



## ZAMAN TANIM BÖLGESİNDE SONLU FARKLAR YÖNTEMİ İLE PYTHON DİLİNDE GRAFİK KULLANICI ARAYÜZLÜ BİR ELEKTROMANYETİK BENZETİM YAZILIMININ GELİŞTİRİLMESİ

<sup>1</sup>Murat Can SERKUŞ , <sup>2</sup>Serhan YAMAÇLI 

<sup>1,2</sup>Nuh Naci Yazgan Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Kayseri, TÜRKİYE  
<sup>1</sup>[muratacanserkus@gmail.com](mailto:muratacanserkus@gmail.com), <sup>2</sup>[syamacli@nny.edu.tr](mailto:syamacli@nny.edu.tr)

(Geliş/Received: 24.07.2020; Kabul/Accepted in Revised Form: 04.11.2020)

**ÖZ:** Rastgele geometrilerin söz konusu olduğu durumlarda elektromanyetik çözümler için sayısal yöntemler kullanılmaktadır. Zaman bölgesinde sonlu farklar (ZTBSF) metodu da bu amaçla kullanılan popüler yöntemlerden biridir. Bu çalışmada, Python programlama dilinde ZTBSF yöntemi ile rastgele yapıların elektromanyetik benzetimlerini yapmak için bir yazılım geliştirilmiştir. Bu yazılıma Tkinter modülü kullanılarak bir grafik kullanıcı arayüzü de eklenerek kullanıcı dostu bir yapı elde edilmiştir. Çeşitli elektromanyetik problemlerin hem geliştirilen yazılım ile hem de piyasada ticari olarak bulunan bir yazılım ile yapılan benzetim sonuçları karşılaştırılmış ve geliştirilen yazılımın beklendiği şekilde çalıştığı gösterilmiştir.

**Anahtar kelimeler:** Zaman bölgesinde sonlu farklar metodu, ZTBSF, Python, Tkinter, elektromanyetik benzetim.

### Development of an Electromagnetic Simulation Software with Graphical User Interface in Python Programming Language Using Finite Difference Time Domain Method

**ABSTRACT:** Numerical methods are utilized for solving electromagnetic problems having random geometries. Finite difference time domain (FDTD) is a popular method used for these types of simulations. In this study, a software in Python programming language is developed implementing FDTD method for electromagnetic simulations. The software is made as user-friendly by adding a graphical user interface with the Tkinter module. Simulation results of various electromagnetic problems obtained with our software and a commercial software are compared and it is shown that our software can be used to simulate electromagnetic problems as expected.

**Keywords:** Finite difference time domain method, FDTD, Python, Tkinter, electromagnetic simulation.

### GİRİŞ (INTRODUCTION)

Hem haberleşme hem de savunma sanayilerindeki gelişmeler neticesinde elektromanyetik benzetimler günümüzde her zamankinden daha fazla önem kazanmıştır. Ancak, Maxwell denklemlerinin doğasından dolayı, rastgele geometriler söz konusu olduğunda elektromanyetik problemlerin çözümünde sayısal yöntemlerin kullanılması gerekmektedir. Bu sayısal yöntemlere örnek olarak moment metodu (Ney, 1985), sonlu elemanlar metodu (Jin, 1993), zaman tanım bölgesinde sonlu farklar metodu (ZTBSF) (Yee, 1966) ve iletim hattı metodu (Christopoulos, 1995) verilebilir.

Bu çalışmada, ZTBSF yöntemi kullanılarak kullanıcı dostu bir elektromanyetik benzetim yazılımı gerçekleştirilmiştir. Diğer yöntemlerle kıyaslandığında, ZTBSF yönteminin iki temel olumlu yanı vardır. Bunlar, Maxwell denklemlerinin zaman tanım bölgesinde kullanması sayesinde geniş frekans aralığında benzetim yapılabilmesi ve problem uzayı büyüdükçe gereken işlem adımı sayısının doğrusal artışı olarak ifade edilebilir (FDTD Method, 2019). ZTBSF yönteminde temel olarak Maxwell denklemleri sonlu farklar denklemlerine dönüştürülmekte ve bu denklemler iteratif olarak çözülmektedir (Yee, 1966). ZTBSF yönteminin birçok programlama dilinde ve ticari yazılımlarda alternatif gerçeklemeleri mevcuttur. Pratikte ZTBSF metodunun temel alındığı elektromanyetik benzetim programlarına örnek olarak XFDTD (XFDTD Software, 2020), Lumerical (Lumerical Inc., FDTD Software, 2020) ve Synopsys (FullWAVE FDTD Software, 2020) yazılımları gösterilebilir. Diğer taraftan ticari olmayan birer modül olarak yazılmış olan Meep (Oskooi vd., 2010) ve Houle ve Sullivan kütüphaneleri (Sullivan, 2020) literatürde bulunmaktadır. Bu çalışmada ise, Elsherbeni ve Demir tarafından MATLAB ortamında geliştirilen ZTBSF çözümü hareket noktası olarak alınmıştır (Elsherbeni ve Demir, 2015). Bunun sebebi, MATLAB dilinin diğer dillere göre Python diline daha fazla benzemesidir. Her ne kadar dil benzerliği olsa da Python değişkenleri ile MATLAB değişkenleri arasındaki veri yapısı farkları ile sınıf-nesne farklılıkları nedeniyle aynı program kodunu Python ortamında yürütmek doğal olarak olanaksızdır. Bu nedenle bu çalışmada ZTBSF algoritmasını gerçekleştiren fonksiyonlar Python programlama dilinde baştan yazılmıştır. Daha sonra ise, kendi yazılımımıza Python'ın Tkinter modülü ile bir grafik kullanıcı arayüzü eklenerek daha da geliştirilmiş ve kullanıcı dostu haline getirilmiştir. Ayrıca, eklenebilen geometrilere dikdörtgen prizması ve küre haricinde, silindir, halka ve kesik piramit nesnelere de eklenmiştir. Bu sayede, geliştirilen yazılımla kodlama bilgisine ihtiyaç olmaksızın ZTBSF tabanlı elektromanyetik benzetimler kolaylıkla yapılabilmektedir. Aynı zamanda, yazılım geliştirilirken kullanılan bütün modül ve kütüphaneler ücretsizdir ve bu sayede geliştirilen yazılım sınırsız şekilde dağıtılabilecek potansiyele sahiptir. Geliştirilen arayüz ile kullanıcı, elektromanyetik benzetimini yapmak istediği yapıyı geometrik olarak kısa sürede tanıtabilmekte, tasarımına toplu devre elemanları, gerilim kaynakları ve elektriksel portlar ekleyebilmektedir. Kullanıcının eklediği nesnelere 3 boyutlu sahnede OpenGL tabanlı olarak çizilmekte ve problem uzayı üzerinde döndürme, sürüklenme, uzaklaştırma ve yakınlaştırma gibi 3 boyutlu işlemler yapılabilmektedir. Geliştirilen yazılımın ayrıntılı anlatımı ilerleyen bölümlerde bulunmaktadır.

### ZTBSF YÖNTEMİ (FDTD METHOD)

Öncelikle, ZTBSF yönteminin başlangıç noktası olan Maxwell denklemleri hatırlatılmıştır. Diferansiyel formdaki Maxwell denklemleri (1) ve (2)'de gösterildiği gibi yazılabilir. Sürekli zaman ve uzay bileşenlerini içeren bu denklemler daha açık halde (3)-(8)'deki gibi ifade edilebilmektedir (Elsherbeni ve Demir, 2015).

$$\vec{\nabla} \times \vec{H} = \varepsilon \frac{\partial \vec{E}}{\partial t} + \sigma^e \vec{E} + \vec{J}_i \quad (1)$$

$$\vec{\nabla} \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} - \sigma^m \vec{H} - \vec{M}_i \quad (2)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon_x} \left( \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma_x^e E_x - J_{ix} \right) \quad (3)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon_y} \left( \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma_y^e E_y - J_{iy} \right) \quad (4)$$

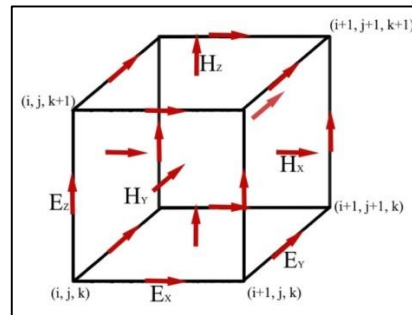
$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon_z} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma_z^e E_z - J_{iz} \right) \quad (5)$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu_x} \left( \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \sigma_x^m H_x - M_{ix} \right) \quad (6)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu_y} \left( \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \sigma_y^m H_y - M_{iy} \right) \quad (7)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu_z} \left( \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma_z^m H_z - M_{iz} \right) \quad (8)$$

Herhangi bir geometriye ilişkin olarak bu denklemlerin bilgisayar ortamında ZTBSF yöntemi ile sayısal olarak çözülmesi için problem uzayı Şekil 1’de gösterilen Yee hücrelerine bölünmekte ve bu hücrelerde bulunan elektrik ve manyetik alan bileşenleri (9)-(14)’deki gibi ifade edilmektedir (Yee, 1966, Elsherbeni ve Demir, 2015).



Şekil 1. Yee hücresi (K.S. Yee, 1966)

Figure 1. Yee cell (K.S. Yee, 1966)

$$E_x(i, j, k) \rightarrow ((i-0.5)\Delta x, (j-1)\Delta y, (k-1)\Delta z) \quad (9)$$

$$E_y(i, j, k) \rightarrow ((i-1)\Delta x, (j-0.5)\Delta y, (k-1)\Delta z) \quad (10)$$

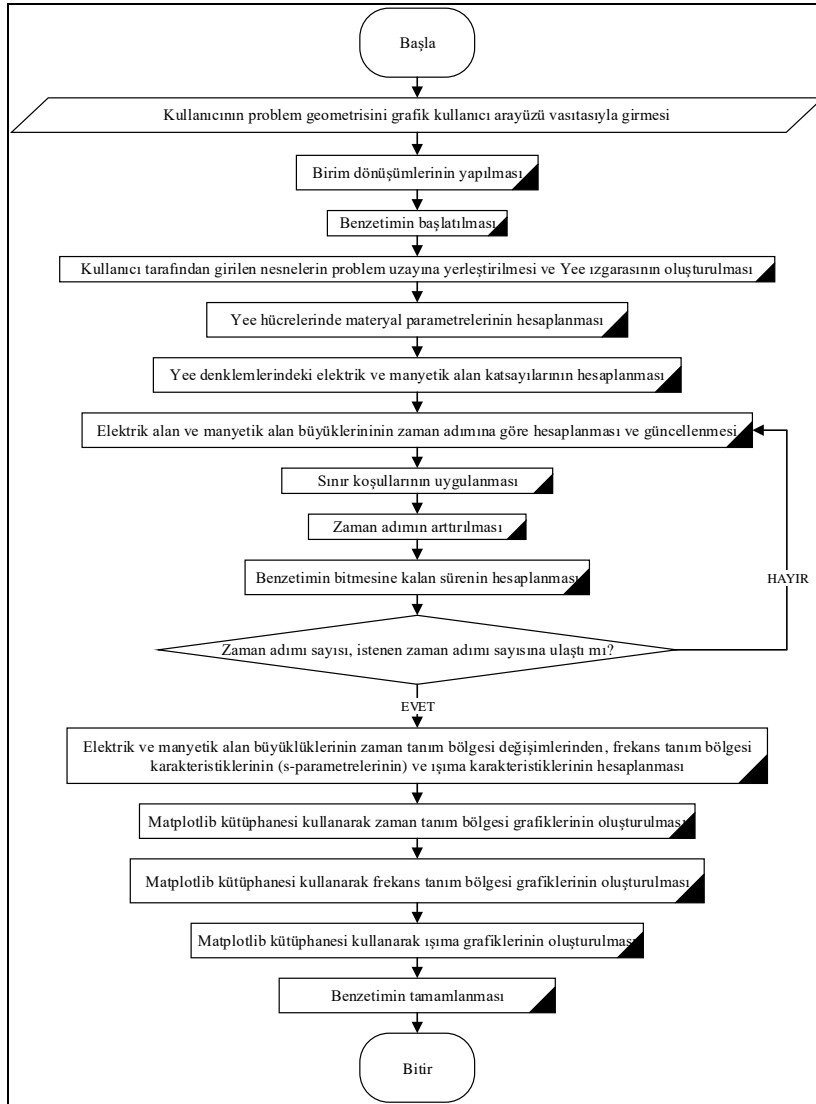
$$E_z(i, j, k) \rightarrow ((i-1)\Delta x, (j-1)\Delta y, (k-0.5)\Delta z) \quad (11)$$

$$H_x(i, j, k) \rightarrow ((i-1)\Delta x, (j-0.5)\Delta y, (k-0.5)\Delta z) \quad (12)$$

$$H_y(i, j, k) \rightarrow ((i-0.5)\Delta x, (j-1)\Delta y, (k-0.5)\Delta z) \quad (13)$$

$$H_z(i, j, k) \rightarrow ((i-0.5)\Delta x, (j-0.5)\Delta y, (k-1)\Delta z) \quad (14)$$

Yee hücrelerindeki elektrik alan ve manyetik alan bileşenleri, zaman tanım bölgesinde kullanıcı tarafından girilen benzetim frekans aralığına bağlı olan belli bir zaman adımı ile iteratif olarak hesaplanmaktadır. Problem uzayındaki sınır koşulları uygulanarak elektromanyetik problemin çözümü sonuçlandırılmakta ve benzetim sonucunda istenen büyüklükler, elde edilen elektrik ve manyetik alan bileşenlerinden yola çıkılarak hesaplanmaktadır. ZTBSF yönteminin kısa özeti bu şekilde olsa da, bu algoritmanın tam bir yazılım şeklinde gerçekleştirilmesi için, ayrıntıları Şekil 2’deki işaret-akış diyagramında verilen adımlar kullanılmıştır.



Şekil 2. Geliştirilen elektromanyetik benzetim yazılımının işaret-akış diyagramı.

Figure 2. Flowchart of the developed electromagnetic simulation software

## GELİŞTİRİLEN ELEKTROMANYETİK BENZETİM YAZILIMI (DEVELOPED ELECTROMAGNETIC SIMULATION SOFTWARE)

Bu çalışmada, Python programlama dilinde bir elektromanyetik benzetim yazılımı geliştirilmiştir. Kısaca belirtmek gerekirse, Python nesne tabanlı üst seviye bir programlama dilidir. Açık kaynaklı ve modüler yapısı sayesinde bilimsel hesaplamalar için yoğun bir şekilde kullanılmaktadır. Python diline gösterilen rağbet son yıllarda gittikçe artmış ve 2020 yılının en popüler programlama dillerinden biri olmuştur (Eastwood, 2020).

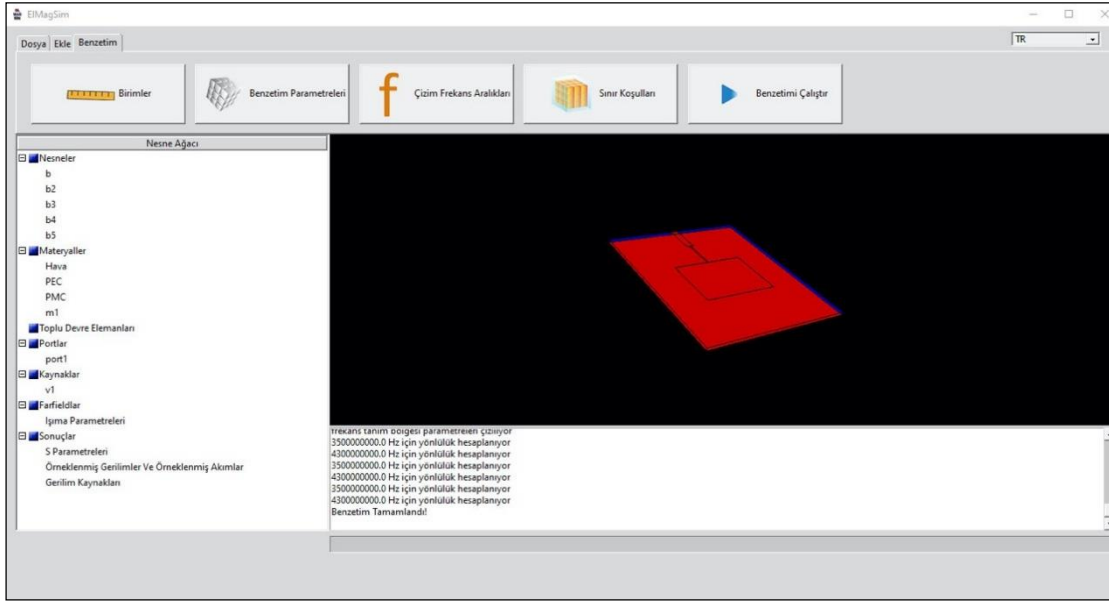
Geliştirilen yazılımın merkezinde, ZTBSF yönteminin gerçekleştirildiği ZTBSF kütüphanesi bulunmaktadır. Bu kütüphane, bu çalışma kapsamında Python programlama dilinde yazılmıştır ve içerisinde elektromanyetik benzetim fonksiyonları bulunmaktadır. Bu fonksiyonlar, çağrılabilir ve benzetim sonuçlarını liste olarak geri döndürebilir olarak yazılmıştır. Bu sayede, ZTBSF yöntemini gerçekleştiren ana fonksiyonlar arayüzden bağımsız olup, bu fonksiyonların başka kütüphanelerde yazılabilecek olan arayüzlerle de ileriye yönelik uyumlu olması hedeflenmiştir.

Yazılımda kullanılan grafik kullanıcı arayüzü Python programlama dilinde Tkinter arayüz kütüphanesi kullanılarak yazılmıştır. Tkinter, Python programa dilinin içerisinde bulunan bir arayüz oluşturma kütüphanesidir. Tkinter kullanılarak Windows, Mac OS veya Linux ortamında pencere oluşturulabilmekte ve pencere içerisine amacına uygun olarak butonlar, metin kutuları, açılır menüler gibi standart nesnelere eklenebilmektedir. Ayrıca, pencere içerisinde 2 boyutlu veya 3 boyutlu çizimler OpenGL tabanlı kütüphaneler ile yapılabilmektedir.

Bu çalışmada geliştirilen yazılımda kullanım kolaylığı sağlamak amacıyla kullanıcı arayüzü, ana pencere ve diyalog pencerelerinden oluşturulmuş ve modüler olarak yazılmıştır. Tkinter ile pencerelerin görünüşleri, boyutları, pencerelerin içerisinde bulunacak olan buton, menü, metin kutusu gibi nesnelere, gerçekleştirdikleri olaylar ve pencere içindeki konumları kod ile yazılmakta ve kullanıcıya gösterilmektedir. Oluşturulan projelerin saklanması için gereken dosya formatı ve ZTBSF metodunda gereken sayısal işlemler için numpy (Oliphant, 2006) kütüphanesi kullanılmıştır. Numpy bir bilimsel hesaplama kütüphanesi olup, bu kütüphane ile büyük boyutlu dizilerle matematiksel işlemler yapılabilmektedir. Diğer taraftan, elektromanyetik benzetim sonucu ortaya çıkan zaman tanım bölgesi dalga şekilleri, frekans tanım bölgesi grafikleri ve polar grafikler için ise matplotlib (Hunter, 2007) kütüphanesinden yararlanılmıştır. Matplotlib, 2 boyutlu ve 3 boyutlu grafik çizme imkânı sağlayan bir kütüphanedir. Geliştirilen yazılımda bulunan sonuç grafiklerine yakınlaştırma, uzaklaştırma, taşıma, grafik görüntüsünü resim biçiminde kaydetme ve grafik ayarları butonları da eklenmiştir.

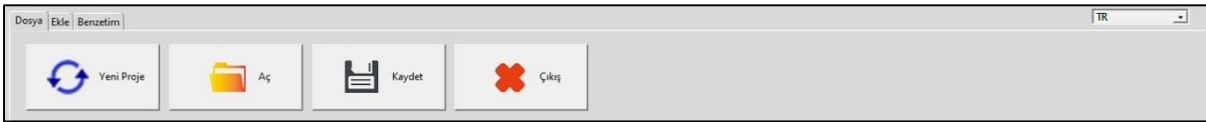
Kullanıcı tarafından girilen elektromanyetik problem geometrisinin 3 boyutlu olarak gösterilmesi için PyOpenGL (Fletcher ve Liebscher, 2005) ve PyOpenGLTk (Wright, 2018) kütüphaneleri kullanılmıştır. PyOpenGL 2 boyutlu ve 3 boyutlu gösterimler ve animasyonlar yapabilme imkânı sağlayan bir kütüphanedir. Gösterilen tasarımlara gölgelendirme ve ışıklandırma gibi birçok görsel işlem de uygulanabilmektedir. PyOpenGLTk ise PyOpenGL ve Tkinter arasına köprü görevi görmektedir. Grafik kullanıcı arayüzündeki "canvas" nesnesi PyOpenGL için sahne olarak kullanılmıştır. Gösterilen 3 boyutlu problem geometrisi yakınlaştırılabilir, uzaklaştırılabilir, taşınabilir ve döndürme işlemi yapılabilir. Kullanıcının girdiği nesnelere anında sahneye çizdirilmekte ve bu sayede sahne sürekli güncellenmektedir.

Hazırlanan arayüz ile benzetim yapacak olan kullanıcı, problem geometrisini grafiksel olarak tanıtmakta, daha sonra ise benzetim parametrelerini de girerek benzetimi gerçekleştirmekte ve sonunda da benzetim sonuçlarını görüntülemektedir. Kullanıcı arayüzünün genel görünümü, benzetim sekmesi aktifken Şekil 3'de gösterilmiştir. Arayüzün solundaki çerçevede nesnelere, materyallerin, devre elemanlarının, elektriksel portların, kaynakların ve sonuçların listelendiği nesne ağacı bölümü vardır. Arayüzün üst kısmında ise çeşitli aşamalar arasında geçiş yapmaya yarayan sekmeler ile gösterim dilinin seçildiği bölüm (Türkçe veya İngilizce) bulunmaktadır. Pencerenin ortasında ise girilen problem geometrisinin bulunduğu 3 boyutlu sahne ve bu sahnenin altında ise çalışmakta olan benzetim hakkında bilgi veren bir bölüm bulunmaktadır. Şekil 4'te ise genel dosya işlemlerinin yapıldığı sekme gösterilmiştir.



Şekil 3. Grafik kullanıcı arayüzünün genel görünümü

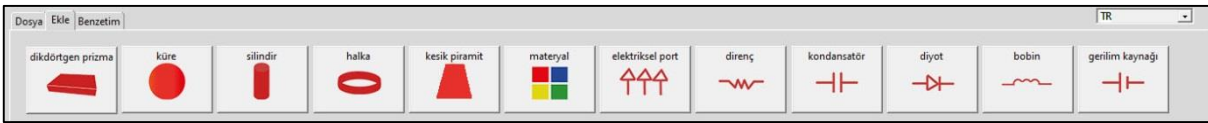
Figure 3. General view of the graphical user interface



Şekil 4. "Dosya" sekmesi

Figure 4. "File" tab

Kullanıcının benzetimini yapacağı problem geometrisi ise, Şekil 5'te gösterilen "Ekle" sekmesi vasıtasıyla yazılıma girilmektedir. Bu sekmede sırasıyla dikdörtgen prizma, küre, silindir, halka, kesik piramit geometrileri ekleme; yeni fiziksel materyal ekleme, elektriksel port ekleme, direnç, kondansatör, diyot, bobin ve gerilim kaynağı şeklindeki toplu devre elemanları ekleme kısımları bulunmaktadır.



Şekil 5. "Ekle" sekmesi

Figure 5. "Add" tab

Geometri ekleme butonlarına basıldığında bir diyalog penceresi çıkmakta ve bu pencerede, eklenecek olan geometrinin parametreleri kullanıcı tarafından girilmektedir. Dikdörtgen prizma ekleme penceresi Şekil 6'da, küre ekleme penceresi Şekil 7'de, silindir ekleme penceresi Şekil 8'de ve halka ekleme penceresi Şekil 9'da ve kesik piramit ekleme penceresi Şekil 10'da gösterilmiştir. Nesne ekleme pencerelerinde nesnenin kartezyen koordinat sistemindeki yeri, nesne ağacında görünecek olan ismi, materyal tipi gibi gerekli olan parametreler istenmektedir. Bu nesneler eklendiğinde, pencerenin merkezindeki 3 boyutlu sahne anlık olarak güncellenmektedir.

Bir Dikdörtgen Prizma Ekle

Dikdörtgen Prizma İsmi:

Minimum x:  mm

Minimum y:  mm

Minimum z:  mm

Maksimum x:  mm

Maksimum y:  mm

Maksimum z:  mm

Malzeme Tipi:

TAMAM İptal

Şekil 6. Dikdörtgen prizma ekleme penceresi  
Figure 6. Brick add window

Bir Küre Ekle

Küre İsmi:

Merkez x:  mm

Merkez y:  mm

Merkez z:  mm

Yarıçap:  mm

Malzeme Tipi:

TAMAM İptal

Şekil 7. Küre ekleme penceresi  
Figure 7. Sphere add window

Bir Silindir Ekle

Silindir İsmi:

Merkez x:  mm

Merkez y:  mm

Merkez z:  mm

Yarıçap:  mm

Yükseklik:  mm

Yön:  p:pozitif

Malzeme Tipi:

TAMAM İptal

Şekil 8. Silindir ekleme penceresi  
Figure 8. Cylinder add window

Şekil 9. Halka ekleme penceresi

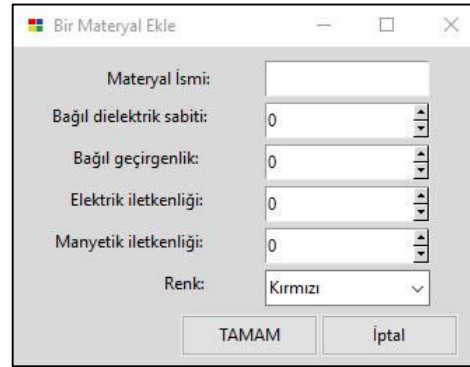
Figure 9. Torus add window

Şekil 10. Kesik piramit ekleme penceresi

Figure 10. Frustum pyramid add window

Yeni fiziksel materyal ekleme penceresi ise Şekil 11’de gösterilmiştir. Bu pencerede materyalin ismi, bağıl dielektrik sabiti, bağıl manyetik geçirgenliği, elektrik iletkenliği ve manyetik iletkenliği girilmektedir. Ayrıca, 3 boyutlu sahnede bu materyalden yapılmış olan nesnelerin çiziminde kullanılacak renk seçilebilmektedir.

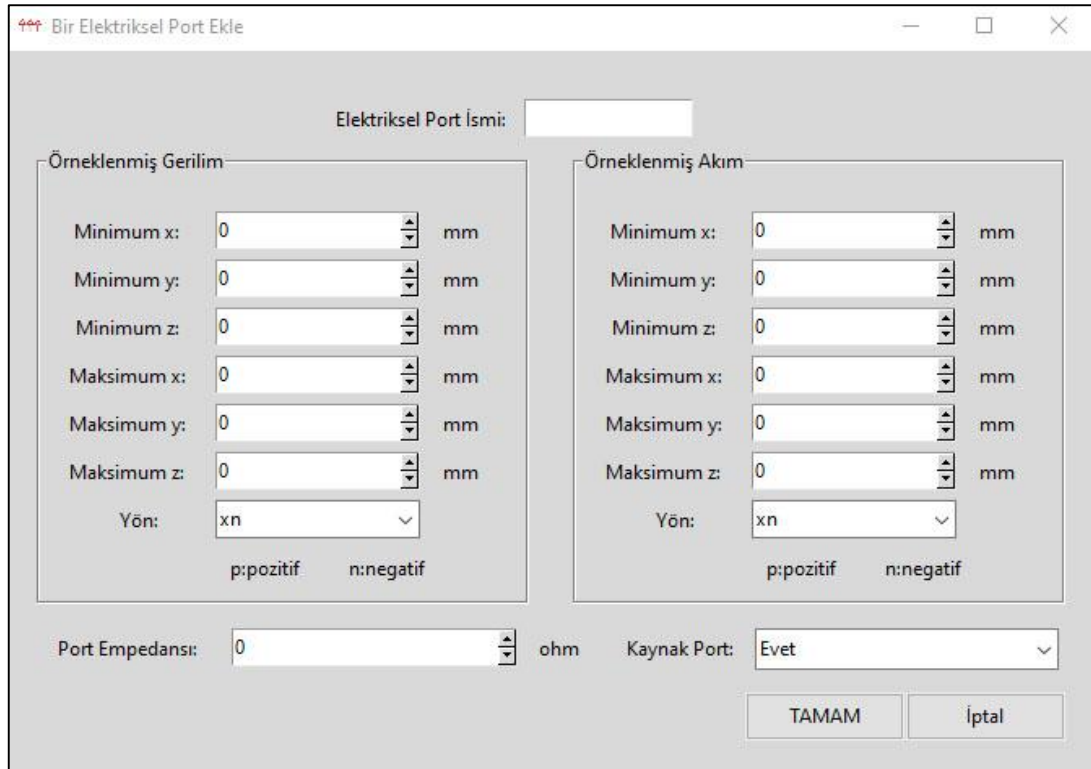




Şekil 11. Materyal ekleme penceresi

Figure 11. Material adding window

Elektriksel port ekleme penceresi ise Şekil 12’de gösterilmiştir. Pencerede, elektriksel portun örneklenmiş akım ve örneklenmiş gerilim için x, y ve z koordinatındaki minimum ve maksimum noktaları ile porta verilecek olan isim girilmekte ve yönü seçilmektedir. Aynı diyalog penceresinde port empedansı da girilmekte ve kaynak port olup olmadığı da işaretlenmektedir. Eklenmiş olan elektriksel portlar ana pencerenin sol kısmında bulunan nesne ağacında “Portlar” bölümü altında kullanıcının portlara verdiği isimlerle listelenmektedir.



Şekil 12. Elektriksel port ekleme penceresi

Figure 12. Electrical port adding window

Direnç ekleme penceresi Şekil 13’te gösterilmiştir. Pencerede direncin kartezyen koordinat sistemindeki yeri, yönü, direnç değeri ve nesne ağacında görülecek ismi girilir. Kondansatör, diyot ve bobinde de dirençte olduğu gibi açılan pencerelerde kartezyen koordinat sistemindeki yeri, ismi ve yönü girilmektedir. Kondansatör ekleme penceresinde kapasitans değeri ve bobin ekleme penceresinde bobin değerleri belirtilmektedir. Herhangi bir toplu devre elemanı eklendiğinde, ana penceredeki nesne ağacında “Toplu devre elemanları” bölümü altında kullanıcının elemanlara verdiği isimlerle listelenmektedir.

**Şekil 13.** Direnç ekleme penceresi

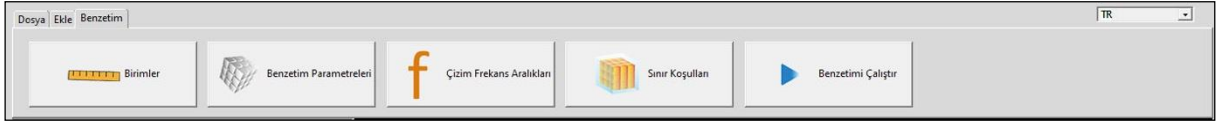
*Figure 13. Resistor add window*

Gerilim kaynağı ekleme penceresi ise Şekil 14’te gösterilmiştir. Pencerede gerilim kaynağının kartezyen koordinat sistemindeki yeri, ismi, dalga formu tipi ve yönü seçilir. Gerilim kaynağının iç direnç ve genliği de girilmektedir. Eklenen gerilim kaynakları, genel penceredeki nesne ağacında “Kaynaklar” bölümü altında kullanıcının verdiği isimlerle listelenir.

**Şekil 14.** Gerilim kaynağı ekleme penceresi

*Figure 14. "Add voltage source" window*

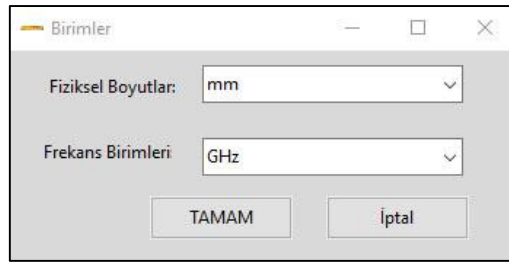
Benzetimi yapılacak olan problem geometrisi tanımlandıktan sonra ise son olarak ana penceredeki “Benzetim” sekmesi kullanılarak benzetim adımına geçilmektedir. Benzetim sekmesinde Şekil 15’de gösterildiği gibi sırasıyla “Birimler”, “Benzetim parametreleri”, “Çizim frekans aralıkları”, “Sınır koşulları” ve “Benzetimi çalıştır” butonları bulunmaktadır.



Şekil 15. Benzetim sekmesi

Figure 15. Simulation tab

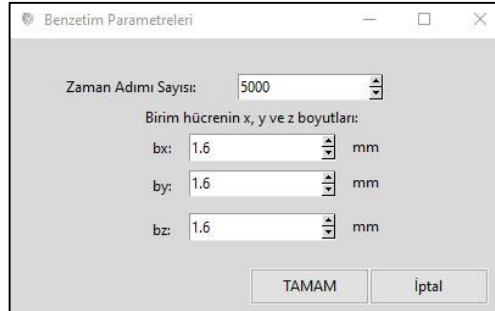
Birimler butonuna basıldığında gelen Birimler penceresi Şekil 16’da gösterilmiştir. Bu pencerede fiziksel boyutların ve frekansın birimleri seçilmektedir.



Şekil 16. Birimler penceresi

Figure 16. Units window

“Benzetim parametreleri” penceresi Şekil 17’de gösterilmiştir. Bu pencerede benzetimin zaman adım sayısını ve birim hücrenin (Yee hücresinin) boyutları girilmektedir.



Şekil 17. Benzetim parametreleri penceresi

Figure 17. Simulation parameters window

“Çizim frekans aralığı” butonuna basıldığında gelen çizim frekans penceresi ise Şekil 18’de gösterilmiştir. Bu pencerede sonuç grafiklerinde kullanılacak olan alt ve üst frekans değerleri girilmektedir.

Şekil 18. Çizim frekans aralığı penceresi

Figure 18. Plotting frequency range window

“Sınır koşulları” penceresi ise Şekil 19’da gösterilmiştir. Bu pencerede sınırın x, y ve z koordinatındaki pozitif ve negatif yönündeki hava boşluğu ve kalınlık değerleri girilir. Dış sınırdaki hücre sayısı girilir. Hava boşluğu nesnelere ve sınırlar arasındaki hücre sayısıdır. Kalınlık ise hücre kalınlığıdır. PEC (Perfect Electrical Conductor-Mükemmel Elektrik İletkeni) ve CPML (Convolutional perfectly matched layer-Evrişimli mükemmel uyumlu katman) olmak üzere iki çeşit sınır şartı kullanılabilir.

Şekil 19. Sınır koşulları penceresi

Figure 19. Boundary conditions window

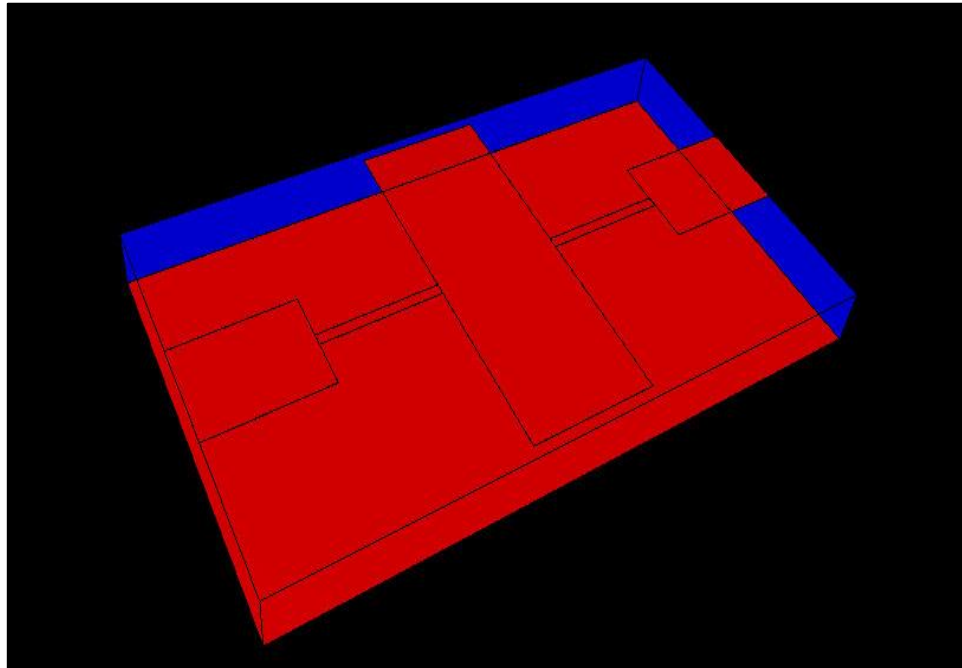
Son olarak ise “Benzetimi çalıştır” butonu ile elektromanyetik benzetim başlatılabilmektedir. Benzetim başladıktan sonra zaman adımları ve benzetime ilişkin anlık parametreler, ana penceredeki sahnenin altında akıcı şekilde gösterilmektedir. Benzetim sona erdikten sonra ise, hesaplanan elektrik ve manyetik alan büyüklüklerinden yola çıkılarak s-parametreleri, ışımaya karakteristikleri ve zaman tanım bölgesindeki dalga formları hesaplanarak matplotlib kütüphanesi vasıtasıyla çizdirilmektedir.

#### YAZILIMIN DOĞRULANMASI (VALIDATING THE SOFTWARE)

Yazılımı test etmek amacıyla farklı örnek yapıların elektromanyetik benzetimleri, geliştirilen yazılım ve ZTBSF metodunu temel alan ticari bir yazılım ile gerçekleştirilmiştir.

#### Mikroşerit Alçak Geçiren Süzgeç Benzetimi (Simulation of a Microstrip Lowpass Filter)

Alçak geçiren süzgeç, kesim frekansından düşük frekanslarda olan işaretleri ileten, daha yüksek frekanslarda olan işaretleri ise zayıflatan bir süzgeç çeşididir. Bu bölümde, bir mikroşerit alçak geçiren süzgecin hem geliştirdiğimiz yazılımda hem de ticari bir yazılımda benzetimleri yapılmıştır. Bu süzgeç devresinde alttaş (substrate) kalınlığı 1.8mm, alttaş bağlı dielektrik sabiti 4.6, süzgecin genişliği 13mm ve boyu ise 23.073mm’dir. Benzetime tabi tutulan mikroşerit süzgeç Şekil 20’de gösterilmiştir.

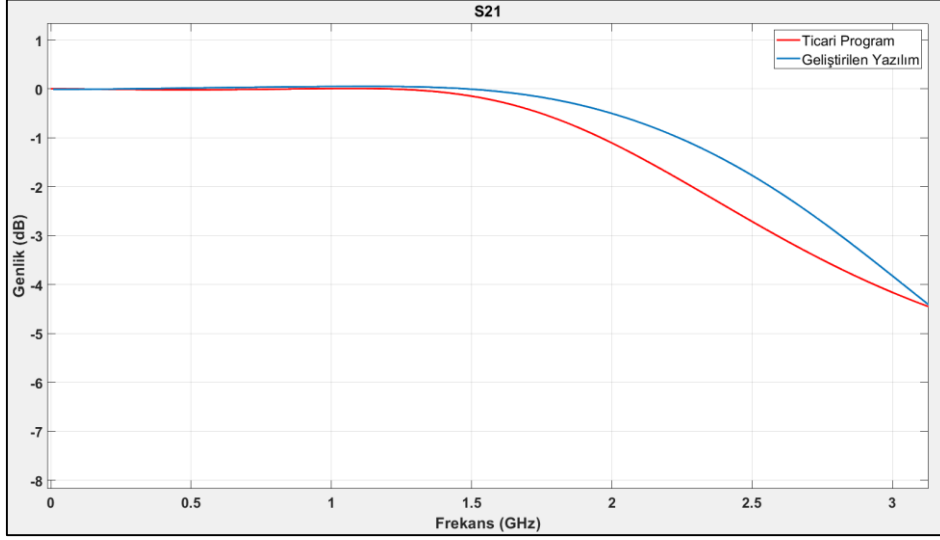


Şekil 20. Mikroşerit alçak geçiren süzgeç devresi.

*Figure 20. Microstrip lowpass filter circuit.*

Elektromanyetik benzetimler sonucunda, ticari yazılımdan ve geliştirilen yazılımdan elde edilen  $S_{21}$  karakteristikleri Şekil 21’de aynı eksen takımında çizdirilmiştir. Saçılma parametrelerinin, doğrusal devrelerin davranışlarını tanımlamak üzere kullanılan parametreler olduğunu hatırlatmakta fayda vardır. Mikrodalga mühendisliğinde yoğun olarak kullanılan ve diğer bir ifadeyle saçılma parametreleri olarak isimlendirilebilen s-parametrelerinden;  $S_{21}$  kompleks doğrusal gerilim kazancını,  $S_{11}$  ise girişteki kompleks gerilim yansımalarını ifade etmek için sıklıkla kullanılmaktadır. Mikrodalga devrelerinin davranışları genel

itibariyle frekansa bağlı olduğundan, s-parametrelerinin de frekansa bağlı olarak ifade edilmesi bir gerekliliktir; bu çalışmada geliştirilen yazılım da s-parametrelerinin benzetim neticesinde frekansa bağlı olarak elde edilmesine olanak sağlamaktadır.

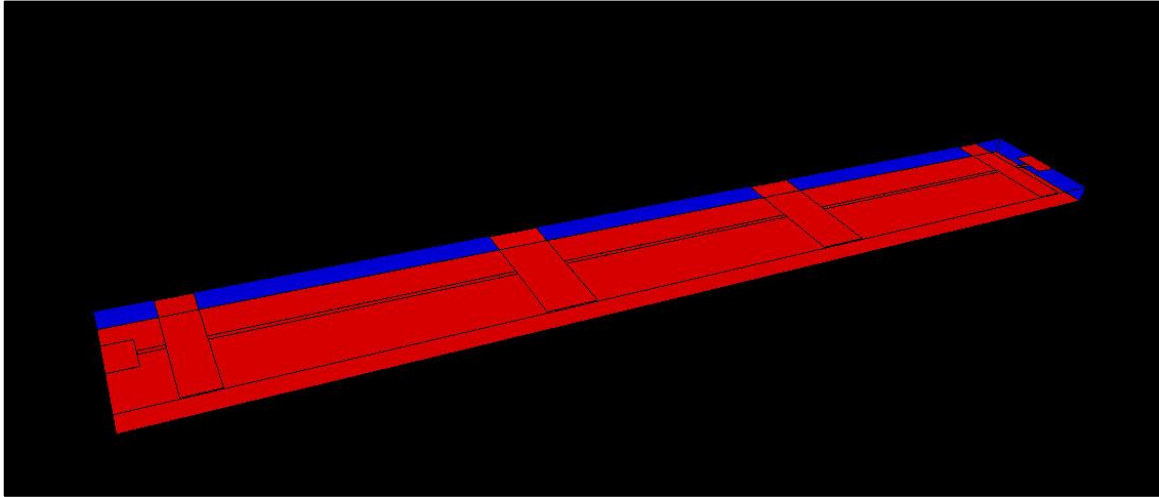


Şekil 21. Mikroşerit alçak geçiren süzgecin  $S_{21}$  karakteristikleri.

Figure 21.  $S_{21}$  characteristics of the simulated microstrip lowpass filter.

#### Mikroşerit Band Durduran Süzgeç Benzetimi (Simulation of a Microstrip Bandstop Filter)

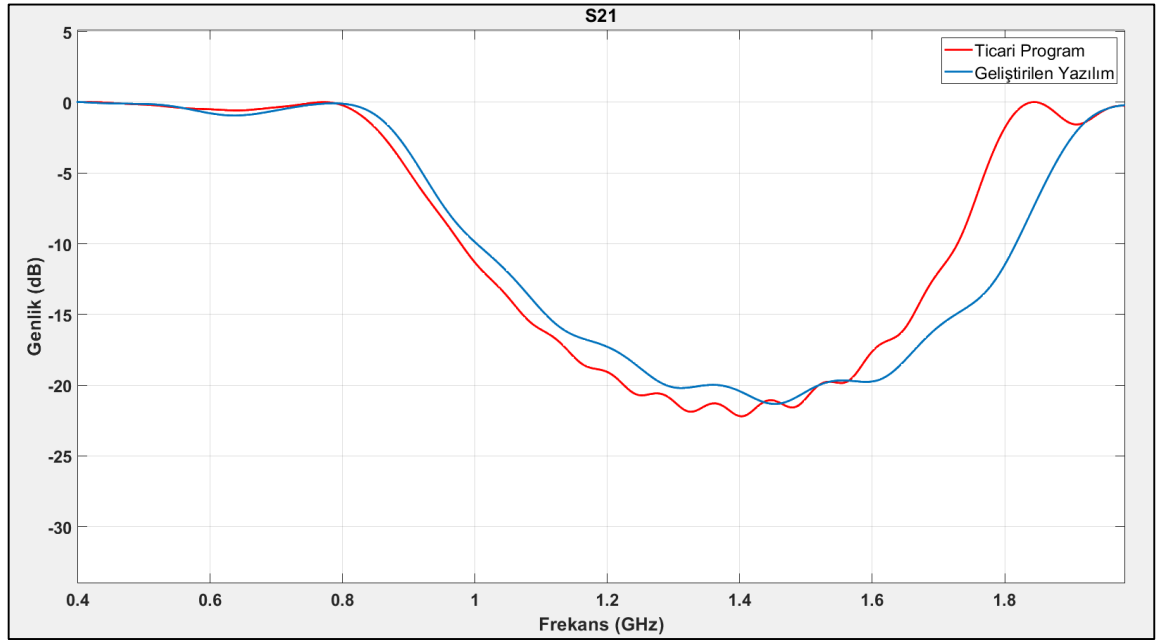
Band durduran süzgeç, belirli iki frekans arasındaki frekanslara sahip olan işaretleri zayıflatan ve o aralık dışındaki frekanslarda olan işaretleri geçiren bir devredir. İkinci örnek olarak, alttaş kalınlığı 0.64mm, genişliği 13.1mm, boyu 92.567mm ve alttaş bağlı dielektrik sabiti 5 olan ve Şekil 22’de gösterilen mikroşerit band durduran süzgeç devresi benzetime tabi tutulmuştur.



Şekil 15. Mikroşerit band durduran süzgeç devresi.

Figure 22. Microstrip bandstop filter circuit.

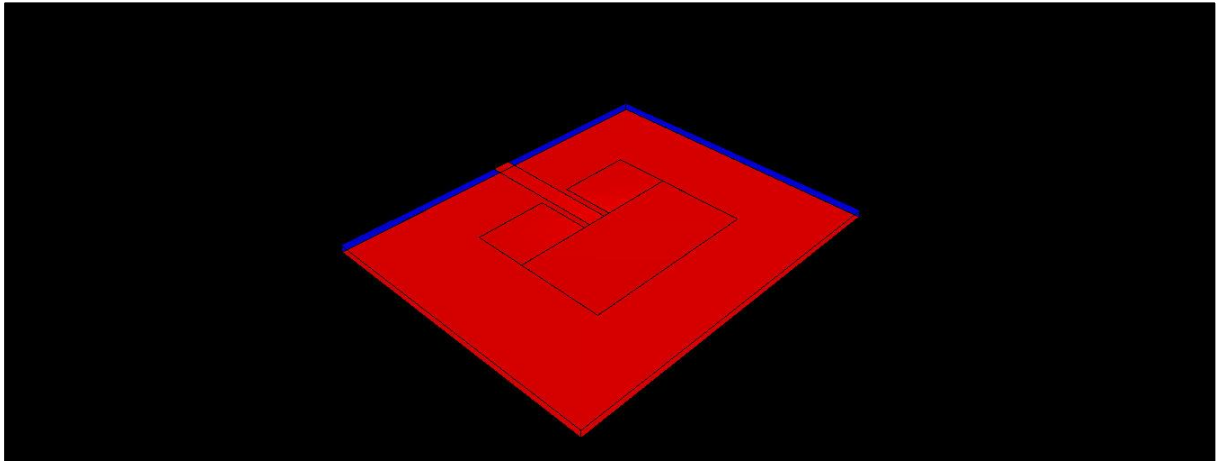
Mikroşerit band durduran süzgeç devresi hem bu çalışmada geliştirilen yazılımda hem de ticari yazılımda benzetime tabi tutulmuştur. Bu benzetimler sonucu elde edilen  $S_{21}$  grafikleri yine aynı eksen takımında çizdirilmiş ve Şekil 23’te verilmiştir.



Şekil 23. Band durdurucu süzgecin  $S_{21}$  grafikleri.  
Figure 23.  $S_{21}$  characteristics of the simulated microstrip bandstop filter.

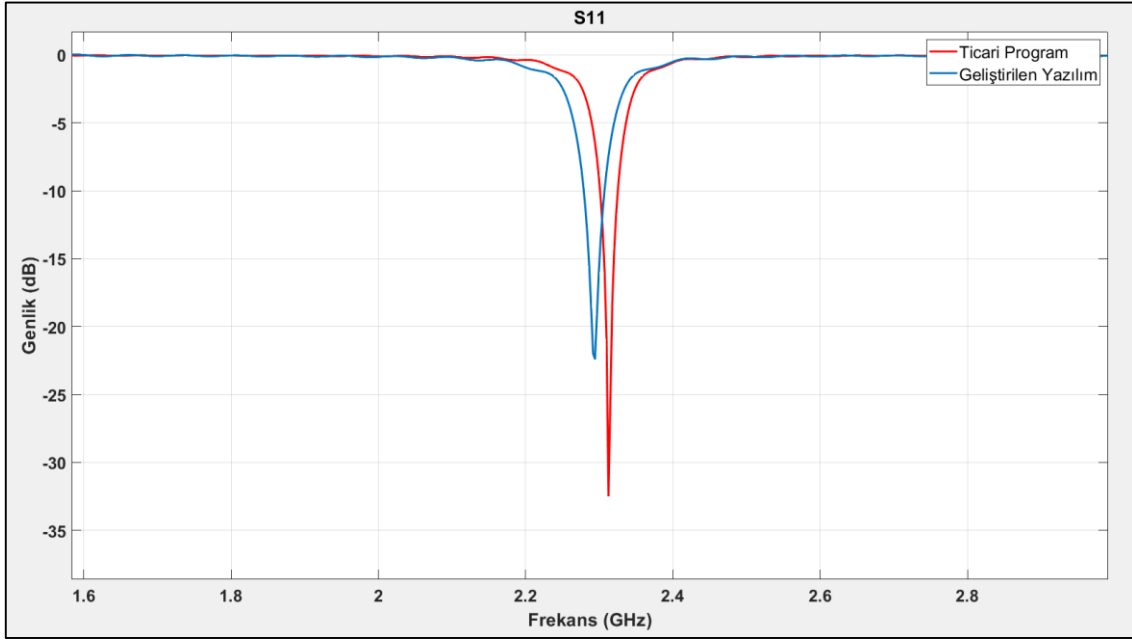
#### Ek Beslemeli Dikdörtgenel Yama Mikroşerit Anten Benzetimi (Simulation of an Inset Fed Rectangular Microstrip Patch Antenna)

Üçüncü olarak ise merkez frekansı 2.4GHz, yama uzunluğu 35.4529mm ve yama genişliği 44.1917mm olan ek beslemeli bir dikdörtgenel yama mikroşerit antenin benzetimleri hem bu çalışmada geliştirilen yazılımla hem de ticari yazılım ile yapılmıştır. Bu mikroşerit anten Şekil 24'te gösterilmiştir.

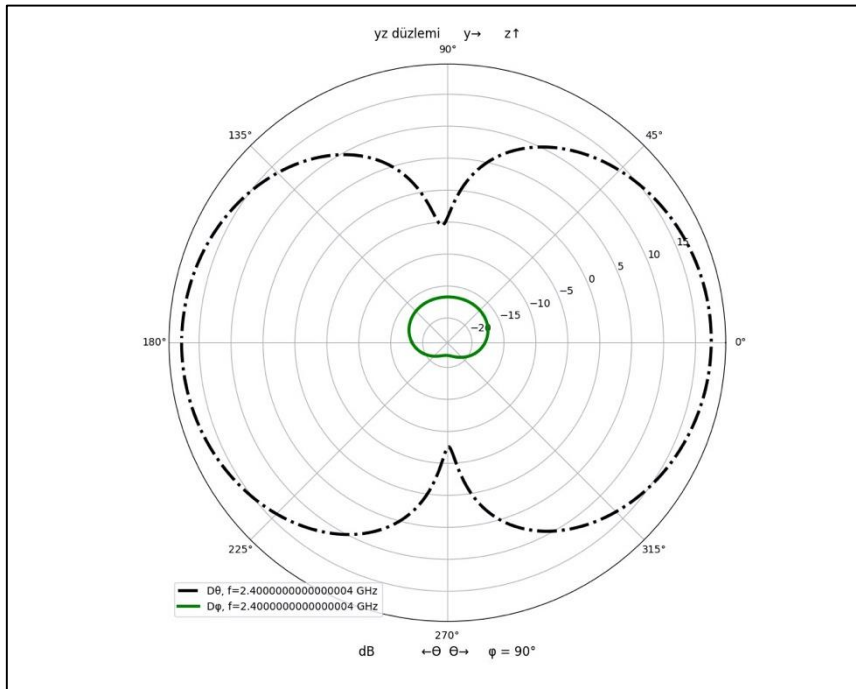


Şekil 24. Ek beslemeli dikdörtgenel yama mikroşerit anten  
Figure 24. Inset fed rectangular patch microstrip antenna

Söz konusu antene ilişkin hem bizim geliştirdiğimiz yazılımla hem de ticari yazılımda yapılan benzetimler sonucunda elde edilen  $S_{11}$  karakteristikleri Şekil 25'te verilmiştir. Ayrıca bu antenin her iki yazılımla elde edilen ışıma karakteristikleri de Şekil 26 ve Şekil 27'de gösterilmiştir. Işıma karakteristiklerinin yönler göre ışıma yoğunluğunu gösteren karakteristikler olduğunu hatırlatmakta fayda vardır.

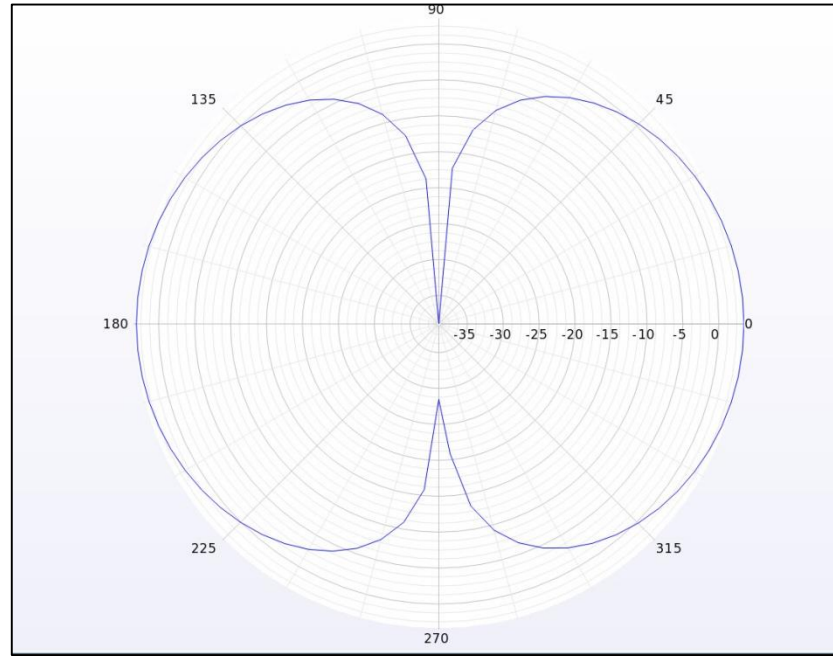


Şekil 25. Dikdörtgenel mikroşerit yama antenin  $S_{11}$  karakteristikleri.  
 Figure 25.  $S_{11}$  characteristics of the rectangular microstrip antenna.



Şekil 26. Geliştirilen yazılımdan elde edilen ışıma karakteristiği.  
 Figure 26. Radiation characteristics obtained from the developed software.





Şekil 27. Ticari yazılım ile elde edilen ışıma karakteristiği.  
Figure 27. Radiation characteristics obtained from a commercial software

## SONUÇLAR (RESULTS)

Bu çalışmada, Elsherbeni ve Demir tarafından MATLAB ortamında geliştirilen ZTBSF çözümünden hareketle, Python programlama dilinde rastgele geometrilerin elektromanyetik benzetimlerini icra etmek için bir yazılım geliştirilmiştir. Bu yazılım, Python'ın numpy nümerik kütüphanesi ve ilgili veri yapıları kullanılarak en baştan yazılmıştır. Ayrıca, kullanıcı kolaylığı sağlamak için Python'ın Tkinter kütüphanesi kullanılarak bir grafik kullanıcı arayüzü de geliştirilmiştir. Geliştirilen yazılımın temel faydaları şu şekilde izah edilebilir: i) ZTBSF yönteminin kullanıcının girdiği herhangi bir elektromanyetik probleme uygulanabilmesi, ii) kullanımı kolay bir arayüze sahip olması, iii) dikdörtgen prizması, küre, silindir, halka ve kesik koni gibi nesnelere, direnç, kondansatör, diyot, bobin gibi toplu devre elemanlarını, istenilen dalga formuna sahip gerilim kaynağı, elektriksel port ve istenen parametrelere sahip materyal ekleme olanağı, iv) elektromanyetik problem uzayındaki cisimlerin sahneye 3 boyutlu olarak çizilmesi ve bu şeklin yakınlaştırma, uzaklaştırma, sürükleme, bırakma ve çevirme özellikleri ile detaylı izlenebilmesi, v) benzetim sırasında kullanıcıya yapılan işlem, bitime kalan süre ve süreç durumu hakkında verilmesi ve vi) s-parametreleri, ışıma karakteristikleri ve zaman tanım bölgesindeki dalga şekillerinin kolaylıkla çizdirilebilmesi ve çizdirilen bu grafiklerin resim biçiminde kaydedilmesine olanak tanınmasıdır. Ayrıca, oluşturulan projeler istenildiği takdirde kaydedilip daha sonra tekrar açılabilir.

Yazılımın test edilmesi için üç farklı yapı, geliştirilen yazılım ve ticari olan bir programda elektromanyetik benzetimlere tabi tutulmuştur. Bu yapılar mikroşerit alçak geçiren süzgeç, mikroşerit band durdurucu süzgeç ve ek beslemeli dikdörtgensel mikroşerit yama antendir. Bu benzetim sonuçlarının gerek s-parametreleri gerekse de ışıma grafikleri anlamında uyumlu olduğu görülmektedir. Daha belirgin şekilde belirtmek gerekirse, Şekil 21'de verilen mikroşerit alçak geçiren süzgece ait  $S_{21}$  karakteristikleri incelendiğinde kesim frekansı ticari yazılımda 1.09GHz, bu çalışmada geliştirilen yazılımda ise 1.18GHz olarak tespit edilmiştir. Şekil 23'te verilen mikroşerit band durdurucu süzgece ait olan  $S_{21}$  karakteristikleri incelendiğinde ise band durdurucu aralığı ticari yazılımda 0.77GHz-1.84GHz, bu çalışmada geliştirilen yazılımda ise 0.78GHz-2.05GHz'dir. Ek beslemeli mikroşerit yama antenin Şekil 25'te gösterilen  $S_{11}$  karakteristikleri incelendiğinde, geliştirilen yazılımda antenin merkez frekansı 2.29GHz, ticari yazılımda ise merkez frekans 2.31GHz olarak elde edilmiştir. Yine bu yama antene ait olan Şekil 26 ve Şekil 27'de

verilen ışınma karakteristikleri incelendiğinde 2.4GHz frekansında ışınma şekillerinin bu çalışmada geliştirilen yazılımda ve ticari yazılımda elde edilmiş hallerinin benzer olduğu görülmektedir.

Sonuç olarak, bu çalışmada Python programlama dili kullanılarak grafik kullanıcı arayüzüne sahip olan ve ZTBSF yöntemini kullanan bir elektromanyetik benzetim yazılımı geliştirilmiştir. Çeşitli problemler için, bu çalışmada geliştirilen yazılım ve ticari olan bir elektromanyetik benzetim yazılımının benzetim sonuçlarının benzer olduğu gösterilmiş ve böylece geliştirilen yazılımın doğrulaması yapılmıştır. Rastgele problemlerin elektromanyetik benzetimlerinin yapılabildiği bu şekilde bir yazılımın, farklı özellikler eklenerek daha ileri bir noktaya taşınmasının hem ülkemizde hem de dünyada elektromanyetik benzetimlere ihtiyaç duyan araştırmacılara faydalı olacağı değerlendirilmektedir.

#### SEMBOLLER (SYMBOLS)

$E$	: Elektrik alan vektörü
$J$	: Akım Yoğunluğu
$H$	: Manyetik alan vektörü
Hz	: Hertz
dB	: Desibel
mm	: Milimetre
$\sigma$	: İletkenlik
$\epsilon$	: Dielektrik sabiti
$\mu$	: Manyetik geçirgenlik
$t$	: Zaman
ZTBSF	: Zaman bölgesinde sonlu farklar

#### KAYNAKLAR (REFERENCES)

- Elsherbeni A. ve Demir V., 2015. The Finite Difference Time Domain Method for Electromagnetics: With MATLAB Simulations, Scitech Publishing, 2nd ed.
- Christopoulos, C., 1995. The Transmission-Line Modeling Method: TLM. New York: IEEE Press.
- Eastwood, B., 2020. The 10 Most Popular Programming Languages to Learn in 2020, <https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>, erişim tarihi: 15.10.2020.
- FDTD Method, 2019. <https://www.remcom.com/xf-fdtd-method>, erişim tarihi: 14.10.2020.
- Fletcher, M., ve Liebscher, R., 2005. PyOpenGL, <http://pyopengl.sourceforge.net/>, erişim tarihi: 11.07.2020.
- Hunter, J. D, 2007. Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95.
- Jin, J, 1993. The Finite Element Method in Electromagnetics, Wiley.
- Lumerical Inc., FDTD Software, 2020. <https://www.lumerical.com/tcad-products/fdtd/>, erişim tarihi: 11.07.2020.
- Ney, M.M, 1985. Method of moments as applied to electromagnetic problems, IEEE Transactions on Microwave Theory and Techniques, vol. 33, no. 10, 972-980.
- Oliphant, T.E., 2006. A guide to NumPy, Trelgol Publishing USA.

- Oskooi, A., Roundy, D., Ibanescu, M., Bermel, P., Joannopoulos, J., ve Johnson, S. 2010. Meep: A flexible free-software package for electromagnetic simulations by the FDTD method. *Computer Physics Communications*, 181(3), 687-702.
- Remcom Inc., XF7 Software 7.3.0.3, 2020. <https://www.remcom.com/xfdtd-3d-em-simulation-software/>, erişim tarihi: 11.07.2020.
- Sullivan, D., 2020. *Electromagnetic Simulation Using the FDTD Method with Python*, Wiley-IEEE Press.
- Synopsys Inc., FullWAVE FDTD Software, 2020. <https://www.synopsys.com/optical-solutions/rsoft/passive-device-fullwave.html>, erişim tarihi: 11.07.2020.
- Wright, J., pyopenglTk, 2018. <https://github.com/jonwright/pyopenglTk>, erişim tarihi: 11.07.2020.
- Yee, K.S, 1966. Numerical solution of initial boundary problems involving Maxwell's equations in isotropic media, *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, 302-307.