



|   |   |  |  |
|---|---|--|--|
|  | <b>SAKARYA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ DERGİSİ</b><br><i>SAKARYA UNIVERSITY JOURNAL OF SCIENCE</i>                 |  |  |
|   | e-ISSN: 2147-835X<br>Dergi sayfası: <a href="http://dergipark.gov.tr/saufenbilder">http://dergipark.gov.tr/saufenbilder</a> |  |  |
|   | <u>Geliş/Received</u><br>30.04.2016<br><u>Kabul/Accepted</u><br>09.05.2017  | <u>Doi</u><br>10.16984/saufenbilder.221361 |  |

## Elektromanyetik alan teorisinde nümerik tekniklerin görsel programlama yöntemleriyle uygulanması

H.Hakan Genç<sup>1</sup>, Metin Varan<sup>\*2</sup>

### ÖZ

Elektrik-Elektronik mühendisliğinin iki alt teorisinden biri olan alan teorisi lisans öğrencilerinin algılamada güçlük çektiği bir konudur. Lisans döneminde elektromanyetik alanlar kapsamında alan teorisi, parçalı diferansiyel denklemler ve integral yöntemlerinin kullanıldığı analitik yöntemlerle anlatılmaktadır. Alan problemlerinin matematiksel modelinin çıkartılması esasına dayalı olan analitik çözüm yöntemleri lisans öğrencilerinin matematik ve fizik altyapılarına bağlı olarak algılama güçlüklerinin yaşanmasına neden olabilmektedir. Basit modellerde uygulanabilir olan analitik yöntemler daha karmaşık modellerde uygulanabilirliğini yitirmektedir. Bu durumda da daha karmaşık denklemleri çözmek için nümerik yöntemlere başvurulması uzunca zamandır kullanılan bir yöntemdir. Bu çalışmada, alan teorisi kapsamında lisans üstü öğrenim düzeyinde kullanılan bazı nümerik yöntemlerin web tabanlı grafik arayüzlü uygulamalarının, lisans öğrencilerinin bilgisayar programlama kabiliyetleri de göz önünde bulundurularak öğrencilerce geliştirilmesi ve algılama güçlüğü yaşanan alan teorisi çözümlerinin kalıcı öğrenimi sağlayabilmesi hedeflenmiştir.

**Anahtar Kelimeler:** Elektromanyetik alan teorisi, nümerik çözüm, görsel programlama, java tabanlı benzetim.

### Implementation of visual programming methods for numerical techniques used in electromagnetic field theory

#### ABSTRACT

Field theory is one of the two sub-field theories in electrical and electronics engineering that for creates difficulties for undergraduate students. In undergraduate period, field theory has been taught under the theory of electromagnetic fields by which describes using partial differential equations and integral methods. Analytical methods for solution of field problems on the basis of a mathematical model may result the understanding difficulties for undergraduate students due to their mathematical and physical infrastructure. The analytical methods which can be applied in simple model lose their applicability to more complex models. In this case, the numerical methods are used to solve more complex equations. In this study, by preparing some field theory's web-based graphical user interface numerical methods of applications it has been aimed to increase learning levels of field theory problems for undergraduate and graduate students while taking in mind their computer programming capabilities.

**Keywords:** Electromagnetic field theory, numerical solutions, visual programming techniques, java based simulation

\* Sorumlu Yazar / Corresponding Author

<sup>1</sup> Yük.Lisans, Sakarya Univ. Fen Bilimleri Enstitüsü - Elektrik ve Elektronik Mühendisliği, Sakarya, 54187, Türkiye  
hakan.genc@ogr.sakarya.edu.tr

<sup>2</sup> Yrd. Doç. Dr., Sakarya Univ. Teknoloji Fakültesi - Elektrik ve Elektronik Mühendisliği, Sakarya, 54187, Türkiye  
metin.varan@sakarya.edu.tr

## 1. GİRİŞ

Diğer pek çok disiplinde olduğu üzere, elektrik elektronik mühendisliği alanında da ileri düzey mühendislik matematiği konuları bulunmaktadır. "Devre" ve "alan" teorisi olarak iki ana alt disiplinden oluşan elektrik-elektronik mühendisliği elektromanyetik alanlar teorisi kapsamında çeşitli analitik yöntemler (değişkenlerin ayrılması, seri açılımları, vb.) teorik olarak lisans öğrencilerine sunulmaktadır. Bu konuların anlaşılmasında ve somutlaştırılmasında güçlükler yaşanmaktadır. Daha karmaşık alan problemlerinin çözümünde ise bir dizi nümerik teknik (sonlu fark yöntemi, moment yöntemleri, sonlu farkı yöntemi, iletim hatları yöntemi, monte carlo yöntemi, vb.) kullanılmaktadır. Nümerik tekniklerin avantajı bilgisayar programlama yöntemleriyle uygulanabilir olmalarıdır ve bu teknikler uzunca bir süredir bilgisayar programlama dilleri (FORTRAN, BASIC, C, vb.) kullanılarak akademik düzeyde incelenmektedir [1]. Bilgisayarların fizikte kullanımı nümerik analizler, sembolik işletme, görselleştirme, simülasyon ve veri analizi olarak kategorize edilebilir. Nümerik analiz, iyi tanımlanmış matematiksel problemlerin, sembolik çözümler yerine nümerik çözümlerini ifade eder [2]. Nümerik teknikler deterministik olan ve deterministik olmayan şeklinde ikiye ayrılmaktadır. Monte Carlo tekniği deterministik olmayan bir nümerik tekniktir ve elektrik-elektronik mühendisliğinde kullanılan çeşitli üst seviye devre ve alan analiz programlarında da (PSPICE, vb.) kullanılmaktadır. Monte Carlo tekniği bir sistemin rastgelelik durumuna göre davranışını ortaya koyan bir teknik olarak [3], bilgisayar programcılığı açısından rastgele sayı analizini de içermektedir.

Lisans düzeyinde eğitim gören öğrencilerin matematik ve fizik altyapılarına bağlı olarak farklı öğrenme seviyelerinin yaşandığı bu gibi derslerde, bizzat öğrencinin daha modern bir öğrenme süreciyle uygulama geliştirme ve karşılaşılan problemleri bilgisayar programlama teknikleriyle birleştirerek analiz etme fırsatı elde etmeleri sağlanır. Bu bağlamda tüm mühendislik disiplinlerinde uygulama alanı olan bilgisayar programcılığı, elektrik-elektronik mühendisliğinde de farklı seviyelerde üst seviye ve alt seviye programlama dilleri ya da hazır araçlar içeren paket programlar (MATLAB, vb.)

vasıtalarıyla kullanılmaktadır. EM problemlerinin ve alan teorisinin bilgisayar yazılımı tabanlı materyallerle desteklenmesiyle ilgili çeşitli çalışmalar da mevcuttur [4,5]. Bu çalışmanın öncelikli amacı, bazı alan problemlerinin nümerik teknikler kullanılarak, grafik arayüz tabanlı üst seviye bilgisayar programlama dilleri yardımıyla incelenmesini ve öğrencilere bu analiz tekniklerinin bilgisayar programlama yöntemleriyle öğretimini sağlayacak bir uygulamanın geliştirilmesidir. Öğrencilere bu tür grafik arayüzlü programlama teknikleriyle, alan problemlerinin analiz yöntemlerinin öğretilmesinin kalıcı öğrenmeye katkı sağlayacağı düşünülmüştür [6,7]. Çalışmada, Sabit Rasgele Yürüyüş ve Kayan Rasgele Yürüyüş olmak üzere iki önemli Monte Carlo yönteminin Java AWT ile web tabanlı grafik arayüzlü olarak programlaması ayrıntılı olarak sunulmuştur. Programlama platformu olarak Eclipse Java seçilmiş ve Java programlama dilinin Graphic AWT (Abstract Window Toolkit) aracı kullanılmıştır. Bu programlama dili 2000'li yılların başından günümüze çeşitli mühendislik çalışmalarında yaygın olarak kullanılmaktadır. Java programları rahatlıkla web ortamına uyarlanabilmektedir. Java, ses, video ve diğer medya araçlarıyla etkileşim içinde olan grafik kullanıcı arayüzü geliştirmeye imkan sağlayan pek çok güçlü kütüphane sunar [2].

## 2. MONTE CARLO ANALİZİ METODU

Monte Carlo analizi metodu diğer nümerik yöntemlerden farklı olarak rastgelelik ve olasılık durumuna göre matematiksel ve fiziksel analiz ortaya koyan bir nümerik analiz metodudur. Problemlerin rastgele sayı dizileri kullanılarak yaklaşık olarak çözülmesi prensibine dayanır. Matematiksel problemleri, olasılıksal bir benzeşim ve ardından bu benzeşime deneysel örnekleme yöntemiyle yaklaşık cevaplar bularak çözmektedir. Bu haliyle klasik matematiksel tekniklerden ziyade, daha çok fiziksel deneylerde izlenen çözüm modeline benzerliği vardır [3].

Monte Carlo analizi metodu simülasyon ve örnekleme olarak iki şekilde uygulanır. Simülasyon en kısa haliyle rastgele olarak gerçekleşen bir dizi fiziksel olayın matematiksel olarak taklit edilmesidir. Rastgele yürüyüş (random walk) tekniğiyle, bir nükleer reaktör içinde nötron parçacığının gerçekleştirdiği zigzag hareketlerinin simülasyonu buna tipik bir örnek

olarak gösterilebilir. Örneklemeye ise, çok geniş bir eleman kümesinin özelliklerinin, rastgele seçilmiş olan küçük bir kısmının incelenerek belirlenmesidir. Örneğin  $a < x < b$  şeklindeki bir  $f(x)$  fonksiyonunun ortalama değeri, bu aralıktan rastgele seçilecek sonlu adet noktanın ortalama değerinden tahmin edilebilir. Monte Carlo analiz metodları (MCM), öz değerlerin (eigenvalue) bulunması, matris dönüşümleri ve özellikle çok katlı integrallerin ölçülmesi için diferansiyel ve integral denklemlerin çözümünde de kullanılmaktadır. [3].

Özünde rastgele değişen bileşenler içeren bir sistemin ya da prosesin simülasyonu için, öncelikle rastgele oluşan sayılar üretmek ya da elde etmek gerekir. Monte Carlo hesaplamaları, belli olasılık dağılımlarından rastgele olarak çekilmiş gözükten sayı dizilerine sahip olmayı gerektirir. Monte Carlo Analizleri rassal davranış sergileme durumu olan tüm alan ve devre uygulamalarında kullanılan bir yöntemdir. Yarı iletken malzemelerin davranışını izlemeye de kullanılır [8].

Monte Carlo prosedürleri bir dizi testin ortalamasını veren çözümler sunar. Bu nedenle çözümler ortalama bir değerden sapmalar içerir. Elde edilen sonucun %100 güvenilir olduğu söylenemez. Monte Carlo hesaplamalarındaki istatistiksel kararsızlığı ölçmek için, rassal sayılarla ilişkili çeşitli istatistiksel tekniklere başvurmamız gerekir. Kısaca bir hata tahmini gerçekleştirmek için varyans ve merkezi limit teoremi kavramlarına başvurmak gerekir.

X'in rassal bir değişken olduğunu varsayalım. X'in beklenen ya da ortalama değeri  $\bar{x}$  olsun ve şu şekilde ifade edilsin:

$$\bar{x} = \int_{-\infty}^{\infty} xf(x)dx \quad (1)$$

Burada,  $f(x)$ , X'in dağılımının olasılık yoğunluğudur. Bunu sayısallaştıracak olursak,

$$\hat{x} = \frac{1}{N} \sum_{n=1}^N X_n \quad (2)$$

X'in gerçek ortalama değeri  $\bar{x}$  iken gerçek ortalamayı tahmin edici olarak  $\hat{x}$  değişkeni kullanılır.

X değişkeninin varyansı x'in her bir değerinin ortalama değerden sapmalarının karelerinin ortalamasıyla bulunur.

$$\text{Var}(X) = \sigma^2 = \overline{(x - \bar{x})^2}, \quad \text{standart sapma } \sigma \text{ ise,}$$

$$\sigma = \sqrt{\overline{(x - \bar{x})^2}} \text{ olarak elde edilir.}$$

Standart sapma, x'in  $\bar{x}$  ortalama değeri boyunca dağılımının bir ölçümünü verir. Bize hatanın şiddetinin ölçüsünü sunar. x ve  $\hat{x}$  değerlerinin standart sapmaları arasında şu şekilde bir ilişki vardır:

$$\sigma(\hat{x}) = \frac{\sigma(x)}{\sqrt{N}} \quad (3)$$

Bu ilişki bizlere,  $\bar{x}$  'i tahmin etmek için xn'in N değerlerinden elde edilmiş  $\hat{x}$  'i kullanırsak,  $\hat{x}$ 'in bizim sonuçlarımızdaki  $\bar{x}$  'e ilişkin dağılımının  $\sigma(x)$  ile orantılı olduğunu ve örneklerin N sayısı yükseldikçe azaldığını gösterir. N örneklerinin sayısı sonlu olduğu için, Monte Carlo hesaplamalarında mutlak kesinliğe ulaşamaz. Bu nedenle  $\bar{x}$  ortalaması boyunca bir limit ya da aralık tahmin edilmeye çalışılır.  $\hat{x}$  in bu limit içine denk geldiğini kestirebileceğimiz bir güven aralığı belirlenir.  $\hat{x}$  in  $\bar{x} - \delta$  ve  $\bar{x} + \delta$  arasında yer aldığı ihtimalini varsayarız.

Rassal ve bağımsız gözlemler elde etmek için Monte Carlo prosedürünü kullanacak ve bununla ilişkili olarak N değerleri için tekrar eden rastgele  $\bar{x} + \delta$  aralığını ortaya koyacak olursak, yaklaşık olarak bu rastgele aralıkların

$$\text{err} \left( \sqrt{\frac{N}{2}} \frac{\delta}{\sigma(x)} \right) \times \%100 \text{ hata oranı kadar } \hat{x}'i \text{ kapsar.}$$

Rastgele  $\bar{x} + \delta$  aralığı güven aralığı olarak adlandırılır ve  $\text{err} \left( \sqrt{\frac{N}{2}} \frac{\delta}{\sigma(x)} \right)$  hata fonksiyonu güven seviyesini ifade eder. Yüksek güven seviyesi isteniyorsa, iki ya da üç standart sapma kullanılabilir.

### 3. RASTGELE YÜRÜYÜŞ PROBLEMLERİ

Sabit yürüyüş modelli Monte Carlo metodunun aşağıdaki Laplace denkleminde uygulandığını düşünelim.  $\Delta^2 V = 0$  denklemi R bölgesinde, R bölgesini ağ gözeneklerine böldüğümüzde ve yerine sonlu fark eşdeğerini koyduğumuzda iki boyutlu R için sonlu fark gösterimi aşağıdaki şekilde olur.

$$V(x, y) = p_{x+}V(x + \Delta, y) + p_{x-}V(x - \Delta, y) + p_{y+}V(x, y + \Delta) + p_{y-}V(x, y - \Delta) \quad (4)$$

Burada,

$$p_{x+} = p_{x-} = p_{y+} = p_{y-} = \frac{1}{4} \quad (5)$$

$\Delta$  boyutlu ağ gözeneklerinden oluşan kare şeklindeki bir ızgaralı alan içinde yer alan ve rastgele yürüyüş yapacak bir parçacık için aşağıdaki gibi dört farklı hareket ihtimali vardır. Eğer rastgele yürüyüş hareketi yapacak olan parçacık herhangi bir  $(x, y)$  noktasındaysa, parçacığın hareketinin  $p_{x+}$ ,  $p_{x-}$ ,  $p_{y+}$  ve  $p_{y-}$  noktaları için  $(x, y)$ 'den sırasıyla  $(x+\Delta, y)$ ,  $(x-\Delta, y)$ ,  $(x, y+\Delta)$ ,  $(x, y-\Delta)$  koordinatlarına yer değiştirme olasılıkları vardır.

Parçacığın nereye hareket edeceğini belirlemenin bir yolu  $0 < U < 1$  aralığında  $U$  rastgele sayısı üretmek ve parçacığın hareketini Tablo 1'deki şekilde tanımlamaktır.

Tablo 1. Parçacık hareketi için sınır değerleri

|      |                  |               |                                    |
|------|------------------|---------------|------------------------------------|
| Eğer | $0 < U < 0,25$   | $\Rightarrow$ | $(x, y) \rightarrow (x+\Delta, y)$ |
| Eğer | $0,25 < U < 0,5$ | $\Rightarrow$ | $(x, y) \rightarrow (x-\Delta, y)$ |
| Eğer | $0,5 < U < 0,75$ | $\Rightarrow$ | $(x, y) \rightarrow (x, y+\Delta)$ |
| Eğer | $0,75 < U < 1$   | $\Rightarrow$ | $(x, y) \rightarrow (x, y-\Delta)$ |

Eğer kare yerine dikdörtgen şeklinde bir ızgaralı yüzey üzerinde çalışılacaksa,  $p_{x+} = p_{x-}$  ve  $p_{y+} = p_{y-}$  olmakla birlikte  $p_x \neq p_y$ 'dir.  $(x, y)$  noktasındaki potansiyeli hesaplamak için, rastgele yürüyecek parçacığa o noktadan başlama talimatı verilir. Parçacık sınıra ulaşana kadar bir düğümden bir diğer düğüme hareket etmeye başlar. Sınıra ulaştığında, gezinme hareketi sonlandırılır ve daha önceden belirtilmiş o noktadaki  $V_p$  potansiyeli kaydedilir. İlk yürüyüş sonundaki  $V_p$  değerinin şekilde gösterildiği üzere  $V_p(1)$  olarak atandığını varsayalım. Ardından ikinci bir parçacık  $(x, y)$  noktasından gönderilir ve bir sınır noktasına ulaşana kadar gezinmesi sağlanır. Bu gezintinin son bulunduğu nokta da şekilde görüldüğü üzere  $V_p(2)$  olarak adlandırılır. Bu prosedür  $V_p(3)$ .,  $V_p(4)$ ....  $V_p(N)$  şeklinde devam ettirilir.  $(x, y)$  noktası için çözüm eşitliği (6) nolu eşitlikte verildiği üzere toplam fonksiyonuyla gösterilebilir.

$$V(x, y) = \frac{1}{N} \sum_{i=1}^N V_p(i) \quad (6)$$

Burada  $N$  toplam yürüyüş sayısıdır ve büyük bir değerdir. Yakınsama oranı  $\sqrt{N}$  oranında değişir ve doğru sonucun alınması için çok sayıda rastgele yürüyüş yapılması gerekir. Bu MCM adım boyutu  $\Delta$ 'nın sabit olması nedeniyle sabit rastgele yürüyüş türü olarak adlandırılmıştır ve yürüyüş adımları koordinat düzlemine paraleldir. Neumann sınır koşullarına sahip düzgün olmayan şekilli

gövdelerin çözümlerinde kullanımı uygun değildir. Bu sorun rastgele kayan yürüyüş (floating random walk) tekniğini kullanan MCM ile çözülmüştür. Ayrıca kayan rastgele yürüyüş tekniği sabit yürüyüş tekniğine göre çok daha hızlı işler.

Kayan Rastgele Yürüyüş yönteminin matematiksel temeli, potansiyel teorisinin orta değer teoremine (mean value theorem) dayanır. İki boyutlu bir çember için  $x, y$  noktasındaki gerilim,  $V(x, y)$  ile hesaplanır.

$$V(x, y) = \frac{1}{2\pi\rho} \oint_L V(\rho') d\rho' \quad (7)$$

Burada;  $(x, y)$  merkezli ve  $\rho$  yarıçaplı bir çemberin integrali alınmaktadır. (7) nolu eşitlik aşağıdaki şekilde de gösterilebilir.

$$V(x, y) = \int_0^1 V(a, \phi) dF \quad (8)$$

Burada,

$$F = \frac{\phi}{2\pi} \text{dir.} \quad (9)$$

$\phi$  açısı düzgün dairesel koordinat değişkenidir.  $F$  fonksiyonu  $\phi$ 'ye karşılık gelen olasılık dağılımı olarak düşünülebilir ve tüm  $\phi$  açıları eşit olasılıklıdır. Kayan rastgele yürüyüşlü MC modeli (7) nolu denklemin istatistiksel bir boyutta ele alınmasına bağlıdır. İki boyutlu bir problem için,  $i$ 'nci yürüyüşde  $j$  adım sayısı sonrası herhangi  $(x_j, y_j)$  noktasında yer alan rastgele hareket eden bir parçacık düşünün. Bir sonraki  $(j+1)$ 'inci adım aşağıdaki gibi elde edilir. İlk önce,  $(x_j, y_j)$  merkezli ve sınır ile  $(x_j, y_j)$  arasında en kısa mesafeye eşit  $\rho_j$  yarıçaplı bir çember çizilir.  $(0, 2\pi)$  arasında düzgünce dağılmış rastgele bir değişken olarak  $\phi$  koordinatı üretilir. Böylece, parçacığın bir sonraki adımı aşağıdaki şekilde ifade edilir.

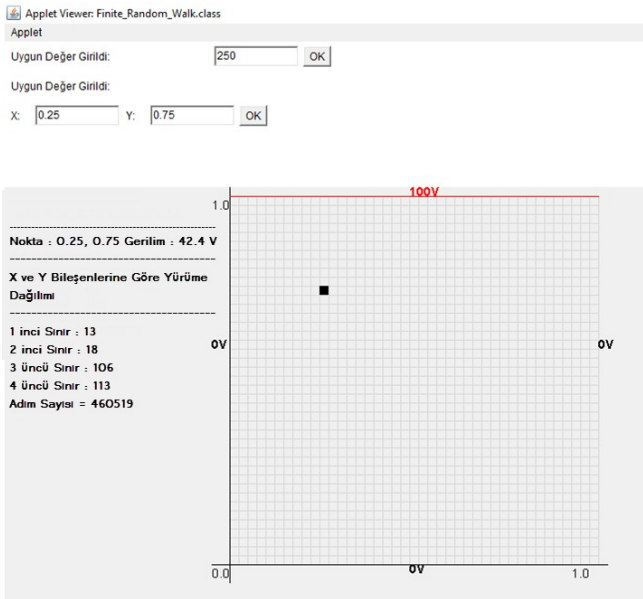
$$x_{j+1} = x_j + \rho_j \cos\phi_j; \quad y_{j+1} = y_j + \rho_j \sin\phi_j \quad (10)$$

Bu prosedür birkaç defa tekrar ettirilir ve sınıra daha önceden tanımlanmış bir değere kadar yaklaşıldığında yürüyüş sonlandırılır. Bu  $i$ 'nci yürüyüş sonunda  $V_p(i)$  potansiyeli Sabit Rastgele Yürüyüş modelinde olduğu gibi  $(x, y)$  için kaydedilir. (6)'da yer alan ortalama değer hesabı formülüyle de o noktanın  $N$  yürüyüş sonundaki  $V(x, y)$  değeri hesaplanır.

## 4. RASTGELE YÜRÜYÜŞ PROBLEMLERİNİN GÖRSEL PROGRAMLAMA TEKNİKLERİYLE ÇÖZÜMÜ

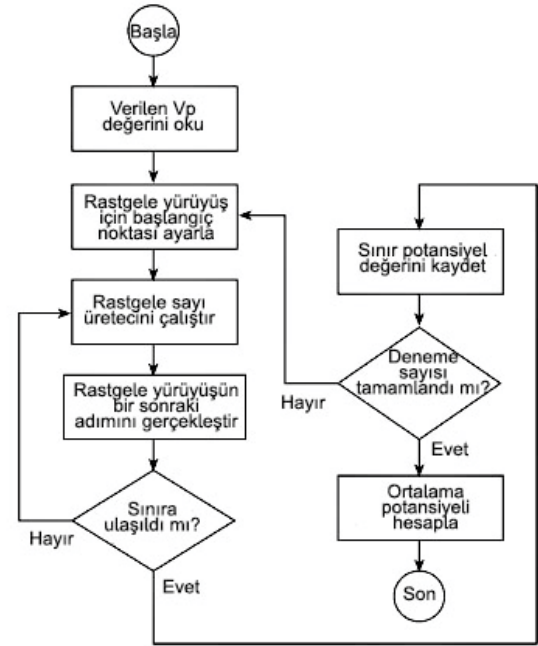
### 4.1. Sabit Rasgele Yürüyüş Yönteminin Görsel Olarak Programlanması

Şekil 1'deki Java Apleti ekranında kesit alanı gösterilen sonlu uzunlukta bir iletkenin bir yüzeyine  $y=1$  koordinatında 100V uygulandığını ve diğer yüzeylerinin topraklandığını düşünelim. Böyle bir iletkenin kesit alanı içinde tanımlanan bir nokta için potansiyel değerini, MC sabit rastgele yürüyüş modelini kullanarak bulabiliriz.



Şekil 1. MC Sabit Rastgele Yürüyüş modeli (N=250)

Şekil 2'de MC rastgele yürüyüş modellerinin bilgisayar analizi için kullanılan akış diyagramı verilmiştir.



Şekil 2. MC rastgele yürüyüş modellerinin bilgisayar analizi için kullanılan akış diyagramı

Sabit rastgele MCM'nin uygulanmasıyla aşağıda JAVA kodları verilmiş olan program elde edilebilir. Adım boyutu  $\Delta$  0,05 olarak seçilmiştir. Programın ekran çıktısında sınıra ulaşmak için gerçekleşen rastgele ortalama m adım sayısı ve potansiyel hesapları listelenir. MCM sonuçlarının hata oranının %1'den daha küçük olması için küçük adım boyutları için büyük miktarda rastgele adım sayısı gerekmektedir. Kullanıcıdan 250-1000 arasında bir N örnek değeri ve x-y koordinat noktası girmesi istenir.

Tablo 2. MC Sabit Rastgele Yürüyüş (Fixed Random Walk) modeli için Java kodu

```
public void run() {
    for (k=1;k<=sayac;k++){
        i=0;
        j=0;
        while (akis){
            r=Math.random();
            m3++;
            if (r>=0 && r<p1)
                i++;
            else if (r>=p1 && r<p2)
                j++;
            else if (r>=p2 && r<p3)
                i--;
            else if (r>=p3)
                j--;
            if (i==0) { // -x sol sınır
                sum+=v4;
                m4++;
                akis=false;
            }
            else if (i>imax) { // +x sağ sınır
                sum+=v2;
                m2++;
                akis=false;
            }
            else if (j==0) { // -y alt sınır
                sum+=v1;
                m1++;
                akis=false;
            }
            else if (j>jmax) { // +y üst sınır
                sum+=v3;
                m3++;
                akis=false;
            }
            x_kor = (int) (i*4*delta/0.01)+250;
            y_kor = (int) (400-j*4*delta/0.01)+10;
            V=(double)sum/this.sayac;
            repaint();
            try{
                Thread.sleep(1, 1);
            }
            catch (InterruptedException e) {
            }
            akis=true;
        }
        v=(double)sum/this.sayac;
        repaint();
    }
}
```

Burada "sayac" değişkeni örnekleme değeridir. Ne kadar büyük olursa gerçek çözüm değerine o kadar yakın sonuç elde edilir, ancak çözüm süresi uzar. Şekil 3'te 500 örnek değeri için analizin bitmiş hali gösterilmiştir. Başlangıç noktası girilen nokta için hareket yönleri rastgele üretilir. Noktanın geldiği her sınır değeri için o sınırdaki gerilim değeri toplama eklenir ve "sayac" değerine ulaşıldığında

toplam değeri "sayac" değerine oranlanarak gerçek gerilim değeri hesaplanır.

Noktanın rastgele hareketi,

$r = \text{Math.random}()$ ; satırında Java Matematik kütüphanesinin  $\text{random}()$  prosedürü kullanılarak 0-1 arasında elde edilen sayının değerine göre sağlanır. Hazır kütüphane prosedürü yerine, ilgili kodda izlenen değişiklik yapılarak programlayıcı kendi rassal üreticini yapabilir.

Tablo 3. Yerel saat değerinden rassal değişken için tohum değerinin elde edilmesi

```
LocalDateTime tarih = LocalDateTime.now();
saat = tarih.getHour();
dakika = tarih.getMinute();
saniye = tarih.getSecond();
degerTohum = saniye*dakika*saat;
```

Tablo 3'te rassal değer elde edilmesi için gerekli tohum değerinin sistem saatinden üretilmesi gösterilmiştir. Bu kodun ilgili programın,

```
public void guncelle(){
...
...
...
}
```

Güncelleme işlemi için gerekli alana yazılması yeterlidir. Yeni bir koordinat ve N örnekleme değeri girildiğinde bu prosedür yeni baştan işletilir. Böylece kullanıcı ilgili simülasyonu her çalıştırdığında yeni bir başlangıç tohum değerine göre rassal üreteç çalışır.

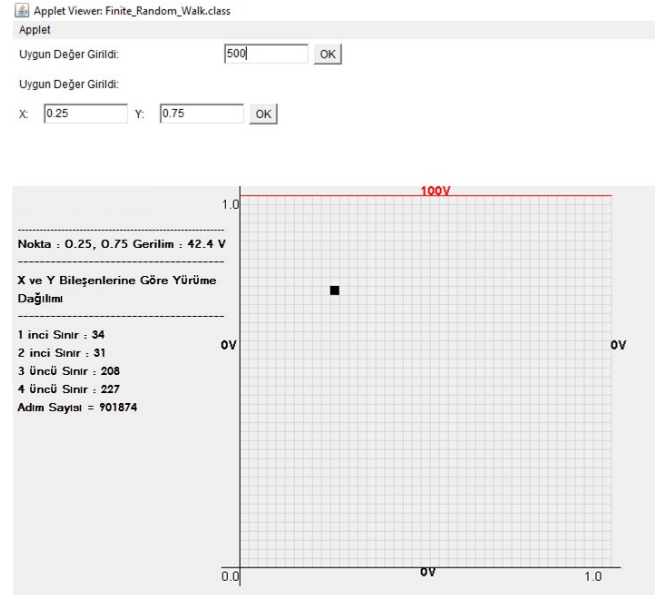
Tablo 4'te gösterilen kullanıcı tanımlı rastgele sayı prosedürü ile "teta" değişkeni adında rassal sayı üretilir.

Tablo 4. Kullanıcı tanımlı rassal sayı üretici

```
public double random(){
degerTohum = (a*degerTohum)%del;
r = degerTohum/del;
teta= Math.acos(1-2*r)/Math.PI;
return teta;
}
```

"del" değikeni  $m=2^{31}-1$  değerine (2147483647), "a" değikeni ise  $7^5$  değerine (16807) karşılık gelen bir sayıdır. "degerTohum" değikeni ise sistem saatinden elde edilen tohum değeridir ve m değerinden küçüktür.

Java programlama dili "Thread" denilen ve birbirinden bağımsız olarak çalışan iş parçacıklarını yürütme imkanı sunar. İlgili simülasyonlarda da "Thread" kullanılarak programın adımlama işini nasıl yaptığı canlandırılmış ve programı kullanacakların süreci yavaş yavaş izlemesi sağlanmıştır.



Şekil 3. : MC sabit rastgele yürüyüş modelinin N=500 için uygulama sonucu

Şekil 3'te simülasyonu sonlanmış ekran görüntüsü görülmektedir.  $x=0,25$  ve  $y=0,75$  koordinat değeri için 41,6 volt değeri elde edilmiştir. 500 örnekleme değerine 901874 adım sayısında ulaşılmıştır. Burada simülasyon adımları uzun sürmektedir ve istatistiksel hata analizi mevcut değildir. Böyle bir koordinat düzlemi içindeki bir noktaya etki eden gerilimin, hata analiziyle birlikte daha hızlı şekilde hesaplanmasını sağlayan, daha gelişmiş bir Monte Carlo yöntemi Kayan Rastgele Yürüyüş (Float Random Walk) modelidir.

## 4.2. Float Random Walk Yönteminin Örnek Üzerinden Görsel Programlanması

Bu yöntemin uygulama prosedürü Şekil-4 üzerinden Tablo-4'te yer alan kodlama ile gösterilmiştir. Sırasıyla  $y= 2.5$  koordinatında 20V,  $x= 2.0$  koordinatında 30V ve  $y= 1.0$ ,  $x= 1.0$  koordinatlarında 0 V olduğu varsayılan iki boyutlu bir potansiyel enerji sistemi tasarlanmıştır. Bu kapalı sistem içinde tanımlı herhangi bir noktanın potansiyel enerjisini bulmak için Kayan Rastgele Yürüyüş tekniği kullanılmıştır. Bu koordinatlardaki gerilim değerlerini değiştirmek suretiyle bu kapalı elektrik alanı içinde tanımlı ( $x$ ,  $y$ ) noktası için farklı gerilim değerleri elde edilebilir.

( $x$ ,  $y$ ) noktasından en kısa  $\rho$  noktasını bulmak için Şekil 4'teki örnek potansiyel alanı 3 adet dikdörtgen çözüm alanına bölünür. Denklem-10'da yer alan  $\rho$  değerlerini bulmak için şu adımlar izlenir:

Eğer  $\{(x, y) : 1 < x < 2, 0 < y < 1\}$ ,



$$\rho = \text{minimum}\{x - 1, 2 - x, y\}$$

Eğer  $\{(x, y) : 0 < x < 1, 1 < y < 2.5\}$ ,

$$\rho = \text{minimum}\{x, y - 1, 2.5 - y\}$$

Eğer  $\{(x, y) : 1 < x < 2, 1 < y < 2.5\}$ ,

$$\rho = \text{minimum} \left\{ 2 - x, 2.5 - y, \sqrt{(x - 1)^2 + (y - 1)^2} \right\}$$

Tolerans  $\tau = 0.05$  olarak seçilir. Böylece rastgele yürüyüş içindeki yeni bir nokta ve sınır arasındaki mesafe tolerans değerinden daha düşükse en yakın sınırdaki potansiyel kaydedilir. 4.1'deki örnek için hazırlanan programa ilave olarak, 4.2'deki örnek için geliştirilen programda hata tahmini analizi de yapılmıştır. Hata hesabı için Denklem-11 kullanılmıştır:

$$\delta = \frac{St_{\alpha/2; n-1}}{\sqrt{n}} \quad (11)$$

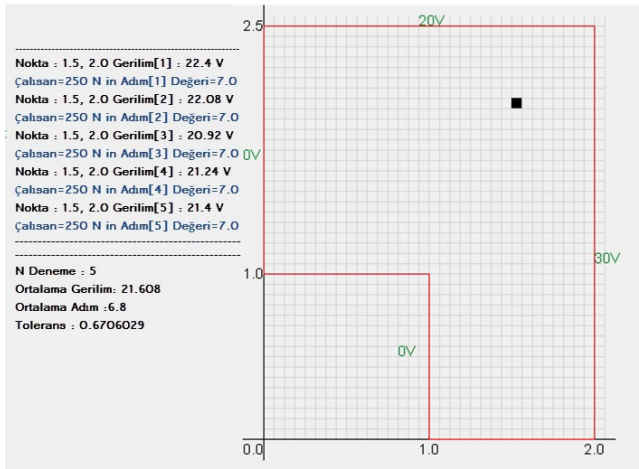
Applet Viewer: Float\_Random\_Walk.class

Applet

Uygun Değer Girildi:

Uygun Değer Girildi:

X:  Y:



Şekil 4. MC Kayan Rastgele Yürüyüş modeli (N=250)

Tablo 5. Kayan Rastgele Yürüyüş (Float Random Walk) yöntemi için Java kodu

```
public void run() {
    while(nrun<sayac){
        nrun+=250;
        for(j=1;j<=ntrials;j++){
            sum=0;
            m=0;
            for (int i=1;i<=nrun;i++){
                x=x0;
                y=y0;
                while(akis){
                    rnd=Math.random();
                    phi=2.0*Math.PI*rnd;
                    rc=Math.sqrt(Math.pow(x-1, 2) +
                    Math.pow(y-1, 2));
                    // En kısa yol algoritması --- en kısa yol
                    hesaplanıyor...
                    if(y>1){
                        if(x>1){
                            r = 2 - x;
                            if(r>rc){
                                r = rc;
                            }
                            if(r>(2.5-y)){
                                r = 2.5 -
                                y;
                            }
                        }
                        else{
                            r = x;
                            if(r>(y-1)){
                                r = y -
                                1;
                            }
                            if(r>(2.5-y)){
                                r = 2.5 -
                                y;
                            }
                        }
                    }
                    else{
                        r = x - 1;
                        if(r>(2-x)){
                            r = 2 - x;
                        }
                        if(r>y){
                            r = y;
                        }
                    }
                    x = x + r*Math.cos(phi);
                    y = y + r*Math.sin(phi);
                    m++;
                    // (x,y) sınırımı kontrol ediyor.
                    if(x<(1+tolerans) && y<(1+tolerans)){
                        akis=false;
                    }
                    else if(x>(2-tolerans)){
                        sum = sum + 30;
                        akis=false;
                    }
                    else if(y>=(2.5-tolerans)){
                        sum = sum + 20;
                        akis=false;
                    }
                    else if(y > 1 && x < tolerans)
                    akis=false;
                    else if(y < 1 && x < (1 - tolerans))
                    akis=false;
                    else if(y <= tolerans && x >= 1)
                    akis=false;
                    else if(y <= (1 + tolerans) && x < 1)
                    akis=false;
                    x_kor = (int)(x*1.6/delta)+250;
                    y_kor = (int)(400-y*1.6/delta)+10;
                    repaint();
                    try{
                        Thread.sleep(1, 1);
                    }
                    catch(InterruptedException e){
                    }
                    akis=true;
                }
                if(nrun==0){
                    v[j]=(float)(sum/nrun);
                    adimlar[j]=(float)(m/nrun);
                }
            }
            // V ve adım sayısının ortalama değerleri
            hesaplanıyor.
            sum = 0;
            suml = 0;
            for(int i=1;i<=ntrials;i++){
                sum = sum + v[i];
                suml = suml + adimlar[i];
            }
            v_ortalama=(float)sum/ntrials;
            adim_ortalama=(float)suml/ntrials;
            // hata oranı hesaplanıyor.
            sum=0;
            for(int i=1;i<=ntrials;i++){
                sum = sum + Math.pow(v[i]-
                v_ortalama,2);
            }
            std = (float)Math.sqrt(sum/(ntrials-
            1));
            error =
            (float)(std*t[ntrials]/Math.sqrt(ntrials));
            x_kor = (int)(x*1.6/delta)+250;
            y_kor = (int)(400-y*1.6/delta)+10;
        }
    }
}
```

Bu analiz yönteminde adım boyutları değişkenlik göstermektedir ve Sabit Rastgele Yürüyüş yöntemindeki pek çok ara değerler işletilmez. Böylece sınır değerlere erişmek için daha az adım gerekir ve analiz süresi kısalmır.

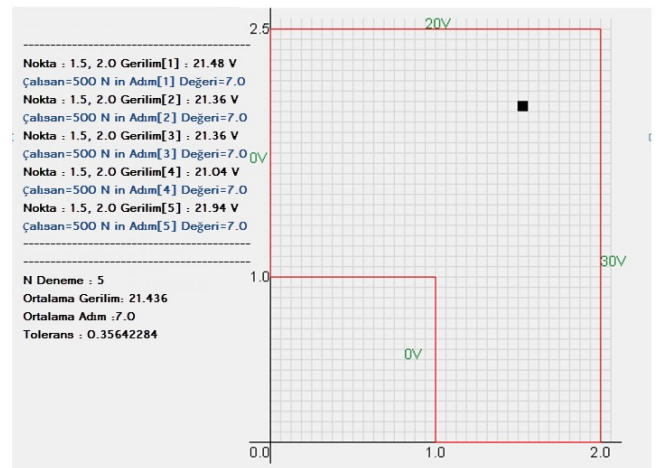
Applet Viewer: Float\_Random\_Walk.class

Applet

Uygun Değer Girildi:

Uygun Değer Girildi:

X:  Y:



Şekil 5. MC kayan rastgele yürüyüş modelinin N=500 için uygulama sonucu

Şekil 5'te simülasyonun sonlanmış ekran görüntüsü görülmektedir.  $x=1,25$  ve  $y=2$  koordinat değeri için 21,436 volt değeri elde edilmiştir. Örneklem değeri yükseldikçe hata payı (tolerans)

düşmektedir. Her bir gerilim değeri için 5 kez farklı tohum değerleriyle yeni baştan örnekleme yapılmaktadır. Bu şekilde daha önceki örnekte elde edilen tek bir ortalama gerilim değerine göre 5 farklı ortalama gerilim değeri elde edilmekte ve bunların da ortalaması alınmaktadır. Böylece rastgelelik durumunun standart sapması da hesaba katılmaktadır.

## 5. SONUÇ VE TARTIŞMA

Elektrik-elektronik mühendisliği eğitimi alan öğrencilerin alan teorisine ilişkin matematiksel düşünme yeteneklerini artırmalarını sağlamak amacıyla söz konusu disiplinle ilgili nümerik analiz tekniklerinden Monte Carlo yöntemi seçilerek programlama örnekleri hazırlanmıştır. Çok sayıda nümerik analiz yöntemi içinden Monte Carlo yönteminin seçilmesinin nedeni deterministik olmayan ve rastgelelik durumlarını içeren bir analiz mantığına sahip olması, istatistiksel çözüm modeli sunması ve elektrik-elektronik devre ve alan analizi programlarında yaygın olarak kullanılan bir yöntem olmasıdır. Uygulamaların grafik tabanlı görsel olarak hazırlanması ve matematiksel gösterimlerin simüle ettirilerek canlandırılmasının matematiksel denklemlerin daha iyi anlaşılmasına katkı sağlayacağı değerlendirilmiştir. Elektromanyetik alanların ve alan teorisinin, çeşitli öğrenme araçları ve uygulama teknikleriyle görselleştirilerek öğretilmesi üzerine yapılan çalışmalar bu sonucu ortaya koymuştur [9].

Lisansüstü öğrenimde "elektromanyetik teoride analitik ve nümerik yöntemler" dersi olup, bu ders kapsamında, elektromanyetik hesaplamalarında kullanılan 2 ve 3 katlı integral hesapları, laplace ve fourier dönüşümleri gibi öğrencilerin lisans döneminde algılama güçlüğü çektikleri analog hesaplama tekniklerinin yerine nümerik hesaplama yöntemleri gösterilmektedir. Bu tür derslerin görsel programlama uygulamalarıyla desteklenerek işlenmesi, belli bir bilgisayar programcılığı altyapısıyla gelen lisans öğrencilerinin de hem geçmiş dönemlerden kalan konu eksiklerini daha iyi tamamlayacakları, hem de nümerik hesapları programlama teknikleriyle gerçekleştirmeleri sonucu bilgisayar programcılığı konusunda da kendilerini geliştirme fırsatı bulacakları değerlendirilmektedir. Elektromanyetik alan teorisi üzerine karmaşık problem egzersizlerinin, öğrencilerin kavramsal ve prosedürel bilgilerinde gelişim sağladığı açıktır

[10]. Çeşitli alan problemleri üzerine yapılan çalışmalar farklı problem çözme tekniklerinin uygulanmasını zorunlu kılmıştır [11], [12].

Bu çalışmada üzerinde durulması gereken iki temel durum yer almaktadır. Birincisi, mühendislik eğitiminde kullanılan alan analizlerinin, grafik arayüz temelli bilgisayar programlama teknikleriyle oluşturulması sırasında, grafik arayüze aktarım sırasında grafik değişkenlerinin iyi ayarlanmasıdır. Bilgisayar programcılığı konusunda zayıf olanlar için Java ve benzer programlama dilleri için geliştirilmiş çeşitli mühendislik kütüphaneleri vardır. Grafik arayüz özelliğinin daha hızlı kullanılmasını sağlayan kütüphaneler yardımıyla hızlı programlama prosedürü gerçekleştirilebilir [2]. Grafik arayüzünün iyi kalibre edilmemesi elde edilecek çıktıların yanlış yorumlanmasına neden olabilir. İkincisi, bu çalışmaların iyi bir vakit planlamasıyla yürütülmesidir ki, programlama konusunda yeterince altyapı sahibi olmayan öğrenciler açısından uygulama güçlüğü yaşanabilir. Alan teorileri ile ilgili olarak bilgisayar programlama yeterliğinin kullanılarak uygulama geliştirilmesi üzerine çalışmalar zenginleştirilebilir. Benzer çalışmalar geliştirilerek ve eksik kalan yanlar güçlendirilerek bilgisayar programlamayla elektromanyetik alan analizlerinin simülasyonlarını yapacak öğrenciler için geniş bir başlangıç kaynağı oluşturulabilir.

## KAYNAKÇA

- [1] P. Paultre, E. Lapointe, C. Carbonneau and J. Proulx, "LAS: A Programming Language and Development Environment for Learning Matrix Structural Analysis", *Computer Applications in Engineering Education*, vol. 24, no. 1, pp. 89-100, 2016.
- [2] H. Gould, J. Tobochnik and W. Christian, *An Introduction to Computer Simulation Methods*, Third Edition, ISBN: 0-8053-7758-1, 2011.
- [3] M.N.O. Sadiku, *Numerical Techniques in Electromagnetics*, CRC Press LLC, 2001.
- [4] F. Sagnard, *Educational Graphical Interfaces to Learn About Radiation and Propagation of Electromagnetic Waves*, *IEEE Transactions on Education*, vol. 47, no. 3, pp. 389-396, 2004.
- [5] Y.J. Dori and J. Belcher, *How Does Technology-Enabled Active Learning*



- Affect Undergraduate Students' Understanding of Electromagnetism Concepts?, *The Journal of The Learning Sciences*, vol. 14, no. 2, pp. 243-279, 2005.
- [6] Y.J. Dori, J. Belcher, M. Bessette, M. Danziger, A. McKinney and E. Hult, *Technology for active learning*, Elseiver, ISSN:1369 7021, 2003.
- [7] J. Guisasola, J.M. Almudi, L. Jose and J.L. Zubimendi, Difficulties in Learning the Introductory Magnetic Field Theory in the First Years of University, *Wiley Periodicals*, Sci Ed 88:443 464, 2004.
- [8] Y. Chen and R. Xu, Analysis of the RF and noise performance of junctionless MOSFETs using Monte Carlo simulation, *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 22, pp. 822-833, 2014.
- [9] J. Gilbert, *Visualization in Science Education*, Springer, 2005.
- [10] J. Leppävirta, H. Kettunen and A. Sihvola, Complex Problem Exercises in Developing Engineering Students' Conceptual and Procedural Knowledge of Electromagnetics, *IEEE Transactions on Education*, vol. 54, no. 1, pp. 63-66, 2011.
- [11] J.R. Cardoso, V.C. Silva, N.M. Abe and L.N. Rossi, Approach to Teaching the Finite Element Method Applied to Electromagnetic Problems with Axial Symmetry to Electrical Engineering Students, *Computer Applications in Engineering Education*, vol. 7, no. 3, pp. 133-145, 1999.
- [12] K. Zuza, J.M. Almudi, A. Leniz and J. Guisasola, Addressing students' difficulties with Faraday's law: A guided problem solving approach, *Physics Education Research*, vol. 10, no. 1, 2014.