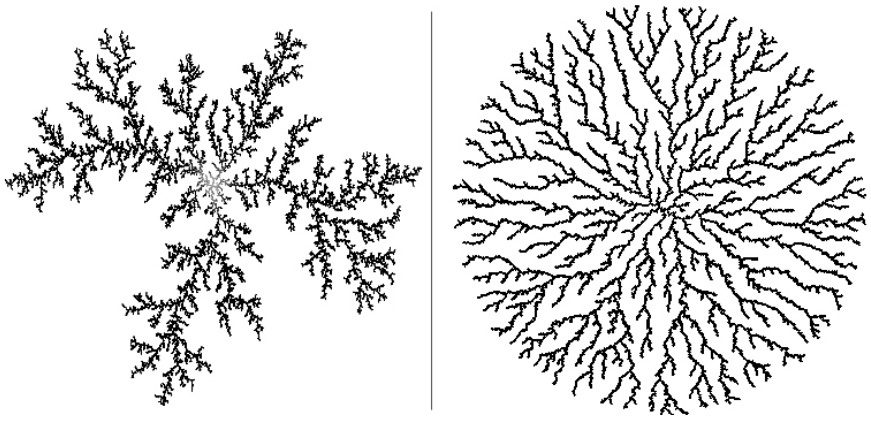
Şekil 4.4 : Benzetim İçin Değişik Yük Biçimleri

Günümüzde yıldırım modellemesinde en yaygın olarak kullanılan yöntem bir Laplasyen büyümesi yöntemi olan Elektriksel Delinme Modeli, (Di-electric Breakdown Model (DBM))'dir [14].

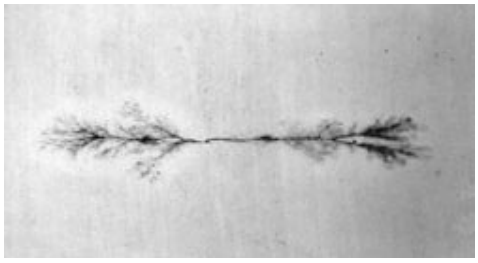
Laplasyen büyümesinde kullanılan yukarıdaki yöntemler, hemen hemen aynı görsel sonuçları vermektedir. Nitekim yıldırım olayı, bilindiği üzere atmosferde meydana gelen elektriksel bir olay olduğundan yıldırımın yolu gibi yıldırımla ilgili bir modellemede, içerdiği elektriksel bağıntılardan dolayı Elektriksel Bozulma Modeli (DBM) tercih edilmektedir.

DBM'nin ilk olarak kullanılma alanı yüksek gerilim kablolarında yaşlanma ve elektriksel zorlanma nedeniyle meydana gelen delinmelerin görsel olarak modellenmesi olmuştur. Şekil 1.a'da görüldüğü üzere, DBM'nin orijinal biçimi, merkezde gri renk ile gösterilen bir $\phi = 0$ noktası, sınır koşulları da siyah renk ile gösterilen ve $\phi = 1$ olan bir daireden oluşmaktadır. Merkezde bulunan $\phi = 0$ noktası, akımın aktığı kablo iletkeni, dolayısıyla delinmenin başlangıç noktası olarak kabul edilmektedir. $\phi = 1$ ile gösterilen daire de yalıtkanın bittiği noktaları veya siperi göstermektedir. Dolayısıyla bu modelde DBM yöntemi kullanılarak merkezden, yalıtkan sınırına yani $\phi = 1$ noktasına doğru ilerleyerek adım adım büyüyen bir delinme olayı benzetimi yapılarak, gerçeğiyle benzeşmesi sağlanmaktadır. Aşağıdaki şekilde DBM kullanılarak elde edilen bir delinme yolu ile gerçekte karşımıza çıkan bir delinme yoluna örnekler sunulmuştur [14].

Aşağıda, Şekil 4.5'te DBM kullanılarak elde edilen benzetimlere ait bazı sonuçlar, Şekil 4.5'te ise gerçekte karşımıza çıkan bir delinme yoluna ait bir örnek sunulmuştur [15,16].



Şekil 4.5 : DBM ile Elde Edilen Bazı Benzetim Sonuçları



Şekil 4.6 : XLPE Tip Bir Kabloda Gerçek Bir Delinme Örneği

Fark edilebileceği üzere, kabloda delinme esnasında zamanla ortaya çıkan "delinme yolu" görsel olarak aslında atmosfer içersinde yol alan yıldırımın yolu ile benzeşmektedir. İkisi arasındaki en büyük fark ise, kablodaki delinme olayında, yolun merkezden kablo yalıtkanın çeperlerine doğru, yani sipere doğru ilerlemesi, yıldırım olayında ise, bulutun alt kesiminde oluşan öncü uçtan yeryüzüne doğru, yani yukarıdan aşağı doğru yönde gerçekleşiyor olmasıdır. Bir başka fark ise delinme olayında, delinmenin meydana geldiği yol merkezden sınıra doğru her zaman tek bir yol şeklinde değil, birden fazla yol oluşturarak ilerleyebilmektedir. Merkezden yukarı doğru bir delinme yolu başlarken, örneğin merkezden aşağı veya başka yönlere doğru da delinme ve buna bağlı olarak da yeni bir delinme yolu eş zamanlı olarak görülebilir. Yıldırım olayında ise ana tek bir yol vardır ve bu ana kola bağlı olarak ikincil, üçüncül kollar oluşabilmektedir.

Yukarıdaki farklar göz önüne alındığında DBM'de kullanılan orijinal biçim, elde edilen görsel sonuçlar göz önüne alındığında şekil olarak yıldırıma benzemektedir fakat yıldırımın özelliklerini ve oluşum şeklini doğru olarak yansıtmayacaktır. Bu nedenle DBM'nin, Şekil 4.4 (a)'da görülebileceği üzere yıldırım modellemesi için de

kullanılabilmesi için, orijinal biçimi üzerinde bazı değişiklikler yapılmış ve yıldırım için daha uygun hale getirilmiştir. DBM'nin ana biçiminin bu şekilde değiştirilerek yıldırım modeli için kullanılması fikri, daha önceden adından söz ettiğimiz L. Niemeyer tarafından ortaya konulmuştur [14].

Günümüzde yıldırım modellemesi konusunda çalışmaları ile öne çıkan iki isimden; L. Niemeyer ve C. Ming Lin'den söz etmiştik. Bu iki araştırmacı da, yıldırımın yolunun modellenmesinde Laplasyen Büyümesi yöntemini ve az önce tanımlamış olduğumuz DBM'nin yıldırım biçimine ilişkin modelini kullanmaktadır.

DBM'nin orijinal biçimin işleyişinden söz etmek gerekirse, içi karelerden oluşan 2D bir ızgara üzerinde, elektriksel potansiyel, ϕ'nin, her noktada izlenmesi esasına dayanmaktadır. İlk olarak negatif bir yük merkeze yerleştirilerek $\phi = 0$ değeri atanmaktadır. Daha sonra merkezi yükün etrafına dairesel formda $\phi = 1$ olan bir pozitif yük dağılımı yapılmaktadır. Daire içersinde kalan diğer noktalardaki potansiyeller de her bir nokta için Laplace denklemi, (4.1) çözülerek ve iterasyonlar gerçekleştirilerek hesaplanmaktadır. Bu çözüm gerçekleştirilirken merkezi ve dairesel yükler sınır koşulları olarak kabul edilmektedir. Ayrıca ızgara sınırları da $\phi = 0$ olarak tanımlanmaktadır [14,17]:

$$\nabla^2 \phi = 0 \qquad \textbf{(4.1)}$$

Laplace denklemi çözüldükten sonra, negatif yüklere komşu olan yani ($\phi = 0$) noktalar işaretlenmektedir. Daha sonra bu noktalardan bir tanesi "büyüme konumu" olacak şekilde tamamen gelişi güzel olarak seçilmektedir (örneğin yüksek gerilim kablosunda delinmenin ilerleyeceği bir sonraki nokta). Seçilen nokta $\phi = 0$ olarak değiştirilmekte ve bundan sonra yapılacak olan iterasyonlarda bu nokta da sanki sınır koşullarından birisiymiş gibi tanımlanmaktadır. Tanımlanacak olan noktanın seçilme olasılığı, potansiyeli ile bağıntılıdır. Seçilme olasılığına ilişkin ağırlık fonksiyonu aşağıda belirtilmiştir [14, 18]:

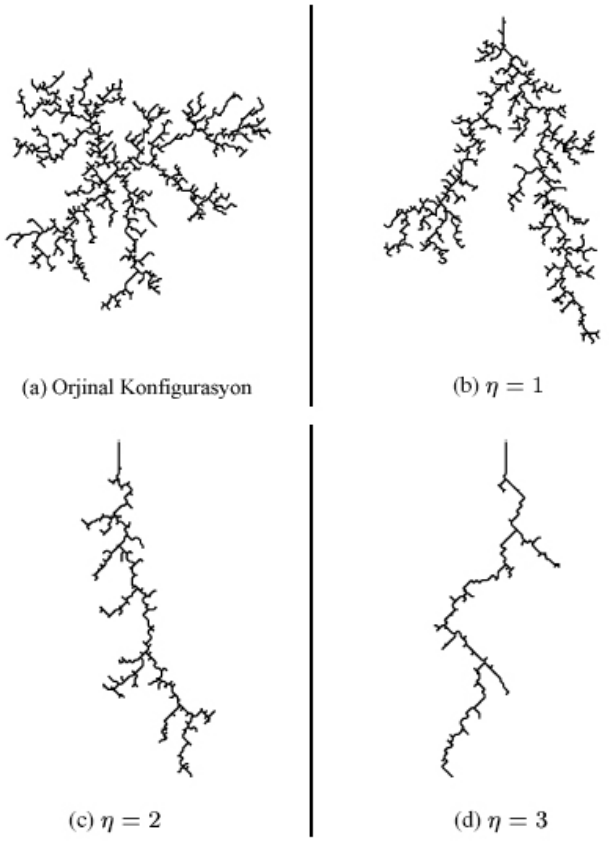
$$p_i = \frac{(\phi_i)^\eta}{\sum_{j=1}^{n} (\phi_j)^\eta} \qquad \textbf{(4.2)}$$

Burada "i", komşu hücreler listesindeki bir hücre, n ise listedeki toplam komşu hücre sayısını ifade etmektedir. Kullanılan η terimi ise modellemedeki kilit değişken

olmakla birlikte "güç katsayısı" (Bkz. 5. Bölüm) olarak adlandırılmaktadır ve ilerideki kısımda ayrıntılı olarak tanımlanacaktır.

Ardışık iterasyonlar, 2D bölgede Laplace denklemi çözülerek ve tekrar bir büyüme noktası seçilerek (4.2) denklemi kullanılarak tekrarlanmaktadır. İterasyonların kullanıcı arzu ettiği değerlere ulaşana kadar tekrarlanması gerekmektedir [14].

DBM'ye ilişkin orijinal biçim uygulandığında sonuç olarak, aşağıdaki Şekil 4.4 (a)'da belirtildiği üzere radyal bir büyüme göstermektedir. Büyümenin yıldırım şeklinde olması için Şekil 4.4 (b)'de görünen biçim kullanıldığında, benzetim sonuçlarında değişik η'lere ilişkin sonuçlar Şekil 4.5 (a), (b), (c) ve (d)'de gösterilmiştir [4].



Şekil 4.7 : Değişik Yük Biçimlerine Göre Benzetim Sonuçları

Şekil 4.7 (a), Şekil 4.4 (a)'nın sonuçları iken; Şekil 4.7 (b), (c) ve (d), Şekil 4.7 (b)'ye ilişkin değişken η terimleri için sonuçlardır. Görüldüğü üzere, DBM'nin yıldırım biçimi kullanıldığında bir başlangıç ve bitiş noktası olan, yukarıdan aşağı doğru yol alan, görsel olarak yıldırımla benzerlik gösteren bir sonuç ortaya

çıkmaktadır. Burada dikkat edilmesi gereken nokta η değerleri, yıldırım biçiminde dallanmanın ve kırıklıkların derecesini etkileyen faktör olarak karşımıza çıkmış olmasıdır.

Yöntemin Üstünlükleri:

- Teknik İçerik: DBM ile yapılan modellemelerde, belirli bir alan içersinde, Laplace denklemi kullanılarak her adımda alana ilişkin potansiyel dağılımı hesabı gerçekleştirilmektedir. Yıldırımın, gerçekte, atmosferde havanın delindiği veya delinmeye en yakın olduğu noktaları izleyerek yeryüzüne ulaştığını biliyoruz. Bu nedenle alan içersinde potansiyel dağılımlarına bağlı olarak elde edilen sonuçlar, Sızma Teorisi veya Rasgele Orta Nokta Kaydırma Algoritması yöntemlerinde olduğu gibi tamamen olasılık hesaplarına ve raslantısal seçimlerine dayalı değildir.
- η katsayısı (kuvvet katsayısı): DBM'de η katsayısı, kendisine değişik değerler atanması ile, modellemede elde edilecek sonuçlara genel olarak müdahale edilmesine olanak tanımaktadır. Yıldırımın daha fazla dallanmasını, daha az kırılmasını, daha karmaşık veya daha yalın olmasını genel anlamda sağlayabilecek bu katsayı ile yıldırım için en optimum değerleri tespit etmek veya istenilen doğrultuda sonuçlar elde edebilmeye fırsat vermektedir. Sızma Teorisi veya Rasgele Orta Nokta Kaydırma Algoritması'nda (RMDA) ise sonuçlar tamamen rasgele şekilde meydana gelmektedir ve bu sonuçlara bir müdahalede bulunması mümkün değildir. Sızma Teorisi'nde seçilecek noktalar için olasılık değeri yüksek veya düşük tutularak dallanmanın azalması veya artması sağlanabilir fakat bu yöntemin de genel olarak yön belirlenmesi konusunda sıkıntıları vardır çünkü ilerleme her doğrultuda gerçekleşebilmektedir.

Yöntemin Eksik Yanları:

- Hız: Yöntem her adımda ve her nokta için Laplace denkleminin hesabını ve iterasyonlarını gerektirdiğinden dolayı, çok küçük ölçekteki (10×10 ızgara) modellemelerde dahi oldukça fazla süreye ve yüksek sistem gereksinimlerine gereksinim duymaktadır.
- Olasılık Temelli: Yöntemde her ne kadar her adımda ve her nokta için potansiyel hesabı ve bu dağılıma ilişkin komşu noktalar için olasılık değerleri

hesap ediliyor olsa dahi, komşu noktalar tamamen rasgele seçildiğinden dolayı, bir noktada bu potansiyel dağılım hesabı önemini yitiriyor gibi görünmektedir.

- Yalnızca dört yönlü ilerleme: DBM'de de Sızma Teorisi ve Laplasyen Büyümesi'nde olduğu gibi ilerlemeler her bir adımda bir nokta şeklinde gerçekleşmektedir. Seçilen noktalar, kendisine komşu olan noktanın altı, üstü, sağı ve solu şeklinde olabilmekte yani ilerleme sadece eksenel doğrultularda gerçekleşmekte ve dört yön ile sınırlı kalmaktadır.

5. ÖNERİLEN MODELLER VE ÖZELLİKLERİ

Bu bölümde, tez çalışmasınd önerilen yöntemlerin iç yüzü, teorik tasarımı ve işleyiş mantığı tanımlanmaktadır. Tez çalışmasının ikinci ana kısmını oluşturacak olan "Benzetim" bölümü ise, Bölüm 6'da, bu bölümde önerilen modellerin, Matlab ortamında program kodu olarak nasıl işlediğini ve nasıl sonuçlar ortaya koyduğunu ortaya koyan bilgileri içerecektir.

5.1 Modeller Hakkında Genel Bilgi ve İşleyiş Mantığı

Tez çalışmasında genel olarak iki model önerilmiştir:

- Dört yönlü ilerleme modeli
- Sekiz yönlü ilerleme modeli

Her iki modelde de işleyiş mantığı oldukça aynıdır, tek fark ilk modelde ilerlemenin her adımda dört yönde (yatay-düşey) ikinci modelde ise sekiz yönde (yatay-düşey yönlere ek olarak çapraz yönler) olmasıdır fakat programlama sırasında ve görsel sonuçlarda bazı farklılıklar ortaya çıkmaktadır. İki modelin karşılaştırılması Bölüm 5.5'te ayrıntılı olarak anlatılacaktır.

Modelin ilkesi, üst sınırına yakın bir yıldırım öncü ucu konumlandırılarak, alttaki sınırla (çoğu zaman toprak) arasındaki alanda oluşan potansiyel farka ilişkin dağılımın hesabı ile yıldırımın her döngüde bir adım ilerleyecek şekilde sınıra adım adım yaklaşması ve temas edip sonlanması şeklindedir.

Modellerin esasını mevcut sınırlar içersindeki potansiyel dağılımını hesaplamak oluşturmaktadır. Yıldırımın ilerlerken, bir sonraki adımlarda seçeceği noktanın neresi olacağına en çok etkiyen faktör bu noktalardaki potansiyel değerleridir.

Model sadece bu mantık üzerine kurulu olsa idi, yıldırım oldukça sade ve düz hatları olan ilerlemeler gösterecekti fakat biliyoruz ki yıldırım dallanmalar ve kırılmalar gösteren bir yapıya sahiptir. Bunun en önemli nedeni havada heterojen olarak dağılmış yüklü parçacıklar bulunması ve bu noktalardaki alan değerleri daha yüksek olduğu için dolayısıyla da delinmeye yani iletken hale geçerek yıldırımın bir parçası

olmaya daha müsait olduğu için yıldırım genellikle bu parçacıkların üzerinden ilerlemeye meyil göstermektedir. Yıldırımdaki dallanmaların ve kırılmaların asıl nedeni budur. Bu bilgi üzerinden yola çıkarak, modele Bölüm 5.3'te ayrıntılı olarak gösterileceği üzere, bu parçacık kavramının katılabilmesi ve daha gerçekçi sonuçlar ortaya konulabilmesi için modellere bir "rasgele sayı algoritması" eklenmiştir.

Yıldırımın gerçek davranışı düşünüldüğü zaman ilerleme yönünü etkileyen faktörler, çevresindeki alan dağılımı ve yüklü parçacıkların bu alana etkisidir. Dolayısıyla model ile gerçek durum arasında oldukça benzer bir yapı vardır, bu nedenle de sonuçlarda büyük oranda gerçekçi elde edilmiştir.

Model, değerleri dışarıdan girilecek olan 4 ana değişkene bağlı olarak çalışmaktadır. Bunlardan ilk ikisi, matrisin yani düzlemin büyüklüğünü belirleyecek olan n1 ve n2 katsayılarıdır. Üçüncü değişken ise Bölüm 5.3'te detaylı olarak anlatılacak olan güç katsayısı, η'dir. Son değişken ise, potansiyel dağılımındaki doğruluğu ve benzetim süresini doğrudan etkileyecek olan "iterasyon sayısı"dır.

Potansiyel dağılımının hesaplanması için uygulanabilecek çeşitli yöntemler vardır. Modelde ve programlamada bu hesap "Merkezi Sonlu Farklar Yöntemi" (MSFY) ile gerçekleştirilmiş, sınır koşullarındaki değerler ise "Neumann Sınır Koşulları"'na göre hesaplanmıştır.

Modelin diğer evrelerinde ise bilinen matematik işlemleri, olasılık hesapları, "güç katsayısı, η'in bu hesaplara harman edilmesi, rasgele sayı üretme algoritmaları gibi yöntemler kullanılarak sonuca gidilmektedir.

5.2 Sonlu Farklar Yöntemi ile Potansiyel Dağılımı Hesabı

Yıldırımın izleyeceği yol, havadaki potansiyel ve alan dağılımı ile birebir bağlantılıdır. Yıldırım yeryüzüne doğru ilerlemekte iken, alanın yoğun yani yüksek olduğu noktalardan geçmeye meyil göstermektedir. Bunun sebebi bu noktalarda havanın delinme değerine yaklaşmış ya da bu değeri geçmiş olmasından kaynaklanmaktadır. Bu şekilde iletken olmaya yaklaşan ya da iletken hale gelmiş olan hava, yıldırımının bu noktalara atlama yaparak ilerlemesini kolaylaştırmaktadır.

Bu nedenden dolayı, yıldırım yolunun benzetiminde ilk olarak hesaplanması gereken büyüklük, öncü uç adı verilen yıldırımın çıkış noktasının ve çevresindeki bölgenin,

toprağa kadar olan kısmındaki alanda potansiyel dağılımının ne şekilde olduğudur, yani bölge üzerinde her noktanın potansiyeli bilinmelidir.

Merkezi Sonlu Farklar Yöntemi (MSFY), bir bölgede potansiyel dağılımının hesaplanmasında kullanılabilecek olan Sonlu Farklar Yöntemi (SFY)'nin en çok tercih edilen türlerinden birisidir. İncelenecek bölge, kare gözleri olan ızgaraya bölünürse, potansiyel dağılımı MSFY ile kartezyen koordinatlarda iki boyutlu Laplace denkleminden de yararlanılarak aşağıdaki şekilde hesaplanabilir. Kartezyen koordinatlarda iki boyutlu (2D) Laplace Denklemi:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \qquad \textbf{(5.1)}$$

olmak üzere, (h x h) kare gözlerden oluşan bir bölgede herhangi bir (x, y) noktası için ikinci mertebeden türevlerin merkezi sonlu fark yaklaşık eşitlikleri aşağıdaki gibidir:

$$\frac{\partial^2 V}{\partial x^2} \cong \frac{V(x+h, y) - 2V(x, y) + V(x-h, y))}{h^2} \qquad \textbf{(5.2)}$$

$$\frac{\partial^2 V}{\partial y^2} \cong \frac{V(x, y+h) - 2V(x, y) + V(x, y-h))}{h^2} \qquad \textbf{(5.3)}$$

Buradan, Laplace denkleminin sonlu fark yazılımı aşağıdaki gibi olur:

$$V(x+h,y) + V(x-h,y) + V(x,y+h) + V(x,y-h) - 4V(x,y) = 0 \qquad \textbf{(5.4)}$$

Yukarıdaki denklem düzenlenirse, herhangi bir noktanın potansiyeli:

$$V(x, y) = \frac{1}{4}[V(x + h, y) + V(x - h, y) + V(x, y + h) + V(x, y - h)] \qquad \textbf{(5.5)}$$

veya alt indisli gösterim ile:

$$V_{i,j} = \frac{1}{4}[V_{(i+1,j)} + V_{(i-1,j)} + V_{(i,j+1)} + V_{(i,j-1)}] \qquad \textbf{(5.6)}$$

şeklini alır [19].

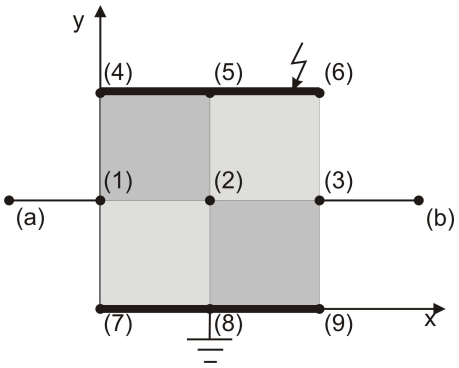
Buradaki son bağıntıları basit ve sözel bir yaklaşımla söyleyecek olursak, MSFY'ne göre, kare gözlerden oluşmuş iki boyutlu bir bölgede, herhangi bir karenin potansiyeli, solundaki, sağındaki, yukarısında ve aşağısındaki, toplamda 4 komşu noktasının potansiyellerinin ortalamasına, yani toplamlarının dörtte birine eşittir.

Sınır koşulları: MSFY ile bir ağın potansiyel dağılımı hesaplanırken sınır koşullarının bilinmesi gerekir çünkü sınır noktalarının hesaplanacağı anda, formülde sınır noktasının dışında kalacak bir noktanın alan değerine gereksinim duyulmaktadır.

Sınır koşulunun bilinmediği durumlarda ki tez çalışmasında bu durum söz konusudur, sınır noktalarındaki potansiyel değerlerinin hesaplanması gerekir. Bu hesap için Neumann sınır koşulu bağıntısı kullanılabilir:

$$\left(\frac{\partial V}{\partial x}\right)_{i,j} = 0 \text{ veya } \left(\frac{\partial V}{\partial y}\right)_{i,j} = 0 \qquad \textbf{(5.7)}$$

Bu durumu bir şekil üzerinde gösterecek olursak;



Şekil 5.1 : Neumann Sınır Koşulu Örneği

Yukarıdaki verilen şekilde, iki boyutlu kare bölgede (4), (5) ve (6) noktalarının potansiyeli ile (7), (8) ve (9) noktalarının potansiyelleri biliniyor olsun. Bu durumda (1) ve (3) noktalarının potansiyellerini Sonlu Farklar Yöntemi ile hesaplamak için, bölgede yer almayan (a) ve (b) simetrik noktalarının da potansiyellerinin bilinmesi gerekmektedir. (2) numaralı düğümün çevresindeki noktalar ise sınır içersinde olduğundan hesabı kolayca yapılabilir. Sınırlar içersinde yer almayan bu (a) ve (b) noktalarının potansiyelleri Neumann sınır koşullarına göre hesaplanabilir ve bölgedeki potansiyel dağılımı hesaplanabilir hale gelmektedir.

$$V_1 = \frac{1}{4}(V_2 + V_4 + V_a + V_7) \quad \textbf{(5.8)}$$

$$V_3 = \frac{1}{4}(V_b + V_6 + V_2 + V_9) \quad \textbf{(5.9)}$$

$$\left(\frac{\partial V}{\partial x}\right)_1 = \frac{V_a - V_2}{2h} = 0 \rightarrow V_a = V_2 \quad \textbf{(5.10)}$$

$$\left(\frac{\partial V}{\partial x}\right)_3 = \frac{V_b - V_2}{2h} = 0 \rightarrow V_b = V_2 \quad \textbf{(5.11)}$$

olarak bulunur. V_2 noktasının potansiyeli ise sınır potansiyellerinden hesaplanabilir durumdadır:

$$V_2 = \frac{1}{4}(V_3 + V_5 + V_1 + V_8) \quad \textbf{(5.12)}$$

Üç bilinmeyenli üç denklem, istenilen yöntemlerden birisi ile rahatlıkla çözülerek 1, 2 ve 3 düğümlerinin potansiyelleri bulunur.

Tez çalışmasında sınır üzerindeki ve sınır içindeki potansiyel değerlerinin hesabı için program kodu yazılırken bu iki yöntem kullanılmış ve potansiyel değerleri bu şekilde hesaplanmıştır. Matlab programında gerçekleştirilen benzetimde yapılan bu işlemlerinden birer tanesini MS Excel programında örnek olması açısından göstermek gerekirse:

99	99,2841	99,2864	99,2886	99,2908
99	99,418	99,4198	99,4216	99,4235
100	99,5584	99,5598	99,5611	99,5625
100	99,7034	99,7043	99,7052	99,7062
100	99,8512	99,8517	x	99,8526
100	100	100	100	100

(a)

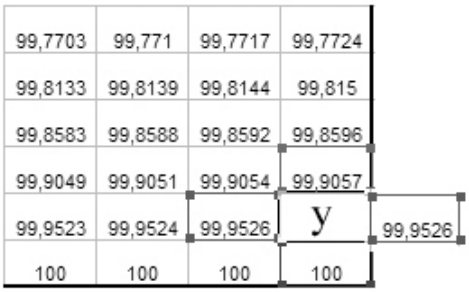
99,7703	99,771	99,7717	99,7724
99,8133	99,8139	99,8144	99,815
99,8583	99,8588	99,8592	99,8596
99,9049	99,9051	99,9054	99,9057
99,9523	99,9524	99,9526	y
100	100	100	100

(b)

Şekil 5.2: İki Farklı Noktanın Potansiyel Hesabı

Şekil 5.2(a)'da görüldüğü üzere, x noktası bir sınır noktası değildir ve potansiyeli hesaplanacağı zaman çevresindeki dört karenin potansiyel değerlerinin ortalaması

alınmaktadır. Şekil 5.2(b)'de ise "y" noktasının bir sınır noktası olduğu görülmektedir ve çevresindeki üç nokta hesaplamakta kullanılabilmektedir. Burada "y" noktasının sınır dışında kalan sağındaki noktanın değeri, yukarıdaki bağıntıdan elde ettiğimiz sonuca göre solundaki noktaya eşit olarak alınacaktır. Dolayısıyla "y" nin sağında, solundaki değerde bir nokta varmış gibi kabul edilir ve hesaplama aşağıdaki şekle göre aynı şekilde yapılır:



Şekil 5.3 : Neumann Sınır Koşulu'nun SFY Üzerinde Uygulanışı.

Yukarıdaki örnekte sınır koşulları 100 ve 0 olan bir bölgeye ilişkin potansiyel dağılımının bir bölümü verilmiştir. Burada 100 ve 0 değeri taşıyan sınır noktalar dışındaki noktalar MSFY kullanılarak hesap edilmektedir. Fark edileceği üzere her bir noktanın hesabı, çevresindeki noktaların değerleri ile gerçekleştirildiğinden, her bir nokta çevresindeki değerlere bağlıdır ve bu yüzden en gerçekçi değerlere ulaşılabilmesi için iterasyonların, artık değerler değişmeyecek duruma gelinceye kadar sürdürülmesi gerekmektedir.

Aşağıdaki şekilde sınır koşulları 0 ve 100 olan iki boyutlu 15 x 15 bir kare bölgedeki SFY ve Neumann Sınır Koşulu'nun uygulandığı örnekte 5 ve 200 iterasyon sonucunda elde edilen sonuçlar gösterilmiştir.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0,09766	0,21973	0,31128	0,36469	0,39139	0,4034	0,40841	0,41038	0,41112	0,41138	0,41148	0,41151	0,41152	0,41152	0,72539
1,75781	2,51465	2,97852	3,22418	3,34167	3,39375	3,41549	3,42414	3,42745	3,42868	3,42912	3,42928	3,42933	3,88266	5,10089
9,8938	11,676	12,6648	13,1655	13,4015	13,5064	13,5508	13,5689	13,576	13,5787	13,5797	13,5801	13,9636	15,1665	17,2667
30,2002	32,3944	33,5632	34,1487	34,4264	34,552	34,6064	34,6291	34,6383	34,6419	34,6433	34,8174	35,5145	36,9585	39,088
62,5748	63,9797	64,7275	65,107	65,2909	65,3763	65,4144	65,4308	65,4377	65,4405	65,4741	65,6653	66,175	67,0755	68,3097
100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Şekil 5.4 : 5 İterasyon Sonucu Hesaplanan Değerler

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7,14285	7,14285	7,14285	7,14285	7,14285	7,14286	7,14286	7,14286	7,14286	7,14286	7,14286	7,14286	7,14286	7,14286	7,14286
14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857	14,2857
21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286	21,4286
28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714	28,5714
35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143	35,7143
42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571	42,8571
50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428	57,1428
64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857	64,2857
71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286	71,4286
78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714	78,5714
85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143	85,7143
92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571	92,8571
100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Şekil 5.5 : 200 İterasyon Sonucu Hesaplanan Değerler

5.3 Dört Yönlü İlerleme Modeli

Tez çalışmasının ilk yarısını yıldırım yolunun sadece dört yönlü olarak ilerleyebildiği varsayımının göz önüne alınarak yapıldığı benzetimler oluşturmaktadır. Sözü edilen bu yönler yukarı, aşağı, sol ve sağ yönlerdir yani ilerleme her bir adımda yatay veya düşey olarak gerçekleşebilmektedir.

Benzetim sırasında yapılan işlemler kabaca şunlardır:

- Potansiyel dağılımının hesabı,
- Sınır koşullarının yıldırım biçimine göre oluşturulması,

- Bölüm 5.3.1 ve Bölüm 5.4.1'de belirtilen algoritmalar yardımı ile bir sonraki adımda seçilecek noktanın belirlenmesi,
- Yıldırıma bu noktanın eklenerek bir adım ilerlemesinin sağlanması,
- Sınıra ulaşıncaya kadar bu işlemlerin tekrarlanması.

5.3.1 Benzetimin Evreleri ve Yapılan Hesaplamalar

Analizde sağlayacağı kolaylıklar, yerden ve zamandan kazandıracak olması nedeniyle, benzetimler 15 x 15 boyutunda, iki boyutlu bir bölge üzerinde gerçekleştirilmiştir. Değişik biçimler için defalarca çalıştırılmış olan benzetimde güç katsayısı $\eta = 1,3$; iterasyon sayısı da 100 olarak alınmıştır. Bu çerçevede dört yönlü ilerlemeye sahip yıldırım benzetiminde kullanılan yöntemler ve yapılan işlemler adım adım aşağıdaki şekilde gösterilebilir:

1. MSFY ve Neumann Sınır Koşulu Yötemleri Kullanılarak Potansiyel Dağılımının Hesaplanması:

   Bölüm 5.2'de 15 x 15 büyüklüğünde iki boyutlu bir bölgedeki potansiyel dağılımının Merkezi Sonlu Farklar ve Neumann Sınır Koşulu yöntemleri kullanılarak nasıl hesaplandığı gösterilmişti. Burada ise yine bölge 15 x 15 göz büyüklüğündedir fakat sınır koşulları Niemeyer'in önerdiği Yıldırım Modeli'ne göre güncellenmiş ve üst sınır değişikliğe uğramıştır. Üst sınır bir düzlem sınırdan farklı olarak, yıldırımın öncü ucunu temsil etmesi anlamında alt sınıra dik bir açı oluşturacak şekilde değiştirilerek yeniden yapılandırılmıştır. Niemeyer'in önerdiği Yıldırım Biçimini kullanan model aşağıdaki şekilde gösterilmiştir:

31,43	30,68	28,36	24,31	18,30	10,09	0	10,09	18,30	24,31	28,36	30,68	31,43
32,19	31,46	29,22	25,30	19,39	11,04	0	11,04	19,39	25,30	29,22	31,46	32,19
34,40	33,75	31,76	28,28	22,93	14,67	0	14,67	22,93	28,28	31,76	33,75	34,40
37,91	37,38	35,80	33,13	29,37	24,72	20,56	24,72	29,37	33,13	35,80	37,38	37,91
42,46	42,07	40,92	39,07	36,70	34,26	32,82	34,26	36,70	39,07	40,92	42,07	42,46
47,80	47,52	46,74	45,53	44,10	42,80	42,20	42,80	44,10	45,53	46,74	47,52	47,80
53,67	53,49	52,98	52,22	51,37	50,66	50,36	50,66	51,37	52,22	52,98	53,49	53,67
59,91	59,79	59,47	59,00	58,50	58,09	57,94	58,09	58,50	59,00	59,47	59,79	59,91
66,37	66,30	66,10	65,82	65,52	65,29	65,20	65,29	65,52	65,82	66,10	66,30	66,37
72,98	72,94	72,82	72,65	72,47	72,34	72,29	72,34	72,47	72,65	72,82	72,94	72,98
79,68	79,65	79,58	79,48	79,38	79,31	79,28	79,31	79,38	79,48	79,58	79,65	79,68
86,43	86,42	86,38	86,32	86,27	86,22	86,21	86,22	86,27	86,32	86,38	86,42	86,43
93,21	93,20	93,18	93,16	93,14	93,12	93,11	93,12	93,14	93,16	93,18	93,20	93,21
100	100	100	100	100	100	100	100	100	100	100	100	100

Şekil 5.6 : 15x15 Matriste Potansiyel Dağılımı

Şekil 5.6'dan görüleceği üzere, Niemeyer'in Yıldırım Biçimi modellenmiş, alt sınır yatay olarak belirlenmiş ve toprağa benzetilmiş, üst sınır ise, yıldırımın öncü ucunu tanımlayacak şekilde konumlandırılmıştır. Burada sınırın 100 olarak gösterilmesi, örneğin 100 kV olarak düşünülebilir.

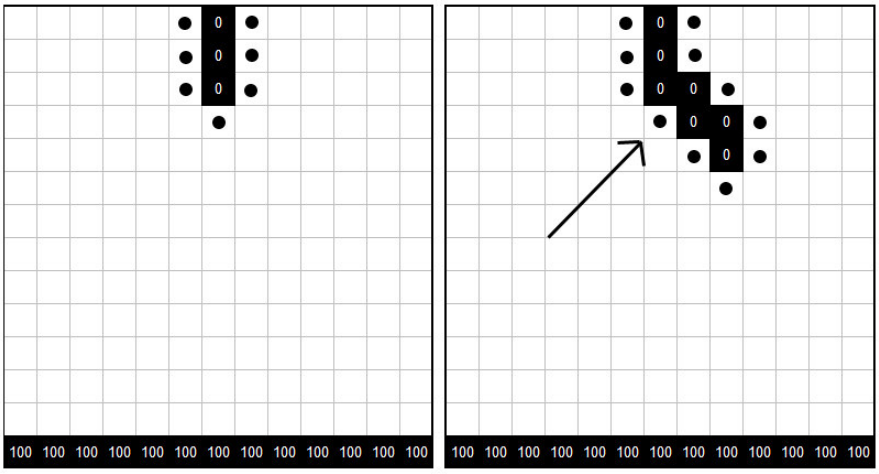
Yıldırım biçimine [14,20] bakıldığında ilk olarak doğaya ters olduğu fark edilmektedir çünkü buradaki düzenlemeye göre yıldırımın potansiyeli 0 V, toprağın potansiyeli ise 100 kV gibi anlaşılmaktadır. Bunun tek sebebi, her bir noktanın yani karenin seçilme olasılığının hesabı sırasında o noktanın potansiyel hesabından yararlanılmakta olmasıdır. Eğer biçim tersi şekilde, yani öncü ucun 100 kV ve toprağın 0 V değerlerini aldığı durumda olasılık hesabı yapılacak olursa, potansiyel dağılımı yukarıdan aşağıya doğru azalacağından dolayı, potansiyel değeri ile doğru orantılı değişen olasılık değerleri de yukarıdan aşağıya doğru gidildikçe azalacaktır. Halbuki yıldırım aşağı doğru gitmeye meyillidir. Yukarıya doğru ilerlemeyi arzu etmez çünkü en büyük potansiyel farkın olduğu yöne doğru yani aşağıya doğru ilerlemek ister. Bu tür bir biçimin tercih edilmesi halinde, yıldırımın bir sonraki adımı için olasılık değerleri en

büyük olan değil, en küçük olanı tercih edilebilir. Bu detay seçilme olasılıklarının hesaplandığı 3. adımda daha net olarak anlaşılacaktır.

Bu nedenle sınır koşulları yıldırım biçiminde yer değiştirmiş, olasılık değerlerinin gerçekle paralel olması, yani yüksek olasılık değeri olan noktanın seçilme olasılığının da yüksek olduğu mantığının korunması için sınırlar bu şekilde yer değiştirmiştir. Önemli olan nokta, sınırlarda neresinin 0 değeri ya da 100 değerini aldığı değil, arada ne kadar bir potansiyel farkın mevcut olduğudur.

2. Bir sonraki adımda gidilebilecek noktaların belirlenmesi:

Çalışmada en kritik ve zorlu olan nokta bu kısımdır. Her döngüde gelişigüzel yönlerde bir adım daha ilerleyen yıldırımın bir sonraki adımda gidebileceği noktaları doğru olarak belirlemek çalışmanın temelini oluşturmaktadır.



Şekil 5.7 : Bir Sonraki Adımda Seçilebilecek Noktalar

Yukarıdaki şekilde basit ve nispeten daha karmaşık durumlar için bir sonraki adımda seçilebilecek noktalar gösterilmiştir. Şekil 5.7'deki soldaki durumda yıldırım biçiminin başlangıç durumu görülmektedir. Burada ilk adımda seçilebilecek noktalar gösterilmiştir. Şekil 5.7'deki sağdaki durumda ise birkaç adım sonunda yıldırımın geldiği nokta sonrasındaki adım için seçilebilecek noktalar belirtilmiştir [17]. Görüldüğü üzere her bir adımdan sonra, yıldırım büyüdükçe seçilebilecek nokta sayısı da artış göstermekte ve noktaların belirlenmesi zorlaşmaktadır.

Burada en çok dikkat edilmesi gereken nokta, her bir olası noktanın bir kere belirlenmesi gerektiğidir. Ok ile gösterilen "aday nokta"ya bakılacak olursa bu

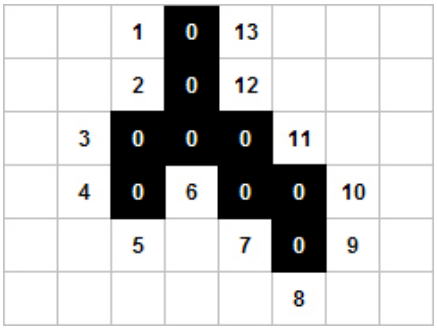
nokta, dört yönlü bir ilerleme göz önüne alındığında hem hemen yukarısındaki 0 noktasının, hem de sağındaki 0 noktasının etki alanındadır ve bu iki nokta kesişmektedir fakat bu aday nokta yalnızca bir kere olasılık hesabına alınmalıdır. Eğer olasılık hesabında hem yukarısındaki 0 noktasının aşağı gidebileceği şekilde, hem de sağındaki noktanın bir sola gidebileceği şekilde düşünülürse ve iki kere hesaba katılırsa, alması gerektiği olasılık değerinin iki misli bir değer alacaktır ki bu durumda toplam olasılık değeri 1'in üzerine çıkacak ve hata yapılmış olacaktır. Dolayısıyla bu nokta yukarıdaki 0 tarafından hesaba katılmış ise, sağdaki 0 tarafından hesaba katılmamalı, ya da tam tersi şekilde sağdaki 0 tarafından hesaba katılmış ise yukarısındaki 0 tarafından hesaba katılmamalıdır. Algoritma bunlardan bir tanesini seçecek şekilde tanımlı olmalıdır. Özetlemek gerekirse, her aday nokta ve bu noktanın sahip olduğu potansiyel değeri, sadece bir kez hesaba katılmalıdır. Tezde bunu sağlayacak algoritma şu şekilde geliştirilmiştir:

a) Her Satır Yatay Olarak Taranmalı: Program her satırı soldan sağa doğru taramaktadır. Tarama sırasında 0 değeri ile (yıldırım) karşılaştığı anda 0 değerinin sağındaki ve solundaki potansiyel değerlerini alacaktır. Eğer 0 değerinin iki sağında başka bir 0 daha var ise, bu sefer sadece ilk karşılaştığı 0 değerinin solundaki değeri alacaktır.

b) Her sıfır değeri ile karşılaşıldığında sağ alt ve sol alt çaprazı sorgulanacaktır. Eğer ki hem sol alt hem de sağ alt değerleri 0'dan farklı ise, 0'ın hemen altındaki değer hesaba alınacaktır.

c) Her sıfır ile karşılaşıldığında bir alttaki satır sorgulanacaktır. Eğer ki 0'ın alt satırında sınır değeri mevcut ise, son satıra gelindiği anlaşılacak ve benzetim sona erecektir.

Şimdi bu algoritma bir örnek ile aşağıdaki biraz karmaşık bir adımda, Şekil 5.7'deki sağdaki durum üzerinde sınanacaktır:



Şekil 5.8 : Başka Bir Durum İçin Komşu Noktaların Belirlenmesi

İlk satır taraması: Algoritma ilk olarak ilk satırdaki sıfıra geldiği anda, 0'ın iki sağında başka bir 0 olup olmadığını sorgulayacaktır. Başka 0 olmadığı için 1 ve 13 noktalarını alacaktır. Daha sonra bu sıfırın sağ alt ve sol alt değerlerine bakacak. Buradaki değerlerin sıfırdan farklı olduğunu görecek ve 0'ın hemen altındaki değeri hesaba katacak. Burası 0 olacağı için sorun yaratmayacaktır. Daha sonra bir alt satırda sınır değeri olan 100'ü arayacaktır, olmadığı için döngü devam edecek ve ikinci satıra geçecektir.

İkinci satır taraması: İlk satır ile aynı durum olduğu için, 2 ve 12 noktaları seçilecek ve hemen altında etki etmeyecek olan 0 değeri alınacaktır.

Üçüncü Satır taraması: Burada yan yana üç tane 0 bulunmaktadır. Tarama soldan sağa doğru yapıldığı için ilk olarak en soldaki 0 ile karşılaşılacaktır. Burada 0'ın iki sağında sıfır olduğu için sadece soldaki değer yani 3 alınacaktır ki burada sağdaki nokta da 0 olduğundan alınması sorun yaratmayacaktır fakat sorun yaratacağı bir durum 4. satırda mevcuttur. Daha sonra 0'ın sağ alt ve sol alt çaprazları sorgulanacak ve 0'dan farklı olduğu için altındaki etki etmeyecek olan 0 değeri de alınacaktır. Ortadaki 0'a geçildiği zaman sağındaki ve solundaki etki etmeyecek olan 0'lar alınacaktır. Sağ alt ve sol alt çaprazlarında sıfır olduğundan ki bir tanesinin olması yeterli idi, altındaki değeri almayacaktır. En sağdaki 0'a geldiği zaman ise, sağındaki 11 değerini ve solundaki etki etmeyecek olan 0 değerini alacak, sağ alt çaprazında 0 olduğu için hemen altındaki değeri almayacaktır.

Dördüncü satır taraması: Burada ilk 0'a gelindiği zaman iki sağ tarafında da 0 olduğu için sadece 4 karesini alacaktır, sağ alt ve sol altında 0 olmadığı için hemen altındaki 5 noktasını da alacaktır. İkinci 0'a geldiği zaman soldaki 6

noktasını ve etki etmeyecek olan sağındaki 0 noktasını alacaktır. Sağ altında bir 0 olduğu için 7 noktasını almayacaktır. En sağdaki 0 noktasına geçtiği zaman ise hemen sağındaki 10 karesini ve etki etmeyecek olan solundaki 0 değerini alacaktır. Sağ altında ve sol altında 0 olmadığı için hemen altındaki etki etmeyecek olan 0 değerini alacaktır.

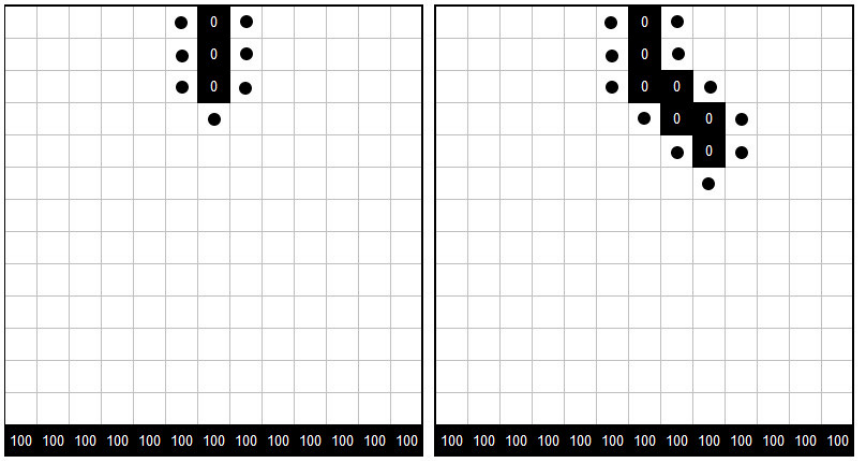
Beşinci Satır Taraması: Burada da 0'a rastladığı zaman solundaki ve az önce alınmamış olan 7 karesini ve sağındaki 9 karesini alacak, sağ alt ve sol altında 0 olmadığı için hemen altındaki 8 değerini alacaktır.

Görüldüğü üzere bütün noktalar hesaba eksiksiz olarak ve sadece bir kere katılmış olmaktadır. Daha önce belirtildiği üzere dikkat edilmesi gereken husus birden fazla sıfırın etki alanında kesişen noktaların olduğu alanlardır. Bu noktaların hesaba bir kez katılması gerekmektedir. Burada örneğin 6 noktası, hem yukarısındaki hem sağındaki hem de solundaki sıfırlar tarafından hesaba katılabilir ve hataya neden olabilirdi. Yazılan algoritmadaki amaç bunun önüne geçecek şekilde tercihlerde bulunmaktır. Bir sonraki adımda bu noktaların sadece bir kere hesaba katılması durumunda komşu noktaların olasılıklarının toplamının 1 olduğunu göreceğiz. Eğer ki herhangi bir nokta birden fazla defa seçilecek olursa, alması gereken olasılık değerinden fazla alacak ve toplam olasılığın 1'in üzerine çıkmasına neden olacak ve hata oluşturacaktır.

3. Bir sonraki adım için komşu olası noktaların potansiyel değerlerinin belirlenmesi ve olasılıklarının hesaplanması:

   Bir önceki adımda, yıldırımın herhangi bir adımında, bir sonraki adım için seçebileceği noktaların nasıl tayin edildiği detaylı olarak gösterilmiştir. Şimdi bu noktaların potansiyel değerlerinin alınması ve bu değerlere bağlı olan seçilme olasılıklarının nasıl hesap edileceği anlatılacaktır.

   Potansiyel dağılımının yapılmasının nedeni, bölgede hangi noktanın hangi potansiyel değerine denk geldiğinin bilinmesi gerekliliğine dayanmaktadır. Aşağıdaki şekilde yıldırımın başlangıç biçimi ve birkaç adım ilerlemiş hali için, bir sonraki adımda tercih edilebilecek noktalar gösterilmiştir:



Şekil 5.9: Komşu Noktanın Seçimi

Yukarıdaki her bir şekilde, bir sonraki adım için gidilmesi olası noktalar gösterilmiştir. Bu noktalardan sadece bir tanesi bir sonraki adım için seçilecektir. Bir sonraki adımda seçilecek noktanın belirlenmesini iki tane ana faktör etkilemektedir.

a) Noktaların potansiyel dağılımındaki değerlerine denk düşen olasılıkları

b) Bu noktalar için üretilen 0 – 1 arasındaki rasgele sayılar

Potansiyel dağılımı hesabı yapıldıktan sonra yukarıdaki her olası nokta için o noktadaki potansiyel değerinin ne olduğuna bakılır. Daha sonra herhangi bir komşu olası noktanın potansiyel değeri, tüm komşu noktaların potansiyel değerlerinin toplamına oranlanarak, her bir noktanın seçilme olasılığı hesaplanmış olur. Böylece yukarıdaki iki adımdan ilki tamamlanmış olur. Aşağıdaki bağıntıda komşu bir noktanın diğer komşu noktalara göre seçilme olasılığı gösterilmektedir [2, 14]:

$$p(i, j) = \frac{V_{i,j}}{\sum V_{i,j}} \qquad \textbf{(5.13)}$$

Bu bağıntıda i, j komşu olası noktaların koordinatlarını göstermektedir. Bağıntıya göre demek oluyor ki, herhangi bir komşu noktanın seçilme olasılığı, o noktanın potansiyel değerinin, tüm komşu noktaların potansiyelleri toplamına oranıdır. Bu manada Şekil 5.9'da vermiş olduğumuz iki durum için komşu noktaların potansiyel farkı ve olasılık değerlerini aşağıdaki şekillerde daha açık anlaşılması bakımından gösterelim:

	10,09	0	10,09	
	11,04	0	11,04	
	14,67	0	14,67	
		20,56		

	0,11	0	0,11	
	0,12	0	0,12	
	0,16	0	0,16	
		0,22		

Şekil 5.10: İlk Adıma İlişkin Potansiyel ve Olasılık Değerleri

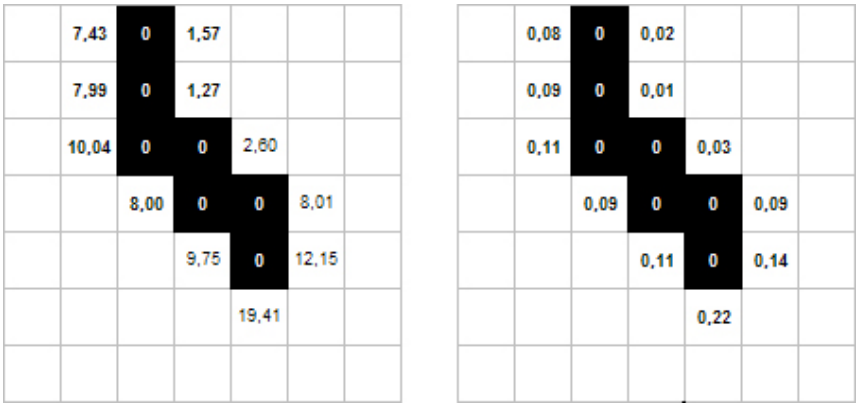
Şekil 5.10'da, Niemeyer'in önerdiği yıldırım biçiminde ilk adıma ilişkin sırasıyla potansiyel ve olasılık değerleri verilmiştir. Yıldırım biçimini 15 x 15 gözlü bölgede daha önceki kısımda oluşturmuş ve potansiyel değerlerini Merkezi Sonlu Farklar Yöntemi'ni kullanarak hesaplamış ve Şekil 5.6'da değerlerini belirtmiştik. Yöntemde 15 x 15 gözlü bölgede sadece yıldırıma dört yönlü olarak komşu olan noktalar gerekli olduğundan bu komşu noktaların potansiyel değerleri Şekil 5.10'da gösterilmiştir. Daha sonra (5.13) bağıntısı kullanılarak, komşu noktaların seçilme olasılıkları hesaplanmıştır. Burada öncü ucun sol üstündeki 10,09 kV değerine sahip noktaya V_1 noktası adı verilirse, bu noktanın seçilme olasılığı, $p(V_1)$ yukarıdaki bağıntı kullanılarak aşağıdaki şekilde hesaplanabilir:

$$p(V_1) = \frac{10{,}09}{(10{,}09 + 11{,}04 + 14{,}67 + 20{,}56 + 14{,}67 + 11{,}04 + 10{,}09)} = 0{,}11 \qquad \textbf{(5.14)}$$

Yukarıdaki bağıntıdan anlaşılmaktadır ki, V_1 noktasının bir sonraki adımda seçilme olasılığı %11'dir. Benzer şekilde tüm komşu noktalar için bağıntı kullanılarak tüm komşu noktaların seçilme olasılığı elde edilmiş olur.

Dikkat edilecek olursa, noktaların seçilme olasılıkları aşağıya inildikçe artmaktadır. Bunun nedeni alt sınırın 100, üst sınırın 0 olmasından kaynaklanmaktadır. Daha önce söz edildiği üzere, eğer bu sınırlar akla daha yatkın olacak şekilde, yıldırım ucu için 100, toprak için 0 seçilecek olsa idi, buradaki olasılık değerleri, altlara inildikçe azalacaktı. Dolayısıyla bu durumda bir sonraki adım için olasılık değeri en büyük olanın değil, en küçük olanın alınması gerekecekti.

Mevcut düzene göre bir sonraki adımda seçilecek olan nokta 0,22 değerine sahip kare olacaktı. Tam bu anda, sürekli olarak en aşağıdaki noktanın seçilmesini engellemek için 5. maddede anlatılacak olan rasgele sayı üretme algoritması devreye girecektir.



Şekil 5.11: 4 Adım Sonunda Potansiyel ve Olasılık Değerleri

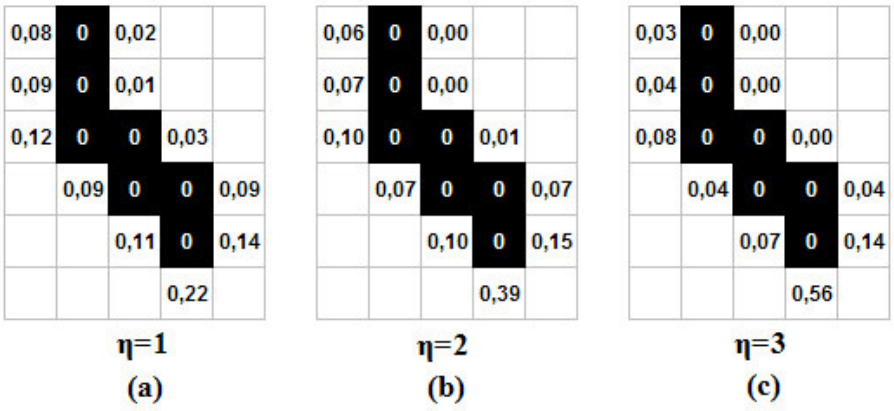
Şekil 5.11'de bir önceki örnekten bağımsız olarak, gelişigüzel bir şekilde seçilmiş, daha karmaşık bir durum için bir sonraki adımda seçilecek olan noktanın olasılık hesapları gösterilmektedir. Şekil 5.11'de soldaki durumda 4. adım sonunda komşu noktalara ilişkin potansiyel dağılımın değerleri, Şekil 5.11'de sağdaki durumda ise bu noktaların hesaplanmış olan seçilme olasılıkları gösterilmiştir. Bu örnekte de seçilecek nokta yine en büyük değere yani 0,22 olasılığına sahip olan en alttaki karedir nitekim bir önceki örnekte de belirttiğimiz üzere bu tekdüzeliğin önüne rasgele sayı üretme algoritması geçecektir. Yani hala bu noktanın seçileceği kesin değildir fakat yüksek olasılığa sahip olacaktır.

4. Güç Yasası ile bu olasılık değerleri arasındaki fark istenilen ölçüde değiştirilir:

   Üçüncü adımda vermiş olduğumuz olasılık hesabına ilişkin bağıntıyı, potansiyel ile olasılık arasındaki bağlantının oranlarını değiştirebilecek olan bir η kuvvet katsayısı kullanarak, güç-yasası'nı bu formüle adapte etmek ve daha kontrol edilebilir bir olasılık dağılımı sağlamak mümkündür [14,21]. Bağıntı η katsayısı ile aşağıdaki şekilde ifade edilebilir:

$$p(i,j) = \frac{(V_{i,j})^{\eta}}{\sum (V_{i,j})^{\eta}} \qquad \textbf{(5.15)}$$

Güç yasasının olasılık formülüne uyarlanması: Bu formülden anlaşılmaktadır ki, herhangi bir komşu noktanın seçilme olasılığı, kendisine denk gelen potansiyel dağılımının η'inci kuvvetinin, tüm komşu noktaların potansiyel değerlerinin η'inci kuvvetlerinin toplamına bölümüne eşittir. Burada η kuvvetinin pozitif değerleri için olasılık dağılımlarında nasıl bir değişiklik görüldüğü, bir önceki örnekteki değerler kullanılarak aşağıdaki şekilde gösterilmiştir.



Şekil 5.12 : η Faktörünün Olasılık Değerlerine Etkisi

Şekil 5.12'den anlaşılabileceği üzere, η değeri büyüdükçe, olasılık değeri büyük olan değerler daha büyürken, olasılık değerleri düşük olan değerler ise gittikçe küçülmektedir. Sürekli olarak en aşağı ucun en büyük olasılık değerini alacağı düşünüldüğünde, η değeri arttıkça, diğer yukarı noktalara göre arasındaki olasılık değeri farkı artacak ve daha seçilebilir hale gelecektir. Her üç koşulda da komşu noktaların olasılıklarının toplamı 1'e eşittir. Buradan çıkartılacak sonuç şudur; eğer ki en alttaki noktaların ağırlıklı olarak seçilmesi isteniyor, fazla dallanma göstermeden, daha kısa sürede ve daha yalın olarak sınıra gidilmek isteniyorsa, η değeri büyük seçilmelidir. Tersi şekilde, yukarıdaki noktaların daha seçilebilir hale getirilmesi isteniyorsa η değeri düşürülerek bu noktaların seçilme olasılığı değerleri en aşağıdaki noktaların seçilme olasılığına yaklaştırılabilir. Böylece yukarıdaki noktalar daha aktif hale gelerek daha fazla seçilecek, dolayısıyla sağ, sol ve yukarı yönlerde daha çok ilerleme göstereceği için, dallanma daha fazla görülecek dolayısıyla yıldırımın doğal görüntüsüne yaklaşacaktır. Tezde yapılan denemeler sonucunda en uygun dallanma miktarını veren değerin 1,3 olduğu kanaatine vardım. 0,8–1,5 arasındaki η değerleri yıldırımın yapısına uygun sonuçlar vermektedir. 1,5'tan itibaren daha yukarı

değerler dallanma miktarını oldukça azaltmakta, hafif sağa sola kaymalar yapan bir çubuk görünümünü vermektedir. İlerleyen bölümlerde değişik η değerleri ile çalıştırılan benzetim sonuçları gösterilecektir.

5. Rasgele sayıları üretme algoritması modele katılır:

   Fark edileceği üzere, bir sonraki adımda seçilecek olan noktanın belirlendiği iki örnekte de, en alttaki, yani en yüksek değere sahip nokta seçilmiştir. Bunun nedeni alan değerlerinin aşağıdaki sınır değerine (100 kV'a) yaklaştığı için olasılık değerlerinin de aşağı yönde gittikçe artmasından kaynaklanmaktadır. Dolayısıyla en büyük olasılık değerini sürekli olarak en aşağıdaki ucun altındaki nokta almaktadır. Eğer ki benzetim yalnızca bu işlemlere dayalı olarak sürdürülseydi, başlangıç yıldırım biçimi, sağa ve sola hiçbir sapma göstermeden, sürekli olarak en aşağıdaki kareyi seçecek ve alt sınıra kadar düz bir çizgi halinde ilerleyecekti.

   Yıldırımın doğası düşünüldüğünde, yıldırım modelinin bir çubuk gibi aşağıya inmesini değil, dallanmalar ve kollar oluşturarak toprakla buluşması arzu edilir. Bu nedenle sürekli olarak en aşağıdaki noktanın seçilmesini engelleyecek, sağdaki, soldaki veya daha yukarılardaki noktaların da seçilmesini sağlayacak, fakat genel olarak yıldırımı aşağı yönlü ilerletecek yeni bir algoritma ihtiyacı ortaya çıkmaktadır. Kısacası, bir sonraki adımın genelde aşağı yönlü olması, ama arada sırada da başka yönlerde olması tercih edilmelidir.

   Bunun için tez çalışmasında şöyle bir düşünce geliştirilmiştir: Potansiyel dağılımı hesabına bağlı olarak olasılık değerleri zaten hesaplanmaktadır. Her bir noktanın seçilme olasılığı 0–1 arasında çıkmaktadır. Bu aşamada aşağıda bulunan noktaların 1'e daha yakın değerler aldığı, üstteki noktaların ise 0'a daha yakın değerler aldığı bilinmektedir. Dolayısıyla her bir nokta için 0–1 arasında bir rasgele sayı üretip, bu noktanın olasılık değeri ile çarpılırsa, aranan sonuca yaklaşılmış olur. Böylece olasılık değeri küçük olan bir noktaya büyük bir rasgele sayı atandığında seçilme şansı doğmuş olmaktadır. Buradan olasılık değeri küçük noktalar ile olasılık değeri büyük olan noktaların seçilme şanslarının eşit ya da eşite yakın bir hale geldiği şeklinde algılanmamalıdır çünkü en alt noktaların aldığı değerler ortalama olarak 0,4–0,7 arasında iken, üstteki satırların aldığı değerler 0,01–0,2 arasında değişmektedir. Dolayısı ile seçilme

olasılığı düşük bir noktanın seçilebilmesi için iki koşulun birden yerine gelmesi gerekmektedir:

a) Olasılığı küçük nokta için 0,8 – 0,99 arası gibi çok büyük bir rasgele sayı üretilmektedir.

b) Olasılığı büyük nokta için de ortalama 0,01 – 0,4 arası çok küçük değerler denk gelmelidir.

Görüldüğü üzere, olasılık değeri zaten büyük olan bir noktaya, ortalama bir rasgele sayı gelmesi halinde bile yine seçilecek nokta o olacaktır. Bu da bizi tam olarak aradığımız noktaya getirecek, genel olarak en aşağıdaki noktaların seçilmesini, nadiren de yukarıdaki noktaların seçilerek dallanma oluşmasını sağlayacaktır.

Rasgele sayıların üretilip, olasılık değerlerine atanmasını fiziksel olarak şu şekilde açıklayabiliriz. Havadaki iyonlar, yükler ve gazlar homojen olarak dağılmamıştır. Havanın içersinde parçacıklar şeklinde uçuşmaktadırlar. Yıldırım da yolunu izlerken, havanın delinmesini kolaylaştıracak olan bu iyonların üzerinden gitmeye meyillidir. Başka bir deyişle, yıldırım, havada bulunan bu parçacıkların üzerinden geçerek toprağa doğru yol almaya meyillidir.

Bu düşünce ile üretilen rasgele sayıları mantıksal olarak, bölgenin o noktasında bir iyon bulunma olasılığı olarak tanımlayabilir ve gerçeğe uygun bir hale getirebiliriz. Benzetimin ikinci kırılma noktası ve gerçekle benzeşmesini sağlayan detay budur.

Aşağıda daha açık anlaşılması için bir bölge üzerinde rasgele sayılar üretilmiştir:

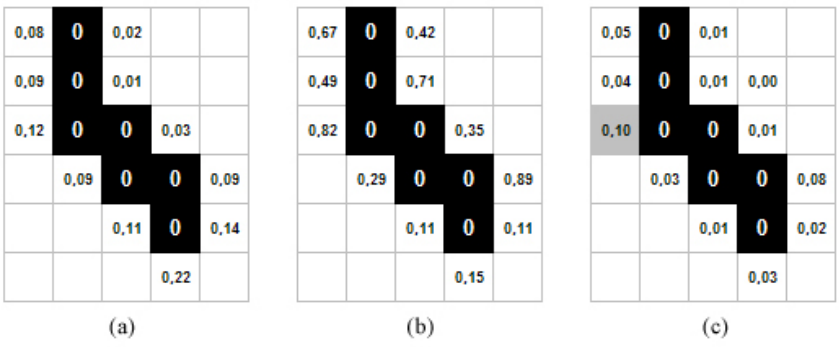
0,44	0,87	0,38	0,24	0,65	0,48	0,13	0,34	0,02	0,40	0,81	0,76	0,51
0,41	0,64	0,83	0,96	0,58	0,94	0,92	0,64	0,08	0,06	0,95	0,45	0,36
0,39	0,62	0,77	0,41	0,09	0,04	0,29	0,66	0,44	0,42	0,22	0,94	0,51
0,60	0,61	0,33	0,25	0,31	0,96	0,91	0,56	0,52	0,44	0,82	0,38	0,29
0,70	0,01	0,15	0,26	0,25	0,74	0,43	0,27	0,67	0,61	0,43	0,96	0,10

Şekil 5.13: 5 x 15'lik Bir Bölgede Üretilen 0–1 Arası Rasgele Sayılar

Yukarıdaki iki boyutlu bölgede, keyfi olarak 0,85 değerlerinden büyük değerler koyu renk ile belirtilmiştir. Bu değerleri, o noktada bir iyon bulunması olasılığı olarak kabul edecek olursak, bu noktaları hava içersinde keyfi olarak dağılmış iyonlar olarak düşünebiliriz. Dolayısıyla eğer ki bu noktalardan biri veya bir kaçı yıldırımın geçeceği doğrultuya komşu olacak olursa, buradaki potansiyele bağlı olarak hesaplanan olasılık değerlerini oldukça yukarıya çekecektir. Eğer ki bu nokta, yıldırımın aşağıdaki bir noktasına denk gelecek olursa, zaten beklenen bir şekilde aşağısı seçilecektir, nitekim eğer yıldırımın daha yukarıdaki komşu noktalarından birisine denk gelir ise, buradaki olasılık değerini yukarı çekecek ve beklenmedik bir şekilde bu noktanın seçilebilir olmasını sağlayacaktır.

6. 3. Adım'da her bir komşu nokta için hesaplanan olasılık değerleri, 4. Adım'daki rasgele sayılar ile çarpılır:

   Bir önceki adımda biraz söz ettiğimiz üzere, üretilen rasgele sayılar, denk geldikleri karelerdeki potansiyel değerlerine bağlı olarak hesaplanan olasılık değerleri ile çarpılırlar. İşte bu son işlemden sonra artık olasılık değerleri biraz değişmiş, kimileri yükselmiş, kimileri alçalmış olur. Bu son durumdan sonra, 0-1 arası değişen bu değerlerden en yüksek olanı seçilerek, yıldırımın bir sonraki adımda gideceği yer belirlenmiş olur. Aşağıdaki şekilde, yıldırımın yukarıdaki örnekle aynı durumdaki hali için sırasıyla, $\eta = 1$ için hesaplanan olasılık değerleri, bu noktalar için üretilen rasgele sayılar ve son olarak da bu iki değerlerin çarpılmış halleri gösterilmiştir:



Şekil 5.14 : Son Olasılık Durumu

7. Olasılık ve rasgele sayıların çarpılmasının ardından en büyük değere sahip olan kare seçilir:

Şekil 5.14 (c)'de gri renkle gösterilen kare bir sonraki adımda seçilecek olan karedir çünkü Şekil 5.14 (a)'daki olasılık değerleri ile Şekil 5.14 (b)'deki rasgele sayılar birbiri ile çarpıldığında (Şekil 5.14 (c)) en büyük değeri alan kare 0,10 değerine sahip olan karedir. Burada dikkat edilirse, seçilmiş olan kare, aslında 0,12 olasılık değerine sahiptir (bkz. Şekil 5.14 (a)) ve 0,22 olan en büyük değerden küçüktür ayrıca 0,02 olan en küçük değerden de büyüktür. Benzer şekilde bu noktaya denk gelen rasgele sayı da (Şekil 5.14 (b)) rasgele sayılar içersinde en büyüğü de değildir (0,89), en küçüğü de değildir (0,11). Bu sonuçlara rağmen ortalama bir olasılığa ve ortalama bir rasgele sayıya denk gelmiş olsa dahi, bu nokta, seçilebilir hale gelmiştir. Benzetimin ana çalışma mantığı bu şekildedir.

Yukarıda verilmiş olan örnek, programın çalışmasının daha rahat anlaşılması için özenle seçilmiştir. Genel olarak seçilen noktalar yıldırımın alt yarısında olmaktadır fakat nadiren bu şekilde yukarıdaki noktalar da seçilebilmektedir, zaten istenen de budur.

8. Son durum için değişen alan hesabı tekrarlanır: Yıldırımın her adımından sonra, son adımda seçilen nokta artık yıldırımın bir parçası haline geldiği için bir sınır değerini almış olur yani artık o noktanın potansiyeli 0 değerini alır. Bu noktanın potansiyeli değiştiği yani 0 olduğu için, bu değişikliğe bağlı olarak, potansiyel dağılımı da değişecektir. Bu nedenle her adımdan sonra son hal için potansiyel dağılımının yeniden hesaplanması gerekmektedir. Bu hesabın nasıl yapıldığı Bölüm 5.2'de detaylı olarak anlatılmıştı.
   Benzetim döngüsü yıldırım ucu diğer sınıra ulaşıncaya dek tekrar eder. Sınıra ulaştığı anda son bulur ve yıldırım görüntüsü elde edilmiş olur.

5.3.2 Yöntem Özeti

Benzetimin çalışma mantığını adım adım özetleyecek olursak, aşağıdaki şekildedir:

1. Adım: Merkezi Sonlu Farklar Yöntemi ve Neumann Sınır koşulları kullanılarak, iterasyon yardımı ile potansiyel dağılımı hesaplanır.
2. Adım: Yıldırımın mevcut durumunun, bir sonraki adımda gidebileceği noktalar, yazılan algoritma sayesinde belirlenir.

3. Adım: Belirlenen bu noktaların, üzerindeki bulundukları potansiyel değerine göre seçilme olasılıkları hesaplanır.
4. Adım: Güç Yasası noktalar arası olasılık değerleri arasındaki fark istenilen ölçüde değiştirilir.
5. Adım: Komşu olası noktaların her biri için 0–1 arasında rasgele sayılar üretilir.
6. Adım: Daha önce 3. Adım'da her bir komşu nokta için hesaplanan olasılık değerleri, 5. Adım'daki rasgele sayılar ile çarpılır.
7. Adım: Komşu olası noktalardan en büyük değere sahip olan nokta, yıldırımın bir sonraki adım için ilerleyeceği nokta olarak seçilir ve yıldırım bir adım ilerlemiş olur.
8. Adım: Son durum için değişen alan hesabı tekrarlanır, sınıra gelinene kadar 1-7 arasındaki adımlar tekrarlanır.

5.3.3 Sonuç

Dört yönlü ilerleme modelinin 15 x 15 matris boyutundaki sonuçlarına bakıldığına dallanmanın çok az meydana geldiği görülmektedir dolayısıyla yıldırım görüntüsüne çok yakın değildir. Bunun nedeni Bölüm 5.6'da belirtilen, benzetime etki eden faktörlerdir. Dört yönlü ilerleme modelinde, yıldırım yön kısıtlamasından dolayı "kırıklı" olarak ilerlemektedir. Bunun önüne geçmenin tek yolu ise, çözünürlüğü, yani matris boyutunu artırmaktır. Matris boyutu yeterince büyük seçildiğinde bu kırıklıklar düzeltilmişcesine ortadan kaybolacak ve gözün algılayamayacağı boyuta inecektir. İlermenin sadece dört yönlü olması, bizim onu sekiz yönlüymüş gibi algılayamayacağımız manasına gelmemektedir, bu sadece düşük mertebedeki matrislerde yani çözünürlüğün düşük olduğu durumlarda kendini belli etmektedir. Yeteri kadar büyük bir bölge oluşturulduğunda, görsel olarak sonuçlar tatmin edicidir. Bölüm 6.4.1'de 64 x 64 boyutlarındaki bir bölgede dört yönlü ilerlemeye ilişkin sonuçlar gösterilmiştir.

5.4 Sekiz Yönlü İlerleme Modeli

Tez çalışmasının ilk kısmında yıldırım yolunun modellenmesi, ilerlemedeki her bir adımın dört yönlü, yani sadece yukarı, aşağı, sağa ve sola olması durumu için gerçekleştirilmişti. Bir başka deyişle, yıldırımın oluştuğu ve yolunun belirlenmeye

başladığı aşamadaki her bir ufak ilerlemeyi bir adım olarak kabul ettiğimizde, bir sonraki her bağımsız adım için seçilebilecek dört yön mevcut idi.

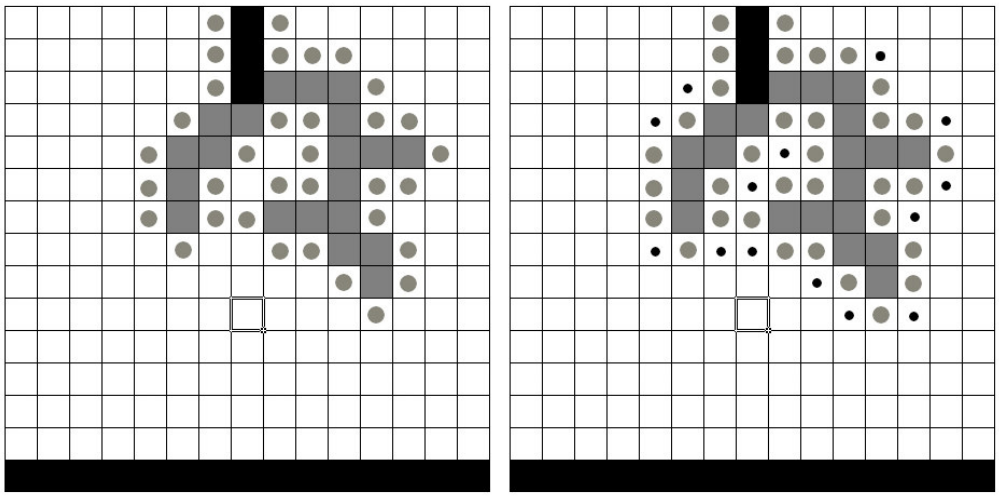
Tez çalışmasının ikinci kısmında ise, bu dört yöne ek olarak, doğada, ilerlemenin çapraz yönlere de gidebileceği göz önüne alınarak, çapraz yönlerin ortaya çıktığı dört yön daha eklenmiştir. Bu durumda her bir bağımsız adımın gitmek için seçebileceği toplamda 8 adet yön ortaya çıkmıştır. Bunlar sağ, sol, yukarı ve aşağı olan dik yönlü adımlara ek olarak, sağ yukarı, sol yukarı, sağ aşağı ve sol aşağı şeklinde sıralayabileceğimiz çapraz adımlardır.

5.4.1 Benzetim Adımları

Sekiz yönlü benzetimde ilerleme mantığı ve adımları dört yönlü benzetimle tamamen aynıdır. Fark, yukarıda anlatıldığı gibi komşu noktaların belirlenmesinde yatmaktadır. Bu da yine Bölüm 5.3.1'de söz edilen, geliştirilmiş algoritma ile programlama sırasında ortaya çıkan bir değişikliktir. Programın çalışma mantığı ve uyguladığı adımlar dört yönlü benzetim ile bire bir aynıdır.

Programlama: Çapraz yönlü ilerlemenin modellenmesi ve dolayısıyla programlanması, düz yönlüye göre bazı farklılıklar ve zorluklar getirmektedir. Bunlardan en önemlisi, daha önceki kısımda söz edilmiş olan, herhangi bir adım sırasında, bir sonraki adımda gidilebilecek noktaların belirlenmesi durumu ve bu noktaların her birinin ayrı ayrı olasılık hesaplarının yapılabilmesidir. Örneğin, dört yönlü modellemede, herhangi bir anda, bir sonraki adım için seçilebilecek olası noktalar aşağıdaki Şekil 5.15'te sol durumda olacağı gibi, ilerlemenin sekiz yönlü olması durumunda Şekil 5.15'deki sağdaki durum gibi olacaktır.

Aradaki farkın daha net anlaşılabilmesi için koşulların eşitliği sağlanmış, ilerlemenin o ana kadar dört yönlü şekilde olmuş, çapraz bir ilerleme göstermemiş olan bir yol seçilmiştir. Böylece bir sonraki adım için seçilebilecek noktalar arasındaki fark daha net açığa çıkacaktır.



Şekil 5.15: Dört ve Sekiz Yönlü İlerlemede Olası Noktalar

Yukarıdaki şekilde, bu durum için, dört yönlü ilerlemede bir sonraki adım için seçilebilecek toplam 34 nokta bulunduğu görülmektedir. Modellemenin mantığı gereği bu noktaların seçilme olasılıkları aynı değildir yine de sonuç olarak 34 olasılık bulunmaktadır. Modele çapraz ilerleme durumu da eklenildiğinde, Şekil 5.15'in sağdaki durumu üzerindeki ufak noktalar ile görülebileceği üzere, ek olarak, 14 nokta yani 14 olasılık daha eklenmiştir. Dolayısıyla toplamda seçilebilecek nokta sayısı 34'ten 48'e yükselmiştir bir başka deyişle, seçilebilecek nokta sayısı yaklaşık olarak %40 oranında artmıştır. Durum böyle olunca, noktaların seçilme olasılıkları toplamı 1 olacağından dolayı, eski yani dört yönlüye ilişkin noktaların seçilme olasılıkları da eklenen çapraz noktalar nedeniyle değişmiş olacaktır ve yeniden bu duruma göre hesaplanması gerekecektir.

Çapraz ilerlemenin modellenmesinde en çok zorluk yaratan nokta da işte hemen bu hesap öncesindeki işlem, yani çapraz ilerlemede ortaya çıkacak olan noktaların belirlenmesi işlemidir. Önceki bölümde, dört yönlü ilerleme modeli için, bir sonraki adım için gidilebilecek olası noktaların belirlenmesini sağlayan algoritmadan söz edilmişti. Dört yönlü ilerleme için tezde ortaya konan en kısa ve sorunsuz çalışan algoritma şu şekildedir:

1) Her Satır Yatay Olarak Taranmalı: Program her satırı soldan sağa doğru taramaktadır. Tarama sırasında 0 değeri ile (yıldırım) karşılaştığı anda 0 değerinin sağındaki ve solundaki potansiyel değerlerini alacaktır. Eğer 0

değerinin iki sağında başka bir 0 daha var ise, bu sefer sadece ilk karşılaştığı 0 değerinin solundaki değeri alacaktır.

2) Her sıfır değeri ile karşılaşıldığında sağ alt ve sol alt çaprazı sorgulanacaktır. Eğer ki hem sol alt hem de sağ alt değerleri 0'dan farklı ise, 0'ın hemen altındaki değer hesaba alınacaktır.

3) Her sıfır ile karşılaşıldığında bir alttaki satır sorgulanacaktır. Eğer ki 0'ın alt satırında sınır değeri mevcut ise, son satıra gelindiği anlaşılacak ve benzetim sona erecektir.

Çapraz ilerlemenin de katıldığı, sekiz yönlü ilerleme için noktaları belirleyecek algoritma ise dört yönlüye göre çok daha karmaşık, dolayısıyla çok daha fazla işlem gerektiren, sonuç olarak da modelleme süresini artıran özelliklere sahip olarak karşımıza çıkmıştır. Seçilebilecek noktaların belirlenmesi için, matris üzerindeki her nokta için toplamda 6 tane durum algoritması tek tek uygulanmakta ve hangisi veya hangilerine sahip olduğuna bakılmaktadır. Sekiz yönlü ilerleme için tezde geliştirilen en kısa ve sorunsuz çalışan algoritma ise aşağıdaki şekildedir:

1 Her Satır Yatay Olarak Taranmalı: Program her satırı soldan sağa doğru taramaktadır. Tarama sırasında 0 değeri ile (yıldırım) karşılaştığı anda 0 değerinin sağındaki ve solundaki potansiyel değerlerini alacaktır. Eğer 0 değerinin iki sağında başka bir 0 daha var ise, bu sefer sadece ilk karşılaştığı 0 değerinin solundaki değeri alacaktır.

2) Sağ alt çaprazdaki noktanın seçilebilme olasılığı durumu: Her nokta kendinin altına ve altının iki sağına bakacaktır. Her ikisi de sıfırdan farklı ise sağ alt çaprazı alınacaktır.

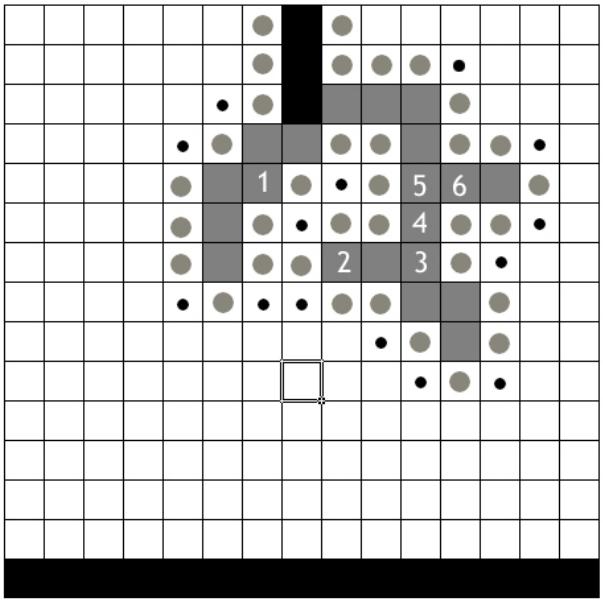
3) Sağ üst çaprazdaki noktanın noktanın seçilebilme olasılığı durumu: Her nokta kendinin üstüne, kendisinin iki üstüne ve üstünün iki sağına bakacaktır. Her üçü de sıfırdan farklı ise sağ üst çaprazı alınacaktır.

4) Sol alt çaprazdaki noktanın noktanın seçilebilme olasılığı durumu: Her nokta kendinin iki soluna, kendisinin altına ve altının iki soluna bakacaktır. Her üçü de sıfırdan farklı ise sol alt çaprazı alınacaktır.

5) Sol üst çaprazdaki noktanın seçilebilme olasılığı durumu: Her nokta kendinin iki soluna, kendisinin üstüne, kendisinin iki üstüne ve kendisinin üstünün iki soluna bakacaktır. Her üçü de sıfırdan farklı ise sol üst çaprazı alınacaktır.

6) En aşağıdaki noktanın sivri uç olması ihtimali durumu: Noktanın tam altındaki kareyi alabilmesi için sıfırın sağı ve solu sıfırdan farklı ise ve sağ altında veya sol altında sıfır yok ise kendisinin altını da alacaktır.

Görüldüğü üzere, sekiz yönlü ilerlemede, bir sonraki adımda seçilebilecek noktaların belirlenmesi olayı, dört yönlüye göre çok daha karmaşık ve fazla işlem gerektiren bir sorgulama algoritmasına sahiptir. Bu algoritmanın bu kadar karmaşık olmasındaki en büyük neden, aday noktaların, çevresinde seçilmiş noktalar ile birden fazla kere seçilme durumunun çok daha fazla ve kolay olmasından kaynaklanmaktadır. Yukarıda daha önce verdiğimiz şekilde örnek vermek gerekirse;



Şekil 5.16 : Aday Noktaların Belirlenmesi

Yukarıdaki şekilde bakıldığı zaman, 1 ve 2 çapraz kutuları arasında kalan, aday nokta, hem 1 kutusunun sağ alt çapraz bölgesine, hem de 2 kutusunun sol üst çaprazı bölgesine girmektedir. Fakat bu noktanın sadece bir kere seçilmiş olması gerekmektedir. Aksi takdirde bu noktadaki gerilimin değerinin tüm aday noktalarının gerilimlerine göre oranlandığı, noktaların olasılık hesabında bu nokta iki kere aynı değerlere hesaba girecek ve alması gerektiği olasılık değerinin iki katını alacaktır ve hataya neden olacaktır.

Daha karışık bir örnek vermek gerekirse, 4 kutusunun sağındaki nokta, hem 4 kutusunun sağ bölgesine, hem 5 kutusunun sağ alt bölgesine, hem 6 kutusunun alt bölgesine, hem de 3 kutusunun sağ üst bölgesine tekabül etmektedir. Yani toplamda 4 aynı noktanın seçim alanındadır ve herhangi bir ölçüt yani algoritma belirtilmez ise bu nokta tam 4 kere hesaba katılarak, alması gereken olasılık değerinin 4 katını alarak hataya sebep olacak, belki hiç seçilemeyeceği bir olasılık değerine sahip olacakken seçilebilir duruma gelecektir.

Bu noktada yukarıda söz edildiği üzere, sekiz yönlü ilerlemede en zor olan kısım, bu noktaların tam, eksiksiz, hatasız en önemlisi de birer kere hesaba katılmasını sağlamaktır. Bu nedenle tezde üretilen algoritma da oldukça karmaşık olmak zorunda kalmıştır.

Sekiz yönlü ilerleme için yazılan program kodunda en önemli değişiklik bu noktaların tespit edilmesini sağlayacak olan algoritmaları bu yöntem için uygulamak olmuştur. Onun dışında potansiyel dağılımı yine aynı şekilde hesaplanmakta ve iterasyonlar da aynı yöntemle tekrarlanmaktadır. Şimdi, dört yönlü ilerleme modelinde kullandığımız örneği (bkz. Şekil 5.17), sekiz yönlü ilerleme modeli için tekrarlayalım ve sorguların nasıl işlediğini adım adım gösterelim:

		1	0	13			
	A	2	0	12	G		
	3	0	0	0	11	F	
	4	0	6	0	0	10	
	B	5	C	7	0	9	
				D	8	E	

Şekil 5.17: Sekiz Yönlü İlerleme Modeli Örneği

- İlk Satır Taraması: İlk satırdaki 0'a gelindiği zaman, ilk sorgu ile solunda ve sağındaki 1 ve 13 noktaları alınacaktır. Alt çapraz taramalarını gerçekleştiren durumlarda hemen altındaki noktanın 0 olmaması koşulu ortak olarak aranmaktadır. Bu yüzden alt çapraz sorguları gerçekleşmeyecektir. İlk satır olduğu için program kodunda yukarı çapraz sorguları da eklenmediğinden dolayı ilk satır sorgusu sona erecektir. Sivri uç sorgusu ise koşulları sağlayacak ve altındaki noktayı alacaktır fakat burası 0 olduğundan sonucu etkilemeyecektir.

- 2. Satır Taraması: İkinci satırdaki 0'a gelindiğinde ilk sorgu ile 2 ve 12 noktaları alınacaktır. Sağ alt ve sol alt sorgusunda altındaki nokta 0 olduğu için bir nokta alınmayacaktır. Benzer şekilde sağ ve sol üst sorgularında da üzerinde 0 olduğu için herhangi bir nokta alınmayacaktır. Sivri uç sorgusu ise alt çaprazlarından birinde 0 olması koşulu yeterli gelecek ve aşağı noktayı almayacaktır.

- 3. Satır Taraması: Burada ilk sorgu ile hemen 3 ve 11 noktaları alınacaktır.

Sağ alt çapraz sorgusunda, soldan ilk 0'ın sağ alt çaprazı, altındaki 0 nedeniyle alınmayacaktır. Soldan ikinci 0'ın sağ alt çaprazı ise altının iki sağındaki 0 nedeniyle alınmayacaktır. Soldan üçüncü 0'ın sağ alt çaprazı ise hemen altındaki 0 nedeniyle alınmayacaktır.

Sağ üst çapraz sorgusunda, soldan ilk 0'ın sağ üst çaprazı sorgu koşullarını sağladığı için alınacaktır fakat 0 olduğu için etki etmeyecektir. Soldan ikinci 0'ın sağ üst çaprazı ise hemen yukarısındaki 0 nedeniyle alınmayacaktır. Soldan 3. sıfırın sağ üst çaprazı ise sorgu ölçütlerinin tümünü sağladığı için yani hem bir yukarısında, hem iki yukarısında hem de bir yukarısının iki sağında 0 olmadığı için G noktası alınacaktır.

Sol alt çapraz sorgusunda, soldan ilk 0'ın ve soldan üçüncü 0'ın hemen altında,0 değerleri bulunduğu için bu iki noktanın sol alt çaprazı alınmayacaktır. Soldan ikinci 0'ın ise sorgu koşulları hemen altının, iki solunun ve bir altının iki solunun, sırasıyla 6, 3 ve 4 noktalarının 0'dan farklı olmasını öngörmektedir. Koşullar sağlandığı için sol alt çapraz noktası alınacak (4–6 noktaları arasında kalan 0) fakat bu noktanın değeri 0 olduğu için sonuca etki etmeyecektir.

Sol üst çapraz sorgusunda, soldan ilk 0 değeri sorgu koşullarını sağladığı için, bu sıfırın sol üst çaprazı olan A noktasının alınmasını sağlayacaktır. Soldan ikinci 0 noktası, hemen üzerinde 0 olduğundan, soldan üçüncü 0 noktası ise iki solunda 0 noktası olduğundan sorguları kıracak ve noktaların alınmasını engelleyecektir.

Sivri uç sorgusunda ise, tüm 0 noktalarının sağında veya solunda 0 noktası olduğu için sorguyu bozacak ve altındaki noktaları aldırmayacaktır.

- 4. Satır Taraması: İlk sorgu ile hemen 4, 6 ve 10 noktaları alınacaktır.

Sağ alt çapraz sorgusunda, soldan ilk 0'ın sağ alt çaprazı, sorguları sağlaması ile C noktasının alınmasını sağlayacaktır. Soldan ikinci 0'ın sağ alt çaprazı da benzer şekilde sorgu koşullarını sağlamaktadır ve sağ alt noktası alınacaktır fakat bu değer 0

olduğu için sonuca etki etmeyecektir. Soldan üçüncü 0'ın ise sağ altı hemen altındaki 0 değeri nedeniyle alınmayacaktır.

Sağ üst çapraz sorgusunda, soldan ilk 0'ın sağ üst çaprazı yukarısındaki veya yukarısının iki sağındaki 0'lardan herhangi birinin varlığı nedeniyle alınmayacaktır. Soldan ikinci 0'ın sağ üst çaprazı ise hemen yukarısındaki 0 nedeniyle alınmayacaktır. Soldan 3. sıfırın sağ üst çaprazı ise sorgu ölçütlerinin tümünü sağladığı için yani hem bir yukarısında, hem iki yukarısında hem de bir yukarısının iki sağında 0 olmadığı için F noktası alınacaktır.

Sol alt çapraz sorgusunda, soldan ilk 0 bütün sorgu koşullarını sağladığı için B noktasının alınmasını sağlayacaktır. Soldan ikinci 0'ın iki solundaki 0, soldan üçüncü sıfırın ise hemen altındaki 0 nedeniyle bu noktaların sol alt çaprazları alınmayacaktır.

Sol üst çapraz sorgusunda, soldan ilk ve soldan üçüncü 0'lar hemen yukarılarındaki 0'lar nedeniyle sorguyu bozacaklardır. Soldan ikinci 0 ise iki solundaki 0 nedeniyle sorguyu bozacak ve değer alınmasını engelleyecektir.

Sivri uç sorgusunda ise, soldan ilk sıfır sorgu koşullarının tümünü sağlamaktadır yani 4, 6, B ve C noktaları 0'dan farklı olduğundan dolayı 5 noktası alınacaktır. Soldan ikinci 0'ın sağındaki ve sağ altındaki 0 nedeniyle, soldan üçüncü 0'ın ise altındaki 0 nedeniyle sorgu bozulacak ve değer alınmayacaktır.

- 5. Satır Taraması: İlk sorgu ile hemen 7 ve 9 noktaları alınacaktır.

Sağ alt çapraz sorgusunda, tüm koşullar sağlandığı için E noktası alınacaktır.

Sağ üst çapraz sorgusunda ise 0'ın hemen üzerindeki 0 değeri nedeniyle sağ üst noktası alınmayacaktır.

Sol alt çapraz sorgusunda tüm sorgu ölçütleri sağlandığı için D noktası alınacaktır.

Sol üst çapraz sorgusu ise 0'ın hemen yukarısındaki 0 değeri nedeniyle bozulacaktır ve bu nokta alınmayacaktır.

Sivri uç sorgusu ise 7, 9, D ve E noktalarının hepsinin birden 0'dan farklı olması ile sorgu ölçütlerinin tümünü sağlayacak ve 8 noktasının alınmasını sağlayacaktır.

Görüldüğü üzere yazılan algoritma ile sekiz yönlü ilerleme modeline uygun olarak alınması ve hesaba kaltılması gereken tüm noktalar eksiksiz ve yalnızca birer kere

alınmıştır. Sorgu ölçütlü, dört yönlü ilerleme modeline göre biraz daha uzun ve zahmetli olmasına rağmen kesin ve eksiksiz sonuç vermektedir.

Sekiz yönlü ilerleme modelinin, dört yönlü ilerleme modeline göre ek olarak bir başka farkı da toplam olasılık değerinin bulunması sırasında ortaya çıkmaktadır. Dört yönlü ilerlemede algoritmalar daha basit olduğu için, olasılık matrisini tek bir işlemle bulabilirken, sekiz yönlüde bu matrisin tek bir defada bulunabilmesi mümkün olamamıştır. Burada 6 durumun kontrolünü yapan algoritmanın her bir durumu için ayrı bir matris ortaya çıkartılmış, ana olasılık değerlerinin bulunması için bu matrisler toplanarak sonuca gidilmiştir.

Aşağıda sekiz yönlü ilerleme modelinin kullanıldığı benzetime ilişkin bazı sonuçlar gösterilmiştir.

5.4.2 Yöntemin Özeti

Sekiz yönlü ilerleme modelinde de görüldüğü üzere, benzer mantık uygulanmaktadır. Yapılan işlemleri adım adım özetleyecek olursak;

Benzetimin çalışma mantığını adım adım özetleyecek olursak, aşağıdaki şekildedir:

1. Adım: MSFY ve Neumann Sınır Koşulları kullanılarak, iterasyon yardımı ile potansiyel dağılımı hesaplanır. Buradaki hesaplama yine her nokta için dört yön baz alınarak yapılmaktadır. İlerlemenin sekiz yönlü olması, potansiyel dağılımının dört yönlü olarak hesaplanmasına engel değildir. Önemli olan her noktadaki potansiyel değerinin doğru olarak hesaplanıp bu noktadaki değerinin bilinmesidir.
2. Adım: Yıldırımın o anki durumuna göre, bir sonraki adımda gidebileceği noktalar, sekiz yönlü ilerleme modeli için yazılan her nokta için 6 sorgu gerçekleştirilen geliştirilmiş algoritma sayesinde belirlenir.
3. Adım: Belirlenen bu noktaların, üzerindeki bulundukları potansiyel değerine göre seçilme olasılıkları hesaplanır.
4. Adım: Güç Yasası, noktaların olasılık değerleri arasındaki fark istenilen ölçüde değiştirilmesine olanak verir.
5. Adım: sekiz yönlü olarak ortaya çıkan komşu olası noktaların her biri için 0–1 arasında rasgele sayılar üretilir.

6. Adım: Daha önce 3.Adım'da her bir komşu nokta için hesaplanan olasılık değerleri, 7. Adım'daki rasgele sayılar ile çarpılır.

7. Adım: Komşu olası noktalardan en büyük değere sahip olan nokta, yıldırımın bir sonraki adım için ilerleyeceği nokta olarak seçilir ve yıldırım bir adım ilerlemiş olur.

a. Adım: Son durum için değişen alan hesabı tekrarlanır, sınıra gelinene kadar 1–6 arasındaki adımlar tekrarlanır.

5.4.3 Sonuç

Modelde görüldüğü üzere, işleyiş açısından dört yönlü ilerleme modeli ile hiçbir fark bulunmamaktadır. Farkın genel olarak ilerlenebilecek açı sayısında, dolayısıyla de komşu nokta sayısındaki artış olduğu söylenebilir bu fark da sadece programlamada bu noktaların belirlenebilir ve seçilebilir hale gelmesi ile ortaya çıkmaktadır. Görsel açıdan ise, daha zengin ilerleme açılarına sahip olduğu için, aynı çözünürlükteki yani aynı matris büyüklüklerinde görsel sonuçlar dört yönlü olana göre daha yumuşak hatlara ve daha gerçekçi bir yapıya sahiptir. Aynı boyutlardaki dört yönlü ilerleme modelinde ilerleme daha kırıklı bir yapıya sahiptir fakat belirli bir seviyeye gelindikten sonra bu kırıklıklar dört yönlüde de seçilemeyecek kadar ufak yapıya sahip olacaktır, nitekim sekiz yönlüde bu durum daha küçük boyutlarda aynı görsel sonucu verecektir.

Benzetim süresine en çok etki eden faktörün matris boyutu büyüdükçe, iterasyonların sayısının katlanarak arttığı göz önüne alındığına, süre bazlı uygulamalarda sekiz yönlü ilerleme modeli daha kısa sürede dört yönlü ilerleme modeli ile benzer görsel sonuçlar ortaya koyacaktır.

5.5 Dört ile Sekiz Yönlü Benzetim Arasındaki Farklılıklar ve Benzerlikler

Bu bölümde aynı parametre ve değerler kullanılarak yapılmış olan benzetimlerde, dört yönlü ile sekiz yönlü ilerleme gösteren yıldırım modelleri karşılaştırılmış ve şu sonuçlara varılmıştır:

5.5.1 Benzerlikler

1) Yöntemin işleyişi: İki yöntemin program kodunun da çalışma şekli farklılıklar gösterse de mantığı ve gerçekleştirdiği adımlar, döngüler ve iterasyonlar tamamen aynıdır.

2) Adım Sayısı: İlk bakıldığında çapraz ilerleme de gösterebilen sekiz yönlü ilerleme modelinde, sanki daha fazla adım, ya da daha fazla dallanma gerçekleşecekmiş gibi görünse de sonuçlara bakıldığında adım sayılarının yaklaşık olarak aynı olduğu görülmüştür.

3) Potansiyel Hesabı: İki yöntemde de her adımda potansiyel dağılımı aynı şekilde, Bölüm 5.2'de anlatıldığı üzere bir noktanın etrafındaki 4 nokta değerlendirilerek yapılmaktadır. Burada karıştırılmaması gereken nokta; dört yönlüde alan dağılımı etraftaki 4 nokta kullanılarak, sekiz yönlüde ise her nokta etrafındaki 8 nokta kullanılarak yapılmamaktadır. Alan dağılımı aynı şekilde hesaplanmakta fakat olası noktaların belirlenmesi ve ilerleme 8 yön göz önüne alınarak hesaplanmaktadır.

5.5.2 Farklılıklar

1) Süre: İki yöntem arasındaki en büyük fark süredir. Sekiz yönlü ilerlemede, her adımda seçilebilecek nokta sayısı daha fazla olduğu için ve bu noktaları belirleyen algoritma dört yönlüye göre çok daha karmaşık olduğu için, aynı büyüklükteki bölgelerde sekiz yönlü ilerleme benzetimi, dört yönlüye göre çok daha uzun sürmektedir. Bunun ana olarak iki nedeni vardır, ikisi de farklı değerlerin sorgulandığı, sorgu aşamalarında ortaya çıkar:

   a) İlk sorgu farklılığı, her adımdan sonra her satırda sıfırların varlığının sorgulandığı durumdur. Program daha önce söz edilen algoritma doğrultusunda her satırdaki 0'ların ve komşu noktalarla ilişkini inceler. Satır taramaları sırasında 0'a rastlanıldığı zaman gerçekleşen sorguların sayısı dört yönlüde sadece 3 iken, sekiz yönlü ilerlemede sorgu sayısı 6'ya çıkmıştır ve sorgu koşulları dört yönlüye göre daha karmaşık ve uzundur.

   b) İkinci sorgu farklılığı ise, dört yönlü ilerlemede, ilk olasılık hesabında (rasgele sayıların hesaba katılmadığı) kullanılacak olan, olası komşu noktaların potansiyellerinin belirlenmesi ve bu noktaların potansiyelleri

toplamı, tek bir matris üzerinde, 3 tane sorgunun aynı anda yapılması ile bir kerede gerçekleştirilebilmektedir. Daha önce söz edildiği üzere, her bir noktanın seçilme olasılığı, (5.15)'ten hatırlanacağı üzere, o noktanın potansiyelinin, tüm toplam komşu noktaların potansiyelleri toplamına oranıdır. Dolayısıyla bu potansiyel toplamının ve her noktanın ayrı ayrı potansiyellerinin bu toplama oranının bulunması gerekir.

Sekiz yönlü ilerlemede, dört yönlünün aksine, bu işlem tek bir kerede, tek bir matris üzerinde hesaplanamayacak kadar karmaşık bir yapıya sahiptir ve ayrı ayrı hesaplanmak durumundadır ve dört yönlüye göre 6 katı işlem ve süreye sahiptir. Dört yönlüde bir kerede ve tek bir matris üzerinde hesaplanabilen olasılık matrisi, "o", sekiz yönlüde 6 ayrı matriste ayrı ayrı hesaplanmakta ve sonra toplanmaktadır. Toplam olasılık değeri bu şekilde bulunmaktadır. Program kodunda "m" matrisi olarak adlandırılan bu matrislerin 4 ve sekiz yönlü ilerleme modellerinde nasıl hesaplandığı "EKLER" kısmında görülebilir.

Yöntemde süreye etki eden bir başka faktör ise iterasyon sayısı ve matris boyutudur. Bu değerler arttıkça, yöntemde süre de doğru orantılı olarak artar. Aynı sistem üzerinde değişik biçimlere ilişkin ortalama süreleri kıyaslayan Tablo 5.1'de aşağıda verilmiştir:

Tablo 5.1 : Mevcut Sistemde Benzetim Süreleri

Matris Boyutu	İterasyon Sayısı	Süre (dk)
15x15	20	1,5
15x15	100	2,5
25x25	20	4,5
25x25	100	11
64x64	20	110
64x64	100	340

2) Sistem Gereksinimi: Yukarıdaki maddede ayrıntılı olarak anlatıldığı üzere, sekiz yönlü ilerlemede yapılan işlem sayısı dört yönlüye göre çok daha

fazladır fakat sistem gereksinimi olarak bakıldığında dört yönlünün sorunsuz olarak çalışabildiği her sistemde sekiz yönlü de rahatlıkla çalışabilir. Sekiz yönlü benzetim daha uzun sorgular içerebilir fakat sorguların tipi dört yönlü ile aynıdır. Gerçekleştirilen işlemler ve iterasyonların da sayısı daha fazladır fakat bu çalışmayı engelleyecek ya da sistemi zorlayacak bir yapıya sahip değil sadece süreyi uzatacak etkiye sahiptir. Bölüm xxx'te söz edildiği üzere, sistem Matlab programını çalıştırabilecek yapıya sahip herhangi bir sistemde rahatlıkla çalıştırılabilir. Tez çalışmalarını yapıldığı Intel Pentium 4 2.4 GHz işlemciye ve 512 MB ram'e sahip Microsoft Windows XP Professional işletim sistemi kullanan bir bilgisayarda Matlab ve benzetim sorunsuz olarak çalışmıştır. Doğal olarak sistem özellikleri arttıkça benzetim süreleri de kısalmaktadır. Burada benzetim süresine bilgisayardaki ram'in değil (ram'in 512 MB ve üzeri değerleri için), işlemci hızının etki ettiği söylenebilir.

Tablo 5.2 : Farklı Sistemlerde Benzetim Süreleri

Matris Boyutu	Ram	Süre (dk)
Celeron 2 GHZ	256 MB	3,5
Celeron 2 GHZ	512 MB	3,4
Pentium IV 2.4 GHZ	512 MB	3,2
Pentium IV 2.4 GHZ	1024 MB	3,2
Centrino 1.83 GHZ	512 MB	2,5

Tablo 5.2'deki sonuçlar incelendiğinde fark edilmektedir ki ram ve işlemci değerleri sonucu doğrudan etkilemektedir fakat ram değeri 512 MB'ın üzerine çıktığında benzetim süresini değiştirmemektedir. İşlemci değeri için ise böyle bir durum yoktur, benzetim sırasında bilgisayarın yüklenme gücü incelendiğinde de tüm yükü işlemcinin sırtlandığı gözlemlenmiştir. Buradan şu sonucu çıkartabiliriz; Tablo 5.1'den hatırlayabileceğimiz üzere, 64 x 64 gibi yüksek çözünürlüklü yani yüksek matris boyutlarına sahip benzetimlerde ortaya çıkan saatler mertebesindeki benzetim sürelerini azaltmanın yolu daha yüksek hızda işlemciler kullanmaktan veya birden fazla işlemciyi paralel bağlamaktan geçmektedir.

3) Görsellik: Kabaca sekiz yönlü ilerleme ile dört yönlü incelendiği zaman, sekiz yönlüde ilerlemenin genellikle çapraz yönü olduğu dikkati çekmektedir. Bunun temel nedeni potansiyel dağılımıdır. Yıldırımın alt ucu etrafındaki alan dağılımı incelendiğinde, ucun alt kısımlarında alan dağılımının yukarı kesimlere göre genel olarak daha yüksek olduğu görülür. En yüksek değeri alt ucun hemen altındaki bölge değil, sağ ve sol uçlarında kalan, alt çaprazdaki bölgelerin aldığını görürüz. Bu alan dağılımını aşağıdaki Şekil 5.18'de gözlemlemek mümkündür.

24,7	24,0	22,0	18,6	13,7	7,4	**0,0**	1,6	3,7	6,4	8,7	10,4	10,9
25,4	24,7	22,7	19,3	14,5	8,0	**0,0**	1,3	3,5	6,5	9,1	10,9	11,5
27,3	26,7	24,8	21,6	16,8	10,0	**0,0**	**0,0**	2,6	6,9	10,4	12,6	13,3
30,5	30,0	28,3	25,3	21,1	15,4	8,0	**0,0**	**0,0**	8,0	13,0	15,7	16,6
34,9	34,4	32,9	30,4	26,9	22,3	16,6	9,7	**0,0**	12,1	17,9	20,7	21,6
40,2	39,8	38,6	36,6	33,8	30,4	26,4	**22,4**	**19,4**	**22,7**	25,8	27,6	28,3
46,3	46,0	45,0	43,5	41,4	38,9	36,3	33,9	32,6	33,4	34,8	35,8	36,2
53,0	52,8	52,1	50,9	49,4	47,7	45,9	44,4	43,6	43,7	44,2	44,7	44,9
60,3	60,1	59,5	58,7	57,6	56,4	55,2	54,2	53,6	53,5	53,6	53,8	53,9
67,8	67,7	67,3	66,7	66,0	65,2	64,4	63,7	63,2	63,0	63,0	63,1	63,1
75,7	75,6	75,3	74,9	74,4	73,9	73,4	72,9	72,6	72,4	72,3	72,3	72,3
83,7	83,6	83,5	83,2	82,9	82,6	82,3	82,0	81,8	81,7	81,6	81,6	81,6
91,8	91,8	91,7	91,6	91,5	91,3	91,1	91,0	90,9	90,8	90,8	90,8	90,8
100	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Şekil 5.18 : Alt ve Çapraz Uçlarda Potansiyel Dağılımı

Yukarıdaki şekilde görülebileceği üzere yıldırımın alt ucunun çapraz noktalarında alan değerleri daha yüksektir. Bu da çapraz gitme şansı olan sekiz yönlü ilerlemede yıldırımın genel olarak çapraz ilerlemesini sağlamaktadır. Düz ilerleme durumları, rasgele sayıların devreye girmesi ve bu değerleri değiştirmesi ile mümkün olmaktadır.

Atmosfer ve yıldırım sırasındaki potansiyel dağılımı düşünüldüğünde gerçekte de bu şekildedir ve yıldırım çapraz ilerlemeye daha meyillidir. Dik ve keskin hareketler yerine yıldırım, zig-zag'lar çizerek ilerlemektedir. Dolayısıyla buradan kolayca söyleyebiliriz ki; sekiz yönlü ilerleme, dört yönlüye göre daha gerçekçi bir benzetim sunmaktadır.

Burada dikkate alınması gereken çok önemli bir husus vardır: Çözünürlük. Yıldırımın görsel olarak güzel ve gerçekçi görünmesini çözünürlük doğrudan etkiyecektir. Burada çözünürlük ile kastedilen şey matrisin büyüklüğüdür. Matrisin boyutu ne kadar büyük olursa atılacak adım sayısı o kadar fazla olacaktır. Aynı boydaki matrislerde sekiz yönlü benzetimler daha güzel ve gerçekçi sonuçlar verecektir fakat bu dört yönlünün daha kötü bir yöntem olduğu anlamına gelmez. Sadece hemen hemen aynı görselliğe sahip olan sonuçların çıkması için, dört yönlü benzetimde sekiz yönlüye göre daha büyük boyutların seçilmesi gerekir. Bunun nedenini kısaca şu şekilde özetleyebiliriz: sekiz yönlü benzetimde bir kerede sağ alt çapraza gidebilecekken, dört yönlü benzetimde bu aşağı ve sağ olmak üzere iki adımda gerçekleşecektir. Buradan, dört yönlü benzetimde seçilecek matrisin sekiz yönlüye göre yaklaşık olarak iki katı daha büyük olması durumunda, görsel olarak yakın sonuçların çıkacağı sonucuna varabiliriz.

5.6 Yöntemlerde Sonucu Etkileyen Faktörler

Dört yönlü ve sekiz yönlü ilerleme modellerinin her ikisinde de, gerek hesaplama yönünden gerek görsel yönden etki eden bazı parametreler bulunmaktadır. Bunlar en genel olarak aslında benzetim sırasında kullanıcı tarafından girilmesi istenen değişkenlerdir. Bunları tanımlayacak olursak, sonuca nasıl etki edeceği de açıkça görülmüş olacaktır:

1. Matris Boyutu: Benzetimin gerçekleştirileceği matrisin boyutu sonucu direk olarak etkilemektedir. Bilindiği üzere matris kare olmak zorunda değildir. Genel olarak matrisin boyutu büyüdükçe görüntü hem görsel olarak daha gerçekçi ve daha yumuşak hatlara sahip olacaktır hem de teorik olarak hesaplama sırasında daha gerçekçi bir modellemeye yaklaşılacaktır. Bunların nedenlerini açıklayacak olursak:

   Matris boyutu arttıkça noktalar arasındaki potansiyel fark değerleri birbirine daha çok yaklaşır ve daha hassas bir dağılım ortaya çıkar. Potansiyel değerler arasındaki farkın azalması demek de noktaların olasılık değerlerinin birbirine yaklaşması demektir. Olasılık değerlerinin birbirine yaklaşması da her ne kadar araya rasgele sayı üretme algoritması girecek olsa da, daha fazla

dallanma ve daha fazla adım anlamına gelecektir. Daha iyi anlaşılması açısından bunu görsel bir örnekle destekleyecek olursak;

22	19	12	0	12	19	22
25	22	14	0	14	22	25
33	30	21	0	21	30	33
46	45	40	35	40	45	46
63	62	60	59	60	62	63
81	81	80	80	80	81	81
100	100	100	100	100	100	100

12	8	0	8	12
15	11	0	11	15
26	20	0	20	26
47	45	40	45	47
73	72	71	72	73
100	100	100	100	100

Şekil 5.19: 7 x 7 ve 5 x 5 Boyutlarda Potansiyel Dağılım

Yukarıdaki şekilde (Şekil 5.19) yıldırım biçimi için, aynı sınır değerlerini kullanan iki değişik boyuttaki matris düzlemindeki potansiyel dağılımı verilmiştir. İlk matriste herhangi iki nokta arasındaki potansiyel fark, ikinci matrise göre daha azdır, çünkü aynı sınır değerleri arasında bu değerler paylaştırılmaktadır. Başka bir deyişle, ilk matristeki potansiyel dağılımı daha hassastır, değerler gerçeğe daha yakındır, hata payı daha azdır. İkinci matriste ise hesaplama daha "kaba" biçimde yapılmıştır.

Örneğin, sekiz yönlü ilerleme modelini göz önüne alalım. Bu biçime göre yıldırımın bir sonraki adımda gidebileceği noktaların, rasgele sayı algoritmasını hesaplamaya katmadan, sadece potansiyel değerlerine göre yapılan olasılık değerlerini bulacağımızda aşağıdaki değerler hesaplanacaktır:

		0,06	0	0,06		
		0,07	0	0,07		
		0,10	0	0,10		
		0,19	0,17	0,19		
100	100	100	100	100	100	100

	0,04	0	0,04	
	0,05	0	0,05	
	0,10	0	0,10	
	0,21	0,19	0,21	
100	100	100	100	100

Şekil 5.20: 7 x 7 ve 5 x 5 Boyutlarda Komşu Nokta Olasılıkları

Yukarıdaki şekilde (Şekil 5.20) ilk matristeki olasılık değerlerinin, ikinci matrise göre birbirine daha yakın değerler oduğunu görmekteyiz. Olasılık değerleri arasındaki fark, daha küçük boyutlu olan ikinci matriste daha artmış durumdadır. Dolayısıyla buradan kolayca söyleyebiliriz ki, matrisin boyutu küçüldükçe, aşağı taraftaki noktaların seçilmesi daha yüksek olasılık taşır hale gelmektedir yani küçük boyutlardaki matrisler aşağıdaki noktaları seçmeye daha meyillidir. Bunun sonucu olarak da daha az dallanma göstermeye yatkınlardır.

Benzer şekilde, matrisin boyutu büyüdükçe, olasılık değerleri birbirine daha fazla yaklaştığından, yukarıdaki noktaların seçilme şansı artmaktadır ve daha seçilebilir hale gelmektedirler. Durum böyle olunca yıldırım daha fazla dallanma ve kırılma gösterebilmekte ve daha gerçekçi sonuçlar elde edilebilmektedir.

Genel olarak bakıldığında ise, gerçeğe paralel olarak, noktaların aşağıya inildikçe olasılık değerlerinin arttığını görmekteyiz. Zaten modelde de bu şekilde olması istenmektedir; genel olarak ilerlemenin aşağı yönlü olması, fakat nadiren de yukarıdaki noktaların seçilerek dallanmanın gerçekleşmesi istenmektedir.

2. Benzetimde dışarıdan alınan değişkenlerden birisi de “kuvvet katsayısı”dır. Daha önce Bölüm 5.3.1’de ayrıntılı olarak söz edildiği üzere, η değeri olasılık değerleri arasındaki dengeye müdahale edebilmek, olasılık değerleri arasındaki farkları azaltıp artırmak için kullanılmakta idi. Bu sayede, bir önceki maddedeki matris boyutunu değiştirdiğimizde olasılık değerleri arasındaki fark değişikliğe uğramaktaydı. Aynı şekilde, bu katsayı ile olasılık değerleri arasındaki fark da etkilenmekte, dolayısıyla ortaya çıkan sonuca aynı şekilde etki etmektedir.

   Şekil 5.12’den hatırlanabileceği üzere, görülmektedir ki kuvvet katsayısı, η değeri arttıkça, olasılık değerleri arasındaki fark açılmakta, küçük olan olasılık değerleri daha küçülmekte, büyük olan olasılık değerleri de daha büyümektedir. Buradan yola çıkarak rahatlıkla söyleyebiliriz ki, daha büyük η değerleri için, daha az dallanma ve daha az adım, nispeten küçük η değerleri için ise daha fazla dallanma, daha fazla adım ve gerçeğe daha yakın görsellik elde edilmektedir.

Benzetimlerde yapılan denemelerde η değeri için en uygun aralığın [1–1,5] olduğu sonucuna varılmıştır. Daha büyük η değerlerinde yıldırım sapmalardan ve dallanmalardan oldukça uzaklaşmıştır.

3. İterasyon Sayısı: Merkezi Sonlu Farklar Yöntemi ile hesaplanan potansiyel dağılımı (Bkz. Bölüm 5.2) yönteminde bir noktanın hesabı, çevresindeki noktalardan etkilendiğinden, iterasyon yani yineleme işlemi gerektirmektedir. İterasyon sayısının az olduğu durumlarda, potansiyel dağılımı sağlıklı olarak hesaplanamayabilmekte, alması gereken değerlerden daha az değerler alabilmektedir. Alan dağılımı hesabının doğru yapılamaması demek, alana bağlı hesaplanan ilk olasılık dağılımının doğru biçimde yapılamadığı anlamına gelecektir ve bir sonraki adımda seçilecek noktalar sağlıklı bir biçimde belirlenememiş olacaktır. Bunun için iterasyon sayısının mevcut matris boyutu için yeterli olduğundan emin olunmalıdır.

   Benzetimler sırasında 15 x 15 matris için iterasyon değerinin 100 olması durumunda, olasılık hesaplarının gerçeğe oldukça yakın olarak, %0,5 mertebelerinde yanılmalarla gerçekleştiği görülmüştür. İterasyon sayısı, benzetim süresini doğrudan ve en çok etkileyen faktör olduğundan, doğru hesaplamaya yetecek kadar yüksek, gereksiz zaman kaybı yaratmayacak kadar da düşük olacak şekilde, bir optimum değerde belirlenmelidir.

   Unutulmaması gereken bir nokta vardır ki, 5 x 5 matris için 10 iterasyon yeterli gelecek iken, 10 x 10 matris için örneğin 25 iterasyon yeterli gelecektir. Hangi boyuttaki matris için hangi sayıda iterasyon belirlemenin yolu, bir başka platformda, örneğin Excel'de alan dağılımı hesabı yapılarak binlerce iterasyon dğerinden sonra aldığı değerle veya başka bir alan dağılımı hesabı yöntemi kullanılarak, Matlab'te alınan değerler karşılaştırılarak sonuca gidilmelidir. Tez çalışmasında bu karşılaştırma için Excel programı kullanılmıştır çünkü Excel'de iterasyon işlemleri Matlab platformuna göre kat ve kat hızlı işlemektedir.

Yukarıda anlatılanları özetleyecek olursak, dört ve sekiz yönlü ilerleme modellerinde daha çok dallanmanın görüldüğü, daha çok adımdan oluşan, gerçeğe daha yakın sonuçların en optimum sürede hesaplandığı sonuçlar elde edebilmek için;

- Matris boyutu mümkün olduğu kadar büyük tutulmalıdır.

- Kuvvet katsayısı, η, mümkün olduğu kadar 1'e yakın seçilmelidir.
- İterasyon sayısı doğru hesaplamayı mümkün kılacak en düşük sayıda seçilmelidir.

6. BENZETİM

6.1 Matlab Programı Hakkında Genel Bilgi

Matlab, yüksek performanslı bir programlama dilidir. Açılımı MATrix LABoratuary sözcüklerinden gelmektedir. Matlab, ilk olarak Fortran dili ile Cleve Moler tarafından yazılmıştır. Şu anki sürümü MathWorks firması taradından C dili kullanılarak yazılmıştır [22].

Matlab'in sahip olduğu fonksiyonlar ve özellikler ile matematikte akla gelebilecek her konudaki işlemler rahatlıkla yapılabilir. Matris işlemleri, türev, integral hesapları, diferansiyel denklemlerin çözümü gibi birçok matematiksel işlem Matlab tarafından basit programlamalarla veya hazır olan kodlarla gerçekleştirilebilir.

Matlab'in ayrıca diğer dilleriyle ve programlarla veri transferi yapması mümkündür. Örneğin Excel ortamından bir dosyanın içeriğini MatLab'a atmak ya da tersini gerçekleştirmek mümkündür. Benzer şekilde C++ ile geliştirilen bir program MatLab platformunda çalıştırılabilir. Yine aynı şekilde tersi de mümkündür [23].

Tez çalışmasında 64 x 64 boyutta olan benzetim denemelerinde, (Bkz. Bölüm 6.4) Matlab'ın veri transferi özelliğinden yararlanılmıştır çünkü bu boyuttaki bir matrisi bir bütün olarak sergileyebilmek kağıt veya bilgisayar ortamında mümkün değildir. Bu nedenle 64 x 64 boyutundaki benzetim sonucuna ilişkin matris, Excel ortamına aktarılmış, burada "0" olan noktalar, yani yıldırım yolu belirlenerek tez çalışmasına eklenmiştir. Matlab'te gerçekleştirilen benzetim sonucunda, yıldırım yolunu gösteren, sonuç matrisi, "x", (6.1) bağıntısında belirtilen kod ile Excel ortamına aktarılmıştır:

Dlmwrite('c:\64sonuc.xls', x, '\t') **(6.1)**

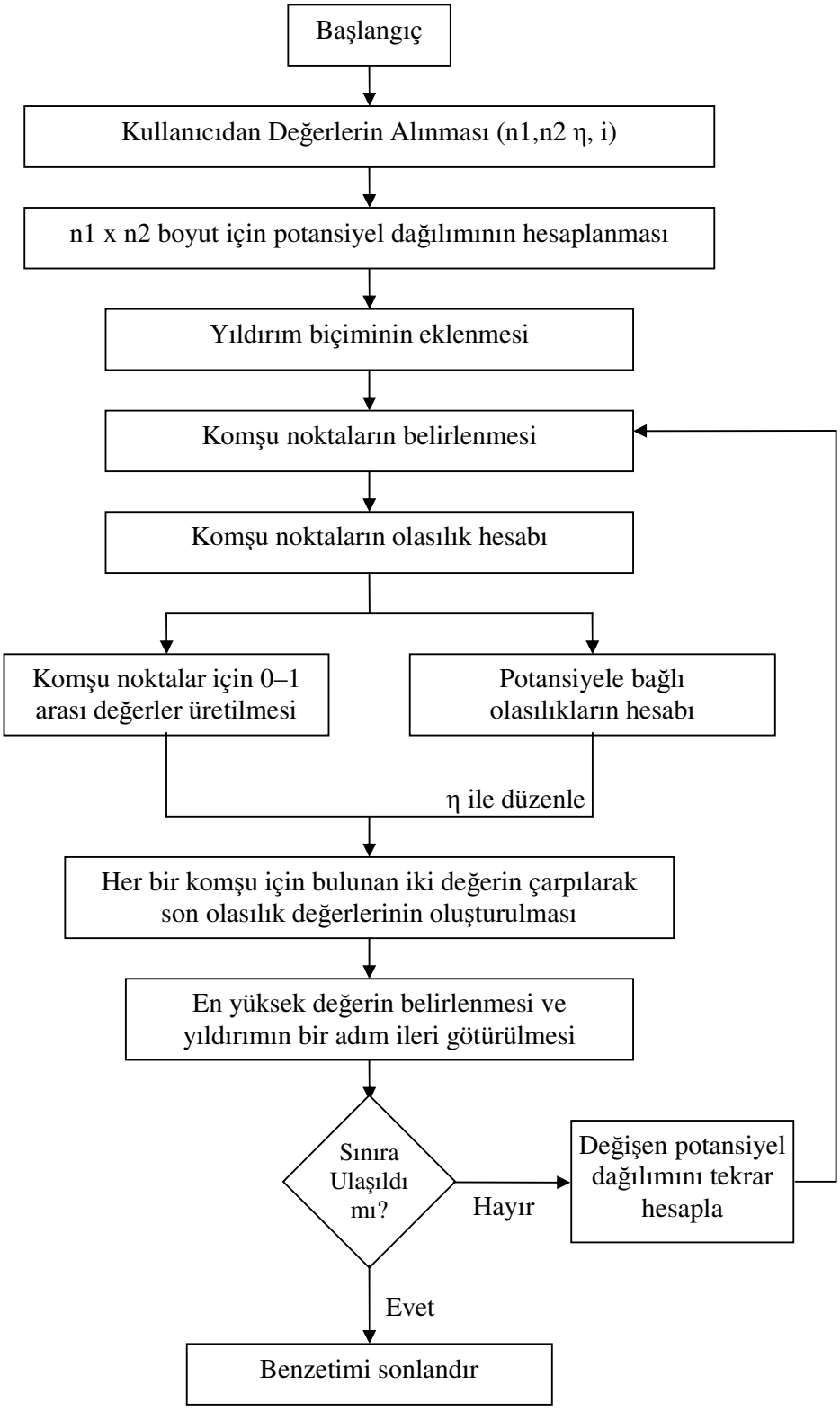
Yukarıdaki bağıntıya göre, o an çalışılmakta olan Matlab dosyasına ilişkin, tanımlanmış "x" matrisinin içeriği, (tez çalışmasında benzetimde bu matris yıldırım yolunu vermektedir) "c:\" klasöründe oluşturulacak olan "64sonuc.xls" dosyasına aktarılacaktır.

İşletim Sistemi Uyumluluğu: Matlab birçok işletim sisteminde hatasız olarak çalışmaktadır. Günümüzde Microsoft Windows 95/98/200 ve XP sürümleri ile Unix tabanlı bir çok işletim sisteminde çalışmaktadır. Herhangi bir işletim sisteminde yazılmış veri dosyaları diğer işletim sistemlerde de sorunsuz olarak çalıştırılabilmektedir [22].

6.2 Modelin Matlab Programı Akış Diyagramı ve Çalışması

Matlab'te yazılmış olan kod, Bölüm 5.3 ve 5.4'te tanımlanmış olan adımları birebir uygulamaktadır. Dört yönlü ve sekiz yönlü ilerleme modelleri için kodlar oldukça benzerdir fakat belirtilen farklılıklar nedeniyle iki ayrı kod biçiminde yazılmıştır. Program kodu EKLER kısmında verilmiştir.

Programın akış diyagramı her iki model için de aynıdır, sadece yapılan işlemler ve gerçekleştirilen sorgular işlem bazında farklılıklar göstermektedir. Akış diyagramı aşağıda verilmiştir:



Şekil 6.1 : Modelin Matlab Program Kodu Akış Diyagramı

6.3 Sistem Gereksinimleri

MS Windows tabanlı işletim sistemleri için minimum sistem gereksimleri:

- Windows 98, Windows Millenium Edition, Windows NT 4.0, Windows 2000, Windows XP Home, Windows XP Professional Edition.
- Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV, Xeon PIII, AMD Athlon veya AMD XP işlemci
- Minimum 128 MB Ram (min. 256 MB önerilen)
- 90 MB boş disk alanı yalnızca Matlab için, 250 MB Matlab ve yardım dosyaları için.
- 8-bit'lik grafik kartı. (256 renk gösterimi için minimum donanım gereksinimi)
- 16, 24, 32-bit Open GL destekli grafik kartı (opsiyonel)
- Microsoft Office 97, Office 2000, Office XP ve Office 2003; Matlab Notebook, Matlab Excel Builder, Excel Link, Database Toolbox, Matlab Web Server uygulamaları için gereklidir.

Unix / Linux işletim sistemi için minimum sistem gereksinimleri:

- 90 MB boş disk alanı yalnızca Matlab için, 215 MB Matlab ve yardım dosyaları için.
- Minimum 128 MB Ram (min. 256 MB önerilen)
- 64 MB takas alanı (önerilen)
- Linux için Matlab 6.0 veyua daha üst versiyonları sadece glibc (libc6) grafik kütüphanesi ile çalışabilir ve Libc5 desteklenmemektedir [22].

Önerilen Sistem:

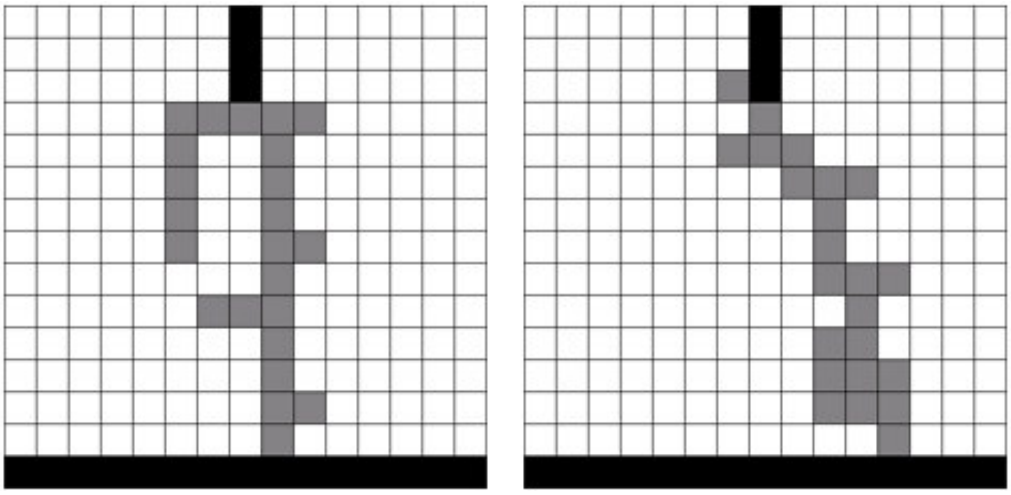
- Pentium IV veya AMD Athlon işlemci.
- 1024 MB ram.
- 20 GB sabit disk alanı
- 128 MB 128 bit ekran kartı

6.4 Örnek Benzetim Sonuçları

Aşağıdaki şekillerde, yıldırım biçimi için çalıştırılan programda, dört yönlü ve sekiz yönlü ilerleme modellerine ilişkin sonuçlar gösterilmiştir. Benzetimlerin tümünde matrix boyu 15 x 15, iterasyon sayısı 100 ve kuvvet katsayısı, $\eta = 1,3$ alınmıştır. Bu bölümdeki tüm şekillerde, alınan Matlab sonuçları, daha iyi anlaşılması ve daha görsel olması amacıyla Excel ortamında kare ızgara bölge oluşturularak belirtilmiştir. Siyah ile gösterilen alanlar yıldırım biçimini, gri renkle gösterilen renkler ise, Matlab benzetimindeki matrislerde 0 olan noktaları, yani yıldırımın ilerlediği noktaları temsil etmektedir. Bu bölümdeki şekillerle ilgili tüm 15 x 15 boyuttaki Matlab sonuçları EKLER bölümüne konulmuştur. 64 x 64 boyuttaki benzetimlere ilişkin sonuçlar çok büyük olduğundan, değerler Excel ortamına aktarılarak (bkz. **Bölüm 6.1**), küçültülerek gösterilmiştir.

6.4.1 Dört Yönlü İlerleme Örnekleri

15 x 15 matris üzerinde, 4 yönlü ilerleme modeline ait rasgele bazı sonuçlar aşağıda gösterilmiştir:

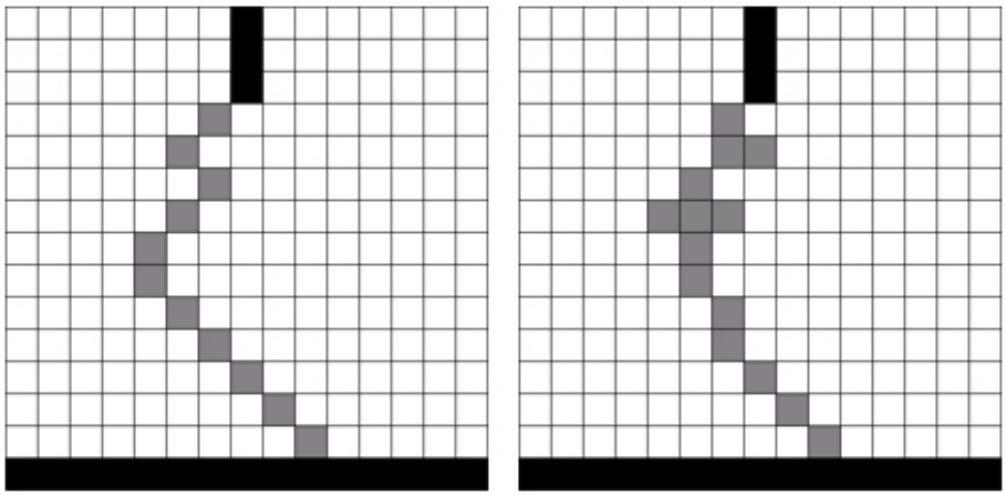


Şekil 6.2: Dört Yönlü İlerleme Modeli İçin 64 x6 4 Rasgele İki Örnek

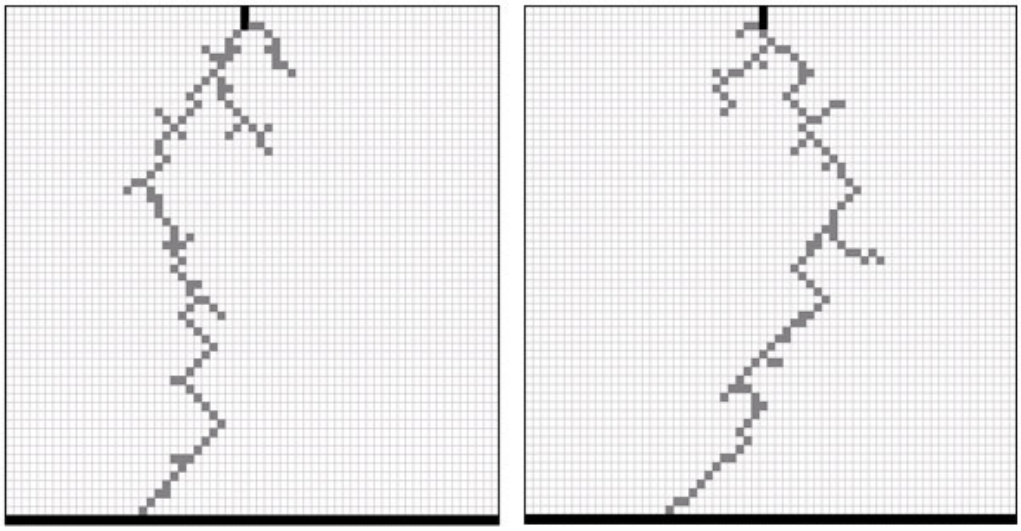
Şekil 6.3: Dört Yönlü İlerleme Modeli İçin 64 x 64 Rasgele İki Örnek

6.4.2 Sekiz Yönlü İlerleme Örnekleri

Aşağıda, 15 x 15 ve 64 x 64 matris üzerinde, dört yönlü ilerleme modeline ilişkin rasgele seçilmiş bazı sonuçlar gösterilmiştir:



Şekil 6.4: Sekiz Yönlü İlerleme Modeli İçin 15 x 15 Rasgele İki Örnek



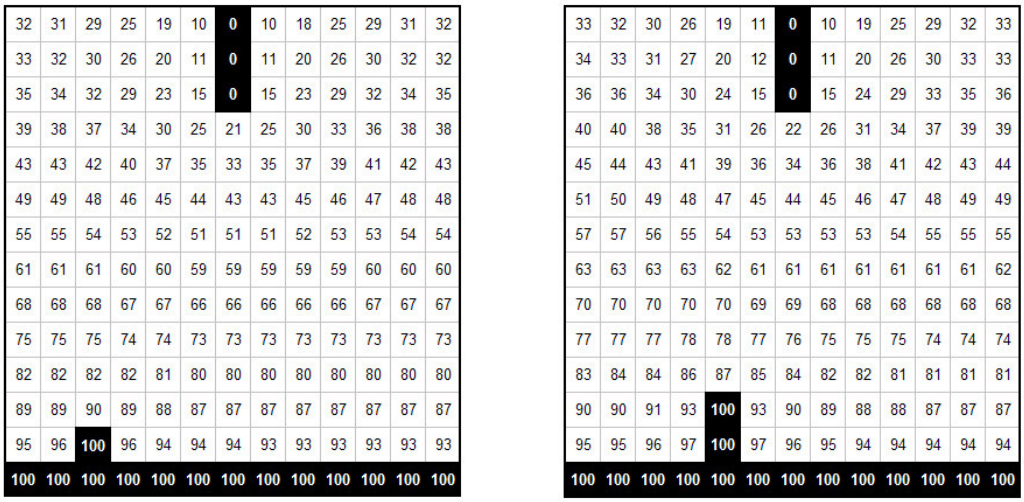
Şekil 6.5: Sekiz Yönlü İlerleme Modeli İçin 64 x 64 Rasgele İki Örnek

6.5 Değişik Uygulamalar

Önceki kısımda yıldırım biçimi için tanımlanmış ve tez kapsamında geliştirilen, defalarca çalıştırılmış olan Matlab koduna ilişkin sonuçlar gösterilmişti. Tez çalışmasının daha sonraki kısmında, bu benzetim, yıldırım biçiminden bir adım daha ileri götürülerek bazı değişik ve özel koşullar altındaki sonuçları görüntülemek için kullanılmıştır.

Bu özel koşulları genel olarak "Apartman Benzetimi" olarak adlandırmak mümkündür. Hatırlanacağı üzere, yıldırım biçimi, üst sınır olarak, bir potansiyel değere sahip yıldırım öncü ucu, alt sınır olarak da düz bir düzlem (toprak) kabul edilerek, yıldırımın toprağa ilerleyişi benzetilmeye çalışılmıştı. Bu kısımdaki uygulamada ise, yıldırımın düştüğü bu alanda değişik yerlerde ve değişiklik yüksekliklere sahip evlerin, apartmanların olduğu durumlar oluşturulmuş ve yıldırımın davranışındaki değişiklikler gözlemlenmeye çalışılmıştır.

Toprak üstüne bir katlı, iki katlı, üç ve dört katlı apartmanlar yıldırım modeline uydurularak, farklı konumlara yerleştirilmiş ve benzetim bu durumlar için çalıştırılmıştır. Burada herhangi bir yere konumlandırılan eve, sınır değeri olarak 100 V atanmıştır, çünkü ev, toprak ile aynı potansiyele sahiptir ve toprak değerini alacaktır. Toprağa neden 100 V, yıldırım öncü ucuna neden 0 V değerleri atandığı daha önce Bölüm 5'te ayrıntılı olarak anlatılmıştır. Aşağıdaki şekilde, bir katlı ve iki katlı evlerin yıldırım biçimi üzerinde nasıl gösterildiği ve tanımlandığı belirtilmiştir:



32	31	29	25	19	10	0	10	18	25	29	31	32
33	32	30	26	20	11	0	11	20	26	30	32	32
35	34	32	29	23	15	0	15	23	29	32	34	35
39	38	37	34	30	25	21	25	30	33	36	38	38
43	43	42	40	37	35	33	35	37	39	41	42	43
49	49	48	46	45	44	43	43	45	46	47	48	48
55	55	54	53	52	51	51	51	52	53	53	54	54
61	61	61	60	60	59	59	59	59	59	60	60	60
68	68	68	67	67	66	66	66	66	66	67	67	67
75	75	75	74	74	73	73	73	73	73	73	73	73
82	82	82	82	81	80	80	80	80	80	80	80	80
89	89	90	89	88	87	87	87	87	87	87	87	87
95	96	100	96	94	94	94	93	93	93	93	93	93
100	100	100	100	100	100	100	100	100	100	100	100	100

33	32	30	26	19	11	0	10	19	25	29	32	33
34	33	31	27	20	12	0	11	20	26	30	33	33
36	36	34	30	24	15	0	15	24	29	33	35	36
40	40	38	35	31	26	22	26	31	34	37	39	39
45	44	43	41	39	36	34	36	38	41	42	43	44
51	50	49	48	47	45	44	45	46	47	48	49	49
57	57	56	55	54	53	53	53	53	54	55	55	55
63	63	63	63	62	61	61	61	61	61	61	61	62
70	70	70	70	70	69	69	68	68	68	68	68	68
77	77	77	78	78	77	76	75	75	75	74	74	74
83	84	84	86	87	85	84	82	82	81	81	81	81
90	90	91	93	100	93	90	89	88	88	87	87	87
95	95	96	97	100	97	96	95	94	94	94	94	94
100	100	100	100	100	100	100	100	100	100	100	100	100

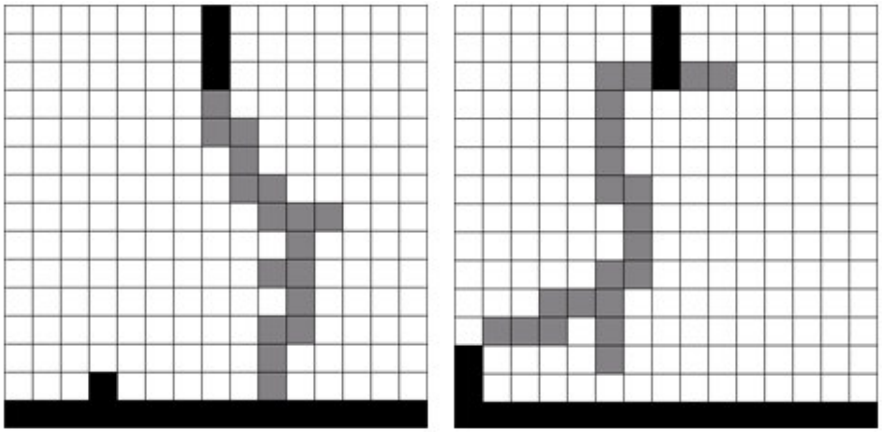
Şekil 6.6: Yıldırım Biçimi Üzerinde Apartman Modeli

Şekil 6.6'da sırasıyla tek ve iki katlı apartman modeli benzetimi yapılmıştır ve sınır değeri olarak da temas halinde olduğu için toprağın değeri atanmıştır. Görüldüğü üzere, düz sınır üzerinde çıkıntı oluşturan sembolik apartmanlar çevresinde alan dağılımı yoğunlaşmış ve burada nispeten daha büyük değerler almış durumdadır. Gerçekte de bu şekilde olması beklenir. Yine, fark edileceği üzere, iki katlı sembolik apartman modelinde, alan, soldaki şekildekine göre daha fazla etkilenmiş ve burada biraz daha fazla yoğunlaşmıştır. Benzer şekilde, üç ve dört katlı apartman modellerinde, alan, apartman etrafında daha da fazla yoğunlaşmaktadır.

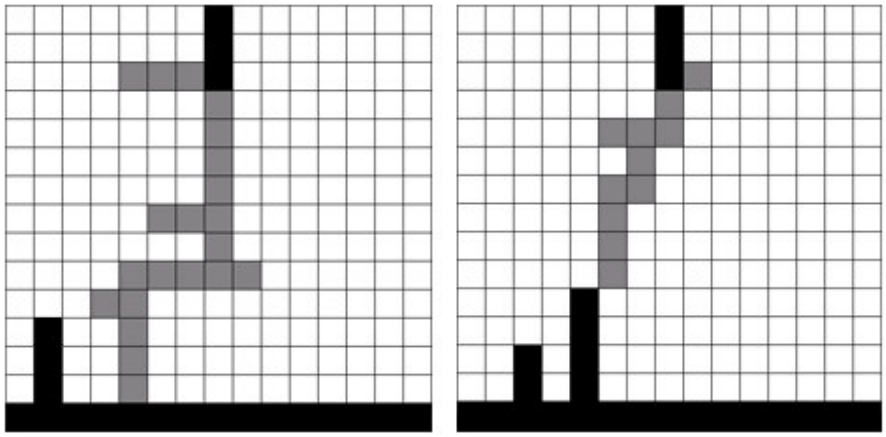
Bu uygulamadaki amaç, düzgün alan dağılımını bozan apartman modelinin, yıldırımın izlediği yol üzerindeki etkisini izleyebilmektir. Yıldırımın, toprak düzlemine dik olarak normali düşünüldüğünde, bu dik eksene ne kadar yaklaştığında, ne kadar yüksekliğe sahip olduğunda ve benzeri değişken durumlarda, yıldırımı üzerine çekme riskinin ne kadar olduğunu ve nasıl değiştiğini gözlemlemek için benzetim aynı durumlar için ve farklı durumlar için ayrı ayı defalarca çalıştırılmıştır.

6.5.1 Dört Yönlü İlerleme Modeli İçin Apartman Modeli

Aşağıdaki şekillerde, daha önceki benzetim ölçütleri ile aynı olacak şekilde, 15 x 15 matriste, kuvvet katsatısı, $\eta = 1{,}3$ değeri ve 100 iterasyon için sonuçlar gösterilmiştir. Şekiller, gelişigüzel olarak, değişik yükseklikteki ve konumlardaki apartman modellerine ilişkindir.



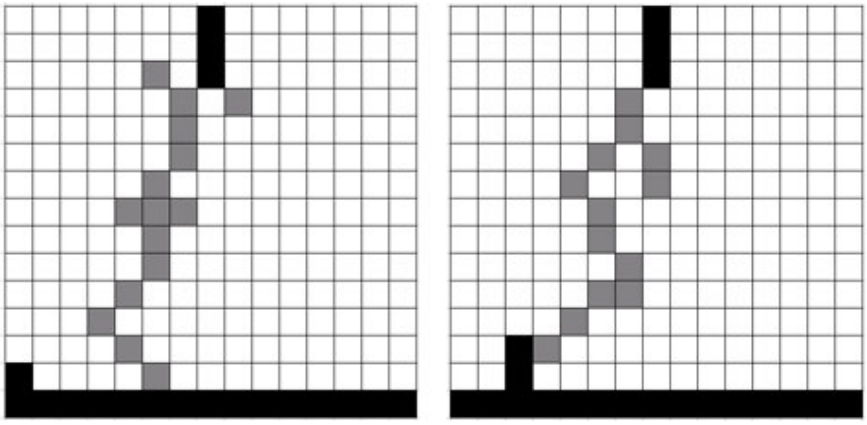
Şekil 6.7: Dört Yönlü 1 ve 2 Katlı Apartman Uygulaması



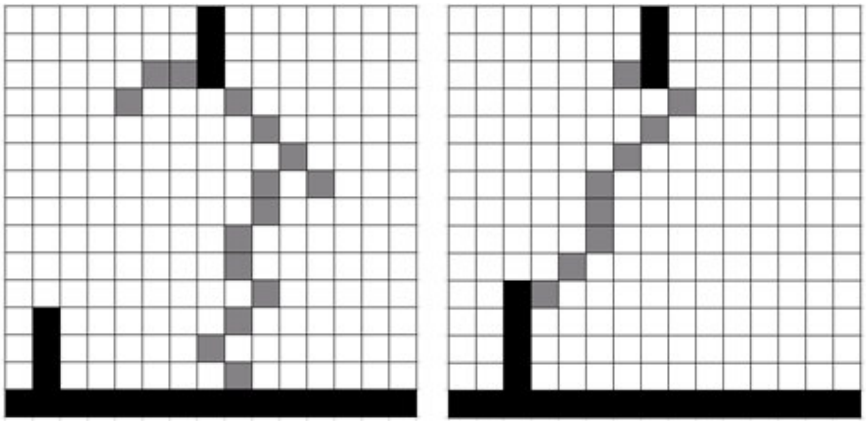
Şekil 6.8: Dört Yönlü 3 ve 2-4 Katlı Apartman Uygulaması

6.5.2 Sekiz Yönlü İlerleme Modeli İçin Apartman Modeli

Aynı ölçütler için, sekiz yönlü ilerleme modeline adapte edilmiş apartman modeline ilişkin bazı benzetim sonuçları aşağıda gösterilmiştir:



Şekil 6.9: Sekiz Yönlü 1 ve 2 Katlı Apartman Uygulaması



Şekil 6.10: Sekiz Yönlü 3 ve 4 Katlı Apartman Uygulaması

6.5.3 Sonuçlar

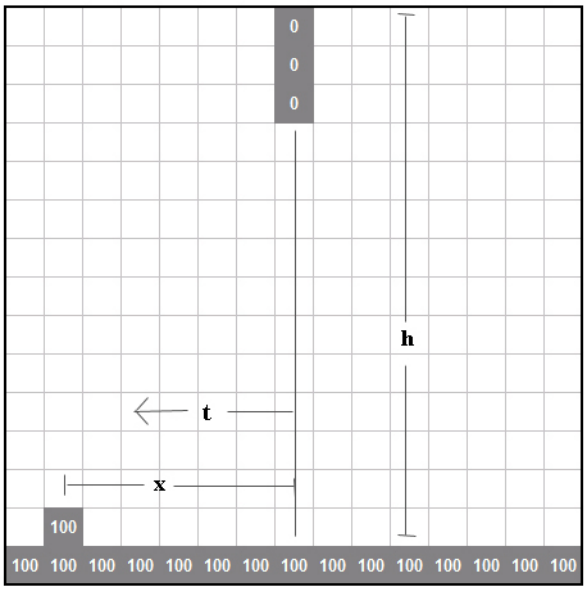
Yukarıda verilmiş olan apartman denemelerinde görüldüğü üzere, yıldırım, alan dağılımını bozan ve kendi üzerinde yoğunlaştıran çıkıntılara (burada apartman) doğru ilerlemeye meyil göstermektedir. Dört yönlü ve sekiz yönlü ilerleme modellerinde, eğer ki rasgele sayı üretme algoritması kullanılmasaydı, yıldırım, alanın en yüksek olarak toplandığı ve kendisine en yakın olduğu bu çıkıntılara doğrusal olarak ve direk yönde (en kısa yol) olarak gidecekti. Bu çıkıntıların olduğu yöne zıt yönde olan kıvrılmalar ve dallanmalar, tamamen, rasgele sayı algoritması tarafından sağlanmaktadır. Fark edileceği üzere bu algoritma yıldırıma dallanma ve zig-zag anlayışını getirebilmesinin yanı sıra, apartmanın yüksek ve yıldırıma eksenel olarak yakın olduğu durumlarda, üzerinde düşmesini engelleyebilecek kadar güçlü bir etkiye sahip değildir. Yıldırımın karakteri ve yapısı düşünüldüğünde, bu sonuç gerçekle oldukça paralel sonuçlar yansıtmaktadır.

Şekil 6.9'da, tabloda kullanılan terimleri açıklamada kullanılacak olan değişkenler belirtilmiştir. Burada;

x: Apartmanın, yıldırımın öncü ucunun normaline olan yatay uzaklığını

t: Yıldırımın düştüğü noktanın, yıldırımın öncü ucunun normaline olan yatay uzaklığını göstermektedir. Pozitif değerler yakınlaşma yönünü, negatif değerler ise uzaklaşma yönünü göstermektedir.

h: Yıldırım çıkış noktasının toprağa göre yüksekliğini göstermektedir.



Şekil 6.11: Apartman Modelinde Değişkenlerin Tanımlanması

- Kat Sayısı: Apartmanın sembolik olarak kaç katlı olduğunu tanımlar.
- Konumu: Yıldırmın oluştuğu nokta ile toprak arasındaki düşey uzaklığın, yani yıldırımın başlangıç noktasının, toprağa göre yüksekliğinin; apartmanın, yıldırımın toprağa göre normaline olan uzaklığına oranıdır (x/h).
- Çarpma Durumu: Yıldırımın apartmana düşme durumunun yüzde olarak değeridir.
- Çekme Oranı: Yatay düzlemde, yıldırımın düştüğü noktanın, yıldırımın çıkma noktasının normaline uzaklığının oranıdır (t/h). Yatay düzlemde yıldırımı kendisine ne kadar çektiğini gösterir.
- Adım Sayısı: Yıldırımın kaç adımda sınıra ulaştığını belirtmektedir.

Aşağıdaki Tablo 6.1 ve Tablo 6.2'de her bir durum için en az 5 kere çalıştırılmış olan benzetimlere ilişkin sonuçlar gösterilmiştir. Çarpma yüzdeleri, aynı kat ve konum için benzetimin en az 5 kere çalıştırılmasıyla elde edilmiştir, bir başka deyişle, her bir durum ve biçim için benzetim en az 5 kere çalıştırılmıştır. Burada kat sayısında 4–2 olarak belirtilen kısım, bir tane iki katlı, bir tane de dört katlı, iki tane yan yana apartman olması durumunu belirtmektedir ve apartmanların konumları, konum kısmında sırası ile belirtilmiştir.

Tablo 6.1: Dört Yönlü İlerleme Modeli İçin Apartman Denemeleri

Kat Sayısı	Konumu	Çarpma (%)	Ort. Çekme Oranı (%)	Ort. Adım Sayısı
1	0,5	0	1,44	17
1	0,428	0		
1	0,357	0		
1	0,285	0		
1	0,214	0		
2	0,5	20	3,70	17
2	0,428	0		
2	0,357	20		
2	0,285	0		
2	0,214	20		
3	0,5	0	6,75	16
3	0,428	0		
3	0,357	20		
3	0,285	40		
3	0,214	60		
4	0,5	0	7,98	16
4	0,428	0		
4	0,357	40		
4	0,285	40		
4	0,214	80		
2-4	0,357 & 0,214	60	7,1	15

Tablo 6.2: Sekiz Yönlü İlerleme Modeli İçin Apartman Denemeleri

Kat Sayısı	Konumu	Çarpma (%)	Ort. Çekme Oranı (%)	Ort. Adım Sayısı
1	0,5	20	5,68	13
1	0,428	0		
1	0,357	0		
1	0,285	20		
1	0,214	40		
2	0,5	20	8,96	11
2	0,428	0		
2	0,357	40		
2	0,285	40		
2	0,214	80		
3	0,5	20	14,24	11
3	0,428	40		
3	0,357	20		
3	0,285	60		
3	0,214	100		
4	0,5	20	16,54	11
4	0,428	20		
4	0,357	60		
4	0,285	100		
4	0,214	100		
2-4	0,357 & 0,214	60	15,7	10

Yukarıda, Tablo 6.1'de dört yönlü, Tablo 6.2'de ise sekiz yönlü ilerleme modeli kullanılarak çalıştırılan apartman durumuna ilişkin benzetim sonuçları gösterilmiştir. Tablolar incelendiğinde şu sonuçlar çıkartılabilmektedir:

- Aynı kat değerlerinde, konum ve çarpma yüzdesi karşılaştırıldığında, bazı değerler için gidişat bozulsa da, genel olarak, konum değeri azaldığında, yani, apartmanın yatay düzlemde yıldırımın normaline göre uzaklığı azaldığında, çarpma yüzdesinin arttığı görülmektedir. Bu beklenen bir durumdur, çünkü yatay olarak yıldırımın çıkış noktasına yaklaşıldığında, yıldırımı üzerine çekme olasılığı artmaktadır çünkü yıldırımdan yatay olarak uzaklaşıldıkça, yıldırım çevresindeki alan dağılımına etki de azalmaktadır.
- Kat değerleri ile çarpma yüzdesi karşılaştırıldığında ise, yine bazı değerler için akış bozulsa da, genel olarak, kat sayısı arttıkça çarpma yüzdesinin de arttığı görülmektedir. Bu da benzer şekilde beklenen bir durumdur, çünkü kat değerinin artması demek, **Bölüm 6.5**'te ayrıntılı olarak anlatıldığı üzere, yüksek alan değerlerinin burada daha çok toplanması ve yoğunluk oluşturması anlamına gelmektedir. Dolayısıyla yıldırım ilerleme modelinde,

alana bağlı olasılık değerlerinin artmasına neden olmaktadır. Bu şekilde de yıldırım alan değerinin daha yüksek noktalara ulaştığı yüksek katlı apartman durumlarında, apartmana doğru ilerlemeye meyillidir. Yıldırımın gerçekteki davranışı da düşünüldüğünde, yıldırımın sivri noktalara düşmeyi tercih ettiği bilinmektedir, dolayısıyla benzetim sonuçlarının gerçeğe uygun olduğu söylenebilir.

- Kat değerleri ile ortalama çekme miktarı karşılaştırıldığında ise, kat değerleri arttıkça genel olarak çekme miktarının da arttığı görülmektedir. Bu durum bir önceki maddenin içeriği ile bağlantılı bir durumdur ve bu şekilde olması beklenmektedir. Apartman kat değerin yükselmesi demek, az önce belirtildiği gibi daha sivri bir durumun yaratılması anlamına gelmektedir. Bu sivri alan çevresinde yoğunlaşan yüksek alan değerleri sürekli olarak yıldırımı yatay doğrultuda (benzetimlerde sola doğru) çekmeye zorlamaktadır. Dolayısıyla kat sayısı ne kadar fazla olursa, alan değerleri burada o kadar yüksek değerler ile yoğunlaşacağından dolayı, yıldırımın yön olarak apartmana yaklaşması da o kadar olağan bir duruma gelmektedir.

  Burada dikkati çeken bir istatistik, dört yönlü ilerleme modelindeki çekme oranı değerlerinin sekiz yönlü ilerleme modeline göre daha düşük olmasıdır. Bunun nedeni ise modelden değil, potansiyel dağılımının yapısından kaynaklanmaktadır. Bölüm 5.5.2'den veya Şekil 5.18'den hatırlanacağı üzere, yıldırımın herhangi bir durumdaki alt ucunun çevresindeki alan dağılımı incelendiğinde, alt çaprazlarındaki noktalarda alan değerlerinin, ucun hemen altındaki noktadan daha büyük değerler aldığı görülmüştü ve buradan yola çıkarak yıldırımın sekiz yönlü ilerleme modelinde çapraz ilerlemeye daha meyilli olduğu belirtilmişti. Bu nedenle şu sonuca varılabilir ki, sekiz yönlü ilerleme modelinde, yıldırım zaten çapraz ilerlemeye daha meyilli olduğu için, dört yönlü ilerlemeye göre çekme miktarları daha yüksek değerlerde çıkmıştır.

  Tablodaki sonuçlara açıklayıcı olması bakımından bir örnek vermek gerekirse, dört yönlü ilerleme modelinde 2 katlı bir apartman yıldırımın üzerine düşebileceği bir konumda olması durumunda, yıldırımın topraktan 100 metre yüksekliği için, kendisine doğru yatay uzaklıkta yaklaşık olarak 4 metre çekebilecek etkiye sahip olduğu söylenebilir. Benzer şekilde, sekiz

yönlü ilerleme modelinde 2 katlı bir apartman, 100 metrelik bir yükseklik için yıldırımı yatay eksende kendisine doğru ortalama 9 metre çekebilecek etkiye sahiptir.

- Kat sayısı ile ortalama adım sayıları karşılaştırıldığında ise, kat sayısı arttıkça adım sayısının genel olarak az da olsa azaldığı görülmektedir. Bu da beklenen bir durumdur çünkü yıldırımın apartman üzerine düştüğü durumlar göz önüne alınırsa, yüksek apartman değerlerine yıldırım daha az adım atarak ulaşacaktır.

  Burada yine dört yönlü ve sekiz yönlü ilerleme modeli ile adım sayılarını karşılaştıracak olursak dört yönlü ilerleme modelinde daha fazla adım atıldığını görmekteyiz. Bunun nedeni bir önceki maddede de belirtildiği üzere yıldırımın alan dağılımının özelliği nedeniyle çapraz yönlü ilerlemeye meyilli olmasından kaynaklanmaktadır. Dört yönlü ilerleme modelinde iki adımda gerçekleştirilebilen bir çapraz ilerleme, sekiz yönlü ilerleme modelinde tek adımda gerçekleştirilebilmektedir.

.

7. SONUÇ VE ÖNERİLER

Yakın geçmişe kadar yıldırımın anlaşılması ve modellenmesi anlamında her ne kadar belli bir yol kat edilmiş olsa da, son dönemlerde yıldırım yolunun modellenmesi çalışmaları oldukça hız kazanmıştır. Yıldırımın oluşumu ve davranışı, anlaşılabilir hale gelmesi açısından, yüksek bilgi ve teknoloji isteyen bir konu olduğundan, incelemelerin bu kadar gecikmesi ne yazık ki kaçınılmaz olmuştur. İleri teknoloji ürünü olan uydular vasıtasıyla anlık görüntüleri ve bulundukları ortamlar daha kolay, daha hızlı ve daha net algılanılabilir duruma gelmesi ile yıldırımın artık daha detaylı olarak incelenebilmesi olanağı ortaya çıkmıştır.

Yıldırımı, davranışını ve izlediği yolları daha iyi anlamak, özellikle yıldırımından korunma tekniklerinin ve ürünlerinin geliştirilmesinde yarar sağlayacaktır. Bu yarar, hiç kuşkusuz en başta insanların yaşamı olmak üzere, tüm canlıları korumaya yöneliktir. Ayrıca, yıldırımdan korunma tekniklerinin uygulandığı bölgedeki cihazların daha iyi korunmasını, dolayısıyla bu cihazlar tarafından yürütülen işlemlerin de sürekliliğini sağlayacaktır. Bu da hem cihazların zarar görmesinden dolayı oluşacak, hem de duracak üretimden dolayı ortaya çıkacak maddi kayıpların önüne geçilmesine yardımcı olacaktır.

Tez çalışmasının sonuçları incelendiğinde, günümüz araştırmalarında öne çıkan yıldırım yolu benzetimlerinin ortaya koyduğu görüntüleri sağladığı ve hatta çok yönlülük ve apartman modelleri gibi bazı ek özellikler getirdiği, bu özellikleri sunarken de tutarlı sonuçlar ortaya koyduğu görülmektedir.

Tez çalışmasında önerilen modellerin sonuçlarının gerçekçi olmasının nedeni, modelin, mevcut yöntemlerin aksine fiziksel tabanlı ve elektriksel potansiyel hesapları kullanılarak geliştirilmiş olmasıdır. Benzetimlerde gerçekleştirilen her algoritmanın ve basamağın fiziksel olarak bir açıklaması mevcuttur. Örneğin, yıldırıma yön değişikliği kazandıracak olan yüklü parçacıkların temsil edilmesi için rasgele sayı üretme algoritması geliştirilmiştir. Olasılık hesaplarının gerçekleştirilmesinde ise ana ölçüt olarak potansiyel dağılımdan yararlanılmıştır.

Potansiyel dağılımdan yararlanan bazı mevcut yöntemlerde dahi nokta seçiminin tamamen rasgele olarak seçildiği görülmüştür. Çalışma bu noktada, temel olasılık hesapları veya gelişi güzel nokta seçimleri ile yıldırım yolu benzetiminin yapıldığı mevcut çalışmalardan ayrılmaktadır.

Tez çalışmasında önerilen iki modelden biri olan dört yönlü ilerleme modeli, mevcut yöntemlerle benzer sonuçlar vermektedir fakat program kodu diğer çalışmalardan bağımsız olarak ve yukarıda sözü edilen bazı ileri algoritmaların veya apartman modeli gibi ek uygulamaların da hesaba katılması ile geliştirilmiştir. Potansiyel dağılımı da dahil olmak üzere tüm hesaplamalardaki programlama mantığı, işleyiş sırası ve kullanılan teknikler tamamen sunulan bu tez çalışmasına özgüdür.

Tezde önerilen modellerden ikincisi olan sekiz yönlü ilerleme modeli ise tez çalışmasının, mevcut yöntemlere göre en fazla öne çıktığı ve farklılık getirdiği modeldir. Modelde şimdiye kadarki çalışmalarda gerçekleştirilmemiş olan, çapraz yönlü ilerleme mümkün kılınmış, hatta ve hatta yıldırımın çapraz olarak ilerlemeye daha meyilli olduğu ortaya konularak mevcut çalışmalar bir adım daha ileriye götürülmüştür. Benzer olarak sekiz yönlü ilerleme modeli için de program kodu ve üretilen algoritmalar tamamen tez çalışmasına özgüdür.

Tezde önerilen iki model göz önüne alındığında, yüksek çözünürlüklerde gerçekçi sonuçlar ortaya koymaktadır. Güç katsayısı ile oynanarak yıldırımın dallanma derecesine müdahale edilebilmektedir. Potansiyel dağılım ve yüklü parçacık varsayımları ile model gerçeğe yakın sonuçlar üretmektedir, nitekim daha fazla parametre hesaba katılarak çalışmayı daha ileri götürebilmek de mümkündür.

Gerçek bir yıldırım bulutu ortamını ve atmosferi daha da detaylı olarak tanımlayacak olan her bir parametre, modeli gerçeğe daha da yakın hale getirecektir. Örneğin, o anki hava sıcaklığının, basıncın veya rüzgarın, yıldırımın izlediği yol üzerindeki etkileri belirli şekillerde tanımlanarak ve buna bağlı ek algoritmalar üretilerek, model daha ileri ve gerçekçi bir boyuta getirilebilir.

KAYNAKLAR

[1] **Uman, M. A., Krider, E. P.**, 1982. A Review of Natural Lightning: "Experimental Data and Modeling" Electromagnetic Compatibility, *IEEE Transactions*, No. 2, Part I, pp. 79 – 112.

[2] **Sosorbaram, B., Fujimoto, T., Muraoka, K., Chiba, N.**, 2001. Visual Simulation of Lightning Taking into Account Cloud Growth, *Computer Graphics International 2001*. Proceedings, 3-6, pp. 89 – 95.

[3] http://tr.wikipedia.org/wiki/Y%C4%B1ld%C4%B1r%C4%B1m, Mayıs, 2007.

[4] **Kim, T., Lin, C.**, 2004. Physically Based Animation and Rendering of Lightning, *Proc. of Pacific Graphics*, Department of Computer Science, University of North Carolina, Chapel Hil.

[5] **Nguyen, D. T., Deegan, G., D'Alessandro, F.**, 2001. Fractal Nature of Probabilistic Model of Lightning Discharge, Electrical and Electronic Technology, TENCON. *Proceedings of IEEE Region 10 International Conference* on Volume 2, 19-22 Aug., pp. 814 – 818.

[6] **Bunde, A. and Havlin**, **S.**, *Fractals and Disordered Systems*. New York, Springer-Verlag, 1991, 350 pp.

[7] **Mendes, O., Margerete, O.**, Elbert E. N., Paula A., 2003. Studies on Lightning Flashes by Using Fractal Analyses and Methods of Geometrical Statistics, *12th International Conference on Atmospheric Electricity (ICAE)*, Instituto Nacional de Pesquisas, SP, Brazil, June 9-13, 2003.

[8] http://www.ipechv.co.uk/technical/partial-discharge.html, Introduction to Partial Discharge, June, 2007.

[9] **Feder, J.**, 1988. *Fractals,* Plenum Press, New York, 283 pages.

[10] **Vicsek, T.**, 1992. *Fractal Growth Phenomena,* World Scientific, River Edge, NJ, (2nd ed.), 488 pages.

[11] **Cannons, J., Kinsner, W.**, 1999. Modeling of Lightning Discharge Patterns as Observed from Space", 12th International Conference on Mathematical and Computer Modeling and Scientific Computing, Chicago, Illinois, August 2-4.

[12] **Canan, S.**, "Kaos'un Resmi: Fraktallar", *Kırmızı Çizgi Dergisi*, Sayı: 3, 2005.

[13] **Mani, G. S.**, 1999. Modeling of Lightning Using Fractals, International Conference on Electromagnetic Interference and Compatibility, Institute of Armament Technology, Pune, 6-8 Dec. 1999, pp. 493 – 496.

[14] **Niemeyer, L., Pietronero, L., Wiessman, H. J.**, 1984. Fractal Dimension of Dielectric Breakdown, *Physical Review Letters*, Volume 52, Number 12, Brown Boveri Research Center, Baden, Switzerland, March, 1984, PACS numbers: 77,50 + p, 02.50+s, pp. 1033 – 1036.

[15] **Chaomei Chen; Lobo, N.**, 2003. Semantically modified diffusion limited aggregation for visualizing large-scale networks, Information Visualization Proceedings, *Seventh International Conference* on 16-18 July 2003 Page(s):576 – 581.

[16] **Trevor, L.**, 2005. A Discussion Paper with Particular Reference to MV XLPE Power Cable Assets, *A Quality Assurance-Based Approach Toward Power Industry Asset Management*, Lord Consulting, Australia, November.

[17] **Kim, T., Sewall, J., Sud, A., Ming, C.**, 2007. Fast Simulation of Laplacian Growth, *Computer Graphics and Applications*, IEEE, Vol. 27, No. 2, March-April, pp. 68 – 76.

[18] **Kim, T., Ming**, **C.**, 2007. Fast Animation of Lightning Using an Adaptive Mesh, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 2, March/April, pp. 390-402.

[19] **Kalenderli, Ö.**, 1995. Elektrik Mühendisliğinde Sonlu Elemanlar Yöntemi, İ.T.Ü. Elektrik-Elektronik Fakültesi, İstanbul.

[20] **Bickel, B., Wicke, M., Gross, M.**, 2006. Adaptive Simulation of Electrical Discharges, VMV, Aachen, Germany, November 22-24.

[21] **Chen, G., Ping, Y., Tao, S., Zhang, S.**, 2005. "Fractal Stochastic Modelling of Breakdown in Dielectric, *14th International Symposium on High Voltage Engineering*, Tsinghua University, Beijing, China, August, 25-29, Page(s), 1-5.

[22] **Uzunoğlu, M., Kızıl, A., Onar, Ö.**, 2003. Her Yönü ile Matlab, İstanbul.

[23] **Uysal, M.**, 2004. Matlab ile Matematiksel Uygulamalar ve Mühendislik Uygulamaları, İstanbul.

Ek A DÖRT ve SEKİZ YÖNLÜ İLERME MODELLERİNE İLİŞKİN MATLAB KODLARI

A.1 Dort_Yonlu_Model.m

```
% Değerlerin dışarıdan alınması
        satir=input('Satır sayısını giriniz:');
        sutun=input('Sutun sayısını giriniz:');
        kuvvet=input('Kuvveti giriniz:');
        iter=input('İterasyon sayısını giriniz:');
% Sadece 100 V sınır değerleri için Laplace denklemi başlangıcı
        x((satir),1:(sutun))=100
        for itteration=1:20
            % Köşe değerler için formülasyon
            x(1,1)=[x(1,2)+x(2,1)]/2
            x(1,(sutun))=[x(1,(sutun-1))+x(2,sutun)]/2
            % Sol kenar değerleri için formülasyon
            for i=2:(satir-1)
               for j=1:1
                  x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j+1)]/4
               end
            end
            % Sağ kenar değerleri için formülasyon
            for i=2:(satir-1)
               for j=sutun:sutun
                  x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j-1)+x(i,j-1)]/4
               end
            end
            % Üst orta kenar değerleri için formülasyon
            for i=1:1
               for j=2:sutun-1
                  x(i,j)=[x(i+1,j)+x(i+1,j)+x(i,j+1)+x(i,j-1)]/4
               end
            end
             % Sınırlara komşu olmayan değerler için formülasyon
             for i=2:(satir-1)
             for j=2:(sutun-1)
                  x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j-1)]/4
               end
             end
end
% Sadece 100 sınır değerleri için Laplace denklemi sonu
% Yıldırım biçimi için Laplace denklemi başlangıcı
```

```
        x(1,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
        x(2,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
        x(3,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
% Program döngüsü buradan başlıyor.
% ---------------------------------------------------------------
% x matrisinin sondan bir önceki satırındaki
% 0 varlığını kontrol etmek
kontrol=x(satir-1,:)
all(kontrol)
        for i=1:(satir-1)
                for j=1:(sutun)
                if all(kontrol)~=0
                        for itteration=1:(iter)
                                x(1,1)=[x(1,2)+x(2,1)]/2
                                x(1,(sutun))=[x(1,(sutun-1))+x(2,(sutun))]/2
                        for i=2:(satir-1)
                        for j=1:1
                                if x(i,j)~=0
                                        x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j+1)]/4
                                else
                                         x(i,j)=0
                                end
                        end
                        end
                        for i=2:(satir-1)
                        for j=(sutun):(sutun)
                                if x(i,j)~=0
                                         x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j-1)+x(i,j-1)]/4
                                else
                                        x(i,j)=0
                                end
                        end
                        end
                        for i=1:1
                        for j=2:(sutun-1)
                                if x(i,j)~=0
                                        x(i,j)=[x(i+1,j)+x(i+1,j)+x(i,j+1)+x(i,j-1)]/4
                                else
                                        x(i,j)=0
                                end
                        end
                        end
                        for i=2:(satir-1)
                        for j=2:(sutun-1)
                                if x(i,j)~=0
                                        x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j-1)]/4
                                else
                                        x(i,j)=0
                                end
                        end
```

```
                end
        end
% Yıldırım biçimi için Laplace denklemi sonu
% "x" matrisi değerlerinin karelerinin hesaplandığı "k" matrisi
        k=x.^(kuvvet)
% Bir sonraki olasılık durumları için payda değerinin hesaplanması
        b(satir,sutun)=0
% ilk satır için payda değerlerinin hesaplanması
        for j=1:(sutun)
        for i=1:1
                if k(i,j)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0
                        b(i,j)=[k(i,j+1)+k(i,j-1)+k(i+1,j)]
                elseif k(i,j)==0
                        b(i,j)=[k(i,j+1)+k(i,j-1)]
                else
                        b(i,j)=0
                end
        end
        end
% 2. satırdan sondan bir önceki satıra kadar payda değerlerin hesaplanması
        for j=1:sutun
        for i=2:(satir-2)
                if k(i,j)==0 & k(i,j+2)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0 & k(i-1,j-
                1)>0 & k(i-1,j+1)>0
                        b(i,j)=[k(i,j-1)+k(i+1,j)+k(i-1,j)]
                elseif k(i,j)==0 & k(i,j+2)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0
                        b(i,j)=[k(i,j-1)+k(i+1,j)]
                elseif k(i,j)==0 & k(i,j+2)==0 & k(i-1,j+1)>0 & k(i-1,j-1)>0
                        b(i,j)=[k(i,j-1)+k(i-1,j)]
                elseif k(i,j)==0 & k(i,j+2)==0
                        b(i,j)=[k(i,j-1)]
% mevcut sıfır noktasının iki sağında sıfır olup olmadığı yukarıda denetleniyor
% aşağıdan itibaren sağında sıfır olmaması durumu irdeleniyor
                elseif k(i,j)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0 & k(i-1,j-1)>0 & k(i-
                1,j+1)>0
                        b(i,j)=[k(i,j+1)+k(i,j-1)+k(i+1,j)+k(i-1,j)]
                elseif k(i,j)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0
                        b(i,j)=[k(i,j+1)+k(i,j-1)+k(i+1,j)]
                elseif k(i,j)==0 & k(i-1,j+1)>0 & k(i-1,j-1)>0
                        b(i,j)=[k(i,j+1)+k(i,j-1)+k(i-1,j)]
                elseif k(i,j)==0
                        b(i,j)=[k(i,j+1)+k(i,j-1)]
                else
                        b(i,j)=0
                end
        end
        end
% Sonuncu satırın payda değerlerinin hesaplanması
for j=1:sutun
      for i=(satir-1):(satir-1)
```

```
        if k(i,j)==0
          b(i,j)=[k(i,j+1)+k(i,j-1)]
          else
          b(i,j)=0
        end
      end
end
% Matristen payda değerinin ("p") oluşturulması (b matrisinin satır ve sütünlarının
toplanması)
        p1=sum(b(1:(satir-1),:))
        p=sum(p1(:,1:(sutun)))
% Bir sonraki adım için alana bağlı olasılıkların hesabı ve payda matrisi "b"den, "o"
matrisinin elde edilmesi.
% 1. satırdan son satıra kadar soldan tarama
        o1((satir-1),sutun)=0
        for j=1:sutun
        for i=1:(satir-1)
                if k(i,j)==0
                        o1(i,j-1)=[k(i,j-1)/p]
                else
                        o1(i,j)=0
                end
        end
        end
% 1. satırdan son satıra kadar sağdan tarama
        o2((satir-1),sutun)=0
        for j=sutun:-1:1
        for i=1:(satir-1)
                if k(i,j)==0
                        o2(i,j+1)=[k(i,j+1)/p]
                else
                        o2(i,j)=0
                end
        end
        end
% 1. satırdan sondan bir önceki satıra kadar aşağıyı tarama, son satırda aşağıyı
taramaya gerek yok.
        o3((satir-1),(sutun))=0
        for i=(satir-1):-1:1
        for j=1:(sutun)
                if k(i,j)==0 & k(i+1,j+1)>0 & k(i+1,j-1)>0
                        o3(i+1,j)=[k(i+1,j)/p]
                else
                        o3(i,j)=0
                end
        end
        end

% 2. satırdan son satıra kadar yukarıyı tarama, ilk satırda yukarıyı taramaya gerek
yok.
```

```
        o4((satir-1),sutun)=0
        for j=1:(sutun)
        for i=2:(satir-1)
                if k(i,j)==0 & k(i-1,j+1)>0 & k(i-1,j-1)>0
                        o4(i-1,j)=[k(i-1,j)/p]
                else
                        o4(i,j)=0
                end
        end
        end
% sağ sol ve alt olasılıkların toplandığı "o" matrisi
        oara=o1+o2
        oara2=oara+o3
        o=oara2+o4
% Rasgele "r" matrisinin üretilmesi
        r=(1-0)*rand((satir-1),(sutun))
% "o" olasılık matrisi ile "r" rasgele matrisin elemanter çarpımı
% "f" son matrisin elde edilmesi:
        f=o.*r
% "f" son durum matrisinde en büyük değerin koordinatlarını en baştaki "x"
matrisine atama
        m1=max(f)
        m=max(m1)

        for i=1:(satir-1)
        for j=1:(sutun)
                if f(i,j)<m
                        continue;
                else
                        x(i,j)=0
                end
        end
        end
kontrol=x((satir-1),:)
all(kontrol)
% ---------------------------------------------
% Programın burasından itibaren döngü sona eriyor.
        else
                break
        end
        end
        end
```

A.2 Sekiz_Yonlu_Model.m

```
% Değerlerin dışarıdan alınması
        satir=input('Satır sayısını giriniz:');
        sutun=input('Sutun sayısını giriniz:');
        kuvvet=input('Kuvveti giriniz:');
        iter=input('İterasyon sayısını giriniz:');
% Sadece 100 sınır değerleri için Laplace denklemi başlangıcı
        x((satir),1:(sutun))=100
        for itteration=1:20
% Köşe değerler için formülasyon
                x(1,1)=[x(1,2)+x(2,1)]/2
                x(1,(sutun))=[x(1,(sutun-1))+x(2,sutun)]/2
% Sol kenar değerleri için formülasyon
                for i=2:(satir-1)
                for j=1:1
                x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j+1)]/4
                end
                end
% Sağ kenar değerleri için formülasyon
                for i=2:(satir-1)
                for j=sutun:sutun
                        x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j-1)+x(i,j-1)]/4
                end
                end
% Üst orta kenar değerleri için formülasyon
                for i=1:1
                for j=2:sutun-1
                        x(i,j)=[x(i+1,j)+x(i+1,j)+x(i,j+1)+x(i,j-1)]/4
                end
                end
% Sınırlara komşu olmayan değerler için formülasyon
                for i=2:(satir-1)
                for j=2:(sutun-1)
                        x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j-1)]/4
            end
            end
        end
% Sadece 100 V sınır değerleri için Laplace denklemi sonu
% Yıldırım biçimi için Laplace denklemi başlangıcı
        x(1,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
        x(2,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
        x(3,(sutun+1)/2)=0 %Başlangıç sıfır değerleri
% Program döngüsü buradan başlıyor.
% ----------------------------------------------------------------
% x matrisinin 0 varlıklarını ve sınıra komşuluklarını kontrol etmek
        l(satir-1,sutun)=1
        for i=1:(satir-1)
        for j=1:(sutun)
                if x(i,j)==0 & x(i+1,j)==100
```

```
                l(i,j)=0
        elseif x(i,j)==0 & x(i+1,j+1)==100
                l(i,j)=0
        elseif x(i,j)==0 & x(i+1,j-1)==100
                l(i,j)=0
        elseif x(i,j)==0 & x(i,j-1)==100
                l(i,j)=0
        elseif x(i,j)==0 & x(i,j+1)==100
                l(i,j)=0
        else
                l(i,j)=1
        end
        end
end
for i=1:(satir-1)
for j=1:(sutun)
        if all(l)~=0
        for itteration=1:(iter)
                        x(1,1)=[x(1,2)+x(2,1)]/2
                        x(1,(sutun))=[x(1,(sutun-1))+x(2,(sutun))]/2
                for i=2:(satir-1)
                for j=1:1
                        if x(i,j)~=0 & x(i,j)~=100
                                x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j+1)]/4
                        elseif x(i,j)==100
                                x(i,j)=100
                        else
                                x(i,j)=0
                        end
                end
                end
        for i=2:(satir-1)
        for j=(sutun):(sutun)
                if x(i,j)~=0 & x(i,j)~=100
                        x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j-1)+x(i,j-1)]/4
                elseif x(i,j)==100
                        x(i,j)=100
                else
                        x(i,j)=0
                end
        end
        end
        for i=1:1
        for j=2:(sutun-1)
                if x(i,j)~=0
                        x(i,j)=[x(i+1,j)+x(i+1,j)+x(i,j+1)+x(i,j-1)]/4
                else
                        x(i,j)=0
                end
        end
```

```
                end
                for i=2:(satir-1)
                for j=2:(sutun-1)
                        if x(i,j)~=0 & x(i,j)~=100
                                x(i,j)=[x(i+1,j)+x(i-1,j)+x(i,j+1)+x(i,j-1)]/4
                        elseif x(i,j)==100
                                x(i,j)=100
                        else
                                x(i,j)=0
                        end
                end
                end
        end
% Yıldırım biçimi için Laplace denklemi sonu
% "x" matrisi değerlerinin karelerinin hesaplandığı "k" matrisi
        k=x.^(kuvvet)
% ----------------------------------------------------------------
% Bir sonraki olasılık durumları için payda değerinin hesaplanması
% b=b1+b2+b3+b4+b5+b6 şeklinde geliştirildi.
        b1(satir,sutun)=0 % sıfırların sağ ve solları toplamı
        b2(satir,sutun)=0 % sıfırların sağ alt çaprazı sorgulaması
        b3(satir,sutun)=0 % sıfırların sağ üst çaprazı sorgulaması
        b4(satir,sutun)=0 % sıfırların sol alt çaprazı sorgulaması
        b5(satir,sutun)=0 % sıfırların sol üst çaprazı sorgulaması
        b6(satir,sutun)=0 % sıfırların alt uç olması durumu sorgulaması
% b1 ---- sıfırların sağ ve solları toplamı
        for j=1:(sutun)
        for i=1:(satir)
                if k(i,j)==0 & k(i,j+2)==0
                        b1(i,j)=[k(i,j-1)]
                elseif k(i,j)==0
                        b1(i,j)=[k(i,j+1)+k(i,j-1)]
                else
                        b1(i,j)=0
                end
        end
        end
% b2 ---- sıfırların sağ alt çaprazı sorgulaması
% (ilk satırdan sondan iki öncekine kadar)
        for j=1:sutun
        for i=1:(satir-2)
                if k(i,j)==0 & k(i+1,j)>0 & k(i+1,j+2)>0
                        b2(i,j)=[k(i+1,j+1)]
                else
                        b2(i,j)=0
                end
        end
        end
% b3 ---- sıfırların sağ üst çaprazı sorgulaması
% (ilk iki satırda çapraz taramaya gerek yok)
```

```
        for j=1:sutun
        for i=(satir-1):-1:3
                if k(i,j)==0 & k(i-1,j)>0 & k(i-1,j+2)>0 & k(i-2,j)>0
                        b3(i,j)=[k(i-1,j+1)]
                else
                        b3(i,j)=0
                end
        end
        end
% b4 ---- sıfırların sol alt çaprazı sorgulaması
% (1. satırdan sondan iki öncekine kadar)
        for j=(sutun):-1:1
        for i=1:(satir-2)
                if k(i,j)==0 & k(i+1,j)>0 & k(i+1,j-2)>0 & k(i,j-2)>0
                        b4(i,j)=[k(i+1,j-1)]
                else
                        b4(i,j)=0
                end
        end
        end
% b5 ---- sıfırların sol üst çaprazı sorgulaması
% (ilk iki satırda çapraz taramaya gerek yok)
        for j=sutun:-1:1
        for i=(satir-1):-1:3
                if k(i,j)==0 & k(i-1,j-2)>0 & k(i-1,j-2)>0 & k(i-2,j)>0
                        b5(i,j)=[k(i-1,j-1)]
                else
                        b5(i,j)=0
                end
        end
        end

% b6 ---- sıfırların alt uç olması durumu sorgulaması
% (1. satırdan sondan iki öncekine kadar)
        for j=1:sutun
        for i=1:(satir-2)
                if k(i,j)==0 & k(i,j-1)>0 & k(i,j+1)>0 & k(i,j-1)>0 & k(i,j+1)>0
                        b6(i,j)=[k(i+1,j)]
                else
                        b6(i,j)=0
                end
        end
        end
% Değerler toplanır.
        b=(b1+b2+b3+b4+b5+b6)
% Matristen payda değerinin ("p") oluşturulması
% (b matrisinin satır ve sütünlarının toplanması)
        p1=sum(b(1:(satir-1),:))
        p=sum(p1(:,1:(sutun)))
% Bir sonraki olasılık durumları için payda değerinin hesaplanmasın sonu.
```

```
% ------------------------------------------------------------------------
% Bir sonraki adım için alana bağlı olasılıkların hesabı
% payda matrisi "b"den, "o" matrisinin elde edilmesi.
% 1. satırdan sondan bir önceki satıra kadar soldan tarama
        o1((satir-1),sutun)=0
        for j=1:sutun
        for i=1:(satir-1)
                if k(i,j)==0
                        o1(i,j-1)=[k(i,j-1)/p]
                else
                        o1(i,j)=0
                end
        end
        end
% 1. satırdan sondan bir önceki satıra kadar sağdan tarama
        o2((satir-1),sutun)=0
        for j=sutun:-1:1
        for i=1:(satir-1)
                if k(i,j)==0
                        o2(i,j+1)=[k(i,j+1)/p]
                else
                        o2(i,j)=0
                end
        end
        end
% 1. satırdan sondan iki önceki satıra kadar sağ alt çapraz tarama
% Son satırdan bir öncekinde aşağıyı taramaya gerek yok.
        o3((satir-1),(sutun))=0
        for i=(satir-2):-1:1
        for j=1:(sutun)
                if k(i,j)==0 & k(i+1,j)>0 & k(i+1,j+2)>0
                        o3(i+1,j+1)=[k(i+1,j+1)/p]
                else
                        o3(i,j)=0
                end
        end
        end
% 2. satırdan sondan bir önceki satıra kadar sağ üst tarama
% İlk satırda yukarıyı taramaya gerek yok.
        o4((satir-1),sutun)=0
        for j=(sutun):-1:1
        for i=2:(satir-1)
                if k(i,j)==0 & k(i-1,j)>0 & k(i-1,j+2)>0
                        o4(i-1,j+1)=[k(i-1,j+1)/p]
                else
                        o4(i,j)=0
                end
        end
        end
% 1. satırdan sondan iki önceki satıra kadar sol alt çapraz tarama
```

```
% Son satırdan bir öncekinde aşağıyı taramaya gerek yok.
        o5((satir-1),(sutun))=0
        for i=(satir-2):-1:1
        for j=1:(sutun)
                if k(i,j)==0 & k(i+1,j)>0 & k(i+1,j-2)>0 & k(i,j-2)>0
                        o5(i+1,j-1)=[k(i+1,j-1)/p]
                else
                        o5(i,j)=0
                end
        end
        end
% 2. satırdan sondan iki önceki satıra kadar sol üst çapraz tarama
% Son satırdan bir öncekinde aşağıyı taramaya gerek yok.
        o6((satir-1),(sutun))=0
        for i=2:(satir-1)
        for j=1:(sutun)
                if k(i,j)==0 & k(i-1,j)>0 & k(i-1,j-2)>0 & k(i,j-2)>0
                        o6(i-1,j-1)=[k(i-1,j-1)/p]
                else
                        o6(i,j)=0
                end
        end
        end
% 1. satırdan sondan iki önceki satıra kadar alt uc nokta tarama
% Son satırdan bir öncekinde aşağıyı taramaya gerek yok.
        o7((satir-1),(sutun))=0
        for i=(satir-2):-1:1
        for j=1:(sutun)
                if k(i,j)==0 & k(i,j-1)>0 & k(i,j+1)>0 & k(i+1,j-1)>0 & k(i+1,j+1)>0
                        o7(i+1,j)=[k(i+1,j)/p]
                else
                        o7(i,j)=0
                end
        end
        end
% sağ sol ve alt olasılıkların toplandığı "o" matrisi
        oara=o1+o2
        oara2=oara+o3
        oara3=oara2+o4
        oara4=oara3+o5
        oara5=oara4+o6
        o=oara5+o7
% Rasgele "r" matrisinin üretilmesi
        r=(1-0)*rand((satir-1),(sutun))
% "o" olasılık matrisi ile "r" rasgele matrisin elemanter çarpımı
% "f" son matrisin elde edilmesi:
        f=o.*r
% "f" son durum matrisinde en büyük değerin koordinatlarını
% en baştaki "x" matrisine atama
        m1=max(f)
```

```
        m=max(m1)
        for i=1:(satir-1)
        for j=1:(sutun)
                if f(i,j)<m
                        continue;
                else
                        x(i,j)=0
                end
        end
        end
        l(satir-1,sutun)=1
        for i=1:(satir-1)
        for j=1:(sutun)
                if x(i,j)==0 & x(i+1,j)==100
                        l(i,j)=0
                elseif x(i,j)==0 & x(i+1,j+1)==100
                        l(i,j)=0
                elseif x(i,j)==0 & x(i+1,j-1)==100
                        l(i,j)=0
                elseif x(i,j)==0 & x(i,j-1)==100
                        l(i,j)=0
                elseif x(i,j)==0 & x(i,j+1)==100
                        l(i,j)=0
                else
                        l(i,j)=1
                end
                end
        end
        all(l)
% ------------------------------------------
% Programın burasından itibaren döngü sona eriyor.
        elseif all(l)==0
                break;
         else
                break;
        end
        end
        end
```

Ek B TEZ ÇALIŞMASINDA KULLANILAN ŞEKİLLERE İLİŞKİN MATLAB SONUÇLARI

B.1 Şekil 6.2'deki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

    6.2822    6.0388    5.3489    4.3083    3.0771    1.8662    0.8608         0    0.8745    1.8859    3.0866    4.2769    5.2585    5.8940    6.1294
    6.5092    6.2506    5.5144    4.3960    3.0619    1.7607    0.7875         0    0.7997    1.7792    3.0722    4.3545    5.3994    6.0700    6.3179
    7.2285    6.9246    6.0494    4.6897    3.0081    1.3244    0.5280         0    0.5367    1.3419    3.0392    4.6289    5.8650    6.6397    6.9244
    8.5293    8.1533    7.0557    5.2960    2.9517         0         0         0         0         0    3.0792    5.2061    6.7304    7.6621    8.0038
   10.5534   10.0860    8.7100    6.4775    3.4993         0    0.0307    0.0275         0    1.5628    4.0242    6.3191    8.1072    9.2249    9.6400
   13.4810   12.9082   11.2054    8.3950    4.5646         0    0.0953    0.0794         0    2.1921    5.0717    7.8501   10.0480   11.4262   11.9400
   17.5199   16.8395   14.7919   11.3218    6.3602         0    0.2710    0.1948         0    2.0919    6.1393    9.8483   12.6733   14.4126   15.0553
   22.8828   22.1154   19.7828   15.7282    9.5497         0    0.7935    0.4285         0         0    7.4488   12.5926   16.2213   18.4028   19.1979
   29.7418   28.9331   26.4763   22.2451   16.1048    8.3756    2.4737    0.7256         0    4.5358   10.9460   16.6933   21.0312   23.6784   24.6364
   38.1788   37.3752   34.9240   30.6564   24.2397   14.9199         0         0         0    7.1193   14.9760   22.0332   27.3369   30.5411   31.6781
   48.1856   47.4419   45.1689   41.2018   35.2678   27.0601   16.7895    8.7754         0    8.8848   19.6768   28.9604   35.5532   39.3768   40.6877
   59.6463   59.0180   57.0906   53.7003   48.5600   41.2576   31.3194   18.3111         0    8.6747   25.7736   38.4325   46.3734   50.6480   52.0468
   72.3370   71.8779   70.4616   67.9379   64.0061   58.0855   48.9165   33.1487         0         0   36.2267   52.5113   60.7329   64.7409   65.9901
   85.9287   85.6853   84.9316   83.5769   81.4357   78.1582   73.1104   65.3661         0   55.4465   66.5691   74.5831   79.2272   81.5655   82.2938
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.1 : Şekil 6.2'de Soldaki Sonuca İlişkin Matlab Çıktısı

B.2 Şekil 6.2'deki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

   14.1049   13.6952   12.4910   10.5479    7.9845    5.0370    2.1542         0    0.1971    0.4017    0.6134    0.8198    0.9961    1.1149    1.1599
   14.4975   14.0799   12.8492   10.8487    8.1699    5.0008    1.7888         0    0.1915    0.3950    0.6116    0.8294    1.0188    1.1475    1.1961
   15.6981   15.2604   13.9617   11.8155    8.8369    5.0034         0         0    0.1714    0.3710    0.6030    0.8604    1.0944    1.2564    1.3170
   17.7444   17.2824   15.9047   13.6005   10.3495    6.1720    1.5430         0    0.1209    0.3110    0.5642    0.9086    1.2350    1.4632    1.5474
   20.6807   20.1981   18.7547   16.3162   12.7773    7.7866         0         0         0    0.1856    0.4304    0.9697    1.4677    1.8110    1.9357
   24.5438   24.0497   22.5772   20.1137   16.6433   12.1906    7.1042    3.7581         0         0         0    1.0680    1.8498    2.3748    2.5645
   29.3524   28.8520   27.3655   24.8965   21.4747   17.2163   12.4607    7.9247    3.4911         0    0.5027    1.4490    2.4844    3.2715    3.5650
   35.1156   34.6106   33.1090   30.6079   27.1228   22.7244   17.5871   11.9830    6.0374         0    0.5604    1.7379    3.3634    4.6601    5.1458
   41.8409   41.3351   39.8235   37.2781   33.6630   28.9546   23.1682   16.3758    8.6731         0         0    1.5768    4.5683    6.8580    7.6924
   49.5304   49.0352   47.5439   44.9927   41.2750   36.2456   29.7431   21.6712   12.2767    3.0692         0         0    6.4726   10.5100   11.9034
   58.1662   57.7037   56.2984   53.8508   50.1788   44.9937   37.8756   28.2820   15.6904         0         0    2.7027   10.8101   16.8051   18.8978
   67.6890   67.2916   66.0733   63.9135   60.5786   55.6612   48.4737   37.8849   22.2011         0         0         0   17.2588   27.0019   30.0751
   77.9770   77.6825   76.7729   75.1366   72.5482   68.5884   62.4657   52.5783   35.2280         0         0         0   31.2224   43.8682   47.3968
   88.8349   88.6775   88.1891   87.3025   85.8814   83.6724   80.2179   74.7322   66.1318   54.5668   52.1353         0   63.7621   69.8513   71.7750
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.2 : Şekil 6.2'de Sağdaki Sonuca İlişkin Matlab Çıktısı

B.3 Şekil 6.3'teki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

    4.4069    4.2404    3.7813    3.0964    2.2834    1.4511    0.6858         0    4.7064    8.9424   12.5230   15.3646   17.4155   18.6400
    4.5410    4.3639    3.8746    3.1450    2.2821    1.4111    0.6437         0    4.8903    9.1862   12.7831   15.6315   17.6875   18.9153
    4.9814    4.7702    4.1834    3.3077    2.2763    1.2610    0.4762         0    5.5830    9.9995   13.6289   16.5038   18.5852   19.8301
    5.7968    5.5232    4.7571    3.6084    2.2435    0.8761         0    3.0662    7.3306   11.4445   15.0400   17.9562   20.0905   21.3723
    7.1123    6.7401    5.6899    4.1093    2.2051         0    1.2206    4.8426    9.0818   13.2153   16.9031   19.9393   22.1814   23.5330
    9.1237    8.6054    7.1299    4.9189    2.4625    0.6156         0    5.8781   10.7510   15.1949   19.1448   22.4188   24.8499   26.3186
   12.1200   11.3960    9.2808    5.9594    2.1055         0    3.1547    7.7544   12.6187   17.3858   21.7419   25.3952   28.1194   29.7667
   16.5056   15.5423   12.6116    7.5191         0    1.6119    4.7419    9.1683   14.3180   19.6677   24.6813   28.9131   32.0632   33.9622
   22.7496   21.6147   18.0746   11.4934         0    1.6394    4.8970    9.6484   15.5362   21.9463   28.0194   33.1008   36.8305   39.0552
   31.1849   30.0447   26.5438   20.3638   11.1662         0    3.4410    8.7968   15.9625   24.2287   31.9689   38.2286   42.6748   45.2695
   41.8144   40.7839   37.6515   32.2257   24.2896   13.7722         0    5.9844   15.0582   26.7388   37.0495   44.7879   49.9718   52.8894
   54.4198   53.5748   51.0115   46.5686   39.9770   30.7922   18.0519         0   11.3811   30.3794   44.4110   53.5774   59.1945   62.2010
   68.6430   68.0419   66.2166   63.0342   58.2405   51.3586   41.4116   26.2430         0   38.8195   56.4220   65.6718   70.7696   73.3592
   84.0190   83.7065   82.7560   81.0940   78.5802   74.9825   69.9886   63.5570   57.9891         0   76.6529   81.7676   84.6933   86.1820
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.3 : Şekil 6.3'teki Soldaki Sonuca İlişkin Matlab Çıktısı

B.4 Şekil 6.3'teki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

    5.0355    4.8534    4.3429    3.5780    2.6646    1.7147    0.8155         0    3.5472    6.8280    9.6847   12.0060   13.7090   14.7362   15.1361
    5.1995    5.0049    4.4591    3.6435    2.6767    1.6856    0.7722         0    3.6416    6.9752    9.8659   12.2115   13.9318   14.9694   15.3731
    5.7259    5.4912    4.8313    3.8495    2.7055    1.5750    0.5868         0    3.9843    7.4716   10.4702   12.8988   14.6796   15.7541   16.1715
    6.6954    6.3865    5.5117    4.2075    2.7147    1.3190         0    1.1959    4.7591    8.3579   11.5174   14.0852   15.9710   17.1104   17.5518
    8.2564    7.8317    6.6089    4.7449    2.6217    0.9852         0         0    5.4290    9.5787   13.0227   15.7989   17.8402   19.0768   19.5531
   10.6406   10.0595    8.3350    5.5339    2.0389         0    0.8011    3.1906    7.3000   11.3933   15.0562   18.0874   20.3406   21.7133   22.2375
   14.1610   13.4154   11.1264    7.0110         0         0         0    4.6109    9.1016   13.5213   17.5784   20.9906   23.5460   25.1084   25.6968
   19.1464   18.2997   15.7336   11.3791    5.4186         0    2.0836    6.0947   10.8857   15.8940   20.6025   24.5888   27.5708   29.3902   30.0619
   25.7995   24.8885   22.1179   17.3469   10.2928         0    2.2114    6.7420   12.3664   18.4528   24.2118   29.0367   32.5928   34.7380   35.5104
   34.2497   33.3226   30.4916   25.5906   18.4030    9.0153         0    6.2460   13.3073   21.2355   28.6298   34.6118   38.8741   41.3857   42.2638
   44.5313   43.6473   40.9250   36.1137   28.7093   17.6569         0    4.8962   13.3173   24.4642   34.3521   41.7839   46.7745   49.6055   50.5634
   56.5616   55.7991   53.4382   49.2233   42.6598   32.9016   18.7258         0   10.5561   28.8837   42.4443   51.2982   56.7278   59.6508   60.6063
   70.1020   69.5407   67.7985   64.6765   59.8016   52.5620   42.0011   26.4151         0   38.0237   55.1814   64.1647   69.1097   71.6317   72.4315
   84.7555   84.4580   83.5343   81.8795   79.3060   75.5423   70.3007   63.6586   57.9167         0   76.0555   81.0257   83.8675   85.3186   85.7752
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.4 : Şekil 6.3'teki Sağdaki Sonuca İlişkin Matlab Çıktısı

B.5 Şekil 6.7'teki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

  18.7719  18.3253  17.0199  14.8888  11.9924   8.4303   4.3575        0   0.3913   0.7901   1.1872   1.5571   1.8635   2.0666   2.1435
  19.1712  18.7202  17.4013  15.2442  12.3038   8.6714   4.4937        0   0.3841   0.7852   1.1928   1.5790   1.9038   2.1216   2.2042
  20.3997  19.9361  18.5779  16.3459  13.2784   9.4388   4.9392        0   0.3552   0.7662   1.2089   1.6488   2.0359   2.3037   2.4055
  22.4783  21.9964  20.5814  18.2421  14.9915  10.8438   5.8154        0   0.2665   0.7081   1.2172   1.7580   2.2714   2.6426   2.7845
  25.4368  24.9342  23.4564  21.0010  17.5604  13.0998   7.4659        0        0   0.5763   1.1839   1.8805   2.6315   3.2010   3.4185
  29.3069  28.7842  27.2463  24.6854  21.0961  16.4879  10.9281   4.7760        0   0.4079   1.0518   1.9342   3.1539   4.0995   4.4545
  34.1165  33.5760  31.9844  29.3235  25.5809  20.7697  14.9438   8.1591        0        0   0.6732   1.6365   3.9285   5.5751   6.1615
  39.8865  39.3343  37.7030  34.9524  31.0467  25.9904  19.8638  12.8911   5.6395        0        0        0   5.3244   8.0951   8.9959
  46.6259  46.0760  44.4393  41.6310  37.5601  32.1904  25.5600  17.8597   9.6503   2.4126        0   4.3575   9.2446  12.4670  13.5797
  54.3198  53.8029  52.2382  49.4586  45.2613  39.5519  32.2486  23.2875  12.6681        0        0   8.1633  14.7963  18.9295  20.3304
  62.9026  62.4772  61.1454  58.5929  54.3657  48.4102  40.5188  30.3251  17.7175   4.4294        0  13.4756  22.8135  28.1059  29.8234
  72.2045  71.9697  71.1801  69.3063  65.1040  59.1196  51.0249  39.7340  23.4298        0        0  22.9027  34.8447  40.8419  42.6963
  81.8725  81.9504  82.2313  82.2801  77.5550  71.8763  64.6776  54.1256  36.2584        0  26.7697  43.2714  52.7958  57.7094  59.2339
  91.3192  91.6898  93.4803 100.0000  90.9209  86.1158  81.6542  75.8146  67.4734        0  63.7990  70.6042  75.3416  77.9607  78.7916
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil B.5 : Şekil 6.7'teki Soldaki Sonuca İlişkin Matlab Çıktısı

B.6 Şekil 6.7'teki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

   1.6342   1.5566   1.3470   1.0360   0.6750   0.3420   0.1375        0   1.1945   2.9638   5.8001   8.5532  10.6820  11.9960  12.4431
   1.6877   1.6061   1.3841   1.0512   0.6552   0.2753   0.1032        0   0.9058   2.4272   5.8356   8.8570  11.0793  12.4222  12.8760
   1.8697   1.7756   1.5157   1.1183   0.6136        0        0        0        0        0   6.2483   9.9468  12.3407  13.7289  14.1929
   2.2077   2.0915   1.7699   1.2830   0.6779        0   0.9304   2.0846   3.5306   5.5376   9.1965  12.3237  14.5887  15.9486  16.4096
   2.7479   2.5952   2.1757   1.5573   0.8119        0   1.6324   3.8684   6.4875   9.4056  12.6550  15.5392  17.7168  19.0528  19.5120
   3.5659   3.3495   2.7677   1.9506   1.0101        0   1.7235   5.2547   9.1241  12.9162  16.4493  19.4299  21.6541  23.0156  23.4859
   4.7914   4.4544   3.5840   2.4602   1.2755        0        0   6.2832  11.8090  16.6503  20.7574  24.0366  26.4136  27.8471  28.3414
   6.6686   6.0800   4.6437   3.0246   1.6295   0.5460        0   8.0440  15.1414  21.0745  25.8457  29.4967  32.0679  33.5915  34.1143
   9.7040   8.5425   5.8783   3.3599   1.6695   0.5539        0  10.7215  19.5946  26.6091  31.9993  35.9810  38.7147  40.3082  40.8516
  15.0474  12.4997   6.9615   2.8639   1.1333        0        0  15.2130  25.8586  33.7122  39.5029  43.6543  46.4432  48.0457  48.5887
  25.4759  19.4426   6.6010        0        0        0  11.1568  24.2350  34.8655  42.8228  48.5878  52.6324  55.3009  56.8156  57.3255
  47.9653        0        0        0   6.4662        0  20.3700  35.6688  46.5007  54.0758  59.3414  62.9348  65.2618  66.5674  67.0040
 100.0000  65.3583  43.4023  34.2536  25.8646        0  34.6348  51.5404  61.3570  67.5992  71.7267  74.4633  76.2047  77.1717  77.4932
 100.0000  84.8389  73.9972  67.7473  62.7380  57.3391  66.6153  74.4819  79.7649  83.2125  85.4773  86.9615  87.8969  88.4132  88.5841
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil B.6 : Şekil 6.7'teki Sağdaki Sonuca İlişkin Matlab Çıktısı

B.7 Şekil 6.8'deki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

   4.0866   3.8501   3.1853   2.2054   1.1688   0.5563   0.2188        0   2.6842   5.2331   7.5195   9.4279  10.8621  11.7501  12.0
   4.3000   4.0487   3.3308   2.2262   0.9534   0.4173   0.1590        0   2.7473   5.3568   7.6982   9.6528  11.1215  12.0308  12.3
   4.9819   4.6937   3.8481   2.4069        0        0        0        0   2.9412   5.7378   8.2493  10.3465  11.9218  12.8965  13.2
   6.2048   5.8755   4.9458   3.5467   1.9412   1.0476   0.4592        0   3.2728   6.3926   9.2000  11.5445  13.3035  14.3901  14.7
   8.0494   7.6351   6.4955   4.8816   3.1638   1.7865   0.7878        0   3.7499   7.3477  10.5977  13.3096  15.3371  16.5850  17.0
  10.6832  10.0963   8.5007   6.3074   4.0374   2.1428   0.9044        0   4.3708   8.6372  12.5163  15.7391  18.1281  19.5875  20.0
  14.4503  13.5425  11.0848   7.7965   4.5279   1.8394   0.6859        0   5.0869  10.2991  15.0719  18.9800  21.8246  23.5379  24.1
  19.9941  18.5164  14.4823   9.2545   4.4327        0        0        0   5.6675  12.3839  18.4709  23.2596  26.6259  28.6074  29.2
  28.4593  26.0278  19.0594  10.2985   3.9460   1.0523   0.2631        0   5.1889  15.0803  23.1446  28.9349  32.7836  34.9805  35.7
  41.7611  38.0623  25.4196   8.9295        0        0        0        0        0  19.5851  30.0675  36.5240  40.5635  42.8108  43.5
  62.4436  59.0327  35.6227        0        0   7.7239  13.6458  18.2882  23.4198  33.1717  40.9908  46.5026  50.1071  52.1407  52.8
  89.9403 100.0000  58.0378  25.0340        0  17.2436  28.5621  36.0752  42.2022  48.6701  54.1979  58.3635  61.1961  62.8256  63.3
  97.3174 100.0000  71.4939  42.0980        0  32.6813  47.2737  55.2353  60.6280  65.0908  68.7481  71.5375  73.4682  74.5927  74.9
  99.3294 100.0000  85.8395  71.8637        0  66.2022  72.6085  76.9551  79.9731  82.3055  84.1538  85.5574  86.5339  87.1059  87.2
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0
```

Şekil B.7 : Şekil 6.8'deki Soldaki Sonuca İlişkin Matlab Çıktısı

B.8 Şekil 6.8'deki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

  14.7117  14.1762  12.7308  10.5531   7.9183   5.1510   2.5071        0   3.4011   7.9067  12.3265  16.0187  18.7283  20.3646  20.9387
  15.0511  14.4904  12.9732  10.6822   7.9132   5.0463   2.4221        0   2.8345   7.9196  12.6476  16.4579  19.2072  20.8549  21.4322
  16.2065  15.5677  13.8236  11.1605   7.9183   4.6511   2.1208        0        0   8.2437  13.8222  17.8821  20.7047  22.3706  22.9536
  18.3063  17.5413  15.4173  12.0872   7.8667   3.4819   1.4007        0   4.9216  11.1750  16.4392  20.4567  23.2655  24.9186  25.5002
  21.5694  20.6472  18.0322  13.7742   7.9118        0        0        0   8.4615  15.0196  20.2107  24.1379  26.8748  28.4803  29.0500
  26.2799  25.2045  22.0979  16.9392   9.9605   2.4901        0   6.4469  13.8313  20.1348  25.1350  28.8911  31.4932  33.0126  33.5577
  32.7254  31.5445  28.0214  21.7978  12.4476        0        0  11.8925  20.1855  26.4361  31.1746  34.6641  37.0582  38.4481  38.9536
  41.1179  39.9833  36.4582  29.6658  17.9934        0  10.7095  20.8495  28.4652  34.1147  38.3199  41.3863  43.4807  44.6933  45.1426
  51.3900  50.5887  47.9924  42.3094  29.8266        0  21.9188  32.2239  38.5797  43.0925  46.4533  48.9284  50.6345  51.6268  52.0048
  62.9246  62.7985  62.4712  61.6677  58.9758        0  44.6544  47.4310  50.4011  53.0759  55.3225  57.0901  58.3546  59.1036  59.4008
  74.4405  75.0632  77.3219  82.8562 100.0000  73.8522  64.7920  62.3301  62.3882  63.3505  64.5316  65.6173  66.4551  66.9691  67.1855
  84.5201  85.5957  88.8382  92.4194 100.0000  86.1768  78.2499  74.6089  73.3589  73.2889  73.7176  74.2755  74.7648  75.0815  75.2255
  92.3354  93.9133 100.0000  97.9788 100.0000  92.5619  87.3586  84.4191  83.0640  82.6391  82.6845  82.9138  83.1601  83.3311  83.4161
  96.9449  97.7145 100.0000  99.4947 100.0000  96.6841  94.1634  92.5963  91.7856  91.4638  91.4118  91.4802  91.5768  91.6488  91.6875
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil B.8 : Şekil 6.8'deki Sağdaki Sonuca İlişkin Matlab Çıktısı

B.9 Şekil 6.9'daki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

    1.9012    1.8202    1.5881    1.2357    0.8163    0.4172    0.1727         0    3.0337    6.2599    9.3346   11.9425   13.8956   15.0965   15.5024
    1.9817    1.8955    1.6480    1.2691    0.8059    0.3400    0.1367         0    2.9367    6.3343    9.5660   12.2675   14.2689   15.4922   15.9045
    2.2339    2.1319    1.8388    1.3867    0.7983         0    0.0342         0    2.3776    6.5729   10.3249   13.2893   15.4170   16.6970   17.1256
    2.6896    2.5588    2.1886    1.6402    1.0004    0.3754         0    0.7318         0    7.2529   11.8687   15.1448   17.4093   18.7515   19.1983
    3.4065    3.2249    2.7164    1.9850    1.1877    0.5011         0    2.9271    5.8208   10.5682   14.7498   18.0089   20.3206   21.6996   22.1592
    4.4862    4.2174    3.4670    2.3956    1.2643    0.4414         0    5.1551    9.7865   14.4472   18.5507   21.8175   24.1612   25.5655   26.0341
    6.1029    5.6915    4.5385    2.8657    1.0325         0    1.9766    7.9060   13.7216   18.8813   23.1860   26.5462   28.9379   30.3657   30.8412
    8.5421    7.9070    6.1297    3.4963         0         0         0   10.7700   18.3110   24.1686   28.7632   32.2407   34.6757   36.1167   36.5947
   12.2515   11.2644    8.5769    4.9899    1.7404         0    7.7175   16.8618   24.5823   30.7169   35.4554   38.9753   41.4048   42.8295   43.2999
   17.9351   16.3220   11.9235    6.1459    1.9716         0   14.0078   24.3766   32.4381   38.6595   43.3643   46.7979   49.1363   50.4954   50.9421
   26.8446   24.1649   16.6491    5.6988         0   11.5004   23.9364   34.1978   42.1329   48.1173   52.5426   55.7140   57.8449   59.0727   59.4745
   41.1136   36.8438   24.8093         0    5.5162   22.0648   36.0391   46.3445   53.7775   59.1330   62.9735   65.6689   67.4550   68.4754   68.8079
   63.9221   57.2875   45.7444   27.6104         0   35.2031   51.8104   61.3630   67.4988   71.6626   74.5484   76.5321   77.8298   78.5655   78.8043
  100.0000   82.6394   73.2703   64.6973   57.9086         0   74.6363   79.7978   83.1916   85.4698   87.0248   88.0807   88.7657   89.1521   89.2773
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.9 : Şekil 6.9'daki Soldaki Sonuca İlişkin Matlab Çıktısı

B.10 Şekil 6.9'daki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

    5.3403    5.1387    4.5922    3.7805    2.8148    1.8115    0.8614         0    3.6429    7.0040    9.9272   12.3059   14.0601   15.1334   15.4999
    5.4847    5.2709    4.6902    3.8299    2.8139    1.7732    0.8129         0    3.7800    7.2168   10.1912   12.6083   14.3897   15.4791   15.8510
    5.9721    5.7178    5.0235    4.0008    2.8149    1.6420    0.6137         0    4.2547    7.8830   11.0009   13.5328   15.3965   16.5345   16.9225
    6.8831    6.5530    5.6423    4.3028    2.7827    1.3570         0    1.8439    5.3498    9.0501   12.3848   15.1115   17.1136   18.3320   18.7464
    8.3703    7.9182    6.6493    4.7569    2.6404    0.9994         0    2.0226    6.2439   10.5732   14.3644   17.4004   19.5988   20.9255   21.3744
   10.6794   10.0517    8.2426    5.4113    2.0129         0    0.2025         0    7.0234   12.6245   17.0865   20.5120   22.9398   24.3882   24.8755
   14.1655   13.3215   10.8253    6.6155         0    0.2025    0.8099         0    9.2183   15.8044   20.8319   24.6063   27.2439   28.8036   29.3257
   19.2664   18.2019   15.0928   10.2145    4.3702         0    2.8340    7.0171   14.0378   20.5323   25.8171   29.8226   32.6097   34.2486   34.7949
   26.4299   25.0897   21.1029   14.7651    7.2614         0    3.5069   11.1918   19.3756   26.4594   32.0688   36.2427   39.1082   40.7785   41.3325
   36.2167   34.5926   29.4409   20.4682    9.9060    2.4765         0   14.8627   25.8058   33.8506   39.7441   43.9575   46.7877   48.4178   48.9552
   49.2062   47.5984   41.5823   27.7506    9.4141         0         0   22.4485   35.1273   43.3841   49.0888   53.0435   55.6544   57.1439   57.6325
   65.3791   64.9961   61.5284   39.5322         0   19.4830   28.2609   39.7999   48.8647   55.4622   60.1744   63.4633   65.6322   66.8663   67.2703
   82.2989   85.4702  100.0000         0   49.1399   49.6700   53.7587   59.6222   65.0649   69.4196   72.6767   74.9959   76.5368   77.4157   77.7034
   92.8676   94.5844  100.0000   86.7222   78.0400   76.2976   77.4800   79.8633   82.3500   84.4709   86.1128   87.3023   88.0988   88.5546   88.7040
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.10 : Şekil 6.9'daki Sağdaki Sonuca İlişkin Matlab Çıktısı

B.11 Şekil 6.10'daki Soldaki Sonuca İlişkin Matlab Çıktısı

```
x =

   10.3150    9.7944    8.3232    6.1477    3.7616    1.8708    0.7471         0    0.3603    0.7664    1.2223    1.6942    2.1177    2.4141    2.5222
   10.7992   10.2442    8.6546    6.2387    3.5062    1.4839    0.5578         0    0.3367    0.7403    1.2123    1.7159    2.1784    2.5061    2.6261
   12.3360   11.6933    9.7848    6.6287    2.5337         0         0         0    0.2453    0.6441    1.1682    1.7754    2.3701    2.8038    2.9635
   15.0929   14.3679   12.1319    7.9408         0    2.3526    2.3020    1.4518         0    0.4209    1.0389    1.8444    2.7189    3.3734    3.6142
   19.2244   18.5069   16.3978   12.9843    8.7947    7.0953    5.3940    3.5005    1.5714         0    0.7204    1.8415    3.2840    4.3547    4.7400
   24.7041   23.9831   21.9220   18.7683   15.0712   11.8182    8.6627    5.5758    2.7818    0.6955         0    1.5151    4.2173    6.0192    6.6302
   31.5306   30.7393   28.4859   25.0513   20.8674   16.4159   11.8436    7.3470    3.2802         0    1.0317         0    6.0474    8.8726    9.7358
   39.8418   38.8963   36.1754   32.0358   26.8918   21.1043   14.9281    8.6771    2.9893         0    4.1264    7.1320   11.0956   13.6857   14.5610
   49.9515   48.7712   45.2306   39.9779   33.5209   26.1519   18.0682    9.4351         0    3.0808    8.3397   13.3026   17.5130   20.2109   21.1296
   62.3432   60.9580   55.9524   49.0830   41.0273   31.8877   21.7414   10.9896         0    3.9821   12.8465   20.2218   25.4383   28.5128   29.5279
   77.4485   76.7310   68.5031   59.3413   49.5891   38.6088   26.0068   12.7778    3.1944         0   18.8398   29.2959   35.5009   38.8720   39.9488
   93.9597  100.0000   81.9655   70.1661   59.3579   46.9354   30.8893   10.9168         0   18.8142   33.2142   42.6179   48.3936   51.5235   52.5170
   98.3893  100.0000   89.1793   79.9845   70.7268   58.8753   39.6927         0   26.4451   42.0413   52.5828   59.5653   63.9290   66.3102   67.0671
   99.5973  100.0000   94.7598   89.8572   84.6818   78.1409   69.0050   58.1859         0   70.3222   75.5089   79.1300   81.4451   82.7207   83.1274
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.11 : Şekil 6.10'daki Soldaki Sonuca İlişkin Matlab Çıktısı

B.12 Şekil 6.10'daki Sağdaki Sonuca İlişkin Matlab Çıktısı

```
x =

   12.1633   11.7338   10.5518    8.7402    6.4855    4.0322    1.7187         0    3.6082    7.4376   11.0802   14.1608   16.4584   17.8633   18.3516
   12.4722   12.0257   10.7933    8.9036    6.5439    3.9396    1.4146         0    3.4880    7.5139   11.3373   14.5228   16.8716   18.2983   18.7926
   13.4992   12.9962   11.6009    9.4661    6.7997    3.7452         0         0    2.8165    7.7684   12.1984   15.6805   18.1614   19.6417   20.1507
   15.3655   14.7577   13.0620   10.4943    7.4016    4.2261    1.5112    0.3778         0    8.5198   13.9720   17.7967   20.4044   21.9312   22.4527
   18.2924   17.5136   15.3165   11.9883    8.0480    4.2285    1.4349         0    6.0344   12.3105   17.3347   21.0845   23.6783   25.1999   25.7202
   22.6369   21.6047   18.6333   14.0441    8.5431    3.1929         0    5.9255   11.8047   17.3198   21.9298   25.4797   27.9719   29.4423   29.9474
   28.9250   27.5643   23.5109   16.9721    8.8679         0    5.4746   11.8794   17.9109   23.1964   27.5397   30.8818   33.2329   34.6220   35.1009
   37.8351   36.1598   30.8297   21.4372    9.9479         0   10.0056   18.1833   24.7307   29.9747   34.1034   37.2227   39.4010   40.6840   41.1273
   50.0198   48.3677   42.1792   27.9801    9.4820         0   16.3483   26.0915   32.8195   37.8267   41.6291   44.4533   46.4103   47.5591   47.9567
   65.4554   65.0844   61.5202   38.8119         0   17.9151   29.2775   36.9878   42.5946   46.8430   50.0876   52.5024   54.1773   55.1609   55.5025
   81.6014   84.9803  100.0000         0   43.7326   42.3703   45.8396   49.9614   53.6962   56.8262   59.3351   61.2477   62.5906   63.3838   63.6608
   90.9747   93.2325  100.0000   80.4290   66.8076   61.9814   61.7312   63.2993   65.3753   67.3991   69.1447   70.5265   71.5161   72.1064   72.3141
   95.8252   96.9739  100.0000   89.1596   81.0802   77.0057   75.7904   76.1113   77.0856   78.2266   79.2924   80.1706   80.8130   81.2004   81.3375
   98.3752   98.8373  100.0000   95.1260   91.3430   89.1639   88.3041   88.2589   88.6160   89.1148   89.6117   90.0340   90.3479   90.5387   90.6066
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil B.12 : Şekil 6.10'daki Sağdaki Sonuca İlişkin Matlab Çıktısı

Ek C DEĞİŞİK KOMBİNASYONLARA İLİŞKİN SONUÇLAR

C.1 Aynı Kat ve Farklı Konumlar İçin Rasgele Sonuçlar

x =

5.8131	5.6152	5.0585	4.2085	3.1658	2.0515	0.9786	0	3.6300	6.9892	9.9099	12.2800	14.0220	15.0846	15.4488
5.9835	5.7753	5.1884	4.2914	3.1919	2.0249	0.9294	0	3.7604	7.2002	10.1741	12.5808	14.3471	15.4236	15.7924
6.5297	6.2890	5.6068	4.5599	3.2742	1.9208	0.7125	0	4.2038	7.8653	10.9903	13.5041	15.3423	16.4604	16.8426
7.5169	7.2193	6.3688	5.0514	3.4138	1.6667	0	1.2962	5.1819	9.0550	12.4025	15.0850	17.0382	18.2228	18.6268
9.0592	8.6779	7.5775	5.8483	3.6542	1.3302	0	0	6.1644	10.7581	14.4641	17.3775	19.4833	20.7559	21.1886
11.3247	10.8317	9.3956	7.0966	4.0180	0	0.9720	3.8864	8.7093	13.3367	17.3030	20.4599	22.7425	24.1191	24.5858
14.5380	13.9058	12.0582	9.1127	5.3169	1.3292	0	5.8591	11.4406	16.5640	20.9359	24.3995	26.8889	28.3830	28.8872
18.9791	18.1738	15.8019	11.9678	6.8029	0	2.0264	8.1038	14.6212	20.5306	25.4627	29.2966	32.0129	33.6277	34.1693
24.9973	23.9886	20.9922	16.1436	9.9232	3.5274	0	9.9030	18.4011	25.4631	31.0739	35.2957	38.2218	39.9376	40.5084
33.0031	31.7740	28.0214	21.6821	13.2144	4.1855	0	13.1020	23.6091	31.8365	38.0618	42.5765	45.6260	47.3856	47.9656
43.4420	42.0688	37.6260	29.3415	17.0637	0	4.7233	18.8916	31.0897	40.2031	46.7494	51.3105	54.3073	56.0072	56.5624
56.6079	55.4223	51.0642	40.9891	25.6971	6.4243	0	26.6476	41.6493	51.1295	57.4136	61.5992	64.2753	65.7691	66.2529
72.1319	71.9411	70.2143	57.8501	38.3099	0	27.6610	46.0468	57.7261	65.2467	70.1698	73.3903	75.4180	76.5380	76.8987
88.0299	89.9928	100.0000	81.8856	69.6919	58.5720	0	72.1506	77.9589	81.9579	84.6248	86.3701	87.4636	88.0649	88.2579
100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000

Şekil C.1 : Soldan 3 Konumunda Rasgele 1 Kat Sonucu

x =

9.1475	8.8639	8.0588	6.8025	5.2085	3.4305	1.6488	0	2.5381	4.9794	7.2109	9.1044	10.5437	11.4415	11.7494
9.3882	9.0940	8.2575	6.9494	5.2844	3.4226	1.5788	0	2.5846	5.0810	7.3754	9.3261	10.8084	11.7321	12.0487
10.1529	9.8259	8.8925	7.4253	5.5377	3.3859	1.2412	0	2.7165	5.3799	7.8774	10.0087	11.6232	12.6255	12.9680
11.5057	11.1231	10.0262	8.2940	6.0363	3.3333	0	0.7249	2.8984	5.8400	8.7390	11.2002	13.0417	14.1739	14.5588
13.5569	13.0929	11.7596	9.6608	6.9625	3.9046	0.9762	0	2.3096	6.3380	10.0314	13.0034	15.1601	16.4649	16.9050
16.4689	15.8906	14.2233	11.6004	8.2311	4.3392	0	0	0	7.1660	12.0384	15.6133	18.1208	19.6156	20.1164
20.4712	19.7371	17.6089	14.2606	10.0065	5.2158	0	0	4.4614	10.2823	15.3353	19.2817	22.0844	23.7553	24.3138
25.8791	24.9398	22.1828	17.8031	12.3040	6.5125	1.6281	0	7.5601	14.1605	19.7313	24.0848	27.1699	29.0021	29.6125
33.1088	31.9264	28.3515	22.4443	14.8812	6.8969	0	4.1436	11.6144	19.0626	25.3367	30.1473	33.4985	35.4656	36.1168
42.6547	41.2769	36.8293	28.7246	17.8699	6.1917	0	4.9581	15.6875	25.1330	32.3985	37.6605	41.2021	43.2406	43.9090
54.9175	53.6755	48.9468	37.7430	21.6760	0	0	0	21.0408	33.3782	41.4572	46.8865	50.4008	52.3817	53.0249
69.6368	69.5461	67.5285	51.6173	31.0887	7.7722	0	19.4592	35.0943	45.8772	53.1602	58.0213	61.1260	62.8575	63.4164
84.5213	87.3362	100.0000	70.1055	43.2878	0	26.7185	42.7412	53.9975	61.8728	67.2810	70.9078	73.2194	74.5038	74.9173
93.7690	95.2763	100.0000	85.5156	71.9564	59.0222	0	70.7885	76.2800	80.3335	83.1805	85.1065	86.3368	87.0201	87.2399
100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000

Şekil C.2 : Soldan 3 Konumunda Rasgele 2 Kat Sonucu

x =

11.9752	11.6010	10.5593	8.9351	6.8651	4.5374	2.1842	0	3.3974	6.5740	9.3761	11.6814	13.3941	14.4460	14.8090
12.2497	11.8642	10.7882	9.1072	6.9567	4.5275	2.0914	0	3.5021	6.7518	9.6117	11.9624	13.7074	14.7785	15.1478
13.1476	12.7258	11.5422	9.6848	7.2822	4.4996	1.6478	0	3.8507	7.3056	10.3387	12.8281	14.6716	15.8006	16.1892
14.7429	14.2581	12.8914	10.7454	7.9452	4.5200	0	1.1473	4.5859	8.2671	11.5913	14.3182	16.3267	17.5510	17.9707
17.1640	16.5834	14.9436	12.4003	9.1937	5.6200	2.1118	0	5.0695	9.5712	13.4226	16.5049	18.7419	20.0935	20.5541
20.6075	19.8830	17.8264	14.6617	10.7705	6.6346	2.8209	0	6.1115	11.5109	16.0039	19.5143	22.0182	23.5143	24.0206
25.3700	24.4361	21.7503	17.5972	12.5564	7.3083	2.5323	0	7.8559	14.3421	19.5483	23.5079	26.2776	27.9123	28.4613
31.8838	30.6713	27.0824	21.3751	14.5195	7.4959	0	2.7409	10.9599	18.4381	24.3202	28.6690	31.6480	33.3836	33.9623
40.7237	39.2245	34.4844	26.2646	16.6278	8.1486	2.0372	0	14.7947	24.1153	30.6068	35.1784	38.2385	40.0004	40.5841
52.4838	50.9741	45.3296	32.5444	17.5625	6.4278	0	12.7457	24.0936	32.6074	38.7957	43.1795	46.1057	47.7849	48.3397
67.2079	66.8284	63.2925	41.0049	14.6418	0	15.2541	26.8832	36.2169	43.4119	48.7735	52.6203	55.2013	56.6848	57.1749
82.6577	85.8239	100.0000	53.5346	0	21.4220	34.1299	43.3102	50.4703	56.0389	60.2531	63.3122	65.3790	66.5711	66.9654
91.7599	93.8067	100.0000	0	50.9430	51.5565	56.5301	61.7523	66.3087	70.0117	72.8779	74.9855	76.4199	77.2505	77.5256
96.7615	97.6421	100.0000	88.0537	79.0817	77.3296	78.6794	80.8572	82.9959	84.8162	86.2554	87.3252	88.0576	88.4831	88.6241
100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000

Şekil C.3 : Soldan 3 Konumunda Rasgele 3 Kat Sonucu

x =

12.1633	11.7338	10.5518	8.7402	6.4855	4.0322	1.7187	0	3.6082	7.4376	11.0802	14.1608	16.4584	17.8633	18.3516
12.4722	12.0257	10.7933	8.9036	6.5439	3.9396	1.4146	0	3.4880	7.5139	11.3373	14.5228	16.8716	18.2983	18.7926
13.4992	12.9962	11.6009	9.4661	6.7997	3.7452	0	0	2.8165	7.7684	12.1984	15.6805	18.1614	19.6417	20.1507
15.3655	14.7577	13.0620	10.4943	7.4016	4.2261	1.5112	0.3778	0	8.5198	13.9720	17.7967	20.4044	21.9312	22.4527
18.2924	17.5136	15.3165	11.9883	8.0480	4.2285	1.4349	0	6.0344	12.3105	17.3347	21.0845	23.6783	25.1999	25.7202
22.6369	21.6047	18.6333	14.0441	8.5431	3.1929	0	5.9255	11.8047	17.3198	21.9298	25.4797	27.9719	29.4423	29.9474
28.9250	27.5643	23.5109	16.9721	8.8679	0	5.4746	11.8794	17.9109	23.1964	27.5397	30.8818	33.2329	34.6220	35.1009
37.8351	36.1598	30.8297	21.4372	9.9479	0	10.0056	18.1833	24.7307	29.9747	34.1034	37.2227	39.4010	40.6840	41.1273
50.0198	48.3677	42.1792	27.9801	9.4820	0	16.3483	26.0915	32.8195	37.8267	41.6291	44.4533	46.4103	47.5591	47.9567
65.4554	65.0844	61.5202	38.8119	0	17.9151	29.2775	36.9878	42.5946	46.8430	50.0876	52.5024	54.1773	55.1609	55.5025
81.6014	84.9803	100.0000	0	43.7326	42.3703	45.8396	49.9614	53.6962	56.8262	59.3351	61.2477	62.5906	63.3638	63.6608
90.9747	93.2325	100.0000	80.4290	66.8076	61.9814	61.7312	63.2993	65.3753	67.3991	69.1447	70.5265	71.5161	72.1064	72.3141
95.8252	96.9739	100.0000	89.1596	81.0802	77.0057	75.7904	76.1113	77.0856	78.2266	79.2924	80.1706	80.8130	81.2004	81.3375
98.3752	98.8373	100.0000	95.1260	91.3430	89.1639	88.3041	88.2589	88.6160	89.1148	89.6117	90.0340	90.3479	90.5387	90.6066
100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000

Şekil C.4: Soldan 3 Konumunda Rasgele 4 Kat Sonucu

C.2 Farklı Kat ve Aynı Konuma İlişkin Rasgele Sonuçlar

```
x =

  12.4574  12.1424  11.2604   9.8466   7.9497   5.6252   2.9377        0   1.1067   2.3380   3.6691   4.9513   6.0107   6.7050   6.9485
  12.6749  12.3560  11.4630  10.0343   8.1219   5.7793   3.0514        0   1.0433   2.2863   3.6906   5.0590   6.1890   6.9274   7.1857
  13.3853  13.0517  12.1189  10.6363   8.6712   6.2848   3.4765        0   0.7787   2.0707   3.7443   5.4000   6.7528   7.6267   7.9301
  14.6173  14.2545  13.2417  11.6508   9.5876   7.1765   4.5550   1.9321        0   1.4710   3.8119   6.0387   7.7893   8.8930   9.2713
  16.4297  16.0156  14.8598  13.0674  10.7963   8.2395   5.6123   3.1643   1.2246        0   3.9900   7.1480   9.4663  10.8809  11.3586
  18.9270  18.4281  17.0332  14.8934  12.2353   9.3324   6.4644   3.8746   1.7296        0   4.9961   9.0913  12.0397  13.8019  14.3898
  22.2839  21.6502  19.8737  17.1711  13.8661  10.3511   7.0125   4.1259   1.8149        0   6.8990  12.1752  15.7917  17.8929  18.5848
  26.7788  25.9348  23.5685  19.9902  15.6582  11.1578   7.0855   3.7896   1.4011        0  10.4200  16.9123  21.0511  23.3890  24.1507
  32.8455  31.6703  28.4118  23.5098  17.5772  11.5073   6.3648   2.5386        0   9.4971  17.8635  23.9957  28.1032  30.4571  31.2268
  41.1638  39.4291  34.8457  28.0164  19.6011  10.9087   4.3184        0  11.0096  20.1213  27.5356  33.0969  36.9010  39.1054  39.8295
  52.8735  49.9902  43.4846  34.0768  21.8802   8.1972        0  13.1720  23.9147  32.4388  39.0550  43.9485  47.2910  49.2306  49.8683
  70.2957  64.1416  54.9975  42.9043  25.6338        0  15.2414  28.7719  39.0355  46.6603  52.2924  56.3451  59.0778  60.6547  61.1721
 100.0000  81.2650  69.4421  56.8971  37.7471        0  32.1929  47.6370  56.7930  62.8712  67.1053  70.0574  72.0154  73.1364  73.5030
 100.0000  91.4670  84.6001  77.4887  68.4558  58.5871        0  72.7891  77.6267  80.9241  83.1978  84.7608  85.7871  86.3714  86.5619
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil C.5: Soldan 1 Konumunda Rasgele 2 Kat Sonucu

```
x =

   6.6029   6.3774   5.7357   4.7465   3.5150   2.1795   0.9281        0   3.1851   6.1475   8.7342  10.8411  12.3944  13.3429  13.6743
   6.8074   6.5704   5.8962   4.8572   3.5597   2.1332   0.7653        0   3.2880   6.3212   8.9552  11.0953  12.6716  13.6341  13.9701
   7.4532   7.1800   6.4040   5.2129   3.7241   2.0242        0        0   3.6327   6.8738   9.6438  11.8822  13.5289  14.5342  14.8847
   8.6108   8.2708   7.3085   5.8523   4.0908   2.2363   0.5591        0   4.3556   7.8767  10.8372  13.2298  14.9933  16.0717  16.4470
  10.4110   9.9609   8.6879   6.7825   4.5416   2.2675        0   2.3659   5.8983   9.4186  12.5714  15.1747  17.1086  18.2944  18.7062
  13.0722  12.4501  10.6803   8.0343   5.0173   2.2896        0   3.5563   7.4374  11.3058  14.8273  17.7576  19.9376  21.2736  21.7355
  16.9382  16.0633  13.5301   9.6442   5.1970   1.8716        0   4.4122   8.9729  13.5178  17.6472  21.0593  23.5770  25.1098  25.6358
  22.5156  21.3122  17.7155  11.8045   4.2504        0   1.2783   5.1099  10.5086  16.1238  21.1578  25.2254  28.1686  29.9367  30.5369
  30.4641  28.9344  24.2006  15.6003        0   2.6277        0   4.2321  11.8133  19.2909  25.6101  30.4876  33.9047  35.9165  36.5904
  41.4410  39.7437  34.5401  26.3913  16.4091  10.5097   4.6018        0  13.2093  23.5987  31.4820  37.1848  41.0188  43.2211  43.9481
  55.7881  54.0457  47.8148  39.0093  28.7318  18.3982   7.8970        0  17.4152  30.3974  39.5151  45.7289  49.7410  51.9898  52.7221
  73.6029  72.8272  63.6566  53.0939  41.1072  26.4526   8.5874        0  26.0460  41.0486  50.4368  56.4568  60.2075  62.2667  62.9296
  92.9608 100.0000  80.8863  68.5986  56.1482  37.7163        0  27.5336  45.7141  57.3051  64.7155  69.4412  72.3516  73.9340  74.4402
  98.2402 100.0000  91.2878  84.2640  77.1692        0  58.1707  64.4182  71.9676  77.7366  81.6718  84.2328  85.8155  86.6746  86.9488
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil C.6: Soldan 2 Konumunda Rasgele 2 Kat Sonucu

```
x =

   7.0781   6.8850   6.3383   5.4788   4.3619   3.0435   1.5720        0   1.5716   3.3168   5.1925   6.9790   8.4358   9.3804   9.7095
   7.2374   7.0378   6.4732   5.5895   4.4489   3.1111   1.6186        0   1.4834   3.2491   5.2334   7.1390   8.6864   9.6841  10.0304
   7.7467   7.5250   6.8997   5.9347   4.7159   3.3230   1.7875        0   1.1110   2.9592   5.3478   7.6505   9.4791  10.6351  11.0319
   8.6520   8.3861   7.6398   6.5119   5.1407   3.6671   2.2044   0.8664        0   2.1261   5.5431   8.6295  10.9365  12.3410  12.8143
  10.0441   9.6999   8.7369   7.3122   5.6524   3.9898   2.4908   1.2593   0.4198        0   6.0641  10.3807  13.2883  14.9737  15.5299
  12.0835  11.6070  10.2735   8.3293   6.1529   4.1392   2.5043   1.2576   0.4194        0   8.3276  13.5337  16.8534  18.7308  19.3441
  15.0389  14.3486  12.4014   9.5630   6.4791   3.9021   2.1252   0.8457        0   7.1778  13.7065  18.5652  21.8517  23.7472  24.3701
  19.3434  18.3284  15.4044  11.0300   6.2899   2.8599   1.2463        0   8.1328  15.0001  20.7484  25.1606  28.2312  30.0311  30.6269
  25.6520  24.2022  19.8457  12.8540   4.7860        0        0   9.7200  17.5276  23.9356  29.1190  33.0887  35.8716  37.5141  38.0600
  34.8404  32.9715  26.9137  15.7498        0   7.1329  13.5274  21.3494  28.3177  34.0898  38.6957  42.1949  44.6433  46.0891  46.5701
  47.7520  45.9221  39.0827  23.2301        0  15.0031  25.6252  33.8292  40.2992  45.4041  49.3722  52.3441  54.4090  55.6247  56.0289
  64.3140  63.8772  60.2619  38.0870        0  27.2530  40.1390  48.0397  53.6417  57.8499  61.0387  63.3934  65.0167  65.9687  66.2847
  81.7440  85.0086 100.0000        0  50.4013  53.8689  59.6360  64.5464  68.3747  71.3110  73.5348  75.1689  76.2905  76.9463  77.1638
  92.6426  94.4128 100.0000  86.9339  78.8800  78.1845  79.9886  82.1333  83.9977  85.4818  86.6178  87.4536  88.0266  88.3613  88.4721
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil C.7: Soldan 3 Konumunda Rasgele 2 Kat Sonucu

```
x =

   2.5844   2.4726   2.1812   1.7612   1.2815   0.8080   0.3805        0   5.5071  10.4227  14.5213  17.7295  20.0244  21.3994  21.8673
   2.6481   2.5290   2.2170   1.7689   1.2627   0.7760   0.3537        0   5.7954  10.8193  14.9507  18.1676  20.4641  21.8389  22.3066
   2.8805   2.7364   2.3541   1.8080   1.2072   0.6703   0.2560        0   6.8437  12.0907  16.2726  19.5004  21.7977  23.1711  23.6382
   3.3352   3.1427   2.6231   1.8791   1.0743   0.4361        0   4.1075   9.4752  14.4084  18.5251  21.7369  24.0264  25.3944  25.8599
   4.1140   3.8404   3.0890   1.9931   0.7669        0   1.7373   6.9455  12.5258  17.5216  21.6578  24.8678  27.1469  28.5047  28.9666
   5.3851   4.9847   3.8764   2.2256        0        0        0   9.4001  16.1433  21.4722  25.6901  28.9006  31.1575  32.4951  32.9494
   7.4089   6.8099   5.1880   3.0265   1.0499        0   6.6569  14.4982  21.1559  26.5099  30.7025  33.8567  36.0559  37.3526  37.7924
  10.5894   9.6358   7.0239   3.6354   1.1713        0  12.1200  20.7645  27.4516  32.6842  36.7253  39.7378  41.8251  43.0511  43.4666
  15.6435  14.1022   9.6251   3.3151        0  11.0735  21.0475  28.9715  35.1809  40.0252  43.7493  46.5147  48.4252  49.5451  49.9247
  23.7542  21.4915  14.0523        0  12.2686  23.2387  32.0126  38.8764  44.2549  48.4627  51.7061  54.1189  55.7874  56.7658  57.0978
  36.3714  34.0486  25.0897        0  25.8312  37.5914  44.8757  50.2510  54.4809  57.8432  60.4704  62.4429  63.8146  64.6214  64.8957
  53.6205  53.2357        0  51.4293  53.4595  56.4123  59.6369  62.7572  65.5588  67.9409  69.8703  71.3476  72.3859  73.0001  73.2096
  71.6302  73.0136  79.2666 100.0000  80.1611  74.9548  74.4941  75.5719  77.0441  78.4778  79.7079  80.6761  81.3661  81.7771  81.9178
  86.8678  87.9196  91.7966 100.0000  92.2275  88.7478  87.8073  87.9855  88.5606  89.2099  89.7986  90.2734  90.6160  90.8212  90.8916
 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
```

Şekil C.8: Soldan 4 Konumunda Rasgele 2 Kat Sonucu

```
x =

    7.0048    6.7834    6.1593    5.2061    4.0262    2.7220    1.3682         0    1.7301    3.6451    5.6946    7.6373    9.2135   10.2293   10.5927
    7.1999    6.9660    6.3073    5.3061    4.0782    2.7405    1.3729         0    1.6310    3.5656    5.7301    7.7975    9.4667   10.5362   10.9170
    7.8236    7.5489    6.7764    5.6156    4.2278    2.7816    1.3806         0    1.2191    3.2394    5.8378    8.3236   10.2826   11.5120   11.9436
    8.9568    8.6046    7.6127    6.1353    4.4241    2.7705    1.3655    0.3414         0    2.3207    6.0336    9.3437   11.7894   13.2647   13.7714
   10.7538   10.2752    8.9137    6.8727    4.5522    2.5049    0.9676         0    0.6605         0    6.6078   11.1933   14.2256   15.9634   16.5469
   13.4670   12.8043   10.8740    7.8755    4.3986    1.7259         0    1.9662    2.6397         0    9.1776   14.5587   17.9121   19.7927   20.4191
   17.4656   16.5769   13.8842    9.3447    3.4358         0    2.3043    5.2185    7.9257   10.3527   15.5118   19.9103   23.0239   24.8507   25.4694
   23.2025   22.1309   18.7247   12.1744         0    0.9986    3.9917    8.6654   13.4735   17.9455   22.5694   26.5019   29.3728   31.0908   31.6791
   31.0454   29.9982   26.6942   20.6222   11.2203         0    3.9900   11.9619   19.3338   25.3542   30.2783   34.1085   36.8243   38.4351   38.9862
   40.9481   40.1027   37.4173   32.3914   24.2558   12.7023         0   15.8419   26.5203   33.8258   39.0409   42.7840   45.3320   46.8151   47.3192
   52.5110   52.0295   50.4677   47.2619   40.7046   26.5516         0   24.8691   37.0555   44.3562   49.2383   52.6131   54.8606   56.1533   56.5905
   65.0114   65.0227   65.1511   65.4767   64.7452         0   41.8937   46.5641   52.4549   57.2780   60.9111   63.5345   65.3069   66.3304   66.6768
   77.4705   77.8887   79.6297   84.7441  100.0000   77.9951   68.2055   67.0264   68.9053   71.3688   73.5695   75.2805   76.4743   77.1730   77.4108
   89.0817   89.4258   90.7314   93.8689  100.0000   90.9744   85.9012   84.4227   84.7603   85.7091   86.7027   87.5271   88.1195   88.4709   88.5910
  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000  100.0000
```

Şekil C.9: Soldan 5 Konumunda Rasgele 2 Kat Sonucu

ÖZGEÇMİŞ

Cenker Elbirler 1981 yılında İstanbul'da doğdu. Orta öğretimini Özel Alanya Hamdullah Emin Paşa Lisesi'nde birincilik derecesi ile tamamladı. 2005 yılında Yıldız Teknik Üniversitesi'nde Elektrik Mühendisi diploması aldı. 2005 yılında bir sene süre ile taahhüt alanında faaliyet gösteren Net Mühendislik AŞ'de proje mühendisi olarak görev aldı. 2006 yılından beri Alanya'da bir yerel gazete olan Haber Alanya Gazetesi'nde köşe yazarlığı yapmaktadır.