YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NONLİNEER ELEKTROMANYETİK PROBLEMLERİN SONLU ELEMANLAR YÖNTEMİ İLE ÇÖZÜMÜNDE MALZEME KARAKTERİSTİKLERİNİN YAPAY SİNİR AĞLARI YÖNTEMİ İLE MODELLENMESİ

Elek. Yük. Müh. K. Nur DÖNMEZTÜRK BEKİROĞLU

F.B.E. Elektrik Mühendisliği Anabilim Dalında Hazırlanan

DOKTORA TEZİ

Tez Savunma Tarihi : 29 Temmuz 1998
Tez Danışmanı : Y. Doç. Dr. İbrahim ŞENOL (YTÜ)
Jüri Üyeleri : Prof. Dr. Atıf URAL (Kocaeli Üniversitesi)
Prof. Dr. Nusret YÜKSELER (İTÜ)

İSTANBUL, 1998

İÇİNDEKİLER

SİMGE LİSTESİ v

ÖNSÖZ vi

ÖZET vii

ABSTRACT viii

1. ELEKTROMANYETİK ALANLAR 1

1.1 Tarihi Gelişim 1
1.2 Statik Elektrik Alanlar 2
1.2.1 Coulomb yasası 2
1.2.2 Gauss yasası 3
1.2.3 Poisson ve Laplace denklemleri 4
1.2.4 Sınır şartları 5
1.3 Statik Manyetik Alanlar 7
1.3.1 Ampére yasası 8
1.3.2 Bio-Savart yasası 9
1.3.3 Sınır şartları 9
1.4 Maxwell Denklemleri 12
1.4.1 Faraday yasası 14
1.4.2 Elektrik ve manyetik alanlar için Gauss yasası 15
1.4.3 Ampére yasası ve deplasman akımı 15
1.5 Kısmi Türevli Diferansiyel Denklemlerin Çözümünde Kullanılan Sayısal Yöntemler 16
1.5.1 Sonlu farklar yöntemi 16
1.5.2 Monte Carlo yöntemi 18
1.5.3 Yük benzetim yöntemi 21

2. SONLU ELEMANLAR YÖNTEMİ 23

2.1 Sınır Değer Problemleri 23
2.1.1 Sınır değer probleminin çözümünde Ritz yöntemi 23
2.1.2 Sınır değer problemlerinin çözümünde Galerkin yöntemi 25
2.2 Sonlu Elemanlar Yöntemi 26
2.3 Sonlu Elemanlar Yönteminin Temel Adımları 27
2.3.1 Bölgenin ayrıştırılması 28
2.3.2 Enterpolasyon fonksiyonlarının seçilmesi 28
2.3.3 Denklem sisteminin formülasyonu 29
2.3.4 Denklem sisteminin çözümü 29
2.4 Sonlu Elemanlar Yöntemi ile İki Boyutlu Problemlerin Çözümü 30
2.4.1 Sınır değer teoremi 30
2.4.2 Varyasyonel formülasyon 32
2.4.3 Sonlu eleman analizi 36

2.4.3.1	Bölgenin ayrıştırılması	36
2.4.3.2	Eleman enterpolasyon fonksiyonları	38
2.4.3.3	Ritz yöntemi yoluyla formülasyon	40
3.	SONLU ELEMANLAR YÖNTEMİ İLE NONLİNEER PROBLEMLERİN ÇÖZÜMÜ VE υ-B^2 EĞRİLERİNİN SAYISAL MODELLENMESİ	48
3.1	Newton-Raphson Yöntemi ile Nonlineer Sistem Eşitliklerinin Çözümü	48
3.2	Newton-Raphson Yöntemi ile Nonlineer İzotropik Problemlere Birinci Dereceden Elemanlar için Uygulanışı	52
3.3	Elektriksel Malzemelerin υ-B^2 Eğrilerinin Elde Edilmesi	56
3.3.1	Üstel fonksiyon yaklaşımı	57
3.3.1.1	El-Sherbiny yöntemi	57
3.3.1.2	Brauer yöntemi	58
3.3.2	Kübik spline yöntemi	59
3.3.2.1	Parçalı lineer enterpolasyon	59
3.3.2.2	Parçalı kübik spline	61
3.3.2.3	Kübik splineların varlığı	62
3.3.2.4	Kübik splineların oluşturulması	64
3.3.2.5	Kübik splineların uygunluğu	66
4.	YAPAY SİNİR AĞLARI	71
4.1	Yapay Sinir Ağlarının Yapısı ve İşlem Elemanı	71
4.2	Bağlantı Geometrileri	74
4.3	Ağ Tipleri	75
4.4	Eşik Fonksiyonları	75
4.5	Ağırlık Uzayı	76
4.6	Yapay Sinir Ağlarında Eğitme Algoritmaları	77
4.7	Yapay Sinir Ağlarında Bellek	78
4.8	Yapay Sinir Ağlarında Hata Toleransı	79
4.9	Öğrenme Kuralları	79
4.10	Perceptron	80
4.11	Çok Katmanlı Perceptron	82
4.12	Hatanın Geriye Yayılması Algoritması ve Genelleştirilmiş Delta Kuralı	83
4.13	Öğrenme ve Momentum Katsayıları	88
5.	YAPAY SİNİR AĞLARI YÖNTEMİ İLE υ-B^2 EĞRİLERİNİN MODELLENMESİ	93
5.1	Mıknatıslanma Eğrisi ve υ-B^2 Eğrisinin Elde Edilmesi	89
5.2	υ-B^2 Eğrisinin Kübik Spline ile Modellenmesi	92
5.3	Yapay Sinir Ağları ile Modelleme için Hatanın Geriye Yayılması Algoritması	94

5.4 Yapay Sinir Ağları ile Modellemede Kullanılan Ağ Mimarisi..... 100

6. YAPAY SİNİR AĞLARINI SONLU ELEMANLAR YÖNTEMİNDE KULLANARAK DOĞRU AKIM MAKİNASININ MANYETİK ALAN ANALİZİ..... 102 102

6.1 Hava Aralığındaki Akı Yoğunluğu Dağılışının Analitik Olarak Hesaplanması..... 102
6.2 Gerçek Makinada Hava Aralığında Akı Yoğunluğunun Hesabı..... 104
6.3 Kullanılan Sonlu Elemanlar Modeli..... 105
6.3.1 Makinanın geometrisinin tanıtılması..... 107
6.3.2 Sonlu eleman yaklaşımı..... 109
6.3.3 Kullanılan ağ yapısı..... 109
6.4 Elde Edilen Alan Dağılımları..... 111
6.4.1 Lineer çözümle elde edilen alan dağılımları..... 111
6.4.2 Nonlineer çözümle elde edilen alan dağılımı..... 113
6.5 Değişik Hava Aralıkları ve Kutup Yüzey Şekilleri için Alan Dağılımları ve Akı Yoğunlukları 116

7. SONUÇLAR VE ÖNERİLER 119

KAYNAKLAR 121

EKLER 125

ÖZGEÇMİŞ 163

SİMGE LİSTESİ

$\mathbf{A}$	Manyetik vektör potansiyel
$\mathbf{B}$	Manyetik akı yoğunluğu vektörü
{b}	Kaynak vektörü
$\mathbf{D}$	Elektriksel akı yoğunluğu vektörü
D	Endüi dış çapı
$d\Omega$	Üçgen eleman yüzey normal vektörü
$\mathbf{E}$	Elektrik alan vektörü
F	Fonksiyonel
$\mathbf{H}$	Manyetik alan şiddeti vektörü
I	Akım şiddeti
$\mathbf{J}$	Akım yoğunluğu vektörü
$\mathbf{K}$	Katsayılar matrisi
l	Hava aralığı uzunluğu
l_a	Endüi uzunluğu
N	Sarım sayısı
$\mathbf{P}$	Jacobian matrisi
Q	Yük
S	Kutup yüzey kesiti
V	Elektrik potansiyel
W	Enerji
ρ	Hacimsel yük yoğunluğu
ε	Permitivite
μ	Geçirgenlik (permeabilite)
∇	Nabla operatörü
υ	Manyetik relüktivite
α	Momentum katsayısı
δ	Hata terimi
Φ	Akı
μ_o	Boşluğun permeabilitesi
$\Re$	Hava aralığı relüktansı

ÖNSÖZ

Bu tezin ortaya çıkmasında büyük sabır ve özveri göstererek, yardımlarını esirgemeyen sayın hocam Y. Doç. Dr. İbrahim ŞENOL'a, yapıcı önerileri ile çalışmama yön veren sayın hocalarım Prof. Dr. Atıf URAL ve Prof. Dr. Nusret YÜKSELER'e, çalışmam boyunca bana çok yardımcı olan arkadaşım Elektrik Yüksek Mühendisi Ali İhsan ÇANAKOĞLU'na, bana destek olarak büyük bir sabır ve anlayış gösteren sevgili eşim Baybars'a ve başta Sibel olmak üzere tüm arkadaşlarıma teşekkürlerimi sunarım.

Daima yanımda olan, yardımını hiçbir zaman esirgemeyen canım anneme sonsuz teşekkürler.

ÖZET

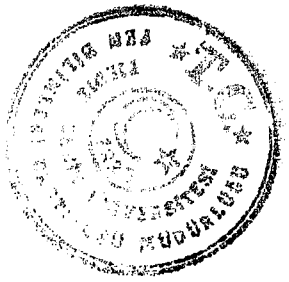
Doğru akım makinalarının kaçak alan analizi, sayısal olarak, sonlu farklar yöntemi ile yapılmaktaydı. Fakat günümüzde, sonlu elemanlar yöntemi için geliştirilen paket programlarının sağladığı kolaylıktan ötürü sonlu elemanlar yöntemi tercih sebebi olmuştur. Tasarım aşamasında doğru akım makinalarının kutup yüzey şekli oldukça önemli bir faktördür. Sonlu elemanlar yöntemi ile yapılan incelemelerde, kutup yüzey şeklinin kübik spline gibi bir eğri modelleme yöntemi ile modellenerek, optimizasyonunun anlatıldığı çalışmalar yapılmıştır. Ancak problem lineer olarak incelenmiştir.

Bu çalışmada yapay sinir ağları sonlu elemanlar yönteminde kullanılarak, doğru akım makinalarının kaçak alan incelemesi yapılmıştır. Kutup ayağı için kullanılan malzemenin mıknatıslanma karakteristiği yapay sinir ağları ile modellenmiştir. Sonlu elemanlar yöntemi ile nonlineer olarak incelenen alan hesabı sonucunda hava aralığı optimize edilmiştir. En uygun kutup yüzey şekli belirlenmiştir. Elde edilen sonuçlar deneysel sonuçlarla karşılaştırılmış ve doğruluğu ispatlanmıştır.

Sonlu elemanlar yöntemi ile nonlineer alan problemlerinin çözümü yapılırken, elektriksel malzemelerin mıknatıslanma eğrilerinin modellenmesinde, kübik spline yöntemi tercih edilmektedir. Yöntem, eğrinin hızlı değiştiği noktalarda büyük hata verdiğinden, eğri bu noktalardan birkaç parçaya bölündükten sonra modelleme işlemi yapılmaktadır. Bu durum modelleme maliyetini yükseltmektedir. Eğrilerin yapay sinir ağları ile modellenmesiyle bu dezavantaj ortadan kaldırılmıştır. Yapay sinir ağları yönteminde öğrenme işlemi için hatanın geriye yayılması algoritması kullanılmıştır.

Doğru akım makinasının kaçak alan çözümü yapıldıktan sonra, boşta çalışma için eşpotansiyel noktalar grafik olarak elde edilmiştir. Hava aralığı boyunca akı yoğunluğu hesaplanarak uygun kutup yüzey şekli optimize edilmiştir.

Birinci bölümde elektromanyetik alanlar hakkında genel bilgi verilip, kısmi türevli diferansiyel denklemlerin çözümünde kullanılan sayısal yöntemler açıklanmıştır. İkinci bölümde, sonlu elemanlar yöntemi ve yöntemin temel adımları hakkında teorik bilgi verilip, iki boyutlu problemlerin sonlu elemanlar yöntemi ile çözümünde bu temel adımlar uygulanarak, örnek bir ağ yapısı üzerinde denklem sistemi çözülmüştür. Üçüncü bölümde nonlineer problemlerin sonlu elemanlar ile çözümünde nonlineer izotropik problemlere Newton Raphson yönteminin birinci dereceden elemanlar için uygulanışı verilmiştir. υ-B^2 eğrilerinin sayısal modellenmesinde üstel fonksiyon yaklaşımları ve kübik spline yöntemi açıklanmıştır. Dördüncü bölümde yapay sinir ağlarının teorisi verilmiştir. Beşinci bölümde seçilen doğru akım makinasının mıknatıslanma eğrisi kullanılarak υ-B^2 eğrisi elde edilmiş ve bu eğri kübik spline ve yapay sinir ağları ile modellenerek sonuçlar karşılaştırılmıştır. Altıncı bölümde, makinanın sonlu elemanlar yöntemi ile modellenmesinde kullanılan ağ yapısı verilmiş, lineer ve nonlineer çözümle elde edilen alan dağılımları analitik çözümle karşılaştırılmış, hava aralığı optimize edilerek, en uygun kutup yüzey şekli belirlenmiştir. Son bölüm olan yedinci bölümde ise sonuçlar tartışılmış ve önerilerde bulunulmuştur.



ABSTRACT

Leakage field analysis of direct current machines used to be calculated numerically by the finite difference method. Today, finite elements method is preferred, as computer programs for finite elements method are available. Shape of pole surface of the direct current machines is an important factor in design stage. In studies using the finite elements method, the optimization of the pole surface is explained by modeling its shape with a curve modeling method as the cubic spline. However, the problem is examined linearly.

In this present study, leakage field of the direct current machines is examined by using artificial neural networks in finite elements method. Magnetization characteristic of the material used in the pole leg is modeled with artificial neural networks. The air gap is optimized as a consequence of field calculation examined nonlinearly with the finite elements method. The most appropriate shape of the pole surface is determined. Results obtained are compared with the experimental results and proved to be accurate.

The cubic spline is a preferred method in the modeling of magnetization curves of electrical materials when solving nonlinear field problems with finite elements method. As the method results with large errors at the points the curve changes drastically, modeling is executed following the division of the curve into several parts at these points. This operation increases modeling cost. This flaw is corrected by modeling the curves with artificial neural networks. For learning processes in the artificial neural networks, the back-propagation training algorithm is used.

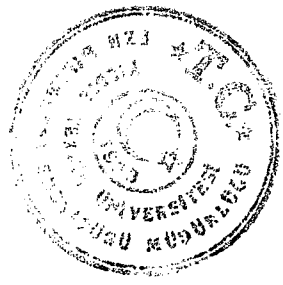
After controlling the leakage field of direct current machine, equipotential points are obtained graphically for working on noload. Appropriate pole surface shape is optimized by calculating the flux density across the air gap.

In the first chapter, general information on electromagnetic field is presented and numerical methods used to solve differential equations with partial derivatives are explained. In the second chapter, theoretical information on the finite elements method and its basic steps are presented and these are applied for solving two dimensional problems with the finite elements method in an example where the equation system is solved on a net system sample. In the third chapter, the application of Newton-Raphson method on first-order elements when solving nonlinear problems with the finite elements method for nonlinear isotropic problems is presented. Exponential functions approach and cubic spline method are explained in numerical modeling of υ-B^2 curves. At fourth chapter, artificial neural networks theory is presented. In the fifth chapter, the υ-B^2 curve is obtained by using the magnetization curve of the chosen direct current machine This curve is then modeled with cubic spline and artificial neural networks and the results are compared. In the sixth chapter, mesh pattern used for modeling the machine with the finite elements method is given; field distributions obtained by the analytic solution and the most appropriate pole shape is defined by optimizing the air gap. In the seventh chapter, which is the last one, results are discussed and suggestions are offered.

1. ELEKTROMANYETİK ALANLAR

1.1. Tarihi Gelişim

M.Ö. 6. yüzyılda Yunanlı filozof Thales, amber ile ipeğin birbirine sürtülmesi durumunda kıvılcım oluştuğunu, tüy ve hasır parçacıklarını çeken bir kuvvetin var olduğunu görmüştür. Amber'in yunancası "elektron" dur ve buradan da elektrik, elektron ve elektronik kelimeleri elde edilmiştir. M.S. 1600 yıllarında William Gilbert, elektrik ve manyetik olayların ilk sistematik deneylerini yapmış ve elektrostatik etkileri ölçen elektroskobu icat etmiştir. Aynı zamanda dünyanın tek başına büyük bir mıknatıs olduğunu da belirtmiştir. 1750 yıllarında Benjamin Franklin yükün korunumu yasasını bulmuş ve polariteleri pozitif ve negatif olarak isimlendirmiştir. Daha sonra Fransız Coulomb elektrik ve manyetik kuvvetleri ölçmüştür. Bu zaman süresince Alman matematikçi Gauss, belli bir hacim ve yüzeyle ilgili diverjans teoremini formüle etmiştir. 1800'de Volta, bugün kendi adını taşıyan pili icat etmiş, piller ile elektrik akımını oluşturmuş ve 1819'da Oersted akım taşıyan bir telin, yakınındaki pusulayı saptırdığını görmüş, elektriğin mıknatıslanma oluşturduğunu keşfetmiştir. Bunu takib eden yıllarda Ampére manyetik alanı oluşturabilmek için selenoid bobinini bulmuş ve bir mıknatıs içindeki atomların, içlerinde dolaşan küçük elektrik akımı tarafından mıknatıslandığını ispat etmiştir. Bu dönemde George Simon Ohm akım, gerilim ve direnç arasında bilinen ohm kanununu yayınlamıştır. 1813 yılında Michael Faraday değişen bir manyetik alanın, elektrik akımına neden olduğunu kanıtlamıştır. 1873'te Maxwell, ışığın elektromanyetik olduğunu söylemiş ve elektromanyetik teoriyi oluşturmuştur. 1888'de Hertz radyo dalgalarını tespit etmiş ve radyo dalgalarının polarizasyonu, yansıması ve kırılmasının ışık ile aynı olduğunu deneyle kanıtlamıştır. Ardından Marconi radyoyu daha kullanışlı hale getirmiştir. Edison elektrik ampulünü icat etmiş ve ilk elektrik güç sistemlerini kurmuştur. Daha sonra Tesla, gücün alternatif akımla iletimini bulmuş ve ilk indüksiyon motorunu yapmıştır. 1905 yıllarında Einstein kendi teorisi doğrultusunda Maxwell eşitliklerini evrensel hale getirmiştir.



1.2. Statik Elektrik Alanlar

Duran elektrik yüklerinin oluşturduğu çekme ve itme kuvvetlerinin neden olduğu olaylar elektrostatik adını alır. Hareketsiz elektrik yükleri tarafından yaratılan alana elektrostatik alan denir. Bir $\vec{E}$ (x, y,z) elektrostatik alanına karşılık öyle bir eğri ailesi bulunabilir ki, bu ailenin teğetleri değme noktalarında elektrostatik alan vektörlerine paraleldir. Bu eğri ailesine elektrostatik alanın alan çizgileri veya kuvvet çizgileri denir.

Bu bölümde, Coulomb'un deneysel kuvvet yasasından başlanarak, Gauss yasası, Poisson ve Laplace denklemleri ve son olarak sınır şartlarından bahsedilecektir.

1.2.1. Coulomb yasası

Coulomb yasasına göre; kuvvet, yükün büyüklüğü ile orantılıdır.

$$F = k.\frac{Q_1.Q_2}{r^2} \qquad (1.1)$$

Burada;

F : İki nokta arasında meydana gelen kuvvet, Newton (N)

Q_1 ve Q_2 : Yükün niceliği, Coulomb (C)

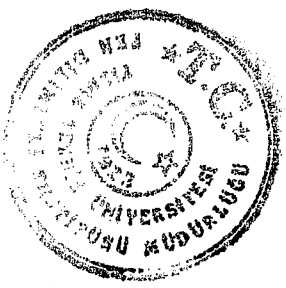
r : Yükler arasındaki uzaklık, metre (m)

k : Orantı sabiti

Eğer Q_1 ve Q_2 aynı işaretli ise, kuvvet itme, ters işaretli ise çekme kuvvetidir. Uluslararası sistemde k orantı sabiti;

$$k = \frac{1}{4\pi\varepsilon} \qquad (1.2)$$

şeklindedir. (Kraus, 1992)



Burada, ε , ortamın permitivitesidir. Boşluk ve hava için ε_0 kullanılacaktır.

$$\varepsilon_0 = 8.85 \times 10^{-12} \quad F / m \tag{1.3}$$

1.2.2. Gauss yasası

D deplasman vektörünün kapalı bir yüzeyden geçirdiği akı, bu yüzeyin kapladığı bölge içinde bulunan yüklerin cebirsel toplamına eşittir. **dS** kapalı yüzeyine ait bir vektör olmak üzere;

$$\oint_{(s)} \mathbf{D}.d\mathbf{S} = \Sigma Q \tag{1.4}$$

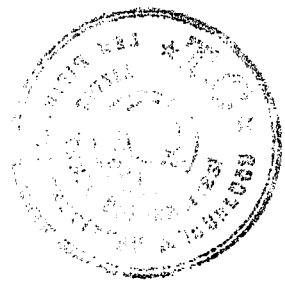
Eğer, kapalı yüzeyin hacmi v ile gösterilir ve bu yüzey içinde bulunan toplam yüklerin birim hacme isabet eden miktarı ρ hacimsel yük yoğunluğu meydana getiriyorsa, toplam v hacmi içindeki yük;

$$\Sigma Q = \int_{(v)} \rho \, d\, v \tag{1.5}$$

şeklindedir. (1.4) eşitliği uyarınca ΣQ yerine yazılırsa, Gauss teoreminin matematiksel ifadesi;

$$\oint_{(s)} \mathbf{D}.d\mathbf{S} = \int_{(v)} \rho dv \tag{1.6}$$

olarak bulunur. (Ergeneli, 1986)



1.2.3. Poisson ve Laplace denklemleri

Gauss teoreminin diferansiyel ifadesi;

$$\nabla . \mathbf{D} = \rho$$

olmak üzere, bu eşitlikte $\mathbf{D} = \varepsilon \mathbf{E}$ ve $\mathbf{E} = - \nabla V$ yazılırsa;

$$\mathbf{D} = - \varepsilon \nabla V$$

$$\nabla . \nabla V = -\frac{\rho}{\varepsilon}$$

olur. Gradyanın diverjansı ∇^2 operatörü ile gösterilir ve bu Laplace operatörü olarak adlandırılır. Böylece, Poisson eşitliği aşağıdaki şekilde yazılabilir.

$$\nabla^2 V = -\frac{\rho}{\varepsilon} \qquad (1.7)$$

Eğer $\rho = 0$ olursa; Poisson eşitliği Laplace eşitliği olarak bilinen şekle gelir.

$$\nabla^2 V = 0 \qquad (1.8)$$

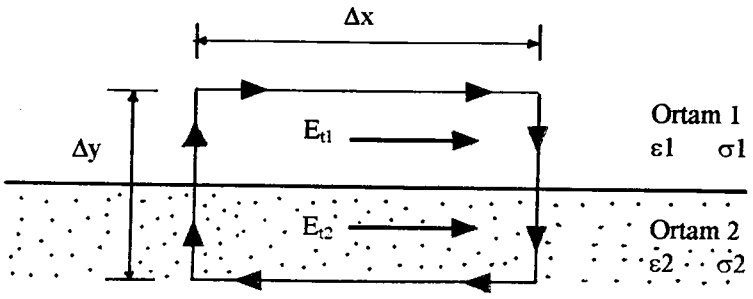
Karesel koordinatlarda;

$$\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} \qquad (1.9)$$

şeklindedir. Laplace denklemi, potansiyeli bulmakta ve alan hesabında sıkça kullanılır.

1.2.4. Sınır şartları

Belirli bir ortamda elektrik alan süreklidir. İki farklı ortam arasındaki sınırda elektrik alan hem büyüklük hem de yön olarak değişebilir. İlki sınıra teğet, ikincisi sınıra normal (dik) alanlar arasındaki bağıntıları gözönüne alarak sınır problemlerini iki kısımda analiz etmek uygundur.



Şekil 1.1. Teğetsel elektrik alanın sınır boyunca sürekliliği

Şekil 1'de sınıra teğet alan bağıntılarının oluşturulmasında permitiviteleri ε_1 ve ε_2 iki dielektrik ortam gözönüne alınmıştır. İki ortam da mükemmel dielektrik olup, iletkenlikleri σ_1 ve σ_2 iki ortam için de sıfırdır. Herbir yarısı bir ortamda bulunan dikdörtgen yolun, sınıra paralel uzunluğu Δx ve sınıra dik uzunluğu Δy'dir.

Ortam 1'de sınıra teğet ortalama alan şiddeti E_{t1} ve ortam 2'de E_{t2}'dir. Bu kapalı yol boyunca iş, yol boyunca **E**'nin yüzeysel integraline eşittir. Δy, yaklaşık olarak sıfır yapılarak sınıra dik yol parçacıkları boyunca iş sıfır olsa da sınıra dik sonlu bir elektrik alan varolabilecektir. Oklar doğrultusunda dikdörtgen etrafında **E**'nin yüzeysel integrali;

$$E_{t1}\,\Delta_x - E_{t2}\,\Delta_x = 0$$

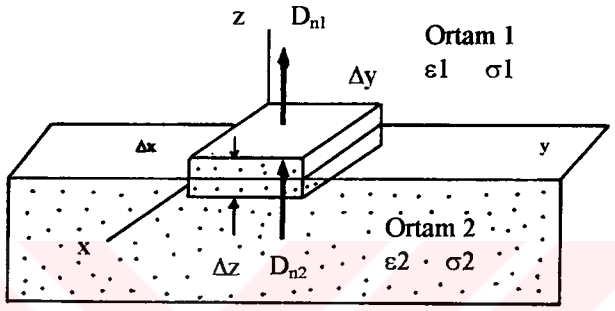
veya

$$E_{t1} = E_{t2} \qquad (1.10)$$

olacaktır. Son eşitliğe göre; iki dielektrik arasındaki sınırın her iki tarafında elektrik alanın teğetsel bileşenleri aynıdır. Diğer bir deyişle, teğetsel elektrik alan bu tip bir sınır boyunca süreklidir. Eğer ortam 2, bir iletken ise ($\sigma_2 \neq 0$), statik koşullar altında E_{t2} alanı ortam 2'de sıfır olmalıdır. Bu durumda (1.10) eşitliği

$$E_{t1} = 0 \qquad (1.11)$$

olacaktır. Bir dielektrik - iletken sınırda teğetsel elektrik alan sıfırdır.



Şekil 1.2. Yüksüz bir sınır boyunca akı yoğunluğunun dikey bileşeninin sürekliliği

Şekil 2'de permitiviteleri ε_1 ve ε_2 olan xy düzlemiyle ayrılmış iki dielektrik ortam gözönüne alınmıştır. Her iki ortam da mükemmel dielektrik olup, $\sigma_1 = \sigma_2 = 0$'dır. Her bir yarısı bir ortamda bulunan kutuların boyutları Δx Δy ve yükseklikleri Δz kabul edilir. D_{n1}, ortam 1'de kutunun üst yüzeyine dik ortalama akı yoğunluğu ve D_{n2} ortam 2'de kutunun alt yüzeyine dik ortalama akı yoğunluğudur. D_{n2} negatifken, D_{n1} pozitiftir. Gauss yasasına göre **D**'nin dik bileşeninin kapalı yüzey boyunca yüzeysel integrali kapalı yüzeydeki yüklerin toplamına eşittir. Δz yaklaşık olarak sıfır yapılarak, kenarlar boyunca yüzeysel integral sıfıra eşitlenir. Sınırda ortalama yüzey yük yoğunluğu, ρ_s olmak üzere, Gauss yasası uygulanarak;

$$D_{n1}\Delta x\, \Delta y - D_{n2}\, \Delta x\, \Delta y = \rho_s\, \Delta x\, \Delta y$$

veya

$$D_{n1} - D_{n2} = \rho_s \qquad (1.12)$$

yazılabilir.

Yukarıdaki eşitliğe göre; akı yoğunluğunun dikey bileşeni, iki dielektrik arasındaki yüklü sınırda yüzeysel yük yoğunluğuna eşit miktar ile değişir.

Eğer sınır yüksüz ise;

$$\rho_s = 0$$
$$D_{n1} = D_{n2} \qquad (1.13)$$

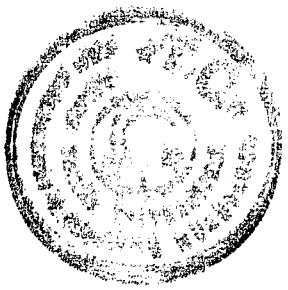
olur. Akı yoğunluğunun dikey bileşeni, iki dielektrik arasındaki yüksüz sınır boyunca süreklidir. Eğer ortam 2 iletken ise;

$$D_{n2} = 0$$
$$D_{n1} = \rho_s \qquad 1.14)$$

olacaktır. Dielektrik - iletken sınırda akı yoğunluğunun dikey bileşeni iletkendeki yüzeysel yük yoğunluğuna eşittir.

1.3. Statik Manyetik Alanlar

Bu bölümde, sabit elektrik akımlarının manyetik alanlarından bahsedilecektir. Üzerinden akım geçen bir iletken, gücü bir mıknatıs veya pusula iğnesine etki eden bir bölge ile çevrelenir. Oersted'in 1819'da keşfettiği bu etki ile, bir iletken etrafında manyetik alanın kapalı dairesel halkalar şeklinde olduğu bulunmuştur. Alanın yönü, pusula iğnesinin kuzey kutbunun yönünde olacak şekilde alınır.



Manyetik alan yönü ile akım yönü arasındaki ilişki sağ el kuralı ile tespit edilir. Baş parmağın akım yönünü göstermesiyle manyetik alan doğrultusunda sağ elin parmakları iletkeni çevreler.

Bu bölümde elektrik alanlardan başlıca bir takım farklılıkları olan ve elektrik akımları tarafından oluşturulan manyetik alanlardan bahsedilecektir. Manyetik alanın yönü ve büyüklüğü Bio-Savart Yasası ile verilecektir. Kapalı yüzeyler için Ampére Yasası kullanılarak, belli bir nokta için yasa geliştirilecektir.

1.3.1. Ampére Yasası

Ampére yasasına göre, kapalı bir eğri boyunca **H** vektörünün eğrisel integrali, integrasyon yolunun çevrelediği toplam akıma eşittir. (Fink vd., 1993)

$$\oint \mathbf{H}.\, d\mathbf{L} = I \qquad (1.15)$$

H : Manyetik alan, A/ m
dL: Eğrinin seçilmiş çok küçük bir parçası, m
I : Akım şiddeti, A

Stokes teoremine göre, bir vektörün sirkülasyonu, o vektörün rotasyonelinin sirkülasyon eğrisinin çevrelediği yüzeyden geçirdiği akıya eşit olduğundan; bu teorem **H** vektörüne uygulandığında;

$$\oint \mathbf{H}.\, d\mathbf{L} = \int_{(s)} (\nabla x \mathbf{H}).d\mathbf{S} = I \qquad (1.16)$$

$$I = \int_{(s)} \mathbf{J}.\, d\mathbf{S} \qquad (1.17)$$

$$\nabla \times \mathbf{H} = \mathbf{J} \qquad (1.18)$$

elde edilir.

J : Akım yoğunluğu, A / m^2

1.3.2. Bio - Savart yasası

k, ortama bağlı bir orantı sabiti olmak üzere, **dL** boyundaki iletkenden geçen I akımının kendisinden R uzaklıkta bir noktada meydana getirdiği **dB** diferansiyel alanı;

$$\mathbf{dB} = k.\frac{I\,\mathbf{dL} \times \mathbf{r}}{R^2} \qquad (1.19)$$

ifadesi ile belirlenir. Burada;

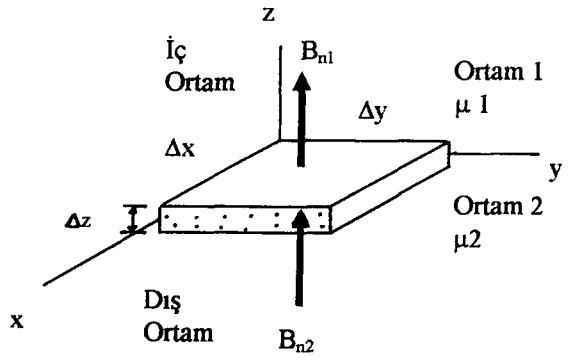
$$k = \frac{\mu}{4\pi} \qquad (1.20)$$

μ : Ortamın geçirgenliği (permeabilitesi), H / m

1.3.3. Sınır şartları

Belirli tek bir ortamda manyetik alan süreklidir. Fakat, iki farklı ortam arasındaki sınırda, manyetik alan hem büyüklük hem de yön açısından değişebilir.

Bu sınır problemlerini biri sınıra teğet diğeri ise sınıra dik alanlar arasındaki ilişkileri ayrı ayrı inceleyerek analiz etmek uygundur. İlk olarak sınıra dik alanlar arasındaki ilişkileri incelemek için, Şekil 3'te gösterildiği gibi xy düzlemi ile birbirinden ayrılmış permeabiliteleri μ_1 ve μ_2 olan iki ortam gözönüne alınmıştır.



Şekil 1.3. **B**'nin normal (dikey) bileşeninin sürekli olduğunu gösteren şekil

Boyutları Δx Δy ve yüksekliği Δz olan ve herbir yarısı bir ortamda bulunan bir kutu düşünülür. Bn_1 ortam 1'de kutunun üst yüzeyine dik **B**'nin ortalama bileşeni ve Bn_2 ortam 2'de kutunun alt yüzeyine dik **B**'nin ortalama bileşenidir. Bn_2 negatifken, Bn_1 pozitiftir. Manyetik alanlar için Gauss yasasına göre kapalı bir yüzey boyunca toplam manyetik akı sıfıra eşittir. Kutunun Δz yüksekliği sıfıra yaklaştırılarak kutunun kenarları boyunca yüzeysel integral sıfır yapılır. Dolayısıyla yüzeysel integral;

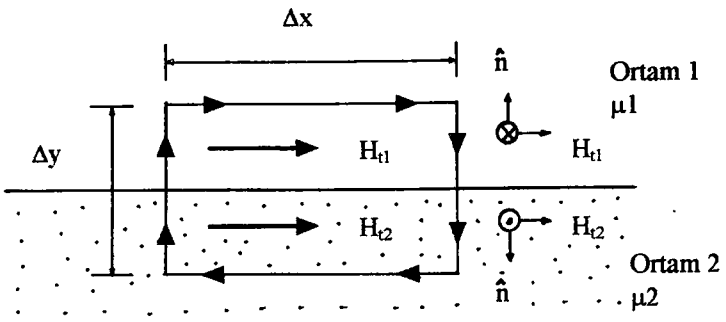
$$B_{n1}\,\Delta x\,\Delta y - B_{n2}\,\Delta x\,\Delta y = 0$$

veya

$$B_{n1} = B_{n2} \qquad (1.21)$$

haline gelir.

İki ortam arasındaki sınır boyunca **B** akı yoğunluğunun (dik) normal bileşeni süreklidir. Sınıra teğet manyetik alanlar için bağıntıları incelemede Şekil 4'te gösterildiği gibi birbirinden düz bir sınır ile ayrılan ve permeabiliteleri μ_1 ve μ_2 olan iki ortam gözönüne alınır.



Şekil 1.4. **H**'ın teğetsel bileşeninin yüzeyde akım yoğunluğu ile değişimi

Sınıra paralel Δx ve sınıra dik Δy uzunluğunda ve herbir yarısı bir ortamda bulunan bir dikdörtgen yol seçilir. Ortam 1'de **H**'nin sınıra teğet ortalama değeri H_{t1} ve ortam 2'de **H**'nin sınıra teğet ortalama değeri H_{t2}'dir. Yol uzunluğu Δy sıfıra yaklaştırılarak ve Ampére yasası kullanılarak eğrisel integral

$$H_{t1}\,\Delta x - H_{t2}\,\Delta x = I \qquad (1.22)$$

veya

$$H_{t1} - H_{t2} = \frac{I}{\Delta x} = K \qquad (Am^{-1}) \qquad (1.23)$$

haline getirilir.

Sınır boyunca **H**'nin teğetsel bileşenindeki değişim, sınırda yüzey akım yoğunluğu K'nın büyüklüğüne eşittir.

$\hat{\mathbf{n}}$ sınıra dik birim vektör olmak üzere,

$$\mathbf{K} = \hat{\mathbf{n}} \times (\mathbf{H}_{t1} - \mathbf{H}_{t2}) \qquad (1.24)$$

olur. Eğer K = 0 ise;

$$H_{t1} = H_{t2} \tag{1.25}$$

yazılabilir.

1.4. Maxwell Denklemleri

1. Maxwell Denklemi:

Faraday'ın indüksiyon elektromotor kuvvetini veren iki formülün vektörel ifadeleri 1. Maxwell denklemini verir.

$$\oint \mathbf{E}.d\mathbf{L} = -\int_{(s)} \mathbf{B}.d\mathbf{S} + \oint (\mathbf{V}x\mathbf{B}).d\mathbf{L} \tag{1.26}$$

Burada **V**, sabit alan içindeki hızdır. Bu denklemin diferansiyel şekli, devrenin sabit **B** alanı içinde hareketinden doğan $\oint$ **(VxB)**.**dL** hesaba katılmadan;

$$\nabla x\mathbf{E} = -\dot{\mathbf{B}} = -\frac{\partial \mathbf{B}}{\partial t} \tag{1.27}$$

şeklinde yazılır.

Faraday kanunundan yola çıkılarak 1. Maxwell denklemi:

$$\iint (\nabla x\mathbf{E}).d\mathbf{S} = -\iint \frac{\partial \mathbf{B}}{\partial t}.d\mathbf{S} \tag{1.28}$$

şeklindedir.

2. Maxwell Denklemi

Ampére teoremine dayanır. $\dot{\mathbf{D}} = \frac{\partial \mathbf{D}}{\partial t}$ deplasman akım yoğunluğu da hesaba katılarak integral şekli;

$$\oint \mathbf{H}.d\mathbf{L} = \int_{(s)} (\mathbf{J} + \dot{\mathbf{D}}).d\mathbf{S} \qquad (1.29)$$

şeklindedir. Diferansiyel formda ise,

$$\nabla \mathrm{x}\, \mathbf{H} = \mathbf{J} + \dot{\mathbf{D}} \qquad (1.30)$$

şeklinde yazılabilir.

3. Maxwell Denklemi

Gauss yasasından türetilen elektrik alan eşitliği 3. Maxwell denklemini oluşturur.

$$\oint \mathbf{D}.d\mathbf{S} = \int_{(v)} \rho \, d\, v \qquad (1.31)$$

D deplasman vektörü ile ρ hacimsel yük yoğunluğu arasındaki bu bağıntının diferansiyel hali;

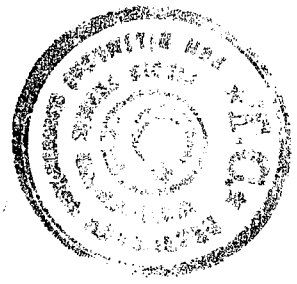
$$\nabla . \mathbf{D} = \rho \qquad (1.32)$$

şeklindedir.

4. Maxwell Denklemi

Manyetik alanlar için kapalı bir S yüzeyi boyunca **B**'nin yüzeysel integrali sıfıra eşittir.

3. Maxwell denkleminde **D** yerine **B** konulduğunda 4. Maxwell denklemi elde edilir.



$$\oint \mathbf{B}.d\mathbf{S} = 0 \qquad (1.33)$$

Diferansiyel şekli;

$$\nabla.\mathbf{B} = O \qquad (1.34)$$

olur.

1.4.1. Faraday yasası

Faraday, zamanla değişen bir manyetik alanın ölçülebilen bir gerilim (e.m.k.) oluşturduğunu belirtmiştir. Bu kural indüktörler için temeldir. Bu kuralı açıklayan temel eşitlik:

$$\text{e.m.k.} = - \frac{\partial \phi}{\partial t} \qquad (1.35)$$

şeklindedir. (Ergeneli, 1988)

Formüldeki - işareti, Lenz kanununa göre, enerji korunumu ilkesini göstermekte, meydana gelen e.m.k.'nın akı değişimine zıt yönde bir akı doğuracağını anlatmaktadır. Vektörel ve diferansiyel gösterimlerde Faraday yasası aşağıdaki gibi yazılır.

$$\oint \mathbf{E}d\mathbf{L} = -\int \frac{\partial \mathbf{B}}{\partial t} d\mathbf{S} \qquad (1.36)$$

$$\nabla \times \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t} \qquad (1.27)$$

1.4.2. Elektrik ve manyetik alanlar için Gauss yasası

Elektrik alanlar için Gauss yasasına göre, kapalı bir yüzey boyunca akı, bu yüzeyin kapladığı bölge içinde bulunan yüklerin cebirsel toplamına eşittir. **D**; yüzey boyunca akı yoğunluğunu vermek üzere, elektrik alanlar için Gauss yasası;

$$\oint_s \mathbf{D}.d\mathbf{S} = \int_{(v)} \rho dv \qquad (1.31)$$

$$\nabla.\mathbf{D} = \rho \qquad (1.32)$$

şeklinde yazılabilir.

Elektromanyetik alanlar için Gauss yasasına göre, manyetik akı yoğunluğu vektörünün kapalı bir yüzeyden geçirdiği akı sıfıra eşittir. **B** manyetik akı yoğunluğu olmak üzere yasanın integral ve diferansiyel formu aşağıdaki gibidir.

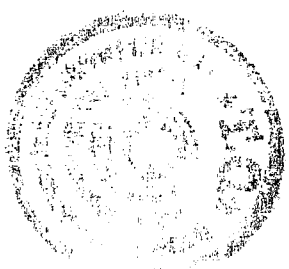
$$\oint_s \mathbf{B}.d\mathbf{S} = 0 \qquad (1.33)$$

$$\nabla.\mathbf{B} = 0 \qquad (1.34)$$

1.4.3. Ampére yasası ve deplasman akımı

Ampére yasasının integral formu bölüm 1.3.1'de;

$\oint \mathbf{H}.\, d\mathbf{L} = I$ şeklinde verilmişti.



Eşitliğin sağ tarafındaki akım, iki bileşenden ibarettir. Bunlardan biri sabit bileşen diğeri ise zamana bağlı olarak değişen ve deplasman akımı olarak adlandırılan bileşendir. Buna göre; Ampére teoreminin integral ve diferansiyel formda yazılışı:

$$\oint \mathbf{H}.d\mathbf{L} = I + \int_s \frac{\partial \mathbf{D}}{\partial t} d\mathbf{S} \qquad (1.29)$$

$$\nabla x\mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \qquad (1.30)$$

şeklinde olacaktır. (Fink ve Beaty, 1993)

1.5. Kısmı Türevli Diferansiyel Denklemlerin Çözümünde Kullanılan Sayısal Yöntemler

1.5.1. Sonlu farklar yöntemi

Sonlu farklar yöntemi, kısmi türevli denklemlerin çözümü ve elektrik ve manyetik alan problemlerinin incelenmesinde kullanılan bir sayısal yöntemdir. Basit aritmetik ve trigonometrik bilgi ile uygulanabilmesi, dirençle benzetim yöntemi ile benzerliği ve basit bilgisayar programları ile çalışabilmesinden ötürü kolay anlaşılabilmesi, yöntemin yaygınlaşmasını sağlamıştır.

Sonlu farklar yöntemi ile alan problemi, sınır koşulları verilen ve potansiyel dağılımının sürekli olduğu kabul edilen, sınırlı bir bölge içinde incelenir. Elektrik alan dağılımının bulunacağı bölgede belirli noktalar için yazılan sonlu fark eşitliklerinin oluşturduğu doğrusal denklem sisteminin çözülmesiyle bu noktalardaki elde edilen potansiyel değerlerinden yararlanarak eşpotansiyel çizgiler çizilebilir ve alan şiddetinin maksimum olduğu nokta bulunabilir.

Sonlu farklar yönteminde, tüm sınır koşullarının verilmesi ve incelenecek bölgenin kapalı bir bölge olması gerekmektedir. İncelenecek bölge, dikdörtgen, kare, üçgen gibi şekillere sahip

gözlerden oluşan bir ağ ile bölünür. Düğüm noktaları belirlenir ve ağ üzerindeki herbir düğümün potansiyeli, komşu düğüm potansiyellerine bağlı olarak yazılarak, düğüm potansiyelleri için çözülmesi gereken bir lineer denklem takımı elde edilir.

Sonlu farklar yöntemi ile çözümde, ileri geri ve merkezi sonlu farklar gibi türleri kullanılan sonlu fark işlemlerinden yararlanılır. İleri fark operatörü Δ, geri fark operatörü ∇ ve merkezi fark operatörü δ ile gösterildiğinde; f(x) fonksiyonunun birinci mertebeden ileri farkı;

$\Delta f_i = f_{i+1} - f_i$

n. mertebeden ileri farkı;

$\Delta^n f_i = \Delta (\Delta^{n-1} f_i) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i$

birinci mertebeden geri farkı;

$\nabla f_i = f_i - f_{i-1}$

n. mertebeden geri farkı;

$\nabla^n f_i = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1}$

birinci mertebeden merkezi farkı;

$\delta f_i = \delta f_{i+1/2} - \delta f_{i-1/2}$

n. mertebeden merkezi farkı;

$\delta^n f_i = \delta^{n-1} f_{i+1/2} - \delta^{n-1} f_{i-1/2}$

olur. Bu gösterimlerde, h adım büyüklüğünü göstermek üzere;

$f_i = f(x)$

$f_{i+1} = f(x+h)$

$f_{i-1} = f(x-h)$

$f_{i+1/2} = f(x+h/2)$

$f_{i-1/2} = f(x-h/2)$ olacaktır.

Fark denklemlerinin sayısal çözümü zaman alıcıdır. Sınırlarında, düzgün olmayan elemanlarla tanımlanmış herhangi bir alan problemi çözülürken, ağ, düzlemin sınırlı küçük

bir bölgesinde daha düzgün bir ağ şekline getirilir. Ağın tamamı V potansiyelleri bilinmeyen n düğümden oluşur.

Denklem sistemi matris şeklinde yazıldığında;

$$[C]\,[V] = [Q] \tag{1.37}$$

halini alır. Burada [C], nxn boyutlu katsayılar matrisi, [V] bilinmeyenlerden oluşan sütun matris ve [Q], bilinen potansiyellerin toplamından oluşan sütun matrisidir [C] matrisi, köşegene göre simetrik bir matristir. Katsayılar matrisi genelde seyrek bir matris olduğundan ve bilinmeyen sayısının çok büyük olmasından ötürü oluşan çok büyük boyutlu doğrusal denklem sistemlerinin doğrudan çözüm yöntemlerinin kullanılması bilgisayarda çözüm süresinin uzamasına ve büyük bellek gereksinimine neden olur. Bundan ötürü genellikle katsayılar matrisinin tamamını bellekte saklamayı gerektirmeyen iteratif yöntemler kullanılır. Sonlu farklar yöntemi iteratif ve ardışık yakınsama yöntemleri ile birlikte kullanıma çok uygundur. (Kardeştuncer, 1988)

1.5.2. Monte Carlo Yöntemi

Monte Carlo Yöntemi, bir noktanın potansiyelini, büyük yaklaşım fonksiyonları kullanılmaksızın, az işlemle ve az bilgisayar belleği kullanarak kısa sürede hesaplama olanağı veren bir yöntemdir. Bu yöntem tüm problemi çözmeksizin alt bölgenin incelenmesine olanak verir.

Monte Carlo yöntemi, rasgele yürüyüş ilkesine dayanır. Bu yöntemle bir noktanın potansiyeli, herbiri bu noktadan başlayan ve sınırlarda son bulan belli sayıda rasgele yürüyüş ile kestirilir. Bu rasgele yürüyüşleri gerçekleştirebilmek için gereken rasgele sayılar aritmetik yöntemle elde edilebildiği gibi, bir bilgisayarın rasgele sayı üretiminden faydalanarak da elde edilebilir. (Beasley, 1979)

En tanınmış Monte Carlo yöntemleri sabit ve serbest yürüyüşlü Monte Carlo yöntemleridir.

a) Sabit rasgele yürüyüşlü Monte Carlo yöntemi

Bu yöntemde rasgele yürüyüşler sabit uzunluktaki adımlarla ve koordinat eksenlerine paralel olarak yapılır. Yöntem, Laplace denklemini çözmek için kullanılabilir. Dirichlet tipi sınır koşullarına sahip bir bölge için Laplace denklemi;

$$\nabla^2 V = 0 \qquad (1.38)$$

olur. Bölgenin sınırında V = Vp yazılabilir.

Sabit rasgele yürüyüşlü Monte Carlo Yöntemi'nde işleme, bölgeyi düzgün bir ağa bölerek başlanır ve ∇^2 yerine sonlu fark eşdeğeri alınır.

Bir (x, y) noktasındaki potansiyeli hesaplamak için bu noktadan başlamak üzere rasgele yürüyen bir parçacık gözönüne alınır. Parçacık ağdaki düğümden düğüme, sınıra erişene kadar ilerletilir. Parçacık sınıra ulaştığında yürüme sona erer ve parçacığın ulaştığı noktadaki Vp sınır potansiyeli kaydedilir. Bu işlem (x, y) noktasından çıkan 1,2,3,...., N'inci parçacıklar için yapılır. Bilinmeyen potansiyel değeri, bilinen potansiyel değerlerinden istatistiksel olarak belirlenir. Buna göre Dirichlet probleminin çözümü olan (x, y) noktasındaki potansiyelin beklenen değeri,

$$V(x, y) = \frac{1}{N}\sum_{i=1}^{N} Vp(i) \qquad (1.38)$$

bağıntısıyla hesaplanır. Burada N toplam yürüyüş sayısıdır. Monte Carlo Yöntemi ile çözüm bölgesi içindeki herhangi bir noktadaki potansiyel, ayrı olarak hesaplanabilir. Yöntem basit ve güvenlidir, yanlızca sınır yüzeylerini gerektirir. Her yürüyüş dizisi yanlızca bir noktanın potansiyelini verir. Bu tür Monte Carlo Yöntemi'nin bir olumsuzluğu yavaş olmasıdır. Bu nedenle, bu yöntemin az sayıda potansiyel değerinin bulunması gerektiği durumlarda kullanılması uygundur.

b) Serbest rasgele yürüyüşlü Monte Carlo yöntemi

Serbest rasgele yürüyüşlü Monte Carlo yönteminin ilkesi, potansiyel teorisinin "ortalama değer teoremine" dayanır. Tamamı bölge içinde kalan, (x,y,z) merkezli, R yarıçaplı bir K küresi için potansiyel,

$$V(x,y,z) = \frac{1}{4\pi R^2} \int_{(s)} V(r)ds \qquad (1.39)$$

dir. Bölge içinde herhangi bir kürenin merkezindeki potansiyel, küre yüzeyinde alınan potansiyelin ortalama değerine eşittir. İki boyutlu potansiyel değişimi için,

$$V(x,y,z) = \frac{1}{4\pi R^2} \oint_{(l)} V(r)d\,l \qquad (1.40)$$

olur. Burada integral (x,y) merkezli, R yarıçaplı bir çember boyunca alınmıştır.

Serbest rasgele yürüyüşlü Monte Carlo yöntemi, son iki denklemin uygulanmasına dayanır. İki boyutlu bir problemde, örneğin rasgele hareket eden bir parçacığın i. yürüyüşünde j. adımdan sonra (x_j, y_j) noktasında bulunduğu varsayılırsa, j+l'inci adım için önce merkezi (x_j, y_j) noktasında olan ve r_j yarıçapı (x_j, y_j) noktası ile sınır arasındaki en kısa uzaklığa eşit olan bir çember çizilir. ϕ koordinatı, $0 < n < 1$ aralığında bir n rasgele sayısı ile $(0, 2\pi)$ arasında düzgün olarak dağılmış bir rastlantı değişkeni olarak $\phi=2\pi n$ bağıntısından hesaplanır. (j+1). adımdan sonra parçacık,

$$x_{j+1} = x_j + r_j \cos\phi_j$$

$$y_{j+1} = y_j + r_j \sin\phi_j$$

noktasında bulunur. Bir sonraki adım, merkezi (x_{j-1}, y_{j-1}) noktasında bulunan ve yarıçapı (r_{j+1}), (x_{j-1}, y_{j-1}) noktası ile sınır arasındaki en kısa uzaklığa eşit olan çemberin çizilmesi ile atılır. Bu işlem birçok kere yenilenir. Yürüyüş, sınıra önceden öngörülen küçük bir d uzaklığına yaklaşana kadar sürer ve bu durumda sınıra ulaşıldığı varsayılır. i yolun sonunda

V_p (i) potansiyeli kaydedilir ve sabit yürüyüşte olduğu gibi herhangi bir (x, y) noktasındaki potansiyel N yürüyüşten sonra (1.38) denklemi kullanılarak bulunur. Ortalama değer teoreminin tekrarlanarak uygulanması, bilinmeyen potansiyel ile bilinen potansiyel arasındaki ilişkiyi verir. Serbest rasgele yürüyüşlü Monte Carlo yöntemi ile yukarıda anlatılanlara benzer işlemler uygulanarak üç boyutlu problemler de çözülebilir.

Serbest rasgele yürüyüşlü Monte Carlo yönteminde hem adım uzunlukları hem de hareket yönleri serbest değerler olarak seçildiğinden bu hareket serbest rasgele hareket olarak gözönüne alınır. Serbest yürüyüşle uzun bir atlayışta sınıra ulaşmak için genelde birkaç adım yeterli olduğundan, hesaplama sabit adımlı yürüyüşten daha hızlıdır.

1.5.3. Yük benzetim yöntemi

Yük benzetim yöntemi elektrik alan hesaplarında en çok tercih edilen yöntemdir. Çünkü çözüm iteratif değil doğrudandır ve bu da uygulamada oldukça kolaylık sağlar. Ayrıca Yük Benzetim Yöntemi elektrik mühendisliğinde üç boyutlu ve çok yalıtkanlı düzenlerin, eksenel simetrisi olmayan problemlerin ve sınır optimizasyonu problemlerinin çözümünde de kullanılmaktadır. (Malik, 1989)

Yük benzetim yönteminin temel ilkesi, elektroteknikteki toplama ilkesine dayanır. Elektrotlar üzerinde fiziksel olarak dağılmış elektrik yükleri yerine konulan ayrık yükler, alan çözümü istenen yalıtkan bölgenin dışına, elektrotların içine veya elektrot gibi bir eş potansiyel yüzey arkasına yerleştirilir ve "benzetim yükü" adını alır. Yüklerin değerleri sınır koşullarından belirlenir. Sınır koşulu olarak elektrot sınırı üzerinde alınan m adet sınır noktasının potansiyeli kullanılır. Eğer bir iletken bölge içerisinde herhangi bir tipten birçok ayrık yük varsa, alan çözümü istenen yalıtkan bölge içerisindeki veya elektrot sınırındaki herhangi bir A noktasındaki elektrostatik potansiyel, A noktasının herhangi bir yükün bulunduğu nokta olmaması koşuluyla, herbir yükün ayrı ayrı A noktasında oluşturduğu potansiyellerin toplamına eşittir. q_j, n adet ayrık yük ve V_i de bu bölge içerisinde herhangi bir A noktasındaki potansiyel olursa toplama ilkesine göre,

$$V_i = \sum_{j=1}^{n} P_{ij}\, q_j \qquad (1.41)$$

olmaktadır. P_j, potansiyel katsayısıdır ve değeri her yük tipi için Laplace veya Poisson denkleminin çözümünden belirlenir. Yük benzetim yönteminde hata, benzetim yüklerinin ve sınır noktalarının yerleriyle elektrotların ve yalıtkanların şekline ve benzetim yüklerinin tip ve sayısına bağlıdır. Son yıllarda noktasal yükler, sonsuz uzunlukta ve yarı sonsuz uzunlukta çizgisel yüklere ilaveten eliptik silindirsel yükler, küresel yükler, düzlemsel yükler, disk tipi yükler, sabit yük yoğunluklu halkasal yükler, hacimsel yükler, tabaka ve halka şeklinde düzlemsel yükler ve değişken yoğunluklu çizgisel yükler kullanılmaktadır. Tek yalıtkanlı bir sistemde, benzetim yükü sayısı arttırılarak potansiyel hatası azaltılabilir. Çok yalıtkanlı sistemlerde eğer yalıtkan sınır karmaşık bir yapıya sahipse potansiyel ayrılığı genellikle büyüktür.

2. SONLU ELEMANLAR YÖNTEMİ

2.1. Sınır Değer Problemleri

Bu bölümde önce sınır değer problemlerini tanımladıktan sonra bunların çözümleriyle ilgili iki klasik metot incelenecektir. Bunlardan bir tanesi Ritz varyasyon metodu, diğeri ise Galerkin metodu olup, bu iki metot, modern sonlu elemanlar metodunun temelini oluştururlar.

Bir sınır değer problemi bölgeyi çerçeveleyen Γ sınırı üzerindeki sınır şartlarıyla birlikte Ω bölgesinde geçerli bir diferansiyel denklemle tanımlanabilir.

$$L.\,\phi = f \tag{2.1}$$

(2.1) ifadesinde L bir diferansiyel işlemci, f kaynak veya uyarma fonksiyonu ve ϕ de bilinmeyen büyüklüktür.

2.1.1. Sınır değer probleminin çözümünde Ritz yöntemi

Ritz yöntemi, sınır değer probleminin fonksiyonel olarak isimlendirilen ve minimumu verilen sınır şartları altında geçerli diferansiyel denkleme karşılık gelen varyasyon ifadeleri cinsinden formüle edildiği bir varyasyon yöntemidir.

$$(\phi, \psi) = \int_{\Omega} \phi.\psi^{*} d\Omega \tag{2.2}$$

Burada (*) kompleks eşleniği göstermektedir. Eğer (2.1) ifadesindeki L özeşlenik ise;

$$(L\ \phi, \psi) = (\phi\ , L\ \psi) \tag{2.3}$$

ve pozitif tanımlı yanı aşağıdaki bağıntıların varolması halinde,

$$(L\,\phi, \phi)\begin{cases} > 0 & \phi \neq 0 \\ = 0 & \phi \neq 0 \end{cases} \tag{2.4}$$

(2.1)'in çözümü fonksiyonel, $\bar{\phi}$' ye göre minimize edilerek elde edilebilir.

$$F(\bar{\phi}) = \frac{1}{2}(L\ \bar{\phi}, \bar{\phi}) - \frac{1}{2}(\bar{\phi}, f) - \frac{1}{2}(f, \bar{\phi}) \tag{2.5}$$

Burada $\bar{\phi}$, deneme fonksiyonunu göstermektedir. ϕ yaklaşık olarak aşağıdaki gibi ifade edilebilir.

$$\bar{\phi} = \sum_{j=1}^{N} c_j v_j = \{c\}^T \{v\} = \{v\}^T \{c\} \tag{2.6}$$

Burada v_j bütün bölge üzerinde tanımlı seçilmiş açılım fonksiyonları, c_j de hesaplanacak olan sabit katsayılardır. {.}, bir sütun vektörünü ve T üst simgesi de vektörün transpozesini belirtmektedir. (2.6)'yı (2.5)'de yerine koyarak aşağıdaki bağıntı elde edilir.

$$F = \frac{1}{2}\{c\}^T \int_{\Omega} \{v\}\, L\ \{v\}^T d\Omega \{c\} - \{c\}^T \int_{\Omega} \{v\} f d\Omega \tag{2.7}$$

$F(\bar{\phi})$'yi minimize etmek için, c_i'ye göre türevlerinin kaybolduğu kabul edilerek, elde edilen lineer cebirsel denklem takımı;

$$\frac{\partial F}{\partial c_i} = \frac{1}{2}\int_{\Omega} v_i\, L\ \{v\}^T d\Omega \{c\} + \frac{1}{2}\{c\}^T \int_{\Omega} \{v\}\ L\ v_i d\Omega\ - \int_{\Omega} v_i f d\Omega$$

$$= \frac{1}{2}\sum_{j=1}^{N} c_j \int_{\Omega} (v_i\, L\ v_i + v_j\ L\ v_i) d\Omega - \int_{\Omega} v_i f d\Omega$$

$$= 0 \qquad i = 1, 2, 3, \ldots\ldots\ldots\ldots, N \tag{2.8}$$

Matris formunda yazılırsa;

$$[S]\ \{c\} = \{b\} \tag{2.9}$$

[S] deki elemanlar;

$$S_{ij} = \frac{1}{2}\int_{\Omega}(v_i L\ v_j + v_j L\ v_i)d\Omega \tag{2.10}$$

ve {b} deki elemanlar;

$$b_i = \int_{\Omega} v_i f d\Omega \tag{2.11}$$

[S] simetrik bir matristir. L operatörünün özeşlenik özelliği kullanılarak S_{ij} aşağıdaki gibi yazılabilir.

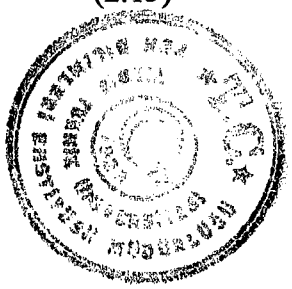
$$S_{ij} = \int_{\Omega} v_i L\, v_j\, d\,\Omega \tag{2.12}$$

Bundan sonra matris eşitliği (2.9) çözülerek (2.1) için yaklaşık bir çözüm elde edilir.

2.1.2. Sınır değer probleminin çözümünde Galerkin yöntemi

ϕ, (2.1) eşitliğinin yaklaşık bir çözümü olarak kabul edilerek (2.1)'de, ϕ yerine $\overline{\phi}$'nin konulması sonucu aşağıdaki gibi sıfırdan farklı bir rezidü elde edilir.

$$r = L\ .\ \overline{\phi} - f \neq 0 \tag{2.13}$$



$\bar{\phi}$ için en iyi yaklaşım rezidü r'yi Ω'nın her noktasında en küçük değerine indireni olacaktır. Weighted rezidü metotları aşağıdaki şartı koşmaktadır.

$$R_i = \int_\Omega w_i . r . d\Omega = 0 \qquad (2.14)$$

Burada R_i ; Weighted rezidü integralini ve W_i seçilen bir Weighting fonksiyonunu göstermektedir.

Galerkin metodunda, Weighting fonksiyonu yaklaşık çözümün açılımı için kullanılanla aynı seçilir. Bu, genelde en doğru çözümü verdiğinden sonlu eleman denklemlerinin çıkarılmasında sıkça tercih edilir. Yöntemi daha açık şekilde göstermek için, (2.6) daki bağıntının çözümü temsil ettiği varsayılırsa, Weighting fonksiyonları öyle seçilir ki,

$$W_i = v_i \qquad i = 1, 2, 3, ..., N \qquad (2.15)$$

Böylece (2.14) denklemi

$$R_i = \int_\Omega (v,\ L\ \{v\}^T\{c\} - v_i f) d\Omega = 0 \qquad i = 1, 2, 3, ..., N \qquad (2.16)$$

şeklini alır.

Bu da bizi, L operatörü özeşlenik olmadıkça [S] matrisinin simetrik olması gerekmemekle beraber yine (2.9)' da verilen matris sistemine götürür. O halde Galerkin metodu, Ritz metodu ile aynı denklem sistemi sonucunu verir.

2.2. Sonlu Elemanlar Yöntemi

Yaklaşık olarak problemin gerçek çözümünün bulunabildiği bütün bir çözüm bölgesi üzerinde tanımlı bir deneme fonksiyonu bulmak Ritz ve Galerkin metotlarında önemli bir adımdır. Fakat özellikle iki ve üç boyutlu problemlerde oldukça zordur. Bu zorluğu

hafifletmek için, bölgenin tamamını küçük alt bölgelere ayırıp her bölgenin üzerinde tanımlı deneme fonksiyonları kullanılabilir. Bu deneme fonksiyonları genellikle çok daha basit bir şekle sahiptir, çünkü alt bölgeler küçüktür ve $\phi(x)$ fonksiyonunun varyasyonu her alt bölge üzerinde daha az etkilidir.

Klasik Ritz ve Galerkin metotlarında deneme fonksiyonu bütün bölge üzerinde tanımlı bir takım temel fonksiyonların kombinasyonu olarak formüle edilir. Bu kombinasyon en azından yaklaşık olarak gerçek çözümü temsil edebilmeli ve uygun sınır şartlarını da sağlamalıdır. Sonlu eleman metodunda deneme fonksiyonu bütün bölgeyi meydana getiren alt bölgeler üzerinde tanımlı bir takım temel fonksiyonların bir kombinasyonudur. Alt bölgeler küçük olduğundan bir alt bölge üzerinde tanımlı temel fonksiyonlar çok basit olabilir.

Bütün bölgenin temel fonksiyonlarını kullanarak problemi çözmek, alt bölge temel fonksiyonlarını kullanarak çözmeye göre çok daha az çaba gerektirir. Bu durumda sonlu eleman metodunu kullanmanın ilk nedeni, düzensiz sınırları olan iki ve üç boyutlu problemlerde bütün bölgeye ait gerekli deneme fonksiyonlarını bulmanın zorluğu, ikincisi ise, karmaşık problemlerin çözümünde, çözüm prosedürünü bilgisayar dilinde tasvir eden bir program yazarak, programı bilgisayar üzerinde çalıştırarak çözüme ulaşmada sonlu elemanlar metodunun en uygun metot olmasıdır.

2.3. Sonlu Elemanlar Yönteminin Temel Adımları

Bir sınır değer probleminin sonlu eleman analizi aşağıdaki temel adımları içerir:

1. Bölgenin ayrıştırılması veya alt bölgelere ayrılması,
2. Enterpolasyon fonksiyonlarının seçilmesi,
3. Denklem sisteminin formülasyonu,
4. Denklem sisteminin çözülmesi

2.3.1. Bölgenin ayrıştırılması

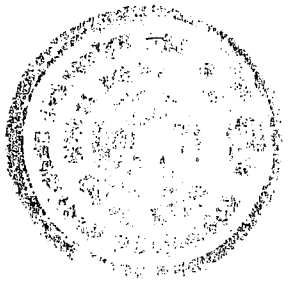
Bu adımda bütün Ω bölgesi, M alt bölgelerinin toplam sayısını göstermek kaydıyla Ω^e (e = 1,2,3,...,M) ile gösterilen küçük bölgelere ayrılmaktadır. Bu alt bölgelere genellikle elemanlar denilmektedir. Çoğu sonlu eleman çözümlerinde problem, elemanlarla bağlantılı düğüm noktalarında bilinmeyen fonksiyon ϕ cinsinden formüle edilir. Düğüm noktasının tam olarak belirtilmesi koordinat değerlerini, yerel sayısını ve global sayısını içerir. Düğümün yerel sayısı eleman içindeki konumunu belirtirken, global sayısı bütün sistem içindeki konumunu belirtir. Sonlu eleman formülasyonunun genellikle bant genişliği bir elemandaki iki düğümün global sayıları arasındaki azami fark tarafından belirlenen bir bant matrisle sonuçlanır.

Bu nedenle, eğer son matris denklemini çözmek için bir bant matris çözüm metodu kullanılacaksa, bilgisayar depolama ve işlem maliyeti bant genişliği minimuma inecek şekilde düğümlere uygun numaralar verilerek, önemli miktarda düşürülebilir. Ancak bant genişliği minimizasyonu gereksiz olduğunda numaralandırma keyfi olabilir ve genellikle programlamayı basitleştirecek şekilde seçilir.

2.3.2. Enterpolasyon fonksiyonlarının seçilmesi

Bir sonlu eleman analizinin ikinci adımı bir eleman içinde bilinmeyen fonksiyona yaklaşım sağlayan bir enterpolasyon fonksiyonunu seçmektir. Enterpolasyon genellikle 1., 2., veya daha yüksek dereceli polinom olarak seçilir. Yüksek dereceli polinomlar sonuçta karmaşık bir formülasyon ortaya çıkardığından basit ve temel 1. dereceden lineer enterpolasyon daha yaygındır. Polinomun derecesi seçildikten sonra bir eleman, örneğin e elemanı, içinde bilinmeyen fonksiyon için aşağıdaki şekilde ifade verilir;

$$\bar{\phi}^e = \sum_{j=1}^{n} N_j^e \phi_j^e = \{N^e\}^T \{\phi^e\} = \{\phi^e\}^T \{N^e\} \qquad (2.17)$$



Burada, ϕ_j^e elemanın j. düğümünde ϕ'nin değeri, N^e_j ise aynı zamanda açılım veya temel fonksiyonu olarak da bilinen enterpolasyon fonksiyonudur. N^e_j'nin en yüksek derecesine elemanın derecesi denir. N^e_j fonksiyonlarının önemli bir özelliği, sadece e elemanının içinde sıfırdan farklı olmaları, bu elemanın dışında ise kaybolmalarıdır.

2.3.3. Denklem sisteminin formülasyonu

Denklem sisteminin formüle edilmesi için önceki bölümlerdekine benzer tarzda hem Ritz hem de Galerkin varyasyon metotları kullanılır. Denklem sisteminin özel bir çözüm için çözülmeye hazır hale gelebilmesi için gereken sınır şartları uygulanmalıdır. Çoğunlukla karşılaşılan iki sınır şartı vardır. Biri, sınırdaki ϕ'yi öngören Dirichlet sınır şartı, diğeriyse ϕ'nin normal türevinin sınırda kaybolmasını öngören homojen Neumann sınır şartıdır.

2.3.4. Denklem sisteminin çözümü

Sonlu eleman analizinin sonucunda ortaya çıkan sistem aşağıdaki iki şekilden birine sahiptir:

$$[K]\{\phi\} = \{b\} \qquad (2.18)$$

veya

$$[A]\{\phi\} = \lambda [B]\{\phi\} \qquad (2.19)$$

(2.18) denklemi ya homojen olmayan bir diferansiyel denklemden veya homojen olmayan sınır şartlarından veya her ikisinden de doğan deterministik türdendir. Elektromanyetikte deterministik sistemler genellikle bir kaynak veya tahrikin mevcut olduğu yerlerdeki dağılma, radyasyon ve diğer deterministik problemlerle ilişkilidir (2.19) ise bunun aksine tam değer türünden olup, olayın bağlı olduğu homojen bir diferansiyel denklemle homojen sınır şartlarından doğmaktadır. Elektromanyetikte tam değer sistemleri genellikle kaynaksız problemlerle ilişkilidir. Bu halde bilinen vektör {b} kaybolur ve [K] matrisi [A] - λ [B] şeklinde yazılır. Burada λ bilinmeyen tam değerleri göstermektedir.

2.4. Sonlu Elemanlar Yöntemi ile İki Boyutlu Problemlerin Çözümü

Sonlu elemanlar yöntemi, birçok fiziksel problemin matematiksel modellerinin genellikle analitik bir çözüm bulunamayacak kadar karmaşık olduğu iki boyutlu problemlerde çok kullanılmaktadır. Bugün artık sonlu elemanlar yöntemi iki boyutlu problemler için çok geliştirilmiş olup, pek çok bilim ve mühendislik dalında önemli rol oynamaktadır. Bu bölümde öncelikle basit lineer üçgen elemanlar kullanılarak genel iki boyutlu sınır değer teoremi için sonlu eleman çözümü formüle edilecektir.

2.4.1. Sınır değer teoremi

2. dereceden sınır değer problemi aşağıdaki ikinci derece diferansiyel denklemle tanımlanmaktadır:

$$-\frac{\partial}{\partial x}\left(\alpha_x \frac{\partial \phi}{\partial x}\right)-\frac{\partial}{\partial y}\left(\alpha_y \frac{\partial \phi}{\partial y}\right)+\beta\phi = f \qquad (x,y) \in \Omega \qquad (2.20)$$

Burada ϕ bilinmeyen fonksiyon, α_x, α_y ve β bölgenin fiziksel özellikleriyle ilişkili bilinen parametreler ve f de kaynak veya tahrik fonksiyonudur. İki boyutlu Laplace denklemi, Poisson denklemi ve Helmholtz denklemi, (2.20) denkleminin özel şekilleridir.

Göz önüne alınacak sınır şartları:

$$\phi = p, \qquad \Gamma_1 \text{ boyunca} \qquad (2.21)$$

ve

$$\left(\alpha_x \frac{\partial \phi}{\partial x}\hat{x}+\alpha_y \frac{\partial \phi}{\partial y}\hat{y}\right).\hat{n}+\gamma\phi = q, \qquad \Gamma_2 \quad \text{boyunca} \qquad (2.22)$$

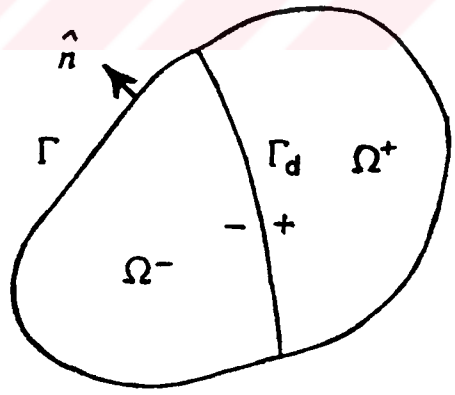
Burada $\Gamma = (\Gamma_1 + \Gamma_2)$, Ω alanını çevreleyen sınırı gösterirken, $\hat{n}$ dışarı yönlenmiş normal birim vektör, γ, p ve q ise sınırın fiziksel özellikleriyle ilişkili bilinen parametreleri göstermektedir. Neumann sınır şartı $\gamma = 0$ yapılarak elde edilen (2.22)'nin özel bir halidir. Eğer α_x ve α_y ile tanımlanan bölgenin özellikleri süreksiz ve ani değişimler gösteriyorsa ve bundan başka eğer süreksizlik ara yüzeyinde herhangi bir yüzey kaynağı yoksa, ϕ, aşağıdaki süreklilik şartlarını sağlar.

$$\phi^+ = \phi^-, \qquad \Gamma_d \quad \text{boyunca} \tag{2.23}$$

ve

$$\left(\alpha_x^+ \frac{\partial \phi^+}{\partial x}\hat{x} + \alpha_y^+ \frac{\partial \phi^+}{\partial y}\hat{y}\right).\hat{n} = \left(\alpha_x^- \frac{\partial \phi^-}{\partial x}\hat{x} + \alpha_y^- \frac{\partial \phi^-}{\partial y}\hat{y}\right).\hat{n} \qquad \Gamma_d \text{ boyunca} \tag{2.24}$$

Burada, Γ_d süreksizlik ara yüzeyini, "+" veya "-" üst sembolü ilişkili büyüklüklerinin Γ_d'nin "+" veya "-" tarafında bulunduğunu, $\hat{n}$ ise Γ_d'ye göre normal birim vektörünü göstermektedir.



Şekil 2.1. Γ_d ile gösterilen bir süreksizlik ara yüzeyine sahip bölge

2.4.2. Varyasyonel formülasyon

Yukarıdaki sınır değer problemine eşdeğer varyasyonel problem şöyle verilir.

$$\begin{cases} \delta F(\phi) = 0 \\ \phi = p \qquad\qquad \Gamma_1 \quad \text{boyunca} \end{cases} \tag{2.25}$$

Burada şu tanım yapılmıştır:

$$F(\phi) = \frac{1}{2}\iint_{\Omega}\left[\alpha_x\left(\frac{\partial\phi}{\partial x}\right)^2 + \alpha_y\left(\frac{\partial\phi}{\partial y}\right)^2 + \beta\phi^2\right]d\Omega$$

$$+ \int_{\Gamma_2}\left(\frac{\gamma}{2}\phi^2 - q\phi\right)d\Gamma - \iint_{\Omega} f\phi\, d\Omega \tag{2.26}$$

Eğer bir süreksizlik ara yüzeyi varsa (2.25)'in mutlaka süreklilik şartı (2.23) ile tamamlanması gerekir. (Jin, 1993)

Varyasyonel formülasyonu uygulayabilmek için önce F (ϕ)'nin ϕ'ye göre ilk varyasyonu alınarak şu ifade bulunur.

$$\delta F(\phi) = \iint_{\Omega}\left[\alpha_x\left(\frac{\partial\phi}{\partial x}\right)\left(\frac{\partial\delta\phi}{\partial x}\right) + \alpha_y\left(\frac{\partial\phi}{\partial y}\right)\left(\frac{\partial\delta\phi}{\partial y}\right) + \beta\phi\delta\phi\right]d\Omega$$

$$+ \int_{\Gamma_2}(\gamma\phi - q)\delta\phi\, d\Gamma - \iint_{\Omega} f\,\delta\phi\, d\Omega \tag{2.27}$$

Bundan sonra aşağıdaki eşitlikler,

$$\alpha_x\left(\frac{\partial\phi}{\partial x}\right)\left(\frac{\partial\delta\phi}{\partial x}\right) = \frac{\partial}{\partial x}\left(\alpha_x\frac{\partial\phi}{\partial x}\delta\phi\right) - \left[\frac{\partial}{\partial x}\left(\alpha_x\frac{\partial\phi}{\partial x}\right)\right]\delta\phi \tag{2.28}$$

$$\alpha_y \left(\frac{\partial \phi}{\partial y}\right)\left(\frac{\partial \delta\phi}{\partial y}\right) = \frac{\partial}{\partial y}\left(\alpha_y \frac{\partial \phi}{\partial y}\delta\phi\right) - \left[\frac{\partial}{\partial y}(\alpha_y \frac{\partial \phi}{\partial y})\right]\delta\phi \tag{2.29}$$

ve α_x ve α_y'nin bütün bölge üzerinde sürekli olduğu farz edilerek diverjans teoremi

$$\iint_{\Omega}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)d\Omega = \oint_{\Gamma}(U\hat{x}+V\hat{y})\hat{n}d\Gamma \tag{2.30}$$

kullanılarak (2.27) şöyle yazılabilir.

$$\begin{aligned}\delta F(\phi) = &\iint_{\Omega}\left[-\frac{\partial}{\alpha x}\left(\alpha_x \frac{\partial \phi}{\alpha x}\right) - \frac{\partial}{\partial y}\left(\alpha_y \frac{\partial \phi}{\partial y}\right) + \beta\phi - f\right]\delta\phi d\Omega \\ &+ \oint_{\Gamma}\left[(\alpha_x \frac{\partial \phi}{\partial x}\hat{x} + \alpha_y \frac{\partial \phi}{\partial y}\hat{y}).\hat{n}\right]\delta\phi d\Gamma \\ &+ \int_{\Gamma_2}(\gamma\phi - q)\delta\phi d\Gamma\end{aligned} \tag{2.31}$$

ϕ, Γ_1 üzerinde sabit bir değere sahip olduğundan $\delta\phi$, Γ_1 boyunca sıfır olur, dolayısıyla karşılık gelen integral de sıfır olur. Böylece (2.31), şöyle yazılabilir.

$$\begin{aligned}\delta F(\phi) = &\iint_{\Omega}\left[-\frac{\partial}{\partial x}\left(\alpha_x \frac{\partial \phi}{\partial x}\right) - \frac{\partial}{\partial y}\left(\alpha_y \frac{\partial \phi}{\partial y}\right) + \beta\phi - f\right]\delta\phi d\Omega \\ &+ \int_{\Gamma_2}\left[(\alpha_x \frac{\partial \phi}{\partial x}\hat{x} + \alpha y \frac{\partial \phi}{\partial y}\hat{y}).\hat{n} + \gamma\phi - q\right]\delta\phi d\Gamma\end{aligned} \tag{2.32}$$

Son olarak, sabitlik şartlarının dikkate alınmasıyla, aşağıdaki denklem elde edilir.

$$\iint_{\Omega}\left[-\frac{\partial}{\partial x}\left(\alpha_x\frac{\partial\phi}{\partial x}\right)-\frac{\partial}{\partial y}\left(\alpha_y\frac{\partial\phi}{\partial y}\right)+\beta\phi-f\right]\delta\phi d\Omega$$

$$+\int_{\Gamma_2}\left[(\alpha_x\frac{\partial\phi}{\partial x}\hat{x}+\alpha_y\frac{\partial\phi}{\partial y}\hat{y}).\hat{n}+\gamma\phi-q\right]\delta\phi d\Gamma=0 \qquad (2.33)$$

(2.33)'ün kabul edilebilir bütün $\delta\phi$ varyasyonları halinde geçerli olabilmesi için alan integrali ile çizgi integralinin birbirinden bağımsız olarak ortadan kalkması gerekir. Sonuç olarak aşağıdaki denklemler elde edilir.

$$-\frac{\partial}{\partial x}\left(\alpha_x\frac{\partial\phi}{\partial x}\right)-\frac{\partial}{\partial y}\left(\alpha_y\frac{\partial\phi}{\partial y}\right)+\beta\phi-f=0 \qquad (2.34)$$

$$\left(\alpha_x\frac{\partial\phi}{\partial x}\hat{x}+\alpha_y\frac{\partial\phi}{\partial y}\hat{y}\right).\hat{n}+\gamma\phi-q=0, \quad \Gamma_2 \quad \text{boyunca} \qquad (2.35)$$

Bunlar (2.20) ve (2.22) eşitlikleri ile aynıdır. Bu problemde (2.21) ile verilen sınır şartı Dirichlet sınır şartı, (2.22) ile verilen sınır şartı Neumann veya doğal sınır şartıdır. Diverjans teoremi (2.30)'un uygulanmasında, α_x ve α_y'nin bütün bölge üzerinde sürekli kabul edildikleri belirtilmelidir. Bununla beraber, eğer α_x ve α_y'de bir süreksizlik veya ani değişim varsa, önce bütün bölgeyi içinde α_x ve α_y'nin sürekli olduğu alt bölgelere ayırıp, daha sonra da bunların her birine diverjans teoremi uygulanabilir. Daha açık ifade etmek için, bölgenin Γ_d ara yüzeyiyle ayrılmış iki alt bölgeye ayrıldığı kabul edilir. Yukarıda açıklanan prosedür takip edilerek (2.33) ifadesi, aşağıdaki iki integral ifadenin sol tarafına eklenerek tekrar elde edilir.

$$\int_{\Gamma_d}\left(\alpha_x^{+}\frac{\partial\phi}{\partial x}\hat{x}+\alpha_y^{+}\frac{\partial\phi^{+}}{\partial y}\hat{y}\right).\hat{n}^{+}\delta\phi^{+}d\Gamma$$

$$+\int_{\Gamma_d}\left(\alpha_x^{-}\frac{\partial\phi^{-}}{\partial x}\hat{x}+\alpha_y^{-}\frac{\partial\phi^{-}}{\partial y}\hat{y}\right).\hat{n}^{-}\delta\phi^{-}d\Gamma \qquad (2.36)$$

Burada, n^+ , Γ_d'ye normal birim vektörü göstermekte olup, "+" taraftan "-" tarafa yönelmiştir n^- zıt yöne sahip birim vektörü göstermektedir. ϕ, (2.23)'de öngörüldüğü şekilde Γ_d'nin bir tarafından diğer tarafına sürekli olduğundan, $\delta\phi^+ = \delta\phi^-$ olur. Böylece (2.36) şu şekilde yeniden yazılabilir.

$$\int_{\Gamma_d}\left[\left(\alpha_x{}^+\frac{\partial\phi^+}{\partial x}\hat{x}+\alpha_y{}^+\frac{\partial\phi^+}{\partial y}\hat{y}\right).\hat{n}^+ +\left(\alpha_x{}^-\frac{\partial\phi^-}{\partial x}\hat{x}+\alpha_y{}^-\frac{\partial\phi^-}{\partial y}\hat{y}\right).\hat{n}^-\right]\delta\phi d\Gamma \qquad (2.37)$$

$\delta\phi$'nin keyfi değişmesinden dolayı, (2.34) ve (2.35)'e ilaveten başka bir Neumann şartı olan aşağıdaki denklem elde edilir.

$$\left(\alpha_x{}^-\frac{\partial\phi^+}{\partial x}\hat{x}+\alpha_y{}^+\frac{\partial\phi^+}{\partial y}\hat{y}\right).\hat{n}^+ +\left(\alpha_x{}^-\frac{\partial\phi^-}{\partial x}\hat{x}+\alpha_y{}^-\frac{\partial\phi^-}{\partial y}\hat{y}\right).\hat{n}^- = 0 \qquad (2.38)$$

Son olarak, (2.26) fonksiyoneli ister reel ister kompleks olsun, α_x, α_y, β ve γ değerleri için geçerlidir. Eğer bu parametreler sadece reel olursa, aşağıdaki fonksiyonel kullanılabilir.

$$F(\phi)=\frac{1}{2}\iint_\Omega\left[\alpha_x\left|\frac{\partial\phi}{\partial x}\right|^2+\alpha_y\left|\frac{\partial\phi}{\partial y}\right|^2+\beta|\phi|^2\right]d\Omega+\frac{1}{2}\int_{\Gamma_2}\left(\gamma|\phi|^2-q*\phi-q\phi*\right)d\Gamma$$
$$-\frac{1}{2}\iint_\Omega\left(f*\phi+f\phi*\right)d\Omega \qquad (2.39)$$

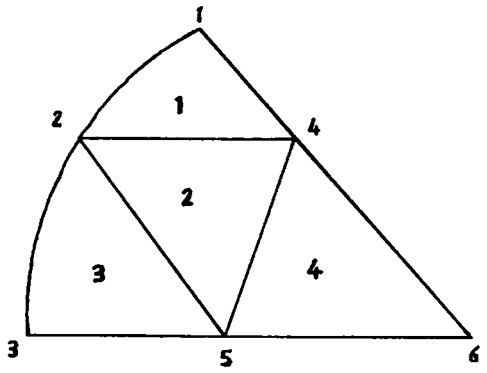
Açıktır ki, bu fonksiyonel reel bir büyüklüktür. Genellik bakımından formülasyonun temeli olarak (2.26)'da verilen fonksiyonel kullanılır. (2.39) denkleminde "*" işareti kompleks eşlenik işlemidir.

2.4.3. Sonlu eleman analizi

Bu bölümde, Bölüm (2.4.1)'de tanımlanan problemin çözümünü formüle etmek için sonlu elemanlar metodunun temel adımları incelenecektir. Basitlik için burada lineer üçgensel elemanlar kullanılacaktır.

2.4.3.1. Bölgenin ayrıştırılması

Sonlu eleman analizinin ilk adımı Ω alan bölgesini iki boyutlu elemanlara ayırmaktır. Yukarıda da belirtildiği gibi burada iki boyutlu elemanlar lineer üçgen elemanlardır. Ayrıştırmanın temel şartı elemanlar arasında üst üste binme ve aralarda boşluk almamasıdır. Elemanlar birbirlerine köşelerinden bağlanmalı, bir elemanın köşesi ancak komşusu olan elemanların köşelerinde olmalı, başka bir elemanın kenarında olmamalıdır. Ayrıştırmada ayrıca dar elemanların veya iç açıları küçük elemanların üretilmesi önlenmelidir. Sonlu eleman çözümünün hatası en küçük iç açının sinüsüyle ters orantılı olduğundan böyle elemanların seçilmesi çözüm hatasını arttırabilir. Bu nedenle, elemanların bütün kenarları mümkün olduğunca birbirine eşit yapılmalıdır. Ayrıca elemanlar ne kadar küçük olursa nümerik çözümün de o kadar iyi olacağına dikkat edilmelidir. Daha küçük elemanlar daha fazla bilinmeyenle sonuçlandığından ve dolayısıyla da hafiza talebini ve hesaplama süresini arttırdığından istenen hassasiyet için eleman sayısını minimumda tutmak gerekir. Her elemanı tanımak için elemanlar bir tamsayı takımıyla etiketlenebilir ve benzer şekilde, elemanların köşeleri olan düğümleri tanımak için de onlar başka bir tamsayı takımıyla etiketlenebilir. Her eleman birden fazla düğümle ilişkili olduğundan, bir düğüm bütün sistem içindeki konumuna ilaveten ilişkili olduğu eleman içinde de bir konuma sahiptir. Bu konum, bütün sistem içindeki konumunu gösteren global numaranın aksine yerel numara olarak anılan bir tamsayıyla da etiketlenebilir. Elemanların bu şekilde yerleşimlerinin tanımlanabilmesi için 3 x m boyutunda i= 1,2,3 ve e = 1,2,3,...., m olmak ve m toplam eleman sayısını göstermek üzere n (i,e) şeklinde gösterilen bir tamsayı matrisi tanımlanır. Burada i lokal (yerel) düğüm numarası, e eleman numarası ve n (i,e)'nin değeri de global düğüm numarasıdır. Bu tamsayı matrisi elemanların ve düğümlerin numaralandırılmasıyla ilgili tüm bilgileri içerir. n (i, e) matrisi, bağlantı matrisi olarak da isimlendirilebilir.



Şekil 2.2. İki boyutlu bir bölgenin alt bölgelere ayrılması

Yukarıdaki örnekte dört eleman ve altı düğüm bulunmaktadır. n (i,e) dizisi aşağıdaki gibi numaralandırılabilir:

e	n(1,e)	n(2,e)	n(3,e)
1	2	4	1
2	5	4	2
3	3	5	2
4	5	6	4

(2.40)

Bu numaralandırma keyfi seçilmiştir. Lokal numaralandırma saat ibresinin tersi yönde olmak şartıyla istenildiği gibi seçilebilir. Bundan başka (2.21) ile verilen Dirichlet sınır şartını koşmak için Γ_1 üzerinde yer alan global düğüm numaralarını depolayan bir vektör oluşturulacaktır.

Yukarıda belirtilen verilere ek olarak sonlu eleman formülasyonunda diğer bazı verilerde gerekli olacaktır.

1. Düğümlerin koordinatlarını veren x_i ve y_i (i=1,2,3,....,N). Burada N, toplam düğüm sayısını verir.
2. Her eleman için α_x, α_y, β ve f
3. Γ_1 üzerinde yer alan düğümler için p değeri (Dirichlet sınır şartı)

4. Γ_2 ile üst üste gelen her parça için γ ve q değerleri (Neumann sınır şartı)

2.4.3.2. Eleman enterpolasyon fonksiyonları

Eğer lineer üçgen elemanlar kullanılırsa, her eleman içinde bilinmeyen fonksiyon ϕ'ye şöyle yaklaşımda bulunulur.

$$\phi^e(x,y) = a^e + b^e x + c^e y \tag{2.41}$$

Burada, a^e, b^e ve c^e belirlenecek sabit katsayılar, e ise eleman numarasıdır. Lineer üçgen bir eleman için üçgenin köşelerinde yer alan üç düğüm vardır. Düğümlerde ϕ değişkeninin değerleri sırasıyla ϕ_1^e, ϕ_2^e, ϕ_3^e olarak gösterilir. (2.41)'e göre düğümlerdeki ϕ değerleri,

$$\phi_1^e = a^e + b^e x_1^e + c^e y_1^e$$

$$\phi_2^e = a^e + b^e x_2^e + c^e y_2^e$$

$$\phi_3^e = a^e + b^e x_3^e + c^e y_3^e$$

olur. Sabit katsayılar olan a^e, b^e ve c^e'yi ϕ_j^e cinsinden çözüp, bunlar (2.41)'de yerine konulursa şu bağıntı elde edilir.

$$\phi^e(x,y) = \sum_{j=1}^{3} N_j^e(x,y)\phi_j^e \tag{2.42}$$

Burada, $N_j^e(x,y)$ aşağıdaki gibi verilen enterpolasyon fonksiyonudur.

$$N_j^e(x,y) = \frac{1}{2\Delta^e}\left(a_j^e + b_j^e x + c_j^e y\right) \qquad j=1,2,3 \tag{2.43}$$

Fonksiyondaki katsayılar şu şekilde hesaplanır.

$$a_1^e = x_2^e y_3^e - y_2^e x_3^e \quad ; \quad b_1^e = y_2^e - y_3^e \quad ; \quad c_1^e = x_3^e - x_2^e$$

$$a_2^e = x_3^e y_1^e - y_3^e x_1^e \quad ; \quad b_2^e = y_3^e - y_1^e \quad ; \quad c_2^e = x_1^e - x_3^e$$

$$a_3^e = x_1^e y_2^e - y_1^e x_2^e \quad ; \quad b_3^e = y_1^e - y_2^e \quad ; \quad c_3^e = x_2^e - x_1^e$$

ve

$$\Delta^e = \frac{1}{2}\begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix} = \frac{1}{2}(b_1^e c_2^e - b_2^e c_1^e) = \text{e'inci elemanın alanı}$$

Yukarıda x_j^e ve y_j^e (j = 1,2,3), e'inci elemanın j'inci düğümünün koordinat değerlerini göstermektedir. (Reddy, 1984)

Enterpolasyon fonksiyonları aşağıdaki özelliğe sahiptir.

$$N_i^e(x_j^e, y_j^e) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad (2.44)$$

Sonuçta i düğümünde, (2.42)'deki ϕ^e düğüm değeri ϕ_i^e'ye indirgenir. N_j^e (x,y)'nin başka bir önemli özelliği gözlem noktası (x,y)'nın j'inci düğümün tam karşısındaki eleman kenarı üzerinde sıfır olmasıdır. Bu nedenle, ϕ^e'nin bir eleman kenarındaki değeri ϕ'nin zıt düğümdeki değeriyle ilişkili değildir ama bağlantılı olduğu iki uç noktasındaki değerlerle belirlenir. Bu önemli özellik, çözümün eleman kenarlarının bir tarafından diğer tarafına sürekliliğini garanti eder.

2.4.3.3. Ritz yöntemi yoluyla formülasyon

(2.42) de verilen ϕ açılımıyla Ritz ve Galerkin yöntemini kullanarak denklem sistemi formüle edilebilir. Önce Ritz yöntemi ile bu denklemlerin elde edilmesi açıklanacaktır.

A. Eleman eşitliklerinin çıkarılması

Basitlik olması açısından, önce (2.26) eşitliğinde verilen fonksiyoneldeki çizgisel integralin sıfırlandığı $\gamma = q = 0$ şartıyla (2.22)'nin özel bir hali olan homojen Neumann sınır şartı gözönüne alınır. Bu durumda fonksiyonel şöyle yazılabilir.

$$F(\phi) = \sum_{e=1}^{M} F^e(\phi^e) \tag{2.45}$$

Burada, M toplam eleman sayısını, F^e ise aşağıdaki gibi verilen alt fonksiyoneli göstermektedir.

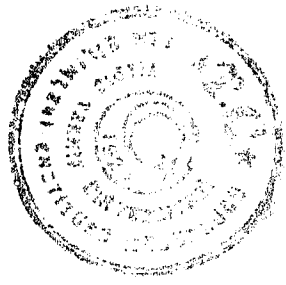
$$F^e(\phi^e) = \frac{1}{2}\iint_{\Omega^e}\left[\alpha_x\left(\frac{\partial\phi^e}{\partial x}\right)^2 + \alpha_y\left(\frac{\partial\phi^e}{\partial y}\right)^2 + \beta(\phi^e)^2\right] d\Omega - \iint_{\Omega^e} f\phi^e d\Omega \tag{2.46}$$

Burada ise Ω^e, e'inci elemanın bölgesini göstermektedir. ϕ^e yerine (2.42)'deki ifade konulup, ϕ_i^e'ye göre türev alınırsa şu bağıntı bulunur.

$$\frac{\partial F^e}{\partial \phi_i^e} = \sum_{j=1}^{3}\phi_j^e \iint_{\Omega^e}\left(\alpha_x\frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} + \alpha_y\frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} + \beta N_i^e N_j^e\right) d\Omega$$

$$- \iint_{\Omega^e} f N_i^e d\Omega \qquad i = 1,2,3 \tag{2.47}$$

veya



$$\left\{\frac{\partial F^e}{\partial \phi^e}\right\} = \left[K^e\right]\left\{\phi^e\right\} - \left\{b^e\right\} \tag{2.48}$$

Burada şu tanımlar yapılmıştır.

$$\left\{\frac{\partial F^e}{\partial \phi^e}\right\} = \left[\frac{\partial F^e}{\partial \phi_1^e}, \frac{\partial F^e}{\partial \phi_2^e}, \frac{\partial F^e}{\partial \phi_3^e}\right]^T \quad ; \quad \left\{\phi^e\right\} = \left[\phi_1^e, \phi_2^e, \phi_3^e\right]^T$$

$[K^e]$ matrisinin elemanları şöyle verilmektedir.

$$K_{ij}^e = \iint_{\Omega^e} \left(\alpha_x \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \alpha_y \frac{\alpha N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} + \beta N_i^e N_j^e \right) dxdy \quad i,j = 1,2,3 \tag{2.49}$$

$\{b^e\}$ vektörünün elemanlarıysa şöyle tanımlanmaktadır.

$$b_i^e = \iint_{\Omega^e} f N_i^e dxdy \qquad i = 1,2,3 \tag{2.50}$$

$[K^e]$ 'nin simetrik matris olduğu açıktır. Şimdi α_x, α_y, β katsayıları ile f kaynağının her elemanın içinde sabit ve sırasıyla α_x^e, α_y^e, β^e ve f^e'ye eşit oldukları farz edilirse, (2.49) ve (2.50) analitik olarak değerlendirilebilir. Bu amaçla kullanılarak temel formül aşağıda verilmiştir.

$$\iint_{\Omega^e} (N_1^e)^l (N_2^e)^m (N_3^e)^n dxdy = \frac{l!m!n!}{(l+m+n+2)!} 2\Delta^e \tag{2.51}$$

Bu formül kullanılarak aşağıdaki sonuçlar elde edilir.

$$K_{ij}^e = \frac{1}{4\Delta^e}\left(\alpha_x^e b_i^e b_j^e + \alpha_y^e c_i^e c_j^e\right) + \frac{\Delta^e}{12}\beta^e(1+\delta_{ij}) \tag{2.52}$$

$$b_i^e = \frac{\Delta^e}{3} f^e \tag{2.53}$$

Diğer taraftan, eğer α_x, α_y, β ve f her elemanda sabit değilse (2.52) ve (2.53) eşitliklerinden hesaplanan değerler, α_x^e, α_y^e, β^e ve f^e'nin ortalama değerleri kabul edilerek kullanılırlar. K_{ij}^e ve b_i^e sayısal olarak hesaplanabilir. Ancak bu durumda, elemanların oldukça küçük ve kullanılan yöntemin iyi sonuçlar vermesi gerekir.

B. Denklem sisteminin elde edilmesi

(2.48)'deki eleman matrisleriyle, bütün M elemanları birleştirilerek ve F'e sabitlik şartı eklenerek, denklem sistemi şu şekilde bulunabilir.

$$\left\{\frac{\partial F}{\partial \phi}\right\} = \sum_{e=1}^{M} \left\{\frac{\partial F^e}{\partial \phi^e}\right\} = \sum_{e=1}^{M} \left(\left[K^e\right]\left\{\phi^e\right\} - \left\{b^e\right\} \right) = \{0\} \tag{2.54}$$

Burada $\left\{\frac{\partial F^e}{\partial \phi^e}\right\}, \left\{\phi^e\right\}$ ve $\left\{b^e\right\}$ vektörleriyle $[K^e]$ matrisi genişletilmiş haldedir. Bu ifade daha basitleştirilmiş haliyle şöyle yazılabilir.

$$[K]\{\phi\} = \{b\} \tag{2.55}$$

Burada [K], $[K^e]$ eleman matrislerinin, {b} ise $\{b^e\}$ kaynak vektörlerinin birleştirilmiş halidir.

$$[K] = \sum_{e=1}^{M} \left[K^e\right] \quad , \qquad \{b\} = \sum_{e=1}^{M} \left\{b^e\right\} \tag{2.56}$$

Birleştirmeyi açıklayabilmek için, altı adet düğümü bulunan Şekil 2.2'deki örnek göz önüne alınabilir. Altı düğüm olduğundan [K] katsayılar matrisinin boyutu 6x6 olur. Daha sonra [K]'ya ilk eleman için eleman matrisi olan $[K^{(1)}]$ eklenir. Önce $K_{11}^{(1)}$ göz önüne alınır. (2.40)'da verilen bağlantı matrisine bakıldığında n (1,1,) = 2 bulunur ve $K_{11}^{(1)}$, K_{22}'ye

eklenir. Daha sonra $K_{12}^{(1)}$ göz önüne alınır. Yine (2.40)'a bakıldığında n (2,1) = 4 bulunur ve $K_{12}^{(1)}$, K_{24}'ün üzerine eklenir. Sürecin genel kuralının $K_{n(i,e),n(j,e)}$'ye K_{ij}^{e} eklemek olduğu açıktır. Bu prosedürü takip ederek [$K^{(1)}$]'in dokuz matris elemanını da [K]'ya ilave ettikten sonra aşağıdaki form elde edilir.

$$[K]=\begin{bmatrix} K_{33}^{(1)} & K_{31}^{(1)} & 0 & K_{32}^{(1)} & 0 & 0 \\ K_{13}^{(1)} & K_{11}^{(1)} & 0 & K_{12}^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_{23}^{(1)} & K_{21}^{(1)} & 0 & K_{22}^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.57}$$

Benzer şekilde, [$K^{(2)}$], [K]'ya eklenirse,

$$[K]=\begin{bmatrix} K_{33}^{(1)} & K_{31}^{(1)} & 0 & K_{32}^{(1)} & 0 & 0 \\ K_{13}^{(1)} & K_{11}^{(1)}+K_{33}^{(2)} & 0 & K_{12}^{(1)}+K_{32}^{(2)} & K_{31}^{(2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_{23}^{(1)} & K_{21}^{(1)}+K_{23}^{(2)} & 0 & K_{22}^{(1)}+K_{22}^{(2)} & K_{21}^{(2)} & 0 \\ 0 & K_{13}^{(2)} & 0 & K_{12}^{(2)} & K_{11}^{(2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.58}$$

Daha sonra [$K^{(3)}$], [K]'ya eklenirse,

$$[K]=\begin{bmatrix} K_{33}^{(1)} & K_{31}^{(1)} & 0 & K_{32}^{(1)} & 0 & 0 \\ K_{13}^{(1)} & K_{11}^{(1)}+K_{33}^{(2)}+K_{33}^{(3)} & K_{31}^{(3)} & K_{12}^{(1)}+K_{32}^{(2)} & K_{31}^{(2)}+K_{32}^{(3)} & 0 \\ 0 & K_{13}^{(3)} & K_{11}^{(3)} & 0 & K_{12}^{(3)} & 0 \\ K_{23}^{(1)} & K_{21}^{(1)}+K_{23}^{(2)} & 0 & K_{22}^{(1)}+K_{22}^{(2)} & K_{21}^{(2)} & 0 \\ 0 & K_{13}^{(2)}+K_{23}^{(3)} & K_{21}^{(3)} & K_{12}^{(2)} & K_{11}^{(2)}+K_{22}^{(3)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.59}$$

ve son olarak ta [$K^{(4)}$]'ü [K]'ya eklersek, sonuçta şu matrisi buluruz.

$$[K]=\begin{bmatrix} K_{33}^{(1)} & K_{31}^{(1)} & 0 & K_{32}^{(1)} & 0 & 0 \\ K_{13}^{(1)} & K_{11}^{(1)}+K_{33}^{(2)}+K_{33}^{(3)} & K_{31}^{(3)} & K_{12}^{(1)}+K_{32}^{(2)} & K_{31}^{(2)}+K_{32}^{(3)} & 0 \\ 0 & K_{13}^{(3)} & K_{11}^{(3)} & 0 & K_{12}^{(3)} & 0 \\ K_{23}^{(1)} & K_{21}^{(1)}+K_{23}^{(2)} & 0 & K_{22}^{(1)}+K_{22}^{(2)}+K_{33}^{(4)} & K_{21}^{(2)}+K_{31}^{(4)} & K_{32}^{(4)} \\ 0 & K_{13}^{(2)}+K_{23}^{(3)} & K_{21}^{(3)} & K_{12}^{(2)}+K_{13}^{(4)} & K_{11}^{(2)}+K_{22}^{(3)}+K_{11}^{(4)} & K_{12}^{(4)} \\ 0 & 0 & 0 & K_{23}^{(4)} & K_{21}^{(4)} & K_{22}^{(4)} \end{bmatrix} \quad (2.60)$$

Benzer bir işlemle, her $b_i^{e)}$yi $b_{n\,(i,e)}$'ye ekleyerek elde edilen {b} şu şekildedir.

$$\{b\}=\begin{Bmatrix} b_3^{(1)} \\ b_1^{(1)}+b_3^{(2)}+b_3^{(3)} \\ b_1^{(3)} \\ b_2^{(1)}+b_2^{(2)}+b_3^{(4)} \\ b_1^{(2)}+b_2^{(3)}+b_1^{(4)} \\ b_2^{(4)} \end{Bmatrix} \quad (2.61)$$

C. Dirichlet sınır şartlarının girilmesi

Denklem sisteminin çözülmeye hazır hale gelmesi için, Γ_1 üzerindeki düğümler için geçerli olan Dirichlet sınır şartı (2.21)'in uygulanması gerekir. Bunu anlayabilmek için yine Şekil 2.2'de verilen örnek göz önüne alınır. 3, 5 ve 6 düğümlerinin Γ_1 üzerinde olduğu ve sırasıyla öngörülen p_3, p_5 ve p_6 değerlerine sahip oldukları kabul edilir. $\phi_3 = p_3$ şartını uygulayabilmek için, aşağıdaki atamalar yapılır.

$$K_{33} = 1, \qquad K_{3i} = 0 \qquad i= 1,2,3,4,5,6 \quad (2.62)$$

Ancak bu atamalar, [K] matrisinin simetrikliğini bozmakta olup, önemli bir özellik olan simetriyi geri getirmek için aşağıdaki atamalar yapılabilir.

$$b_i \leftarrow b_i - K_{i3}p_3, \qquad K_{i3} = 0 \qquad i= 1,2,3,4,5,6 \quad (2.63)$$

Benzer şekilde $\phi_5 = p_5$ ve $\phi_6 = p_6$ kabulüyle [K] matrisi oluşturulur.

$$[K]=\begin{bmatrix} K_{11} & K_{12} & 0 & K_{14} & 0 & 0 \\ K_{21} & K_{22} & 0 & K_{24} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ K_{41} & K_{42} & 0 & K_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.64}$$

{b} kaynak vektörü ise,

$$\{b\}=\begin{bmatrix} b_1 - K_{13}p_3 - K_{15}p_5 - K_{16}p_6 \\ b_2 - K_{23}p_3 - K_{25}p_5 - K_{26}p_6 \\ p_3 \\ b_4 - K_{43}p_3 - K_{45}p_5 - K_{46}p_6 \\ p_5 \\ p_6 \end{bmatrix} \tag{2.65}$$

şeklini alır.

Böylece, çözülmeye hazır halde bulunan denklem sisteminin son şekli elde edilir. Ancak, 3., 5. ve 6. denklemler sistemden çıkarılırsa çözüm değişmeyecektir. Bu çıkarma işlemiyle denklem sistemi şu hale gelir.

$$\begin{bmatrix} K_{11} & K_{12} & K_{14} \\ K_{21} & K_{22} & K_{24} \\ K_{41} & K_{42} & K_{44} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_4 \end{Bmatrix} = \begin{Bmatrix} b_1 - K_{13}p_3 - K_{15}p_5 - K_{16}p_6 \\ b_2 - K_{23}p_3 - K_{25}p_5 - K_{26}p_6 \\ b_4 - K_{43}p_3 - K_{45}p_5 - K_{46}p_6 \end{Bmatrix} \tag{2.66}$$

Sonuç olarak, çok daha küçük bir sistemi çözmek suretiyle çözüm bulunabildiğinden, bu husus özellikle problemde bilinmeyen sayısı çok fazla olduğunda önem kazanmaktadır. Γ_1 üzerinde yer alan N_1 adet düğüme sahip genel bir problem için, eğer bunların global düğüm numaraları nd(i) adı verilen bir vektörde, ϕ'nin öngörülen değerleri de p(i) adı verilen bir

vektör içinde toplanırsa, sınır şartlarının uygulanması basitçe aşağıdaki değerlerin verilmesiyle gerçekleşir.

$$b_{nd\,(i)} = p(i), \quad K_{nd\,(i),\,nd\,(i)} = 1, \quad K_{nd(i),j} = 0, \quad j \neq nd\,(i) \tag{2.67}$$

ve

$$bj \leftarrow bj - K_{j,\,nd(i)}\, p\,(i), \quad K_{j,\,nd\,(i)} = 0 \quad , \quad j \neq nd\,(i), \quad i=1,2,3,\ldots,N_1 \tag{2.68}$$

Yukarıda açıklanan yaklaşıma ek olarak aşağıda açıklanacak bir başka yaklaşım da Dirichlet şartlarının uygulanmasında yaygın olarak kullanılmaktadır. Bunu açıklamak için Şekil 2.2. örneği için çıkarılan sisteme $\phi_3 = p_3$ şartının uygulanması göz önüne alınırsa, (2.62)'de düzenleme yapmak yerine 10^{70} gibi çok büyük bir sayı seçilerek aşağıdaki değerler kabul edilebilir.

$$K_{33} = 10^{70}, \quad b_3 = p_3 \times 10^{70} \tag{2.69}$$

Böyle yapıldığında, içinde ϕ_3'ü bulunduran denklem şu şekli alır.

$$K_{31}\phi_1 + K_{32}\phi_2 + 10^{70}\phi_3 + K_{34}\phi_4 + K_{35}\phi_5 + K_{36}\phi_6 = p_3 \times 10^{70} \tag{2.70}$$

Bütün matris elemanlarının ve bilinmeyenlerin 10^{70}'den çok daha küçük oldukları varsayımıyla (2.70), $\phi_3 = p_3$'e eşdeğer olur. Benzer şekilde $\phi_5 = p_5$ ve $\phi_6 = p_6$ şartları da eklenirse, lineer denklem sistemi şöyle olur.

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & 10^{70} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & 10^{70} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & 10^{70} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ p_3 x 10^{70} \\ b_4 \\ p_5 x 10^{70} \\ p_6 x 10^{70} \end{Bmatrix} \tag{2.71}$$

Burada simetri korunmaktadır. Γ_1 üzerinde yer alan N_1 adet düğüme sahip genel bir problem için, bu yaklaşım sınır şartlarını aşağıdaki değerleri kabul ederek uygulamaktadır.

$$K_{nd(i),\,nd\,(i)} = 10^{70}, \qquad b_{nd\,(i)} = p\,(i) \times 10^{70}, \qquad i=1,2,3,....,N_1 \tag{2.72}$$

Bu yaklaşım oldukça basit ve elverişlidir. Bununla beraber değişkenin değerinin bilindiği düğümlerin elimine edilmemesi dezavantajıdır. Hafiza talebinin azaltılması için geliştirilen çözüm yöntemlerinde kolaylıkla uygulanabilir.

3. SONLU ELEMANLAR YÖNTEMİ ile NONLİNEER PROBLEMLERİN ÇÖZÜMÜ ve υ-B^2 EĞRİLERİNİN SAYISAL MODELLENMESİ

3.1. Newton-Raphson Yöntemi ile Nonlineer Sistem Eşitliklerinin Çözümü

Newton-Raphson yönteminin tek boyutlu problemlere uygulanmasında, g'nin p gibi sabit bir noktasına kuadratik yakınsamayı sağlayacak bir Φ fonksiyonu bulunması gereklidir. Bu fonksiyon,

$$g(x) = x-\Phi(x)\, f(x) \tag{3.1}$$

özelliği taşımalıdır. Bu şarttan yola çıkarak, Newton metodunda $\Phi(x) = 1/f'(x)$ seçilir. Benzer yaklaşımı kullanarak, n boyutlu sistem için Φ fonksiyonunun yerini aşağıdaki matris alır,

$$A(x) = \begin{bmatrix} a_{11}(x) & a_{12}(x) & \ldots\ldots & a_{1n}(x) \\ a_{21}(x) & a_{22}(x) & \ldots\ldots & a_{2n}(x) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ a_{n1}(x) & a_{n2}(x) & \ldots\ldots & a_{nn}(x) \end{bmatrix} \tag{3.2}$$

Matrisin her bir elemanı $R^n \rightarrow R$'ye bir fonksiyondur. Prosedür için A(x);

$$G(x)=x-A(x)^{-1}F(x) \tag{3.3}$$

şeklinde F(x)=0 sistemine kuadratik bir yaklaşımla çözüm getirmelidir (Forghani vd., 1983). Burada A(x), G'nin sabit noktasında nonsingüle (diagonaldaki elemanları sıfırdan farklı) olmalıdır.

A(x)'in, elemanlarının daha sonra seçilecek olan nxn boyutlu bir matris olduğunu düşünelim. Bu matrisin, **F(x)=0** sisteminin p noktasında nonsingüler olduğu da kabul edilerek $b_{ij}(x)$'ler, $\mathbf{A(x)^{-1}}$'in elemanları olsun .

$\mathbf{G(x)=x-A(x)^{-1}F(x)}$ olduğundan;

$$g_i(x) = x_i - \sum_{j=1}^{n} b_{ij}(x).f_j(x) \qquad \text{ve}$$

$$\frac{\partial g_i(x)}{\partial x_k} = \begin{Bmatrix} 1 - \sum_{j=1}^{n} \left(b_{ij}(x)\frac{\partial f_i}{\partial x_k}(x) + \frac{\partial b_{ij}}{\partial x_k}(x) f_j(x) \right), i = k \\ -\sum_{j=1}^{n} \left(b_{ij}(x)\frac{\partial f_j}{\partial x_k}(x) + \frac{\partial b_{ij}}{\partial x_k}(x) f_j(x) \right), i \neq k \end{Bmatrix}$$

eşitlikleri elde edilir. Aynı zamanda p noktasındaki çözüm arandığından her i=1,2,....,n ve k=1,2.....,n için

$$\frac{\partial g_i(p)}{\partial x_k} = 0$$

olmalıdır. i=k için ;

$$0 = 1 - \sum_{j=1}^{n} b_{ij}(p)\frac{\partial f_j}{\partial x_i}(p)$$

$$\sum_{j=1}^{n} b_{ij}(p)\frac{\partial f_j}{\partial x_i}(p) = 1 \qquad (3.4)$$

olur. $i \neq k$ için

$$0=1-\sum_{j=1}^{n} b_{ij}(p)\frac{\partial f_j}{\partial x_k}(p)$$

$$\sum_{j=1}^{n} b_{ij}(p)\frac{\partial f_j}{\partial x_k}(p)=1 \qquad (3.5)$$

olur. Aşağıdaki şekilde bir **J**(x) matrisini tanımlayalım.

$$\mathbf{J}(x)=\begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots\cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots\cdots & \frac{\partial f_2(x)}{\partial x_n} \\ & & & \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \cdots\cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$$

(3.4) ve (3.5) şartlarından ;

$$\mathbf{A}(p)^{-1}\mathbf{J}(p)=\mathbf{I}$$

eşitliğinin sağlanması gerektiği görülebilir. Böylece ;

$$\mathbf{A}(p)=\mathbf{J}(p)$$

olacaktır. **A**(x) matrisinin elemanlarının, **J**(x) matrisindeki elemanlar olacağı görülmektedir.

$$\mathbf{G}(x)=x-\mathbf{J}(x)^{-1}\mathbf{F}(x) \qquad (3.6)$$

(3.6)'daki şekildeki bir **G** fonksiyonu tanımlanır ve iterasyon prosedürü $x^{(0)}$ başlangıç değerleri seçilerek $k\geq 1$ için üretilerek oluşturulur.

$\mathbf{X}^k=\mathbf{G}(x^{(k-1)})=\mathbf{x}^{(k-1)}-\mathbf{J}(x^{(k-1)})^{-1}\mathbf{F}(x^{(k-1)})$

Uygun başlangıç değerlerinin seçimi ve $\mathbf{J}(p)^{-1}$'in mevcut olması ile kuadratik bir yakınsama sağlanır. $\mathbf{J}(x)$ matrisi Jacobian matrisi olarak isimlendirilir. Yöntem uygulanırken her adımda $\mathbf{J}(x)$ matrisinin inversi alınır. Pratik olarak yöntem iki adımda uygulanır.

İlk adımda $\mathbf{J}(x^{(k)}).\mathbf{y} = -\mathbf{F}(x^{(k)})$ eşitliği sağlayacak bir $\mathbf{y}$ vektörü bulunur. İkinci adımda $\mathbf{y}$ vektörü $\mathbf{x}^{(k)}$ vektörüne eklenerek $\mathbf{x}^{(k+1)}$ yaklaşımı hesaplanır. Uygun bir algoritma aşağıdaki gibi verilebilir:

Nonlineer Sistemler İçin Newton-Raphson Algoritması:

INPUT : n bilinmeyen sayısı (denklem sayısı)

$\mathbf{x}=(x_1,x_2,.....,x_n)^T$ başlangıç değerleri

TOL: tolerans

N: Maksimum iterasyon sayısı

OUTPUT: $x=(x_1,x_2,.....,x_n)^T$ yaklaşık çözümü

Step 1: k=1

Step 2: While ($k \leq N$) da step 3-7

Step 3 : $\mathbf{F}(x)$ ve $\mathbf{J}(x)$'i hesapla

$$\mathbf{J}(x)_{ij}=\left(\frac{\partial f_i(x)}{\partial x_j}\right) \qquad i \geq 1 \ , \ j \leq n$$

Step 4 : $\mathbf{J}(x)\ \mathbf{y} = -\mathbf{F}(x)$ denklem sistemini çöz.

Step 5 : $\mathbf{x} = \mathbf{x} + \mathbf{y}$

Step 6 : IF $||\mathbf{y}||<TOL$ then OUTPUT(x)

Step 7 : k=k+1

Step 8 : OUTPUT ('Maksimum iterasyon sayısı aşıldı')

STOP

3.2. Newton-Raphson Yönteminin Nonlineer İzotropik Problemlere Birinci Dereceden Elemanlar için Uygulanışı

Skaler potansiyel P için aşağıdaki nonlineer eşitlik yazılabilir.

$$\nabla . (\mu \nabla P) = 0$$

Bu eşitlik nonlineerdir çünkü malzeme permeabilitesi, manyetik alanın fonksiyonudur.
Buna karşılık vektör potansiyel için,

$$\nabla x (\upsilon \nabla x \mathbf{A}) = \mathbf{J} \qquad (3.7)$$

yazılabilir. Yine malzeme relüktivitesi υ alana bağlı olduğundan (3.7) eşitliği nonlineerdir. (3.7) eşitliğindeki vektör potansiyel A, kartezyen koordinatlarda z yönünde olduğundan, (3.7) eşitliği iki boyutlu formda;

$$\frac{\partial}{\partial x} (\upsilon \frac{\partial A}{\partial x}) + \frac{\partial}{\partial y} (\upsilon \frac{\partial}{\partial y}) = - J \qquad (3.8)$$

şeklinde yazılır. (Silvester ve Ferrari, 1996)

(3.8) eşitliğini sonlu elemanlar yönteminde kullanabilmek için önce uygun bir fonksiyonel tanımlanmalıdır. Bu uygun fonksiyonel lineer durum için olanla aynıdır.

$$F (\mathbf{U}) = \int_{\Omega} \frac{\mathbf{B.H}}{2} d\Omega - \int_{\Omega} \mathbf{J.U} \, d\Omega \qquad (3.9)$$

(3.9) eşitliğinde ilk integral terimi, lineer problemde, depolanmış enerjiye eşittir. Dolayısıyla bu eşitlik,

$$F (\mathbf{U}) = \int W (\mathbf{U}) d\Omega - \int \mathbf{J.U} \, d\Omega \qquad (3.10)$$

şeklinde yazılabilir. Nonlineer malzemeler için de aynı ifade kullanılabilir. Vektör potansiyel problemlerinde,

$$\mathbf{B} = \nabla \times \mathbf{U} \tag{3.11}$$

ve

$$\mathbf{H} = \upsilon \, . \, \mathbf{B} \tag{3.12}$$

olur. Göz önüne alınan problemlerdeki bütün malzemeler için mıknatıslanma eğrilerinin monotonik artımlı, buna karşılık ilk türevlerinin de monotonik azalımlı olması gerekir. (3.10) fonksiyoneline göre enerjinin minimum yapılması,

$$U = \sum_i U_i \alpha_i(x,y) \tag{3.13}$$

olmak üzere,

$$\frac{\partial F}{\partial U_i} = 0 \tag{3.14}$$

olmasını gerektirir.

$$\int \left(\frac{\partial W}{\partial U_i} - J\alpha_i \right) d\Omega = 0 \tag{3.15}$$

Manyetik malzemeler için enerji,

$$W = \int \mathbf{H} . d\mathbf{B} \tag{3.16}$$

olduğundan (3.16) ve (3.12) eşitlikleri kullanılarak,

$$\frac{\partial W}{\partial U_i} = \frac{\partial}{\partial U_i} \int_0^B \upsilon b \, db \tag{3.17}$$

şeklinde yazılabilir. Burada b, integrasyonun yapay değişkenidir. Benzer sonuçlar skaler potansiyel problem için de uygulanabilir. Relüktivite, akı yoğunluğunun karesinin fonksiyonu ise, (3.17) eşitliği,

$$\frac{\partial W}{\partial U_i} = \frac{1}{2} \frac{\partial}{\partial U_i} \int_0^B \upsilon \, (b^2) \, d \, (b^2) \tag{3.18}$$

halini alır. Buradan,

$$\frac{\partial W}{\partial U_i} = \frac{1}{2} \upsilon \, (B^2) \frac{\partial}{\partial U_i} B^2 \tag{3.19}$$

olarak yazılabilir. (3.15) eşitliği matris formunda yazıldığında,

$$[\,S\,][\,U\,] = [\,J\,] \tag{3.20}$$

şeklindedir. Bu nonlineer problemin çözümünde Newton-Raphson yöntemi uygulanabilir.
A , bulunan kesin çözüm ve **U**, **A**'ya yaklaşım olmak üzere,

$$\mathbf{U} = \mathbf{A} - \delta \mathbf{U} \tag{3.21}$$

yazılır. F (**U**) , **U** civarında Taylor serisine açılırsa,

$$\frac{\partial F}{\partial U_i} = \frac{\partial F}{\partial U_i} \Big|_u + \sum_j \frac{\partial^2 F}{\partial U_i \partial U_j} \Big|_u \delta U_j + \ldots \tag{3.22}$$

Fakat (3.14) eşitliği **U** = **A** şartını gerektireceğinden serideki bütün terimlerin eğimi ortadan kalkmalıdır (Chedid, 1994). Serideki ikinci dereceden büyük terimler ihmal edilerek (3.22) ifadesi **U**'nun **A**'dan sapmasını hesaplayabilmek için,

$$\delta \mathbf{U} = -\mathbf{P}^{-1}\,\mathbf{V} \tag{3.23}$$

şeklinde yazılır. Burada **P**, Newton iterasyonunun Jacobian matrisidir.

$$P_{ij} = \frac{\partial^2 F}{\partial U_i \partial U_j} \tag{3.24}$$

V, **U**'da F(**U**)'nun eğimini göstermektedir.

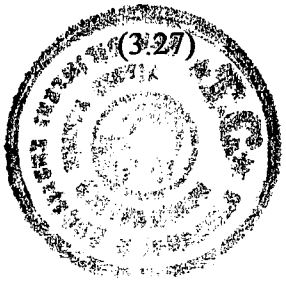
$$V_j = \frac{\partial F}{\partial U_j} \tag{3.25}$$

Şimdi (3.23) eşitliği kullanılarak, belli **U** değerleri başlangıç değerleri olarak seçilmek suretiyle, bir iteratif çözüm başlatılabilir. Sonuçta her adımda bulunan fark değeri, başlangıçta tahmin edilen potansiyel değerlerine eklenir. (3.22) ifadesindeki Taylor serisinde üçüncü ve daha yüksek mertebedeki terimler ihmal edildiğinden, (3.23) ile elde edilen çözüm yaklaşıktır. Bu çözüm ancak F (U) 'nun üçüncü ve daha yüksek mertebeden türevlerinin olmaması durumunda kesin çözümdür. Bu, lineer problemlerde söz konusudur ve iterasyon bir adımda yaklaşır. Nonlineer problemler için, daha iyi bir yaklaşım elde etmek amacıyla, hesaplanan fark **U**'ya eklenir ve işlem tekrarlanır.

$$\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} - [\,\mathbf{P}^{(k)}\,]^{-1}\,\mathbf{V}^{(k)} \tag{3.26}$$

Newton-Raphson yönteminin uygulanabilmesi için, (3.24) ve (3.25) eşitliklerindeki türevlerin bulunması gerekir. Türev işlemleri için aşağıdaki kurallar uygulanır.

$$\frac{\partial^2 F}{\partial U_i \partial U_j} = \int \frac{\partial^2 W}{\partial U_i \partial U_j} d\Omega \tag{3.27}$$



(3.19) ifadesindeki gibi yukarıdaki integral,

$$\frac{\partial^2 F}{\partial U_i \partial U_j} = \frac{\upsilon}{2} \frac{\partial^2 (B^2)}{\partial U_i \partial U_j} + \frac{1}{2} \frac{d\upsilon}{dB^2} \frac{\partial B^2}{\partial U_i} \frac{\partial B^2}{\partial U_j} \tag{3.28}$$

şeklini alır. Akı yoğunluğunun karesi için, (3.11) ve (3.13) eşitliklerini takiben

$$B^2 = \sum_m \sum_n (\Delta\alpha_m \Delta\alpha_n) U_m U_n \tag{3.29}$$

yazılır. Bu ifade (3.28) de yerine yazılırsa,

$$\frac{\partial^2 W}{\partial U_i \partial U_j} = \upsilon(\Delta\alpha_i \Delta\alpha_j) + 2\frac{2\upsilon}{d(B^2)} \sum_m \sum_n (\Delta\alpha_m \Delta\alpha_i)(\Delta\alpha_n \Delta\alpha_j) U_m U_n \tag{3.30}$$

Son eşitlik sağ taraftaki ilk terim, lineer problemdeki terimdir. İkinci ve karmaşık terim sadece nonlineer durumda vardır. Dolayısıyla iki durumda da matris yapısı aynı olmakla birlikte nonlineer problemler daha fazla işlem gerektirir. Ancak hafiza gereksinimi lineer problemlerle aynıdır.

3.3. Elektriksel Malzemelerin υ - B^2 Eğrilerinin Elde Edilmesi

Newton-Raphson yönteminin kullanılabilmesi için, eleman relüktiviteleri ile bunların B^2 'ye göre eğimlerinin bulunması gerekir. (Silvester vd., 1973)

$$W = B.\ H \tag{3.31}$$

şeklindeki enerji ifadesinden faydalanılarak,

$$W = \upsilon \cdot B^2 \tag{3.32}$$

$$B^2 = \frac{W}{\upsilon} \tag{3.33}$$

B-H eğrilerinden $\upsilon = f(B^2)$ eğrileri elde edilir.

3.3.1. Üstel fonksiyon yaklaşımı

3.3.1.1. El - Sherbiny yöntemi

B-H eğrileri için yapılan modelleme yöntemlerinden biridir. Bu yönteme göre B-H eğrilerinin değişimi,

$$B(H)=B_{max}(1-e^{-aH}) \tag{3.34}$$

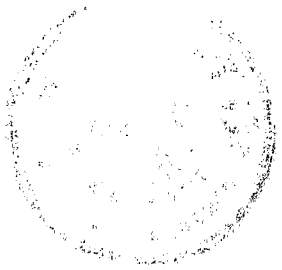
şeklindedir (El-Sherbiny, 1973). Buradan, B-H eğrileri için aşağıdaki üstel ifade yaklaşımı kullanılabilir.

$$B(H)=a_0+a_1.e^{-\alpha_1.H} + a_2.e^{-\alpha_2.H} + a_3.e^{-\alpha_3.H} + a_4.e^{-\alpha_4.H} \tag{3.35}$$

El - Sherbiny değişik malzemeler için a ve α katsayılarını hesaplamıştır. Bu katsayıların analitik olarak elde edilmesi oldukça zordur. Yalnız (3.35) eşitliğine göre olan, problemlerinde gerekli olan bazı değerler, örneğin permeabilite ve birim hacimde depo edilen enerji, hesaplanabilir. Permeabilite,

$$\mu=\frac{dB}{dH} = -(a_1.\alpha_1.e^{-\alpha_1.H} + a_2.\alpha_2.e^{-\alpha_2.H} + a_3.\alpha_3.e^{-\alpha_3.H} + a_4.\alpha_4.e^{-\alpha_4.H}) \tag{3.36}$$

şeklindedir.



3.3.1.2. Brauer yöntemi

Bu yöntem, El - Sherbiny yöntemine göre daha az katsayı içerir. Fonksiyon,

$$H=(k_1.e^{k_2 B^2}+k_3). \qquad (3.37)$$

şeklindedir (Brauer, 1975). Nonlineer problemlerin Newton - Raphson yöntemiyle çözümünde, relüktivite ve bunun B^2' ye göre eğimi gerekmektedir. $\upsilon = H / B$ olduğundan, (3.37) eşitliği,

$$\upsilon=\frac{H}{B}=k_1.e^{k_2.B^2}+k_3 \qquad (3.38)$$

halini alır. Buradan,

$$\frac{d\upsilon}{d(B^2)}=k_1.k_2.e^{k_2.B^2} \qquad (3.39)$$

olarak bulunur.

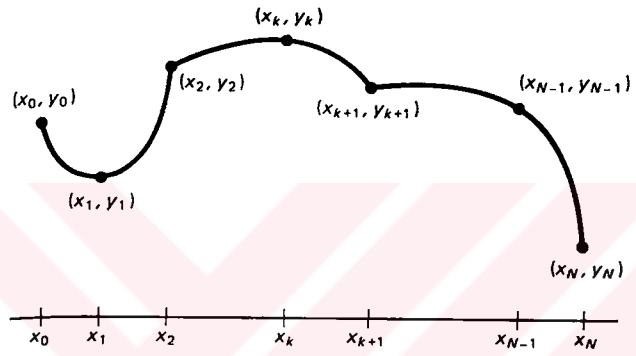
Malzemedeki manyetik enerji yoğunluğu,

$$W_m=\int_0^B H.dB=\frac{k_1}{2.k_2}(e^{k_2.B^2}-1)+\frac{1}{2}k_3 . B^2 \qquad (3.40)$$

olur. İfadelerdeki katsayıların malzemeye göre belirlenmesi iki yöntem için de dezavantaj oluşturur.

3.3.2. Kübik spline yöntemi

Kübik spline yöntemi, sonlu elemanlar yönteminin elektromanyetik problemlere uygulanmasında oldukça tercih edilen bir yöntem olmuştur. N+1 noktadan $\{ (x_k , y_k) \}$ oluşan bir dizinin polinom enterpolasyonu genellikle tatmin edici değildir. N'inci dereceden bir polinom, N-1 adet bağıl maksimum ve minimuma sahip olabilir ve tüm noktalardan geçecek şekilde eğri çizilebilir. Diğer bir yöntem, şekil (3.1)'de görüldüğü gibi, düşük dereceli $S_k(x)$ polinomlarının eğrilerini parça parça birleştirmek ve ard arda gelen (x_k, y_k) ve (x_{k+1}, y_{k+1}) düğümleri arasında enterpolasyon uygulamaktır.

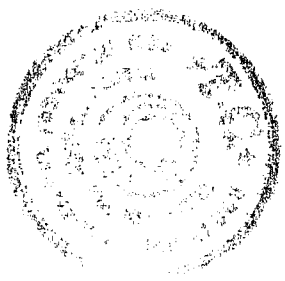


Şekil 3.1. Parçalı polinom enterpolasyonu

$[x_k , x_{k+1}]$ ve $[x_{k+1} , x_{k+2}]$ üzerinde bulunan $y = S_k(x)$ ve $y = S_{k+1}(x)$ eğrisinin iki komşu parçası (x_{k+1}, y_{k+1}) ortak düğümünden geçer. Eğrinin iki parçası (x_{k+1}, y_{k+1}) düğümünde birleşmişlerdir ve $\{ S_k(x)\}$ fonksiyonlar dizisi $S (x)$ olarak gösterilen bir parçalı polinom eğrisi oluşturur. (Mathews, 1992)

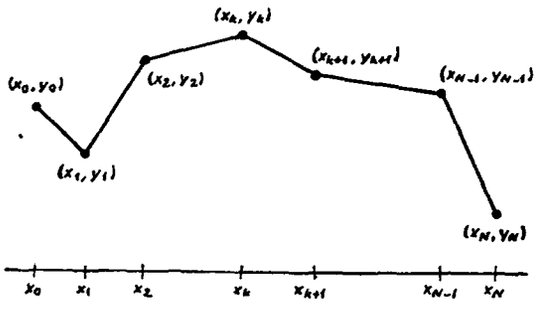
3.3.2.1. Parçalı lineer enterpolasyon

Kullanılabilecek en basit polinom, birinci dereceden bir polinomdur ve noktalardan geçen doğru parçalarından oluşan çok köşeli bir yol oluşturur. Bu lineer eğri Lagrange polinomu kullanılarak tanımlanabilir.



$$S_k(x) = y_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + y_{k+1} \frac{x - x_k}{x_{k+1} - x_k} \quad , x_k \le x \le x_{k+1} \tag{3.41}$$

Burada ortaya çıkan eğri Şekil 3.2'de görüldüğü gibidir.



Şekil 3.2. Parçalı lineer enterpolasyon

Bir $S_k(x) = y_k + d_k(x - x_k)$ doğru parçası için nokta kaydırma formülü kullanılarak eşdeğer bir ifade elde edilebilir. Burada $d_k = (y_{k+1} - y_k) / (x_{k+1} - x_k)$'dır. Buradan elde edilen lineer spline enterpolasyon şu şekilde yazılabilir.

$$S(x)=\begin{cases} y_0 + d_0(x - x_0) & x, [x_0, x_1] \\ y_1 + d_1(x - x_1) & x, [x_1, x_2] \\ \quad . & \quad . \\ \quad . & \quad . \\ y_k + d_k(x - x_k) & x, [x_k, x_{k+1}] \\ \quad . & \quad . \\ \quad . & \quad . \\ y_{N-1} + d_{N-1}(x - x_{N-1}) & x, [x_{N-1}, x_N] \end{cases} \tag{3.42}$$

S (x)'in açık hesabı için (3.42) denkleminin formu (3.41) denkleminin formundan daha iyidir. Apsislerin $x_0 < x_1 < \ldots < x_{N-1} < x_N$ şeklinde sıralandığı farz edilmektedir. x'in sabitlenmiş değeri için, x'i içeren $[x_k, x_{k+1}]$ aralığı, k+1'in $x - x_{k+1} < 0$ olmasını sağlayan en küçük sayı olmasına kadar, $x - x_1, \ldots, x - x_k, x - x_{k+1}$ farklarının ard arda hesaplanmasıyla

bulunabilir. Böylece $x_k \le x \le x_{k+1}$ 'i sağlayan k değeri bulunur ve S (x) spline fonksiyonunun değeri,

$$S(x) = S_k(x) = y_k + d_k(x - x_k) \qquad x_k \le x \le x_{k+1} \tag{3.43}$$

şeklinde olur. Bu teknikler yüksek dereceli polinomlara da uygulanabilir. Örneğin, toplamları tek sayı olan $x_0, \ldots, x_1, \ldots, x_{2M}$ düğümleri verilirse, $k = 0,1, \ldots, M - 1$ için her $[x_{2k}, x_{2k+2}]$ alt aralığında bir parçalı kuadratik polinom oluşturabilir. Elde edilen kuadratik spline'nın bir eksiği, çift x_{2k} düğümlerinde eğilmenin aniden değişmesidir ve bu, eğride istenmeyen eğim veya distorsiyonlara neden olabilir. Bir kuadratik spline'ın ikinci mertebeden türevi çift düğümlerde süreksizdir. Eğer parçalı kübik polinomlar kullanılırsa, birinci ve ikinci mertebeden türevler sürekli kılınmış olur.

3.3.2.2. Parçalı kübik spline

Matematiksel olarak, her $[x_k, x_{k+1}]$ aralığında $S_k(x)$ kübik fonksiyonları oluşturmak mümkündür, öyle ki, elde edilen $y = S(x)$ parçalı eğrisi ve birinci ve ikinci mertebeden türevlerinin her biri daha geniş $[x_0, x_N]$ aralığında sürekli olacaktır. S'(x) 'in sürekliliği $y = S(x)$ eğrisinin keskin köşeleri olmayacağı anlamına gelmektedir. S"(x)'in sürekliliği, "eğilme yarıçapı"nın her noktada tanımlanması anlamına gelmektedir.

Kübik spline arakesiti : $\{(x_k, y_k)\}_{k=0}^{N}$ 'ın $a = x_0 < x_1 < \ldots < x_N = b$ 'de N +1 nokta olduğu kabul edilir. Aşağıdaki özellikleri taşıyan N adet $S_k(x)$ kübik polinom varsa S(x) fonksiyonu kübik spline'dır.

I. $S(x) = S_k(x) = s_{k,0} + s_{k,1}(x - x_k) + s_{k,2}(x - x_k)^2 + s_{k,3}(x - x_k)^3$

her $k = 0,1, \ldots, N-1$ için $x \in [x_k, x_{k+1}]$

II. $S(x_k) = y_k$ $\quad k = 0,1, \ldots, N$ Spline her veri noktasından geçer.

III. $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$ $\quad k = 0,1, \ldots, N-2$ Spline sürekli bir fonksiyon oluşturur

IV. $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1})$ $k = 0,1,\ldots, N-2$ Spline pürüzsüz bir fonksiyon oluşturur.

V. $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1})$ $k = 0,1,\ldots,N-2$ İkinci mertebeden türev süreklidir.

3.3.2.3. Kübik splineların varlığı

I ve V özelliklerine uyan bir kübik spline oluşturmak mümkündür. Her $S_k(x)$ kübik polinomun dört bilinmeyen sabiti olduğuna göre bulunarak 4N katsayı vardır, yani belirtilmesi gereken 4N durum vardır. Veri noktaları N+1 durum verir, III, IV ve V özelliklerinin herbiri N-1 adet durum verir. Buradan, N+1+3(N-1) = 4N-2 durum belirtilir. Bu ise, uç nokta zorlamaları olarak adlandırılan iki ek durum ortaya çıkarır. Bunlar ya x_0 ve x_N'de $S'(x)$ yada $S''(x)$'i içerirler.

S(x) parçalı kübik olduğundan, ikinci mertebeden türevi $S''(x)$, $[x_0, x_N]$'de parçalı lineerdir. Lineer Lagrange enterpolasyon formülü, $S''(x) = S''_k(x)$ için aşağıdaki gösterimi verir.

$$S''_k(x) = S''(x_k)\frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1})\frac{x - x_k}{x_{k+1} - x_k} \qquad (3.44)$$

(3.44) eşitliğinde $m_k = S''(x_k)$, $m_{k+1} = S''(x_{k+1})$ ve $h_k = x_{k+1} - x_k$ yazılırsa;

$$S''_k(x) = \frac{m_k}{h_k}(x_{k+1} - x) + \frac{m_{k+1}}{h_k}(x - x_k) \quad , \quad x_k \le x \le x_{k+1} \quad \text{ve } k=0,1,\ldots,N-1 \qquad (3.45)$$

(3.45) eşitliğini iki kez integre etmek, iki integrasyon sabiti verecektir ve sonuç aşağıdaki forma girecek şekilde işlenebilir.

$$S_k(x) = \frac{m_k}{6h_k}(x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k}(x - x_k)^3 + p_k(x_{k+1} - x) + q_k(x - x_k) \qquad (3.46)$$

(3.46) eşitliğinde x_k ve x_{k+1} yerine konulursa ve $y_k=S_k(x_k)$ ve $y_{k+1}=S_k(x_{k+1})$ değerleri kullanılırsa, p_k ve q_k içeren aşağıdaki denkleme ulaşılır.

$$y_k = \frac{m_k}{6} h_k^2 + p_k h_k \qquad \text{ve}$$

$$y_{k+1} = \frac{m_{k+1}}{6} h_k^2 + q_k h_k \tag{3.47}$$

Bu iki denklem p_k ve q_k için çözülüp, bu değerler (3.46) eşitliğinde yerine yazılırsa, $S_k(x)$ kübik fonksiyonu için sonuç aşağıdaki gibi olur.

$$S_k(x) = \frac{m_k}{6h_k}(x_{k+1}-x)^3 + \frac{m_{k+1}}{6h_k}(x-x_k)^3 + \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right)(x_{k+1}-x) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6}\right)(x-x_k) \tag{3.48}$$

(3.48) eşitliğinin sadece $\{m_k\}$ bilinmeyen katsayılarını içeren bir forma indirgendiğine dikkat edilmelidir. Bu değerleri bulmak için (3.48) eşitliğinin türevi kullanılmalıdır. Bu da,

$$S_k'(x) = -\frac{m_k}{2h_k}(x_{k+1}-x)^2 + \frac{m_{k+1}}{2h_k}(x-x_k)^2 - \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6}\right) \tag{3.49}$$

(3.49) eşitliğini x_k' da değerlendirilerek sonuç basitleştirilirse,

$$S_k'(x_k) = -\frac{m_k}{3} h_k - \frac{m_{k+1}}{6} h_k + d_k \quad , \qquad d_k = \frac{y_{k+1}-y_k}{h_k} \tag{3.50}$$

Benzer şekilde, (3.49) eşitliğinde k yerine k-1 konularak $S'_{k-1}(x)$ ifadesi elde edilir. Bu, x_k' da değerlendirilirse,

$$S'_{k-1}(x_k) = \frac{m_k}{3} h_{k-1} + \frac{m_{k-1}}{6} h_{k-1} + d_{k-1} \tag{3.51}$$

elde edilir. IV numaralı özellik ve (3.50) ve (3.51) eşitlikleri kullanılarak, m_{k-1}, m_k ve m_{k+1} içeren eşitlik aşağıdaki şekilde elde edilir.

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \ , \ k = 1,2,...,N-1 \ \text{için} \ u_k = 6(d_k - d_{k-1}) \tag{3.52}$$

3.3.2.4. Kübik splineların oluşturulması

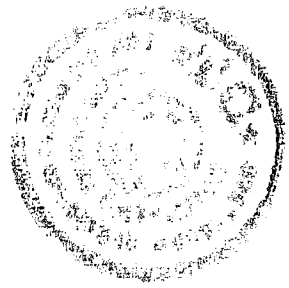
(3.52) eşitliğindeki bilinmeyenler, istenen $\{m_k\}$ değerleri ve diğer terimler ise $\{[x_k, y_k]\}$ veri noktalarıyla yapılan basit aritmetik işlemlerle elde edilen sabitlerdir. (3.52) eşitliği ile verilen sistem N+1 bilinmeyen içeren N-1 lineer denklemden oluşan sistemdir. Sistemin çözülebilmesi için iki ek denklem temin edilmelidir. Bunlar (3.52) denklem sisteminde eşitlik 1'den m_o'ı ve eşitlik N-1' den m_N' i kaldırmak için kullanılır. Uç nokta zorlamaları için standart stratejiler Çizelge 3.1'de özetlenmiştir.

Çizelge 3.1-5'te eğer m_o verilmemişse, $h_o m_o$ hesaplanabilir ve (3.52) denklem sisteminin ilk denklemi (k=1 olduğunda):

$$2[h_o + h_1]m_1 + h_1 m_2 = u_1 - h_o m_o \tag{3.53}$$

Benzer olarak, eğer m_N verilmişse, h_{N-1} m_N hesaplanabilir ve (3.52) denklem sisteminin son denklemi (k=N-1 iken):

$$h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = u_{N-1} - h_{N-1}m_N \tag{3.54}$$



olur. k=2,3,...,N-2 için (3.52) ile kullanılan (3.53) ve (3.54) denklemleri $m_1,m_2,...,m_{N-1}$ katsayılarını içeren N-1 adet lineer eşitliği oluşturur.

Çizelge 3.1. Bir kübik spline için uç nokta zorlamaları.

Stratejinin tarifi	m_o ve m_N içeren denklemler
"Kenetli kübik spline"; $S'(x_o)$ ve $S'(x_N)$'i belirler (türevler Biliniyorsa en iyi seçimdir)	$m_o = \frac{3}{h_o}\left[d_o - S'(x_o)\right] - \frac{m_1}{2}$ $m_N = \frac{3}{h_{N-1}}\left[S'(x_N) - d_{N-1}\right] - \frac{m_{N-1}}{2}$
2. "Doğal kübik spline"	$m_o = 0 \quad , \quad m_N = 0$
3. S" (x) uç noktalara extrapolate edilir.	$m_o = m_1 - \frac{h_o(m_2 - m_1)}{h_1}$ $m_N = m_{N-1} + \frac{h_{N-1}(m_{N-1} - m_{n-2})}{h_{N-2}}$
4. Uç noktalar yakınında S" (x) sabit	$m_o = m_1 \quad , \quad m_N = m_{N-1}$
5. S" (x) her uç noktada belirtilir.	$m_o = S''(x_o) \quad , \quad m_N = S''(x_N)$

Çizelge 3.1 'den seçilen stratejinin hangisi olduğunun önemi olmadan, (3.52) denklem sistemindeki 1 ve N-1 denklemleri yeniden yazılabilir ve $m_1,m_2,...,m_{N-1}$ içeren, HM=V formunda bir tridiagonal lineer sistem elde edilir.

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_1 & b_2 & & & & \\ & c_2 & & & & \\ & . & & & & \\ & . & & & & \\ & & a_{N-3} & b_{N-2} & c_{N-2} & \\ & & & a_{N-2} & b_{N-1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ . \\ . \\ . \\ m_{N-2} \\ m_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_{N-2} \\ v_{N-1} \end{bmatrix} \qquad (3.55)$$

3.5'teki lineer sistem diagonal olarak baskındır ve tek bir çözümü vardır. $\{m_k\}$ katsayıları belirlendikten sonra, $\{s_{k,j}\}$ spline katsayıları aşağıdaki formüller kullanılarak $S_k(x)$ için hesaplanır.

$$s_{k,o} = y_k\,, \quad s_{k,1} = d_k - \frac{h_k(2m_k + m_{k+1})}{6}\,,$$

$$s_{k,2} = \frac{m_k}{2}\,, \quad s_{k,3} = \frac{m_{k+1} - m_k}{6h_k} \tag{3.56}$$

Her $S_k(x)$ kübik polinomu, etkili hesaplama için aşağıdaki gibi yazılabilir.

$$S_k(x) = \left[\left(s_{k,3}w + s_{k,2}\right)w + s_{k,1}\right]w + y_k \quad , \quad w = x - x_k \tag{3.57}$$

$S_k(x)$, $x_k \le x \le x_{k+1}$ aralığında kullanılır.

(3.52) denklem sistemi, Çizelge 3.1'den seçilen bir stratejiyle birlikte uçlarında farklı özellikleri olan bir kübik spline'ı oluşturmak için kullanılabilir. Çizelge 3.1'deki m_o ve m_N için değerler, (3.52)'deki ilk ve sonuncu denklemleri istenilen hale getirmek için kullanılırlar ve (3.55)'te verilen N-1 denklem sistemini oluştururlar. Daha sonra, geriye kalan $m_1, m_2, \ldots, m_{N-1}$ katsayıları için tridiagonal sistem çözümlenir. Son olarak, (3.56)'daki formüller spline katsayılarını belirlemek için kullanılırlar.

3.3.2.5. Kübik splineların uygunluğu

Splineların önemli özelliklerinden biri, sahip oldukları salınım hareketinin minimumudur. Buna bağlı olarak, [a,b] üzerinde iki kat daha sürekli türetilebilir olan ve verilen bir $\left\{\left(x_k, y_k\right)\right\}_{k=0}^{N}$ veri noktaları dizisini değiştiren tüm f(x) fonksiyonları arasında, kübik spline daha az salınımlıdır.

a) Kübik splineların minimum özelliği: $f \in C^2 \,|a,b|$ olduğu ve S(x)' in $S'(a)=f'(a)$ ve $S'(b)=f'(b)$ kenetli uç koşullarını sağlayan, $\left\{ x_k, f(x_k) \right\}_{k=0}^{N}$'dan geçen f(x)' in tek kübik spline değişkeni olduğu kabul edilsin. O halde,

$$\int_a^b \left[S''(x)\right]^2 dx \le \int_a^b \left[f''(x)\right]^2 dx \qquad (3.58)$$

$$\int_a^b S''(x)\left[f''(x)-S''(x)\right] dx = S''(x)\left[f'(x)-S'(x)\right]\Big|_{x=a}^{x=b} - \int_a^b S'''(x)\left[f'(x)-S'(x)\right] dx$$

$$= 0 - 0 - \int_a^b S'''(x)\left[f'(x)-S'(x)\right] dx$$

$[x_k, x_{k+1}]$ aralığında $S'''(x) = 6\, s_{k,3}$ olduğundan,

$$\int_{x_k}^{x_{k+1}} S'''(x)\left[f'(x)-S'(x)\right] dx = 6s_{k,3}\left[f(x)-S(x)\right]\Big|_{x=x_k}^{x=x_{k+1}} = 0 \qquad k=0,1,\ldots,N-1$$

Buradan,

$$\int_a^b S''(x)\left[f''(x)-S''(x)\right] dx = 0 \quad \text{ve}$$

$$\int_a^b S''(x)f''(x)\,dx = \int_a^b \left[S''(x)\right]^2 dx \qquad (3.59)$$

$0 \le [f''(x)-S''(x)]^2$ olmak üzere

$$0 \le \int_a^b \left[f''(x) - S''(x)\right]^2 dx$$
$$= \int_a^b \left[f''(x)\right]^2 dx - 2\int_a^b f''(x)S''(x)dx + \int_a^b \left[S''(x)\right]^2 dx \qquad (3.60)$$

eşitliği (3.60)' da yerine yazılırsa,

$$0 \le \int_a^b \left[f''(x)\right]^2 dx - \int_a^b \left[S''(x)\right]^2 dx$$ elde edilir.

b) Kübik spline için uygun bir algoritma: (x_0 , y_0) , (x_1 , y_1) ,....., (x_N , y_N) gibi N+1 veri noktası için bir kübik spline değişkeni S(x)'in oluşturulması ve değerlendirilmesi . Uç nokta zorlamalarının aşağıdaki seçenekleri için şartlar belirlenmiştir.

(i) " Kenetli kübik spline" ; $S'(x_0)$ ve $S'(x_N)$ ' i belirler.

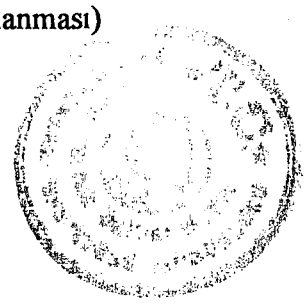
(ii) " Doğal kübik spline"

(iii) $S''(x)$, uç noktalara extrapolate edilir.

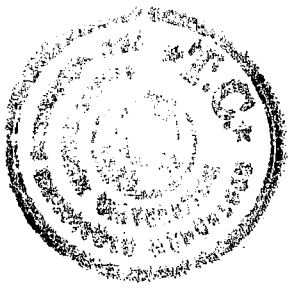
(iv) $S''(x)$, uç noktalara yakın olduğunda sabittir.

(v) $S''(x)$, her uç noktada belirlenir

```
H(0) := X(1) - X(0)                          (Apsisteki fark)
D(0) := [Y(1) - Y(0)]/H(0)                   (Fark bölüm)

FOR K=1 TO N-1 DO
    H(K) := X(K+1) - X(K)                    (Apsisteki fark)
    D(K) := [Y(K+1) - Y(K)]/H(K)             (Fark bölüm)
    A(K) := H(K)                             (Subdiagonal elemanlar)
    B(K) := 2*[H(K-1) + H(K)]                (Diagonal elemanlar)
    C(K) := H(K)                             (Superdiagonal elemanlar)

FOR K=1 TO N-1 DO                            (Sütun vektörünün
    V(K) := 6*[D(K) - D(K-1)]                 hesaplanması)
```



```
CASES (Matris ve/veya sütun vektörünün değiştirilmesi)
|
|  (i)   Set B(1) := B(1) - H(0)/2
|            V(1) := V(1) - 3*[D(0) - S'(x0)]                    (Giriş S'(x0))
|          B(N-1) := B(N-1) - H(N-1)/2
|          V(N-1) := V(N-1) - 3*[S'(xN) - D(N-1)]                (Giriş S'(xN))
|
|  (ii)  Set M(0) := 0 and M(N) :=0
|
|  (iii) Set B(1) := B(1) + H(0) + H(0)*H(0)/H(1)
|            C(1) := C(1) - H(0)*H(0)/H(1)
|          B(N-1) := B(N-1) + H(N-1) + H(N-1)*H(N-1)/H(N-2)
|          A(N-2) := A(N-2) - H(N-1)*H(N-1)/H(N-2)
|
|  (iv)  Set B(1) := B(1) + H(0) and B(N-1) := B(N-1) + H(N-1)
|
|  (v)   Set V(1) := V(1) - H(0)* S''(x0)                        (Giriş S''(x0))
|          V(N-1) := V(N-1) - H(N-1)* S''(xN)                    (Giriş S''(xN))
END

FOR     K=2 TO N-1 DO
|         T := A(K-1)/B(K-1)
|         B(K) := B(K) - T*C(K-1)
|_____    V(K) := V(K) - T*V(K-1)

        M(N-1) := V(N-1)/B(N-1)                          (Bir önceki adım
FOR     K=N-2 DOWNTO 1 DO                                mk'nın bulunmasında
|_____  M(K) := [V(K) - C(K)*M(K+1)]/B(K)                  kullanılır)

CASES {M(0) ve M(N) değerlerinin hesabı}
|
|   (i) Set M(0) := 3*[D(0) - S'(x0)]/H(0) - M(1)/2
|         M(N) := 3*[S'(xN) - D(N-1)]/H(N-1) - M(N-1)/2
|
|   (ii) Set M(0) := 0 and M(N) := 0
|
|   (iii) Set M(0) := M(1) - H(0)*[M(2) - M(1)]/H(1)
|         M(N) := M(N-1) + H(N-1)*[M(N-1) - M(N-2)]/H(N-2)
|
|   (iv) Set M(0) := M(1) and M(N) := M(N-1)
|
|   (v) Set M(0) := S''(x0) and M(N) := S''(xN)
END
```



```
FOR     K=0  TO  N-1  DO
|       S(K,0) := Y(K)                                        (Her bir S_k(x) kübik
|       S(K,1) := D(K) - H(K)*[2*M(K) + M(K+1)]/6             polinomu için
|       S(K,2) := M(K)/2                                      sabitler hesaplanır)
└_____  S(K,3) := [M(K+1) - M(K)]/[6*H(K)]

{[x_0,x_N] üzerinde kübik spline oluşturmak için prosedür}

INPUT   X                                                    (Bağımsız değişken girilir)
FOR     J=1  TO  N  DO
|       IF  X(J-1)≤X≤X(J)  THEN
└_____  └  Set  K := J-1  and  J := N                        (Aralığın bulunması)
        IF  X=X(0)  THEN  Set K := 0
        W := X - X(K)                                        (Spline'nın
        Z  := [[S(K,3)*W + S(K,2)]*W + S(K,1)]*W + S(K,0)    değerlendirilmesi)
PRINT   "Spline S(X)'in değeri" Z                            (Çıkış)
```

4. YAPAY SİNİR AĞLARI

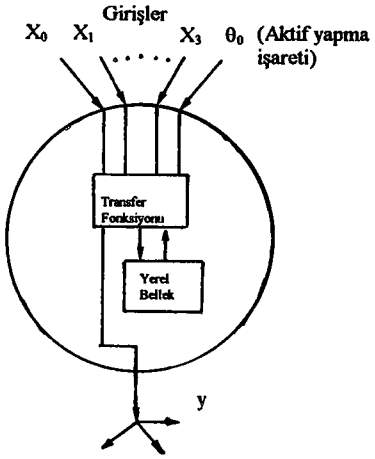
Yapay sinir ağları, insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. Önceleri temel tıp birimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar, bugün fizik, matematik, elektrik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. Mühendislik uygulamalarında yapay sinir ağlarının geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor problemler için uygun bir alternatif oluşturmasıdır. Yapay sinir ağları paralel dağılmış bir bilgi işleme sistemidir. Bu sistem tek yönlü işaret kanalları ile birbirine bağlanan işlem elemanlarından oluşur. Çıkış işareti bir tane olup isteğe göre çoğaltılabilir. Yapay sinir ağları çevre şartlarına göre davranışlarını şekillendirebilir. Girişler ve istenilen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir. Ancak son derece karmaşık bir içyapısı vardır. Onun için bugüne kadar, yapay sinir ağları; biyolojik fonksiyonların temel nöronlarını örnek alarak gerçekleştirilmiştir..

4.1. Yapay Sinir Ağlarının Yapısı ve İşlem Elemanı

Yapay sinir ağları temel olarak, basit yapıda ve yönlü bir graf biçimindedir. Her bir düğüm hücre denilen n. dereceden lineer olmayan bir devredir. Düğümler işlem elemanı olarak tanımlanır. Düğümler arasında bağlantılar vardır. Her bağlantı tek yönlü işaret iletim yolu (gecikmesiz) olarak görev yapar. Her işlem elemanı istenildiği sayıda giriş bağlantısı ve tek bir çıkış bağlantısı alabilir. Fakat bu bağlantı kopya edilebilir. Yani bu tek çıkış birçok hücreyi besleyebilir. Ağdaki tek gecikme çıkışları ileten bağlantı yollarındaki iletim gecikmeleridir. İşlem elemanının çıkışı istenilen matematiksel tipte olabilir. Kısmen sürekli çalışma konumunda "aktif" halde eleman bir çıkış işareti üretir. Giriş işaretleri yapay sinir ağlarına bilgi taşır. Sonuç ise çıkış işaretlerinden alınabilir. Şekil 4.1'de genel bir işlem elemanı (nöron, düğüm) gösterilmiştir. (Karlık, 1994)

Yapay sinir bağları birtakım alt kümelere ayrılabilir. Bu alt kümelerdeki elemanların transfer fonksiyonları aynıdır. Bu küçük gruplara "katman" adı verilir (örn: çok katmanlı perceptron MLP). Ağ katmanların birbirlerine hiyerarşik bir şekilde bağlanmasından

oluşmuştur. Dış dünyadan alınan bilgi giriş katmanı ile taşınır. Bir transfer fonksiyonları yoktur. Yapay sinir ağları transfer fonksiyonu ve yerel bellek elemanı bir öğrenme kuralı ile giriş çıkış işareti arasındaki bağıntıya göre ayarlanır. Aktif yapma girişi için bir zamanlama fonksiyonu tanımlaması gerekebilir.

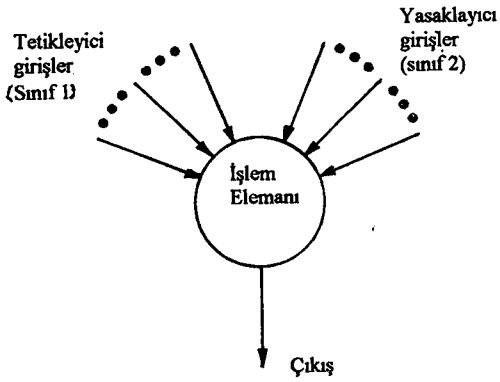


Şekil 4.1. Genel işlem elemanı yapısı

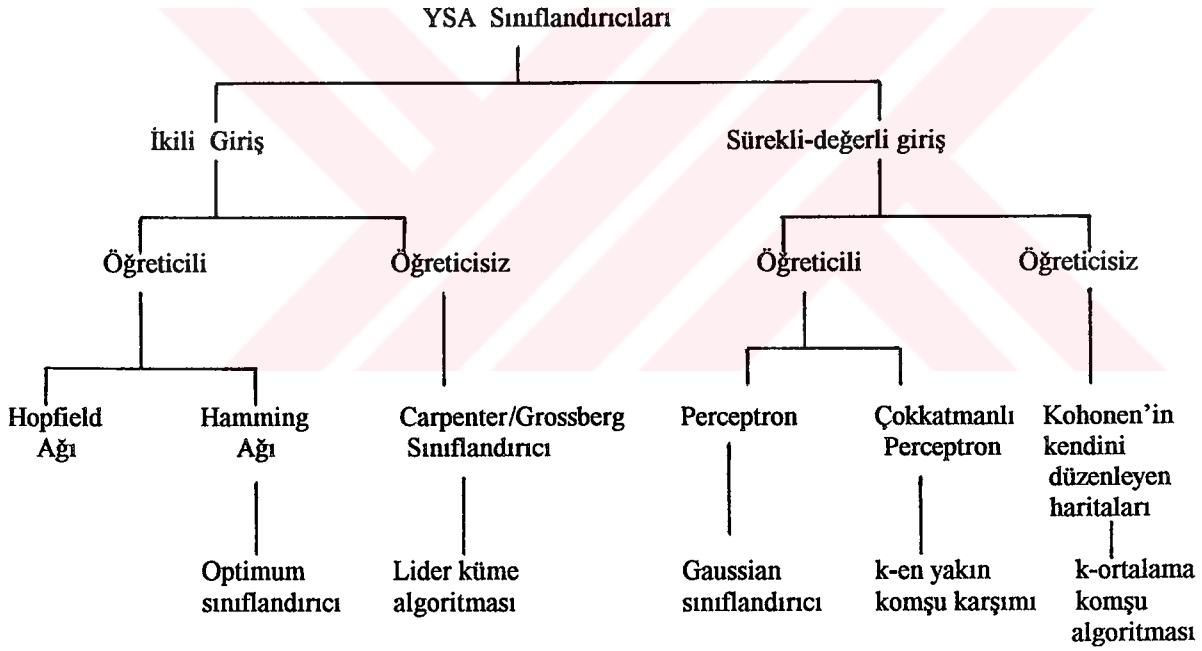
İşlem elemanının transfer fonksiyonu gelen bütün giriş işaretleri için tanımlanır. Bazen değişik katman davranışları farklı olabilir. İşaretlerin hangi bölgelerden geldiğinin bilinmesi gerekir. Değişik bölgelere göre işaretlerin sınıfları tamamlanabilir.

İşlem elemanı tetikleyici girişlerin kendine yakın komşu girişlerden yasaklanan girişlerini daha uzaktan alır. Böylece işlem elemanına gelen girişler sınıflarına göre değerlendirilmiş olur. Tetikleyici bölgeden gelen girişler yasaklanan sınıfı oluşturur. Şekil 4.2 böyle bir işlem elemanını gösterir.

Bir işlem elemanına gelen girişler matematiksel tiplerine göre etiketlendirilerek sınıflandırılır. Yapay sinir ağları giriş veri tiplerine göre ikili giriş (0,1) ve sürekli değerli giriş olmak üzere aşağıdaki gibi sınıflandırılır.



Şekil 4.2. Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı



Her işlem elemanı kendisine verilen yerel veriye göre, kendisini ayarlayacak bütün yapay sinir ağlarının enformasyon bölgesinin öğrenmesini sağlar. Enformasyon bölgesinde birçok uygulamada, gerçek değerin "0" ile "1" arasında normalize edilmesi gerekir. Bu gerçek

değeri 85 olan bir girişi 0.85 şeklinde ağa uygulamaktır. Normalizasyon aynı anda bütün girişlere uygulanabilir.

4.2. Bağlantı Geometrileri

Bağlantılarda taşınan işaret verisinin cinsi tanımlanmalıdır. Bağlantı geometrisi yapay sinir ağları için çok önemlidir, bağlantı işareti her cinsten olabilir. Bağlantının nerede başlayıp nerede bittiğinin bilmesi gerekir. 1'den N'e kadar olan bir işlem elemanı kümesinin bağlantıları aşağıda tanımlandığı gibi NxN boyutlu matris biçiminde gösterilebilir.

$$\left[w_{ij}\right] = \begin{bmatrix} w_{11} & w_{12} & \cdots\cdots & w_{1n} \\ w_{21} & w_{22} & \cdots\cdots & w_{2n} \\ . & . & . & . \\ . & . & . & . \\ w_{n1} & w_{n2} & & w_{nn} \end{bmatrix} \qquad (4.1)$$

$w_{ij} = w_{ji} = 1$ i. işlem elemanı j işlem elemanına bağlı

$w_{ij} = w_{ji} = 0$ bağlı değil.

En fazla N^2 bağlantı olur. Bağlantılar çeşitli geometrik bölgeler arasında demetler halinde düşünülebilir. Bu bağlantı demetlerinin uyması gereken kurallar şunlardır.

1. Bağlantı demetini oluşturan işlem elemanları aynı bölgeden çıkmalıdır.
2. Bağlantı demetinin işaretleri aynı matematiksel tipten olmalıdır.
3. Bağlantı demetinin işaretleri aynı sınıftan olmalıdır.
4. Bağlantı demetinin bir seçim fonksiyonu (σ) olmalıdır.

$\sigma : T \rightarrow 2^S$ T: Hedef bölgesi S: Kaynak bölgesi

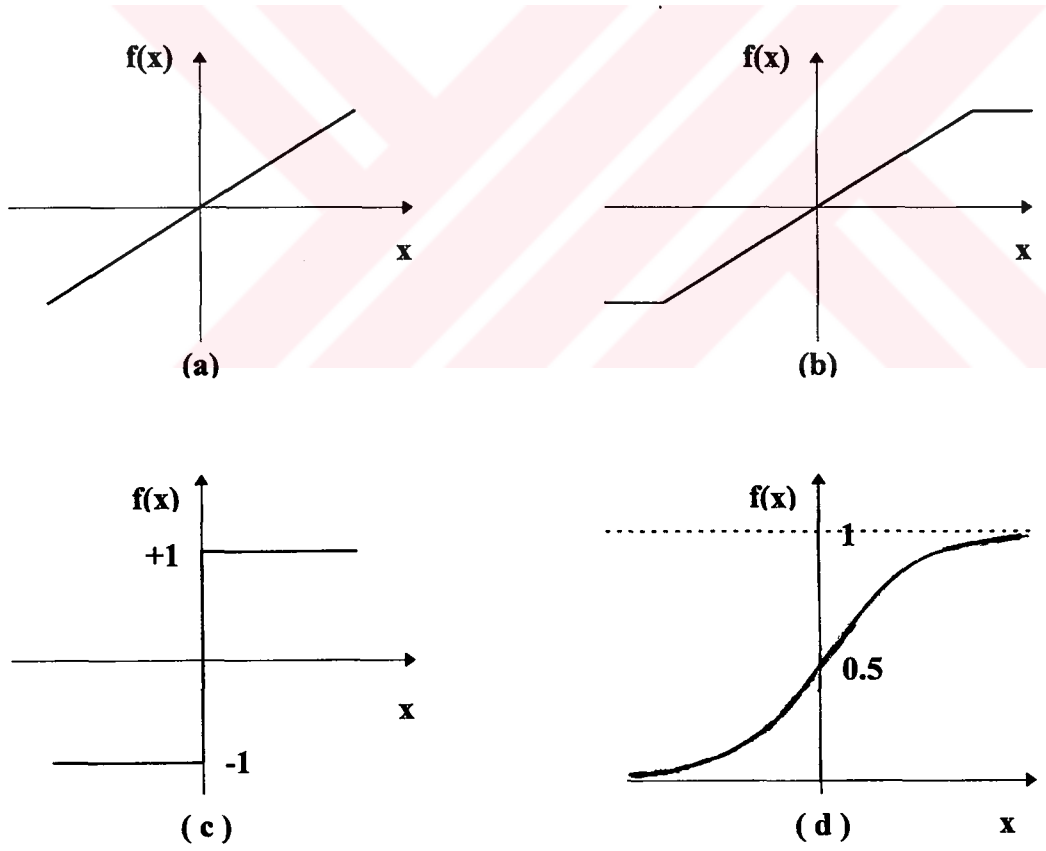
Hedef bölgesindeki her işlem elemanı kaynak bölgesindeki her elemana giderse "tam" bağlıdır (örn: çok katmanlı perceptron). Eğer her hedef bölgesi elemanı N kaynak bölgesi elemanına bağlı ise "düzgün dağılmış" olması olasıdır. Ayrıca her bir elemana, yine bir kaynak elemanı bağlı ise buna "bire-bir" bağlı denir.

4.3. Ağ Tipleri

1. İleri beslemeli ağ : Her bir katmandaki hücreler sadece bir önceki katmanın hücrelerince beslenir.
2. Kaskat bağlantılı ağ: Hücreler sadece önceki katmanlardaki hücrelerce beslenir.
3. Geri beslemeli ağ : En az bir hücre sonraki katmanlardaki hücrelerce beslenir.

4.4. Eşik Fonksiyonları

Eşik fonksiyonları, transfer veya işaret fonksiyonları olarak da adlandırılır. Bunlar, muhtemel sonsuz domen girişli işlem elemanlarını önceden belirlenmiş sınırda çıkış olarak düzenler. En yaygın eşik fonksiyonları Şekil 4.3.'te gösterilen, rampa, basamak ve sigmoid fonksiyonlarıdır.



Şekil 4.3. Sıkça kullanılan dört eşik fonksiyonu

Şekil 4.3. (d)'de kullanılan ve en önemli eşik fonksiyonu olan S biçimindeki sigmoid fonksiyonu, seviyeli, lineer olmayan çıkış veren, sınırlı, monoton artan fonksiyondur ve denklemi,

$$f(x) = \frac{1}{1+e^{-x}} \qquad (4.2)$$

biçimindedir. (Naylor, 1990)

4.5. Ağırlık Uzayı

Birçok yapay sinir ağları öğrenme işlemi, tanımlanan ağırlık değiştirilerek öğrenmede iyi bir model kullanıp, ağırlıkların bu modele göre değiştirilmesi esasına dayanır. Basit bir matematiksel model olarak her bir işlem elemanının "n" adet gerçek ağırlığı olduğu düşünülerek ve N adet işlem elemanı göz önüne alınarak;

$$w = (w_{11}, w_{12}, \ldots, w_{1n}, w_{21}, w_{22}, \ldots, w_{2n}, \ldots, w_{N1}, w_{N2}, \ldots, w_{Nn})^T \qquad (4.3)$$

$$w = (w_1^T, w_2^T, w_3^T, \ldots, w_N^T) \qquad (4.4)$$

$w_1, w_2, \ldots, w_N$: işlem elemanlarının ağırlık vektörleridir.

$$w_1 = \begin{bmatrix} w_{11} \\ w_{12} \\ . \\ . \\ w_{1n} \end{bmatrix} \ldots\ldots\ldots\ldots\ldots w_N = \begin{bmatrix} w_{N1} \\ w_{N2} \\ . \\ . \\ w_{Nn} \end{bmatrix} \qquad (4.5)$$

Yapay sinir ağları ağırlık vektörü N, n boyutlu orkid uzayında yayılır. Yapay sinir ağlarının enformasyon işleme performansı, ağın ağırlık vektörünün belirli bir değeri ile bulunur.

Hata değişimini inceleyen iki çeşit kural vardır. Bunlardan ilki, her bir giriş örüntüsünde ağırlıkları yeniden ayarlayarak çıktı hatasını en aza indirmeye çalışan hata düzeltme kuralları, ikincisi ise ağırlıkları yeniden ayarlanarak ortalama karesel hatanın en aza indirgenmesine çalışılan gradyen kurallarıdır.

Ağırlık vektörü ile çalışan yapay sinir ağlarında önemli noktalardan birisi, bir öğrenme kuralı geliştirip, enformasyon bölgesi kullanarak (eşik fonksiyonu ile) ağırlık vektörü "w" yı istenilen yapay sinir ağlarının performansı verecek noktaya yöneltmektir. Genellikle öğrenme kuralı için bir performans ya da maliyet fonksiyonu tanımlanır. Minimizasyon veya maksimizasyon ile "w" vektörü bulunur. Bir performasyon çeşidi olarak bilinen, karesel ortalama hata şu şekilde tanımlanır.

$$F(w) = \oint_A |f(x) - G(x,w)|^2 \rho(x) dv(x) \qquad (4.6)$$

Burada amaç F'i küçültmeye çalışmaktır.

y = G(w,x): sistemin giriş çıkış fonksiyonu.

y: çıkış işareti vektörü

x: giriş işareti vektörü

w: ağırlık vektörü

ρ (x): olasılık yoğunluk fonksiyonu

4.6.Yapay Sinir Ağlarında Eğitme Algoritmaları

Eğitme algoritması, eldeki problemin özelliğine göre öğrenme kuralının yapay sinir ağlarına nasıl uyarlanacağını belirtir. Öğreticili eğitme, skor ile eğitme ve kendini düzenleme ile eğitme şeklinde adlandırılan ve en yaygın olarak kullanılan üç çeşit eğitme algoritması bulunmaktadır. Öğreticili eğitmede, elde doğru örnekler vardır. Yani, $(x_1, x_2, ..., x_n)$ şeklindeki giriş vektörünün, $(y_1, y_2, ..., y_n)$ şeklindeki çıkış vektörü, tam ve doğru olarak bilinmektedir. Her bir (x_1, y_1), (x_2,y_2)....(x_N,y_N) çifti için ağ doğru sonuçları verecek şekilde seçilen bir öğrenme kuralıyla beraber eğitilir.

Skor ile eğitmede giriş işaretlerine karşılık gelen çıkış işaretleri tam olarak bilinmemektedir. Çıkış işareti yerine skor verilir ve ağın değerlendirilmesi yapılır. Özellikle kontrol uygulamaları için idealdir.

Kendini düzenleyen ağ, giriş işaretine göre kendini düzenleyerek organize eder. Olasılık yoğunluk fonksiyonlarına, sınıflandırma ve şekil tanıma problemlerine uygulanabilir.

Ne tür eğitme yöntemi kullanılırsa kullanılsın, herhangi bir ağ için gerekli karakteristik özellik, ağırlıkların verilen eğitme örneğine nasıl ayarlanacağının belirtilerek öğrenme kuralının oluşturulmasıdır. Öğrenme kuralının oluşturulması için bir örneğin ağa defalarca tanıtılması gerekebilir. Öğrenme kuralı ile ilişkili parametreler ağın zaman içinde gelişme kaydetmesiyle değişebilir.

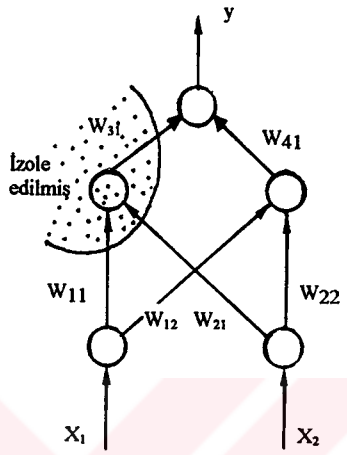
4.7. Yapay Sinir Ağlarında Bellek

Bilgiyi saklama şekli yapay sinir ağlarının önemli bir özelliğidir. Yapay sinir ağlarında bellek, birçok yerel bellekler oluşturularak dağıtılır. Bağlantı ağırlıkları yapay sinir ağları bellek biçimleridir. Ağırlıkların değerleri, ağın o anki bilgi durumunu temsil eder.

Bazı yapay sinir ağları bellekleri ilişkilidir. Öyle ki eğitilen ağa bir kısmı uygulanırsa, ağ bu giriş için belleğindeki en yakın çıkışı seçer ve tam girişe bağlı çıkış ortaya çıkar. Eğer yapay sinir ağları oto-ilişkili ise, kısmi giriş vektörlerinin ağa verilmesi bu girişlerin tamamlanması ile sonuçlanır. Yapay sinir ağları belleğinin yapısı, eksik, gürültülü ve tam seçilemeyen bu giriş uygulandığı zaman bile mantıklı çıkış üretmeye uygundur. Bu kurala "genelleme" adı verilir. Bir genellemenin kalitesi ve anlamı, uygulama çeşidine, ağın tipine ve karmaşıklığına dayanır. Lineer olmayan çok katmanlı ağlar (özellikle geriye yayınım ağları) gizli katmandaki özelliklerden öğrenirler ve bunları çıkışlar üretmek için birleştirirler. Gizli katmandaki bilgi, yeni giriş örüntülerine akılcı çözümler oluşturmak için kullanılabilir.

4.8. Yapay Sinir Ağlarında Hata Toleransı

Yapay sinir ağlarının klasik hesaplama sistemlerinden farkı, hata toleranslı olmasıdır. Yapay sinir ağları paralel dağılmış parametreli bir sistem olduğundan her bir işlem elemanı izole edilmiş bir oda gibi düşünülebilir. Şekil 4.4'de çok katmanlı perceptron (MLP) için bu durum gösterilmiştir.



Şekil 4.4. MLP'nin izole edilmiş hali

Daha çok işlem elemanının zarar görmesi ile sistemin davranışı biraz daha değişir. Performans düşer ama sistem hiç bir zaman durma noktasına gelmez. Yapay sinir ağları sistemlerinin hata toleranslı olmasının nedeni bilginin tek bir yerde saklanmayıp, sisteme dağıtılmasıdır. Bu özellik sistemin durmasının önemli bir zarara neden olacağı uygulamalarda önem kazanır.

4.9. Öğrenme Kuralları

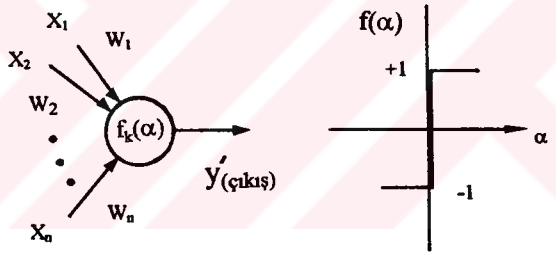
Klasik sistemlerde bilgi kurallar şeklinde açıklanır. Yapay sinir ağları ise gösterilen örnekten öğrenerek kendi kurallarını oluşturur. Öğrenme; giriş örneklerine veya (tercihen) bu girişlerin çıkışlarına bağlı olarak ağın bağlantı ağırlıklarını değiştiren veya ayarlayan öğrenme kuralı ile gerçekleştirilir. Öğreticisiz öğrenmede her giriş işareti için istenen çıkış sisteme tanıtılır ve yapay sinir ağları giriş/çıkış ilişkisini gerçekleştirene kadar kademe kademe

kendini ayarlar. Günümüzde kullanılan birçok öğrenme kuralı vardır. En çok kullanılan öğrenme kuralları şunlardır:

-Kompetif öğrenme
-Performans öğrenme
-Spotitemporal öğrenme
-Raslantısal öğrenme
-Filtreleme
-Genelleştirilmiş Delta kuralı öğrenme

4.10. Perceptron

Perceptron ağı ilk olarak McCulloch ve Pitts tarafından saptandı. Bu yapay sinir ağları tipi Şekil 4.5'da gösterilmiştir.



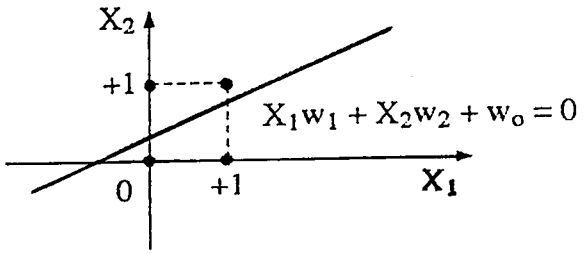
$$y' = f_k\left(\sum_{j=1}^{n} w_j x_j - \theta\right)$$

Şekil 4.5. Tek katmanlı perceptron yapısı

$\sum_{j=1}^{n} w_j x_j - \theta = 0$ n boyutlu uzayda n-1 boyutlu bir düzlem belirler.

Bu ilk perceptron modeline göre, ağ, giriş bilgisinin mevcut iki sınıftan hangisine eşit olabileceğini bulacak şekilde eğitilen basit bir ağdır. Daha sonra Rosenblatt bu ağ tipini biraz daha geliştirmiş fakat Minsky ve Papert bu tek katmanlı perceptronun XOR (ayrıcalıklı

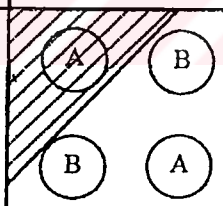
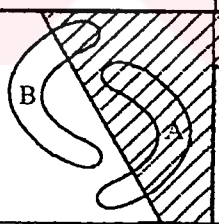
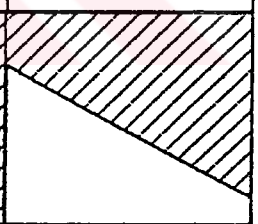
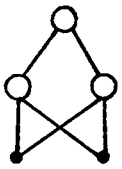
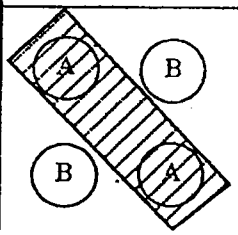
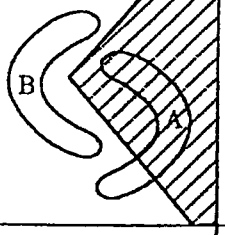
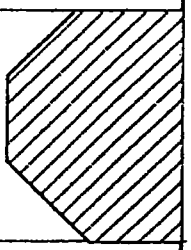
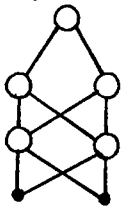
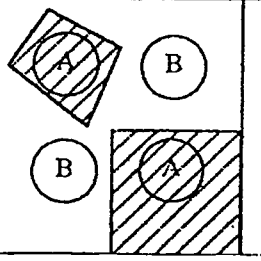
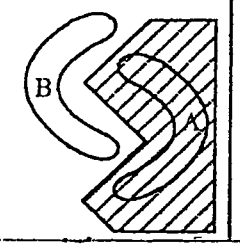
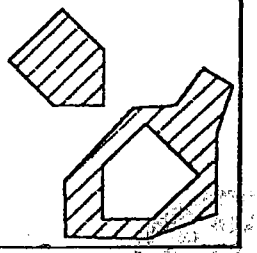
veya) işlemini gerçekleştiremediğini ispatlamışlardır. 0'ların bir tarafta, 1'lerin bir tarafta ayrılacağı şekilde bir bölge oluşmadığı şekil 4.7 ve Çizelge 4.1'de görülmektedir. XOR gibi üç veya daha fazla sınıfa ihtiyaç duyulan problemleri çözmek için yapılması gereken işlem, yapay sinir ağlarına yeni katmanlar eklemektir. Çizelge 4.1'de gösterildiği gibi içbükey ayrılabilir fonksiyonlar gerçekleştirilebilir.



X_1	X_2	f(x)
0	0	0
1	0	1
0	1	1
1	1	0

Şekil 4.6. Lineer yayılabilirliğin gösterimi

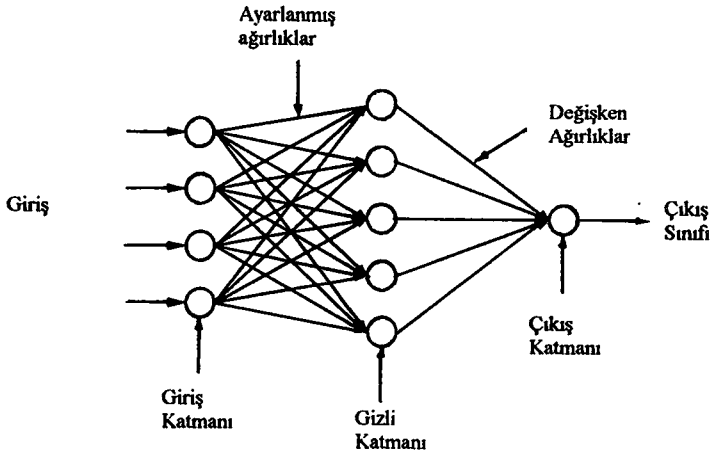
Çizelge 4.1. Çok katmanlı perceptronda gizli katmanın rolü

YAPI	Karar Bölgeleri tipi	XOR Problemi	Bölgelere Dayalı sınıflar	En iyi ayırdığı bölge şekilleri
Tek katman 	**Doğrusal ayrıştırılabilir.Ayrıcalıklı veya işlevini gerçekleyemez**			
İki katman 	**Konveks(iç bükey) açık veya kapalı bölgeler**			
Üç katman 	**İç bükey olmayan hatta bağlantılı olmayan bölgeler**			

Bilgi lineer yayılmıyorsa, gereken katman sayısı üçtür, çünkü ikinci katmanın çıkışı konveks bölgededir ve bunun sonucu olarak üçüncü katmandan gelecek olan çıkış bilgisinin şekli, herhangi bir bölgenin şeklinde olabilir. Çizelge 4.1'de görüldüğü gibi içbükey olmayan hatta basit bağlantılı olmayan bölgeleri kümelemek ancak üç katmanlı ağ ile mümkündür. Üç katmanlı perceptronun ikinci katmanında düğüm sayısı, en kötü durumda giriş bilgilerinin dağılımını yapan bölgenin bağlanmamış sayısına eşit olmalıdır. Birinci katmandakilerin sayısı her iki katmandaki değişim ile üç ya da dört köşeli dışbükey bir alan oluşturabilecek seviyede olmalıdır. Dolayısıyla en az birinci katmandakinden üç kat fazla miktarda olmalıdır. Bununla beraber Gutierrez ve arkadaşlarının perceptron ağlarının ihtiyacı olan düğüm sayıları ile ilgili çalışmalarının sonucunda çok fazla düğümün de, çok az sayıda olduğu gibi zararlı etkisi olduğu ispatlanmıştır. (Gutierrez vd., 1989)

4.11. Çok Katmanlı Perceptron

Çok katmanlı perceptron, giriş ve çıkış katmanları arasında birden fazla katmanın kullanıldığı yapay sinir ağları sistemleridir. Gizli katman olarak isimlendirilen bu katmanlarda düğümleri aracısız giriş olmayan ve aracısız çıkış veremeyen üniteler vardır. Şekil 4.7'de çok katmanlı perceptronun genel yapısı verilmiştir. Çok katmanlı perceptron genel olarak ileri beslemeli sistem gibi davranır, hatalar yapar, bu hatalardan bir şeyler öğrenip istenileni bulana kadar işleme devam eder. Bu yönteme hatanın geriye yayılması algoritması denir. Çok katmanlı perceptron ve algoritma Bölüm 5.3.'te detaylı olarak incelenecektir.



Şekil 4.7. Çok katmanlı perceptron yapısı

4.12. Hatanın Geriye Yayılması Algoritması ve Genelleştirilmiş Delta Kuralı

Hatanın geriye yayılması algoritması, karesi alınmış hata fonksiyonunu minimize eden kodlu bir algoritma olup, genelleştirilmiş delta kuralını eğitme için kullanılır. Algoritma, ana hatlarıyla şöyledir. Her bir j biriminin çıkışı olan o_j şu şekilde tanımlanır.

$$o_j = f(net_j) = f(x) \qquad \text{ve} \qquad net_j = \sum_{i}^{j} w_{ji} o_i + \theta_j \tag{4.7}$$

Burada;

o_i : i. biriminin çıkışı

w_{ji} : i biriminden j birimine bağlantının ağırlığı

θ_j : j biriminin kutbu

Σ_i : çıkışı j birimine akan her i biriminin toplamı

f(x) : monoton artan ve türevi alınabilen bir fonksiyondur.

f(x) için genelde daha önceki bölümlerde açıklanan sigmoid fonksiyon kullanılır. ($f(x) = 1/1+e^{-x}$)

m boyutlu giriş örüntüleri oluşturulduğunda $\{i_p = (i_{p1}, i_{p2},...., i_{pn}); p \in P\}$'dir. Benzer şekilde istenilen n boyutlu çıkış örüntüleri $\{T_p = (t_{p1}, t_{p2},..., t_{pn}); p \in P\}$ şeklindedir. Burada P: yapay sinir ağlarında uygulanan işaret, şekil vb. örüntülerini verir.

Bir örüntü için karesel hata fonksiyonu E_p şu şekilde tanımlanır:

$$E_p = \frac{1}{2} \sum_{j \in \text{çıkış katmanı}} (t_{pj} - o^{pj})^2 \tag{4.8}$$

Amaç uygun w_{ji} ve θ_j seçimiyle, $E = \sum_p E_p$ toplam hatayı yeterince küçük yapmaktır. Bu amacı gerçekleştirmek için, bir $p \in P$ örüntüsü ard arda ve rasgele biçimde seçilir. Daha sonra w_{ji} ve θ_j şöyle değiştirilir;

i_{pi} : Giriş işaretinin i bileşeni;

t_{pj} : Çıkış vektörünün j bileşeni;

o_{pj} : Yapay sinir ağlarında uygulanan P örüntü setinin ürettiği çıkış ise;

$$\delta_{pj} = (t_{pj} - o_{pj}) \tag{4.9}$$

$$\Delta_p w_{ji} = -\varepsilon\left(\frac{\partial E_p}{\partial w_{ji}}\right) \tag{4.10}$$

$$\Delta_p \theta_j = -\varepsilon\left(\frac{\partial E_p}{\partial \theta_j}\right) \tag{4.11}$$

Burada ε: öğrenme oranı adı verilen küçük bir pozitif sabit sayılır. Şayet gizli katman yok ise: (4.10) ve (4.11)'in sağ tarafi hesaplanır, o zaman;

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \cdot \frac{\partial o_{pj}}{\partial w_{ji}} \tag{4.12}$$

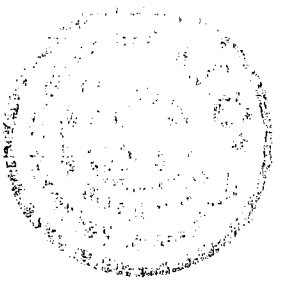
$$\frac{\partial E_p}{\partial o_{pji}} = -(t_{pj} - o_{pj}) = -\delta_{pj} \tag{4.13}$$

$$o_p = \sum_i w_{ji} . i_{pi} \tag{4.14}$$

ise;

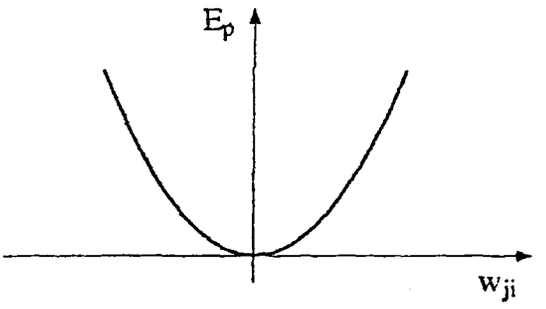
$$\frac{\partial o_{pi}}{\partial w_{ji}} = i_{pi} \tag{4.15}$$

elde edilir. (4.13) ve (4.15) ifadeleri (4.12)'de yerine konulursa,



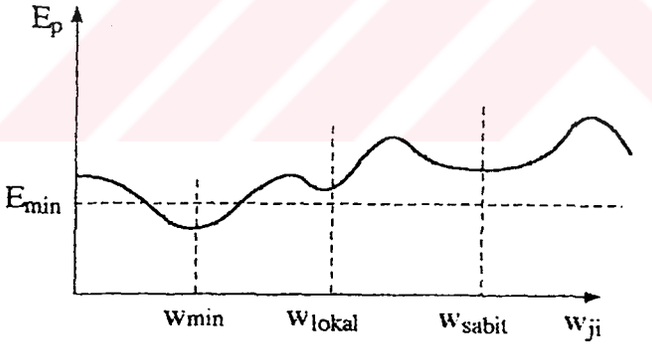
$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} i_{pi} \qquad (4.16)$$

olur. Hata fonksiyonu Şekil 4.8'da gösterildiği gibidir.



Şekil 4.8. Gizli katmanı olmayan ağın hata fonksiyonu

Gizli katman olduğu zaman, hata düzeyi Şekil 4.8'de olduğu gibi sadece bir minimumdan oluşmamaktadır. Şekil 4.9'daki gibi çeşitli minimumlardan öğrenmede en küçük minimuma ulaşılmak istenir.



Şekil 4.9. Gizli katmana ait ağın hata fonksiyonu

Bu durumda j. düğümün lineer olmayan çıkışı;

$$o_{pj} = f_j(net_{pj}) \Rightarrow net_{pj} = \sum_i w_{ji} o_{pj} \qquad (4.17)$$

şeklindedir. Bu durumda;

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial netp_j} \frac{\partial netp_j}{\partial w_{ji}} \Leftrightarrow \frac{\partial netp_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \sum_k w_{jk} o_{pk} = o_{pi} \quad (4.18)$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial netp_j} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial netp_j} = -\frac{\partial E_{pj}}{\partial o_{pj}} f_j^{\acute{y}}(netp_j) \quad (4.19)$$

olur. İki durum vardır.

1. o_{pj} , yapay sinir ağlarının çıkışı ise;

(4.13) ifadesi (4.19)'da yerine konularak,

$$\delta_{pj} = (t_{pj} - o_{pj}) \; f_j^{\acute{y}}(netp_j) \quad (4.20)$$

bulunur.

2. Eğer gizli katmanların çıkış işaretinden bahsediliyorsa yani eleman çıkış elemanı değilse;

$$\sum_k \frac{\partial E_p}{\partial netp_k} \frac{\partial netp_k}{\partial p_j} \quad (4.21)$$

şeklinde ise,

$$\sum_k \frac{\partial E_p}{\partial netp_k} \frac{\partial}{\partial o_{pj}} \sum_i w_{ki} o_{pj} = -\sum_k \delta_{pk} w_{kj} \quad (4.22)$$

olur. Bulunan son ifade (4.19)'da yerine konulursa,

$$\delta_{pj} = f^{\acute{y}}(net_{pj}) \sum_k \delta_{pk} w_{kj} \quad (4.23)$$

elde edilir. (4.22) denklemindeki (-) işareti, ağırlıkların ters yönde değiştiğini belirtir.

Bütün yapılan işlemler kısaca özetlenirse;

1. Genelleştirilmiş Δ (delta) kuralı;

$\Delta_p w_{ji} = \varepsilon \, \delta_{pj} \, i_{pi}$

2. Çıkış katmanı elemanları için;

$\delta_{pj} = (t_{pj} - o_{pj}) \, f'_j \, (net_{pj})$

3. Gizli katman elemanları için;

$\delta_{pj} = f'_j(netp_j) \sum_k \delta_{pk} w_{kj}$

olur. İşlem elemanında, transfer (eşik) fonksiyonu olarak "sigmoid" fonksiyonu; kullanılıp, türev alınarak gerekli kısaltmalar yapılırsa,

$$\frac{\partial o_{pj}}{\partial net_{pj}} = o_{pj}(1 - o_{pj}) \qquad (4.24)$$

bulunur. Bu (4.20)'de yerine konulursa, çıkış elemanı için;

$$\delta_{pj} = (t_{pj} - o_{pj}) \, o_{pj} \, (1 - o_{pj}) \qquad (4.25)$$

(4.24) eşitliği (4.23) de yerine konulursa, gizli katman elemanı için;

$$\delta_{pj} = o_{pj} \, (1 - o_{pj}) \sum_k \delta_{pk} w_{kj} \qquad (4.26)$$

bulunur. Yukarıda toplam içersinde gösterilen k'nın, j çıkış birimine akan k olduğuna dikkat edilmelidir. Hesaplamayı hızlandırmak için momentum terimleri (α) eklenirse, en genel halde çıkış ve gizli katman ifadeleri şu şekilde olur:

$$\Delta_p w_{ji}(t+1) = \varepsilon\, \delta_{pj}\, o_{pi} + \alpha \Delta_p w_{ji}(t) \quad (4.27)$$

$$\Delta_p \theta_j(t+1) = \varepsilon\, \delta_{pj} + \alpha \Delta_p \theta_j(t) \quad (4.28)$$

Burada t, öğrenme döngülerinin sayısını gösterir. (α) küçük pozitif bir sayıdır.

4.13. Öğrenme ve Momentum Katsayıları

Yapay sinir ağlarında çok yüksek tutmak davranışın bozulmasına neden olduğundan, öğrenme katsayısı (ε) küçük tutulmalıdır. Öğrenme katsayısı, 0.01 ile 10 aralığında seçilen sabit bir sayıdır. Ancak çok küçük bir öğrenme oranının da, öğrenme işleminin yavaşlamasına neden olacağına dikkat edilmelidir.

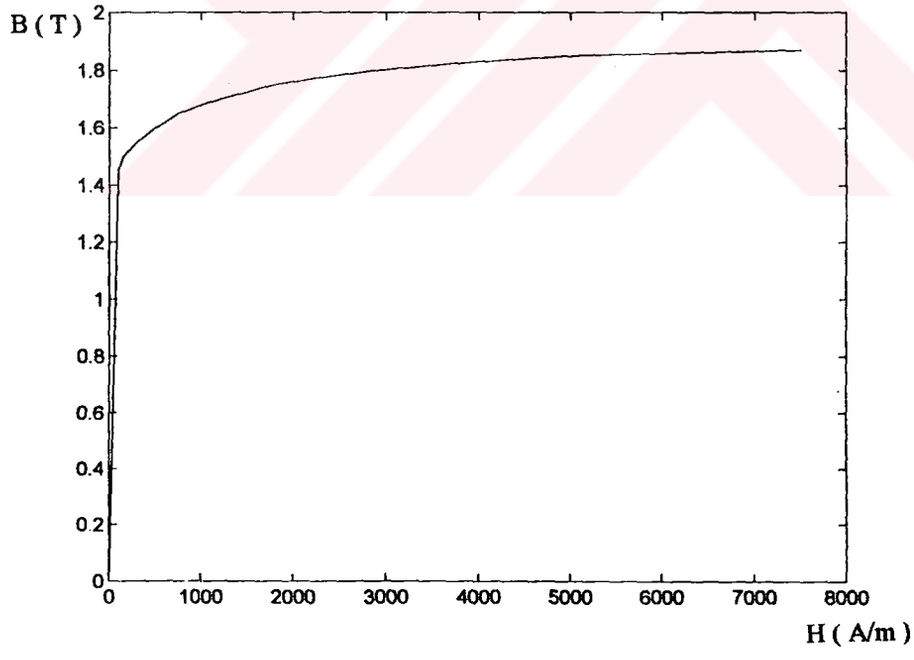
Momentum (α) fikri bu noktadan hareketle ortaya atılmıştır. Momentum, mevcut delta ağırlığı üzerinden önceki delta ağırlığının belli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme elde edilir. Momentum katsayısı genellikle $0 < \alpha < 1$ aralığında değişen sabit bir sayıdır.

5. YAPAY SİNİR AĞLARI YÖNTEMİ ile υ-B^2 EĞRİLERİNİN MODELLENMESİ

Bu bölümde, analizi yapılacak olan doğru akım makinasının mıknatıslanma karakteristiğinin, kübik spline ve yapay sinir ağları ile modellenmesine ait sonuçlar verilecektir. Yapay sinir ağları ile modelleme sonuçları değişik iterasyon değerlerinde ve öğrenme katsayılarında alınmıştır. İterasyon sayısının artması ile öğrenmenin iyileştirilmesi arasında lineer bir bağıntının olmamasından dolayı, uygun iterasyon sayısı araştırılmış ve 40000 iterasyonun uygun olduğu görülmüştür. Kübik spline algoritması olarak Bölüm 3.3.2'de verilen algoritma kullanılmıştır.

5.1. Miknatıslanma Eğrisi ve υ-B^2 Eğrisinin Elde Edilmesi

Bölüm 6'da analizi yapılacak olan 3,5 kW, 440 V, 9.5 A, 2900 d/d etiket değerlerine sahip, 2 ana, 2 yardımcı kutbu bulunan doğru akım şönt motorda kullanılan manyetik malzemenin mıknatıslanma karakteristiği Şekil 5.1'de gösterilmiştir. Kullanılan malzeme boyuna haddelenmiş olup, 0.5mm kalınlığındadır.



Şekil 5.1. Manyetik akı yoğunluğunun manyetik alan şiddetine bağlı olarak değişimi

Bölüm 3'te anlatıldığı gibi relüktiviteyi H ve B cinsinden bulmak için (3.32) eşitliği (3.31) eşitliğinde yerine yazıldığında;

$$B.H = \upsilon.B^2$$

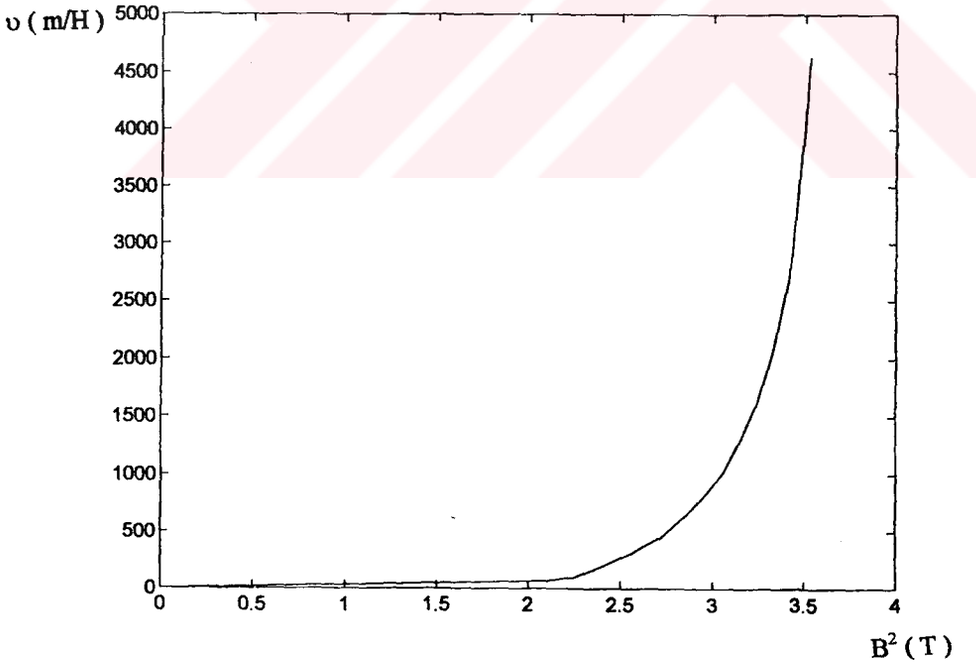
elde edilir. Gerekli sadeleştirmeler yapıldığında, manyetik relüktivite;

$$\upsilon = H / B \qquad (5.1)$$

şeklinde bulunur.

Şekil 5.1' deki mıknatıslama eğrisinden okunan B ve H değerleri (5.1) eşitliğinde yerine yazıldığında elde edilen relüktivite değerleri Çizelge 5.1'de gösterilmiştir.

Relüktivitenin, manyetik akı yoğunluğunun karesine bağlı olarak değişimi Şekil 5.2'de verilmiştir.



Şekil 5.2. Şekil 5.1'deki mıknatıslanma eğrisi kullanılarak elde edilen υ-B^2 karakteristiği

Çizelge 5.1. B ve H'a bağlı olarak hesaplanan relüktivite değerleri.

H (A/m)	B (T)	υ (A/mT)
0	0	0
100	1.45	69
150	1.5	100
300	1.55	193.6
500	1.6	312.5
600	1.62	370.4
750	1.65	454.6
1000	1.68	595.2
1200	1.7	705.9
1500	1.725	869.6
1800	1.75	1028.6
2300	1.775	1295.8
2900	1.8	1611.1
3800	1.825	2082.2
5000	1.85	2702.7
7500	1.87	4010.7
8700	1.88	4627.7
11500	1.9	6052.6
13200	1.91	6911

5.2. υ-B^2 Eğrisinin Kübik Spline İle Modellenmesi

Bölüm 3.3.2.5'te verilen algoritma kullanılarak yazılan bilgisayar programı ile elde edilin $S_{k,0}$, $S_{k,1}$, $S_{k,2}$ ve $S_{k,3}$ katsayıları Çizelge 5.2'de verilmiştir.

Çizelge 5.2. υ-B^2 Eğrisinin kübik spline ile modellenmesinde kullanılan algoritmaya göre hesaplanan katsayılar.

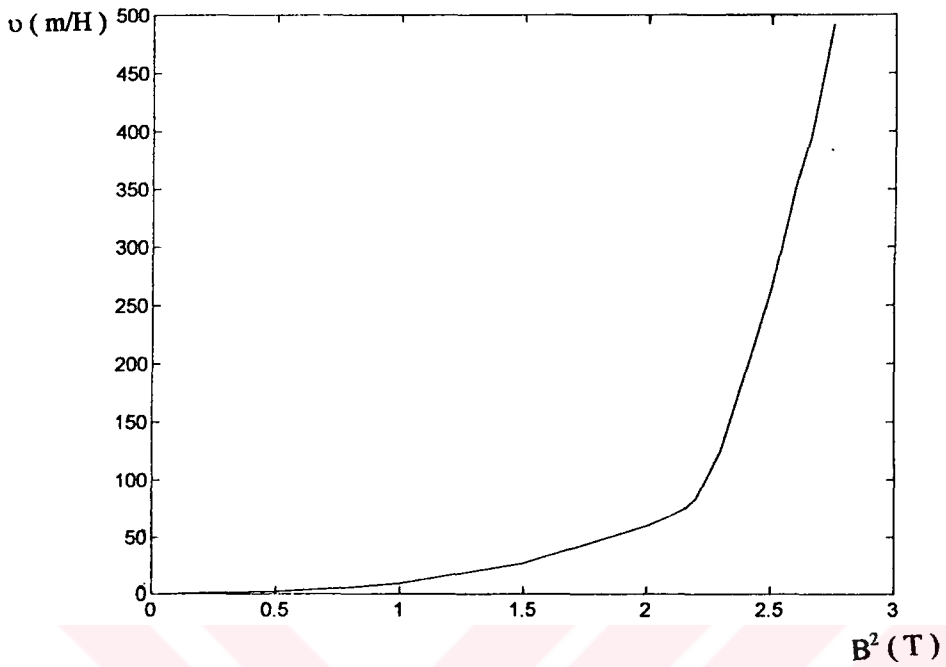
B^2_k	$S_{k,0}$	$S_{k,1}$	$S_{k,2}$	$S_{k,3}$
0.00	0.00	2.92	0.00	6.78
2.10	68.97	92.68	42.74	4789.84
2.25	100.00	428.82	2198.17	-5994.67
2.40	193.55	683.63	-499.43	5457.57
2.56	312.50	942.96	2120.20	-29352.37
2.62	370.37	880.38	3163.22	27774.72
2.72	454.55	1080.97	5169.19	-19099.14
2.82	595.24	1541.84	-560.55	15012.95
2.89	705.88	1697.28	2781.17	-15002.11
2.98	869.57	1811.47	-1512.41	46410.01
3.06	1028.57	2460.56	9626.00	-44198.32
3.15	1295.77	3119.22	-2307.55	73116.26
3.24	1611.11	4480.58	17433.84	-100667.25
3.33	2082.19	5172.46	9746.32	320897.04
3.42	2702.70	11215.92	76895.88	-158998.35
3.50	4010.70	20466.49	38736.28	-1181385.17
3.53	4627.66	19600.93	-67588.19	565354.33
3.61	6052.63	19641.59	68096.65	-567472.08
3.65	6910.93	0.00	0.00	0.00

Kübik spline ile modelleme sonucunda ara değerler için elde edilen relüktivite değerleri ise Çizelge 5.3'de verilmiştir.

Çizelge 5.3. Kübik spline ile modelleme sonucunda elde edilen υ-B^2 değerleri.

B^2	υ
0.50	2.31
0.75	5.05
1.00	9.71
1.50	27.28
2.00	60.12
2.10	68.97
2.15	74.31
2.20	83.46
2.30	126.19
2.45	227.17
2.50	262.38
2.60	351.73
2.66	394.68
2.75	491.12
2.85	641.42
2.94	795.71
3.00	905.57
3.10	1139.57
3.20	1455.10
3.30	1920.96
3.47	3435.86
3.57	5339.74
3.62	6255.29
3.65	6910.93

Kübik spline ile modellemede eğrinin başlangıç değerlerinde aşırı sapma olduğu görülmektedir. Değişim, Şekil 5.3'te gösterilmiştir.



Şekil 5.3. Kübik spline ile modelleme sonucunda elde edilen υ-B^2 eğrisi

Yöntem eğrinin hızla değiştiği noktalarda büyük hata verdiğinden, eğri bu noktalardan birkaç parçaya bölündükten sonra modelleme işlemi yapılmalıdır. Bu durum modelleme maliyetini yükseltmektedir. Eğrilerin yapay sinir ağları ile modellenmesiyle bu dezavantaj ortadan kaldırılmıştır. Yapay sinir ağları yönteminde öğrenme işlemi için hatanın geriye yayılması algoritması kullanılacaktır.

5.3. Yapay Sinir Ağları ile Modelleme için Hatanın Geriye Yayılması Algoritması

Çok katmanlı perceptronlar giriş ve çıkış düğümleri arasında bir ya da daha fazla düğüm katmanı içeren ileri beslemeli ağlardır. (Widrow and Lehr, 1990).

Bu fazladan katmanlar giriş ve çıkış düğümlerine direkt olarak bağlı olmayan gizli üniteler ya da düğümler içerirler. Eğitme algoritmalarının gelişmesiyle, çok katmanlı perceptronlar kullanılmaya başlanmış ve tek katmanlı perceptronların birçok kısıtlamalarının üstesinden gelinmiştir.

Çok katmanlı perceptronların başarısı düğümler içinde kullanılan nonlineerlikten kaynaklanır. Eğer düğümler lineer elemanlar olmuş olsalardı, doğru seçilmiş ağırlıkları olan tek katmanlı bir ağ herhangi bir çok katmanlı ağ tarafından yapılan hesaplamaların tamamen aynısını oluştururdu. Bir tek katmanlı perceptron yarı-düzlem karar bölgeleri oluşturur, iki katmanlı perceptron ise girişler tarafından uzayda herhangi bir olasılıkla sınırsız bir konveks bölge oluşturabilir. Burada konveks terimi bir bölgenin sınırında noktaları birleştiren herhangi bir hat, sadece o bölgedeki noktalardan geçer anlamındadır. Konveks bölgeler, çok katmanlı perceptronun ilk katmanındaki her düğüm tarafından oluşturulan yarı-düzlem bölgelerinin kesişmelerinden oluşur. İlk katmandaki her düğüm tek katmanlı perceptron gibi davranır ve ağırlıkları tarafından oluşturulan hiper-düzlemin bir tarafındaki noktalar için yüksek çıkışa sahiptir. Eğer N_1 birinci katman düğümlerinden bir çıkış düğümüne ağırlıkların tümü 1.0 ise ve $0 < \varepsilon < 1$ olmak üzere çıkış düğümündeki eşik $N_1 - \varepsilon$ ise, bu durumda çıkış düğümü ancak tüm ilk katman düğümleri çıkışları yüksek olduğunda yüksek olur. Bu, çıkış düğümünde bir mantıksal AND işlemi yapmaya karşılık gelir ve ilk katmanda oluşan tüm yarı-düzlem bölgelerin kesişimi olan bir son karar bölgesi verir. Bu tür yarı düzlemlerin kesişimleri yukarıda tanımlandığı gibi konveks bölgeler oluştururlar. Bu konveks bölgeler en çok ilk katmandaki düğüm sayısı kadar yüze sahip olabilirler.

İki katmanlı perceptronda düğüm sayısı, eldeki problem tarafından gereksinim duyulan karmaşıklıkta bir karar bölgesi oluşturmaya yetecek kadar fazla olmalıdır. Fakat eldeki eğitme verilerinden, gerekli birçok ağırlıkların güvenilir şekilde tahmin edilemeyeceği kadar da fazla olmamalıdır (Lippmann, 1987).

Üç katmanlı perceptron keyfi olarak kompleks karar bölgeleri oluşturabilir ve iç içe geçmiş (ağ şekline getirilmiş) sınıfları ayırabilir. Karma dağılımlar ve en yakın komşu sınıflandırıcılar kullanılarak oluşturulanlar kadar kompleks bölgeler oluşturulabilir ve konstrüksiyonla kanıtlanabilir. Kanıt, istenen karar bölgesinin hiperküplere (iki giriş olduğunda kareler) bölünmesine dayanır. Her hiperküp ilk katmanda 2N düğüm gerektirir (iki giriş olduğunda dört düğüm), hiperkübün herbir yüzü için bir ve ikinci katmanda ilk katman düğümlerinin çıkışlarından mantıksal AND alan bir düğüm. İkinci katmanın çıkışları sadece her hiperküpteki girişler için yüksek olacaktır. Hiperküpler, uygun karar

bölgelerine her bir ikinci katman düğümünü sadece düğümün hiperkübün içinde bulunduğu karar bölgesine karşılık gelen çıkış düğümüne bağlayarak ve her çıkış düğümünde bir mantıksal OR işlemi yapılarak atanırlar. Bir mantıksal OR işlemi eğer ikinci gizli katmandan çıkış katmanına bağlantı ağrılıkları bir ise ve çıkış düğümlerindeki eşikler 0.9 ise yapılabilecektir. Bu konstrüksiyon prosedürü, küçük hiperküpler yerine keyfi olarak şekillendirilmiş konveks bölgeler kullanmak için genelleştirilebilir.

Bu analiz, üç katmanlı ağ keyfi olarak seçilmiş karmaşık karar bölgeleri oluşturabileceği için, perceptron benzeri ileri-beslemeli ağlarda üç katmandan daha fazlasının gerekli olmadığını göstermektedir. Ayrıca, üç katmanlı perceptronlarda kullanılacak düğüm sayısını seçme problemine de bazı iç görüler kazandırmaktadır. Karar bölgeleri bağlantısız ya da ağ şekline getirilmiş olduğunda ve bir konveks alandan oluşturulamadığından, ikinci katmandaki düğümlerin sayısı birden büyük olmak zorundadır. En kötü durumda, gerekli ikinci katman düğümlerinin sayısı giriş dağılımlarındaki bağlantısız bölgelerin sayısına eşittir. İlk katmandaki düğüm sayısı tipik olarak her bir ikinci katman düğümü tarafından oluşturulan her konveks alan için üç ya da daha fazla kenar oluşturabilecek yeterlilikte olmalıdır. Bu açıdan, ikinci katmanda, ilkinde olduğundan üç kat fazla düğüm olmalıdır.

Sigmoidal nonlineerlikler kullanıldığında ve karar kuralı en büyük çıkışlı çıkış düğümüne karşılık gelen sınıfı seçmek olduğunda çoklu çıkış düğümlü çok katmanlı perceptronlar benzer davranış gösterirler. Bu ağların davranışı daha karmaşıktır, çünkü karar bölgeleri tipik olarak doğru parçaları yerine düzgün eğrilerle sınırlanmıştır ve bu da analizi daha zorlaştırır. Öte yandan, bu ağlar geriye yayılma algoritmasıyla eğitilebilirler.

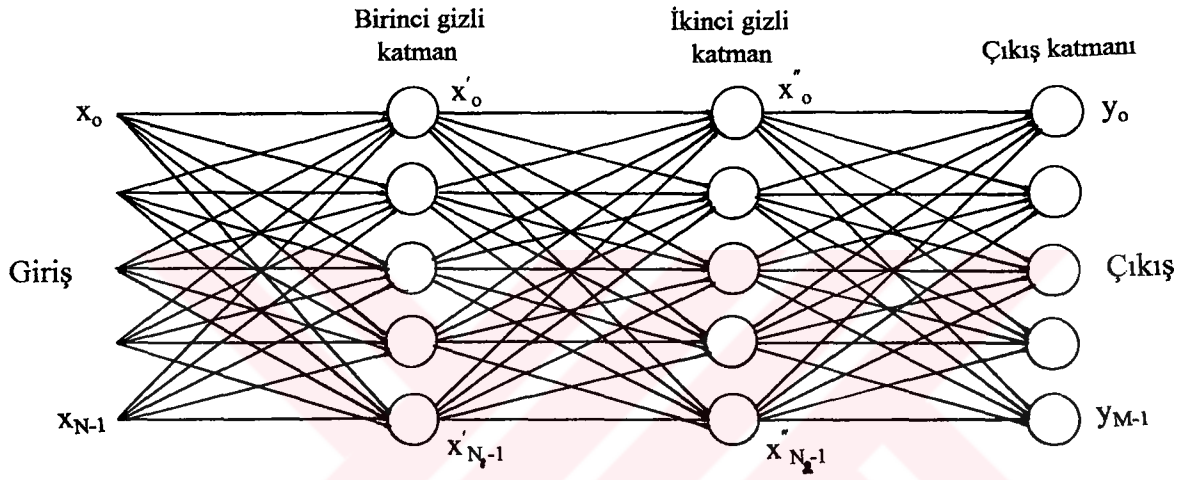
Geriye Yayılma Eğitme Algoritması

Geriye yayılma algoritması, bir çok katmanlı ileri beslemeli perceptronun gerçek çıkışıyla istenen çıkışı arasındaki karesi alınmış hata fonksiyonunu minimize etmek için kullanılan bir algoritmadır.

ADIM 1. Ağırlıkların ilk değerlerini küçük rasgele değerler olarak ata.

ADIM 2. Giriş ve istenen çıkışları gir. Bir $x_0, x_1, \ldots, x_{N-1}$ sürekli değerli giriş vektörünü sun ve istenilen $d_0, d_1, \ldots, d_{M-1}$ çıkışlarını belirt. Eğer ağ, bir sınıflandırıcı olarak kullanılıyorsa girişin geldiği sınıfa karşılık gelen çıkış dışındaki tüm istenen çıkışlar sıfır değerine ayarlanırlar. Bu istenen çıkış 1'dir. Giriş her denemede yeni olabilir ya da ağırlıklar dengelenene kadar bir eğitme setinden örneklemler tekrar tekrar sunulabilir.

ADIM 3. Gerçek çıkışları hesapla, $y_0, y_1, \ldots, y_{M-1}$ çıkışlarını hesaplamak için aşağıdaki şekilde verilen formüller kullanılır.



Şekil 5.4. N sürekli değerli girişli, M çıkışlı ve iki gizli ünite katmanlı bir üç katmanlı perceptron.

Birinci gizli katman $: x'_j = f\left(\sum_{i=0}^{N-1} w_{ij} x_i - \theta_j\right) \qquad 0 \le j \le N_1 - 1$

İkinci gizli katman $: x''_k = f\left(\sum_{j=0}^{N_1-1} w'_{jk} x'_j - \theta'_k\right) \qquad 0 \le k \le N_2 - 1$

Çıkış katmanı $: y_l = f\left(\sum_{k=0}^{N_2-L} w''_{kL} x''_k - \theta'_L\right) \qquad 0 \le l \le M - 1$

Formüllerde $x_j^{'}$ ve $x_k^{''}$, birinci ve ikinci gizli katmanlardaki düğümlerin çıkışlarıdır. $\theta_k^{'}$ ve $\theta_L^{''}$ bu düğümlerdeki iç kaymalardır, w_{ij} girişten ilk gizli katmana bağlantı kuvvetidir ve $w_{jk}^{'}$ ve $w_{kl}^{''}$ sırasıyla birinci ile ikinci ve ikinci ile çıkış katmanları arasındaki bağlantı kuvvetidir.

ADIM 4. Ağırlıkları ayarla. Çıkış düğümlerinden başlayarak geriye doğru çalışarak ilk gizli katmana kadar geri dönerek tekrarlayan bir algoritma kullan. Ağırlıkları;

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x_j^{'}$$

formülünü kullanarak ayarla. Bu eşitlikte, $w_{ij}(t)$ gizli i düğümünden ya da t zamanında j düğümüne bir girişten ağırlıktır. $x_j^{'}$ ya j düğümünün çıkışı ya da bir giriştir, η bir kazanç terimidir ve δ_j, j düğümü için hata terimidir. Eğer j bir çıkışı düğümüyse, o halde;

$$\delta_j = y_j\,(1-y_j)\,(d_j-y_j)$$

olur. Burada d_j, j düğümünün istenen çıkışı, y_j ise gerçek çıkıştır.

Eğer j düğümü bir içsel gizli düğümse, bu durumda,

$$\delta_j = x_j^{'}(1-x_j^{'})\sum_k \delta_k w_{jk}$$

olur. Burada k, j düğümünün üzerindeki katmanlardaki tüm düğümlerdir. İç düğüm eşikleri, yardımcı sabit değerli girişlerin bağlantıları üzerinde bağlantı ağırlıkları oldukları farz edilerek benzer şekilde uyarlanırlar. Yaklaşım bazen bir momentum terimi eklendiğinde ve ağırlık değişimleri,

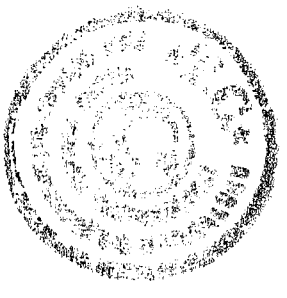
$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x_j^{'} + \alpha(w_{ij}(t) - w_{ij}(t-1))$ $0 < \alpha < 1$ olmak üzere,

ile düzeltildiğinde daha hızlı olacaktır.

ADIM 5. Adım 2'ye dönerek tekrarla.

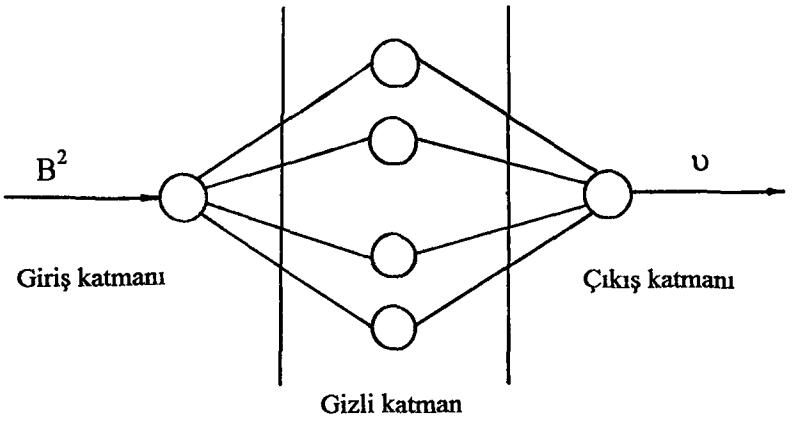
Yukarıda tanımlanan geriye yayılma algoritması, istenen ve asıl ağ çıkışları arasındaki ortalama kare farkına eşit olan bir maliyet fonksiyonunu minimuma indirgemek için bir ortalama arama tekniği kullanır. Düğüm o anki girişin sınıfına karşılık gelmedikçe - ki bu durumda yüksek (1.0 ya da > 0.9) olur- tüm düğümlerin istenen çıkışı düşük (0 ya da < 0.1) olur. Ağ öncelikle küçük rasgele ağırlıklar ve iç eşikler seçilerek ve sonra da tüm eğitme verileri tekrar tekrar sunularak eğitilir. Ağırlıklar, her denemeden sonra yaklaşım sağlanana kadar doğru sınıfı belirten yan bilgiler kullanılarak ve maliyet fonksiyonunu kabul edilebilir bir değere düşürülene kadar ayarlanır. Algoritmada tanımlanan iteratif metot, geriye doğru çıkış katmanındaki düğümlerden daha alttaki katmanlardaki düğümlere doğru ağırlıkları uyarlamak için gereken hata terimlerini üretir.

Geriye yayılmada, performansı arttırmak ve lokal minimumların oluşumunu azaltmak için önerilenler, fazladan gizli ünitelere izin vermek, ağrılıkları ayarlamak için kullanılan kazanç terimini düşürmek ve farklı rasgele ağırlık setleriyle başlayan bir çok eğitme tekrarları yapmak gibi yöntemlerdir. Sınıflandırma problemleriyle kullanıldığında, düğüm sayısı yukarıda tanımlanan noktalar kullanılarak ayarlanabilir. Bu durumda lokal minimumlar, problemi iki ya da daha fazla ayrık sınıf bölgelerini bir bölgede toplamaya karşılık gelir. Bu, farklı rasgele ağırlıklarla birçok başlangıçlar ve ağırlıkları adapte etmek için düşük bir kazanç kullanılarak en aza indirgenebilir. Geriye yayılma algoritmasında fark edilen bir güçlük, birçok durumda yaklaşım için gereken eğitme verileri sunumlarının sayısının yüksek oluşudur (bütün eğitme verilerinde 100 geçişten fazla). Yaklaşımı hızlandırmak için bazı daha karmaşık algoritmalar önerilmişse de sınıf bölgelerinin bağlantıları kesildiğinde az sayıda denemeyle çok katmanlı perceptronlar tarafından oluşturulan karmaşık karar bölgelerinin üretilmesi olası görülmemektedir.



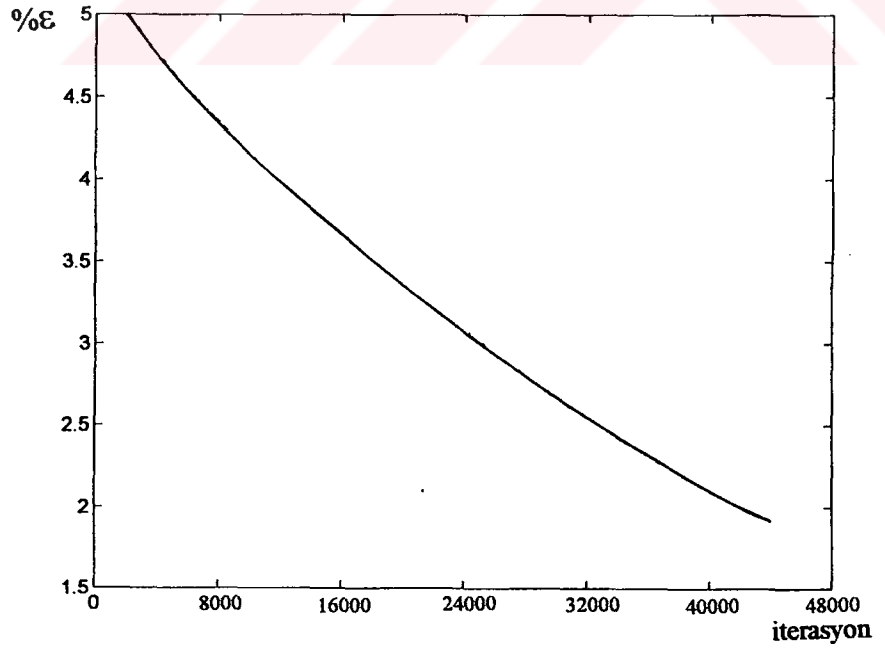
5.4. Yapay Sinir Ağları ile Modellemede Kullanılan Ağ Mimarisi

υ -B^2 eğrisinin modellenmesinde yapay sinir ağları için tek giriş ve tek çıkış bulunacağından giriş ve çıkış katmanlarında düğüm sayısı aynıdır ve bire eşittir. Gizli katmanda ise dört düğüm seçilmiştir. Bu mimariye ait ağ yapısı Şekil 5.5'de görülmektedir.



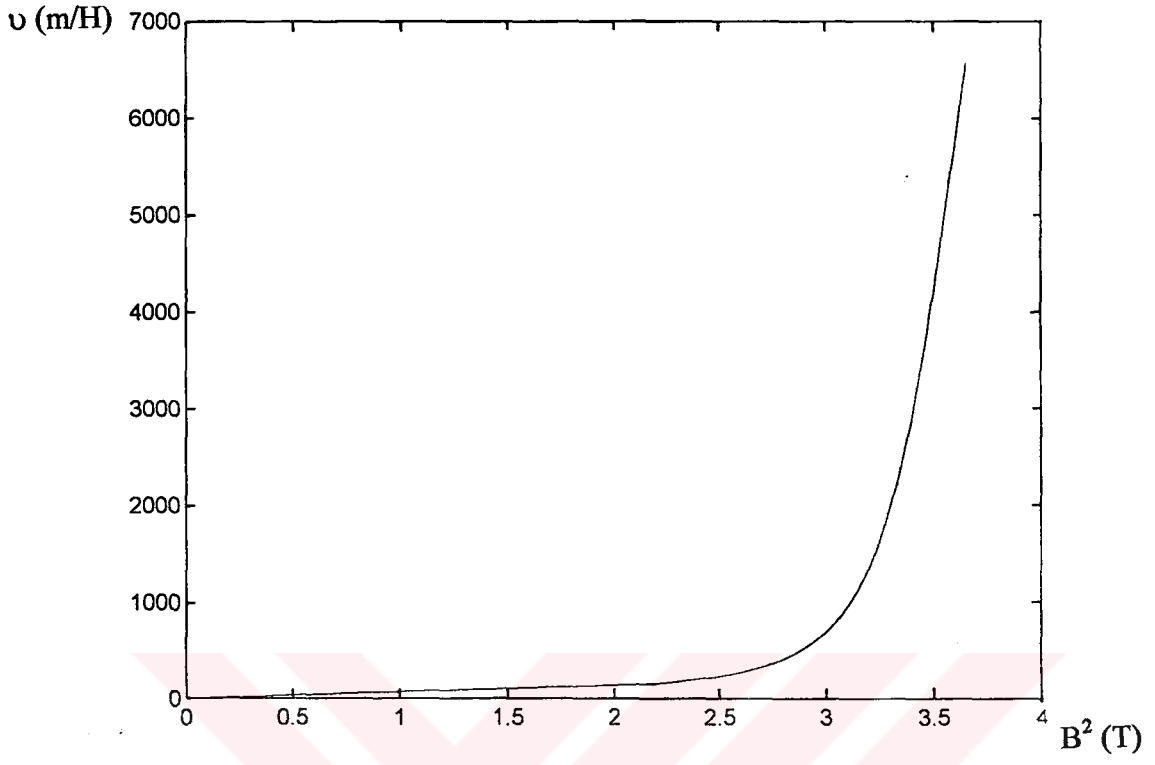
Şekil 5.5. Modellemede kullanılan ağ mimarisi

Yapay sinir ağları ile yapılan modelleme sonucunda iterasyona bağlı olarak hata değişimi aşağıdaki şekilde verilmiştir. 40000 iterasyon civarında mutlak hatanın %2 civarında olduğu görülmektedir.



Şekil 5.6. Yapay sinir ağları ile yapılan modellemede iterasyona bağlı olarak hata değişimi

Bu modelleme sonucunda elde edilen υ-B^2 eğrisi Şekil 5.7'deki gibidir.



Şekil 5.7. Yapay sinir ağları ile yapılan modelleme sonucunda elde edilen υ-B^2 eğrisi

Bu eğri kübik spline yöntemi ile modellenen eğri ile karşılaştırıldığında, yaklaşımın daha başarılı olduğu görülmektedir.

6.YAPAY SİNİR AĞLARINI SONLU ELEMANLAR YÖNTEMİNDE KULLANARAK DOĞRU AKIM MAKİNASININ MANYETİK ALAN ANALİZİ

6.1. Hava Aralığındaki Akı Yoğunluğu Dağılışının Analitik Olarak Hesaplanması

Bölüm 1.3.1' de bahsedilen Ampére teoremine göre, **H** vektörünün kapalı bir eğri boyunca hesaplanan eğrisel integrali, integrasyon yolunun çevrelediği toplam akıma eşittir. Teoremin matematiksel ifadesi;

$$\oint \mathbf{H} \,.\, d\mathbf{L} = N.I$$

şeklindedir. Burada N akım devresindeki sarım sayısı, N.I ise toplam akımdır. Dolayısıyla toplam akımın birimi Amper-sarım' dır. Sarım sayısı için bir birim yoktur ve dolayısıyla Amper-sarım yerine sadece Amper de kullanılabilir. Bu durumda **H**'ın birimi Amper-sarım/metre (AT/m) veya Amper/metre (A/m) olur.

Bir vektörün akısı, o vektörün akıyı geçirdiği yüzeye dik bileşeninin bu yüzey boyunca hesaplanan integrali olduğuna göre, **B** vektörünün akısı da,

$$\phi = \int \mathbf{B} \,.\, d\mathbf{S}$$

şeklinde olacaktır. Halbuki **B** vektörünün kapalı yüzeyden geçirdiği akı sıfır olacağı için,

$$\nabla \,.\, \mathbf{B} = 0$$

şeklinde yazılır. Ampére teoreminin yukarıda verilen ifadesindeki **H** vektörünün sirkülasyonu yerine, Stokes teoreminden yararlanarak;

$$\oint \mathbf{H} \,.\, d\mathbf{L} = \int_{(s)} (\nabla x \mathbf{H}) \,.\, d\mathbf{S} = N.I \qquad (1.16)$$

elde edilir.

Bölüm 5'de bahsedilen B=f(H) mıknatıslanma eğrileri kullanılarak herhangi bir H değerine karşılık gelen B ve dolayısıyla $\mu=B/H$ olduğundan μ geçirgenliği de bulunabilir. Manyetik malzemelerde doyma arttıkça geçirgenlik küçülecektir. Ortam boşluk veya pratik olarak hava ise, mıknatıslanma eğrisi bir doğrudur. Yani, boşluğun geçirgenliği olan μ_o sabit olduğundan, B ile H doğru orantılı olarak değişir. μ_o sabit ve $4\pi x10^{-7}$' ye eşit olduğundan boşlukta;

$$H = B / \mu_o = B / (4\pi x10^{-7}) \approx 0.8x10^{6}B \qquad (6.1)$$

olur (Ergeneli, 1988). Manyetik devrelerde problemlerin çözümü için devre kollarının kesitleri, ortalama uzunlukları ve manyetik malzemenin cinsi yani mıknatıslanma eğrisi verilir. Manyetik malzemeye uygulanan m.m.k.' ler bilindiğinde, kollardan geçen akılar ancak bazı özel hallerde bulunabilir. Çünkü ϕ akısını bulmak için relüktansın bilinmesi gerekir. Relüktans ise geçirgenliğe bağlıdır. Bir malzemenin geçirgenliği B' nin fonksiyonu olarak değişir ve bilinmeyen μ' yü, bilinmeyen B' den bulmak mümkün olmaz. Her ne kadar mıknatıslanma eğrisi verilmemiş ise de eğrinin hangi noktasının probleme uyduğu kestirilemez. Bundan dolayı grafik yöntemle ancak tek cins bir manyetik malzeme ve bir de hava aralığından oluşan bir manyetik devre için bir sonuca ulaşılabilir.

Bir manyetik devreye F=N.I manyetomotorkuvveti uygulandığı takdirde, analitik yöntemle hava aralığındaki akı yoğunluğunu bulmak için; N sarım sayısı, I devreden geçen akım ve l hava aralığının genişliği olmak üzere;

$$F = N.I = H.l \qquad (6.2)$$

eşitliğinde $H = B / \mu$ yazılırsa,

$$N.I = \frac{B}{\mu} l \qquad (6.3)$$

hava aralığında $\mu = \mu_o = 4\pi x10^{-7}$ olduğundan,

$$N.I = \frac{B}{\mu_o} l \tag{6.4}$$

ve buradan da,

$$B = \frac{N.I.\mu_o}{l} \tag{6.5}$$

elde edilir. Akının geçtiği devre kısmının dik kesiti S ile gösterilirse,

$$B = \phi / S \tag{6.6}$$

bağıntısından ϕ akısı hesaplanabilir.

6.2. Gerçek Makinada Hava Aralığında Akı Yoğunluğunun Hesabı

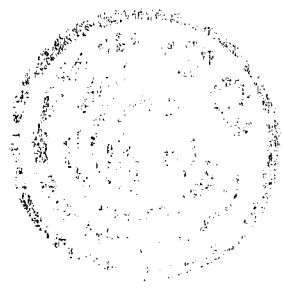
5.Bölüm'de etiket değerleri verilen doğru akım makinasının konstrüksiyon resmi Şekil 6.1'de verilmiştir. Endüinin uzunluğu l_a=0.125m. dir. Bir kutbun sarım sayısı 2600 olup, uyarma akımının sınırları 0.57 – 3.58 A arasındadır. Hava aralığında, minimum uyarmada (6.2) ifadesine göre;

$$H = \frac{N.I}{l} = \frac{2600x0{,}57}{3.10^{-3}} = 494000 \text{ AT/m} = 190 \text{ A/m}$$

olarak bulunur. Kutup yüzey kesiti;

$$S = \frac{\pi.D}{2p} l_a = \frac{\pi.106.10^{-3}}{2}.125.10^{-3} = 0.02\text{m}^2$$

olarak hesaplanır (Berkol, 1974). Buna göre hava aralığı relüktansı;



$$\Re = \frac{l}{\mu_o . S} = \frac{3.10^{-3}}{4\pi.10^{-7}.0.02} = 119366 \ H^{-1}$$

ve oluşacak akı değeri;

$$\phi = \frac{2600.0,57}{119366} = 12.10^{-3} \ Wb$$

elde edilirse, buradan hava aralığındaki akı yoğunluğu;

$$B = \frac{\phi}{S} = \frac{12.10^{-3}}{0.02} = 0.6 \ T$$ olarak bulunur.

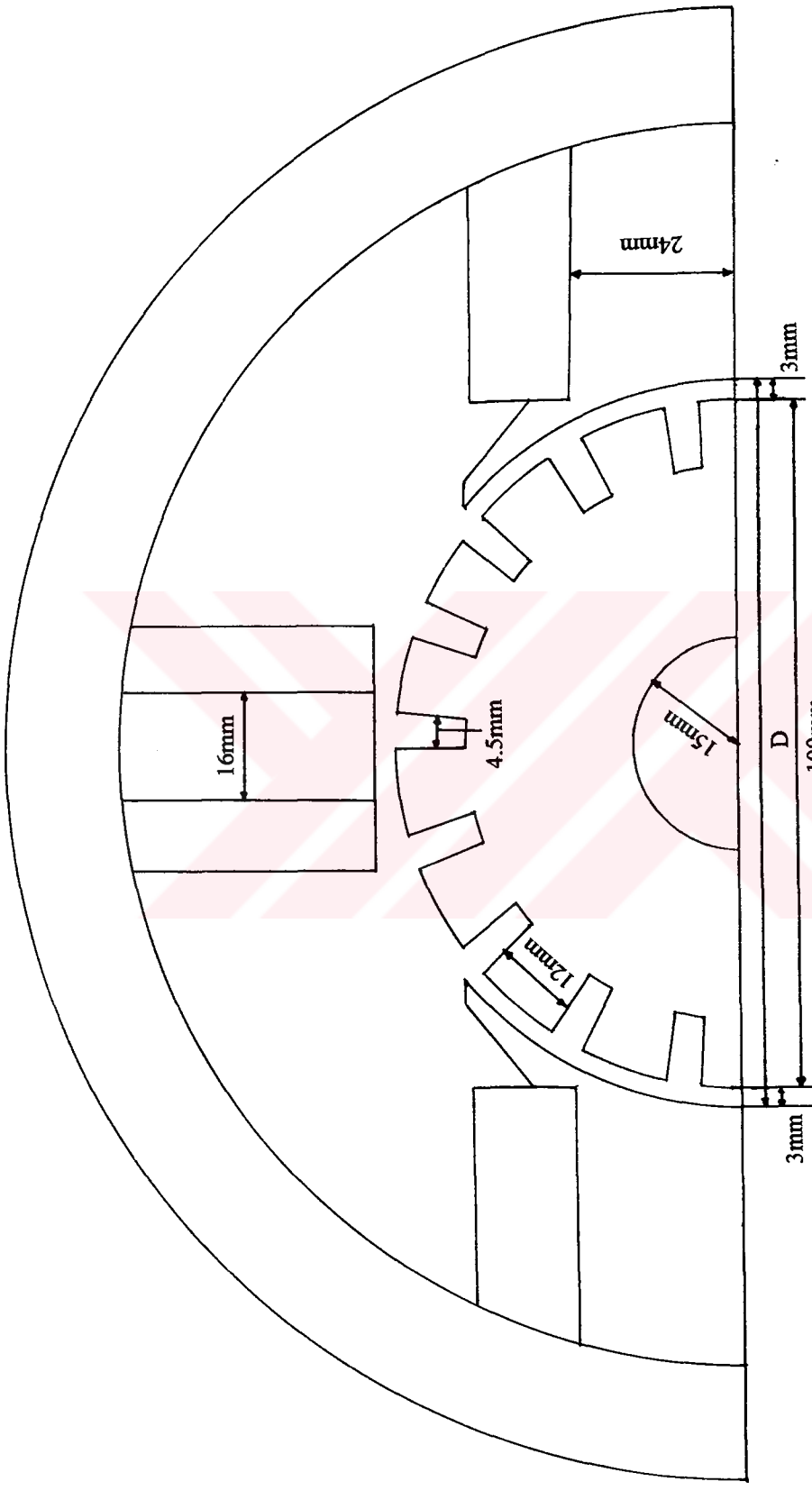
Bu değer, doğru akım makinalarının tasarımında seçilen tipik bir değerdir. Seçilen H değeri için malzemenin mıknatıslanma karakteristiğinden kutuplarda oluşacak ortalama akı yoğunluğu 1,5T olarak okunur. Bu değer malzemenin doyma noktası civarında olduğundan, kutuplar için uygundur. Bu hesaplar makine içersindeki alanın lineer ve nonlineer çözümleri ile karşılaştırılarak, kutuplardaki gerçek akı yoğunluğu değeri hesaplanacaktır.

6.3. Kullanılan Sonlu Elemanlar Modeli

Analiz için iki boyutlu sonlu elemanlar modeli kullanılmıştır. Dolayısıyla makinanın z ekseni yönündeki uzunluğu sonsuz olarak kabul edilir. Bu durumda manyetik vektör potansiyel sadece x ve y' nin fonksiyonudur. (2.20) eşitliği ile verilen genel ikinci dereceden diferansiyel denklem, doğru akım makinasındaki alanı tanımlamak üzere

$$\frac{\partial}{\partial x}\left(\upsilon \frac{\partial \mathbf{A}}{\partial x}\right) + \frac{\partial}{\partial y}\left(\upsilon \frac{\partial \mathbf{A}}{\partial y}\right) = -\mathbf{J} \qquad (6.7)$$

halini alır. Manyetik vektör potansiyelin z bileşeninin sıfır ve **J** akım yoğunluğunun sadece z bileşeninin olması ile (6.7) eşitliği;



Şekil 6.1. Seçilen doğru akım makinasının konstrüksiyon resmi

$$\nabla.(\upsilon\nabla.\mathbf{A})=-\mathbf{J} \tag{6.8}$$

halini alır. (Silvester ve Ferrari, 1996) Bu eşitlik varyasyonel formülasyona göre düzenlenirse, sonlu eleman eşitliği matrisel formda;

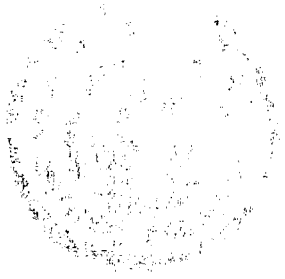
$$\upsilon[K]\{A\}=\{J\} \tag{6.9}$$

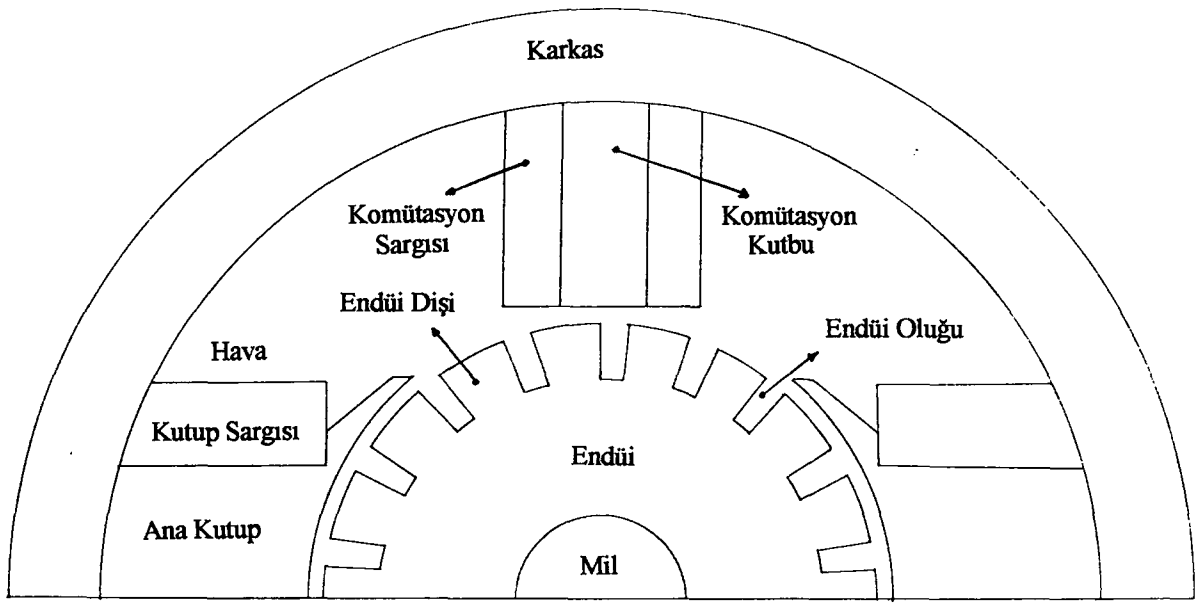
olarak elde edilir. (Chari ve Silvester, 1971)

6.3.1. Makinanın geometrisinin tanıtılması

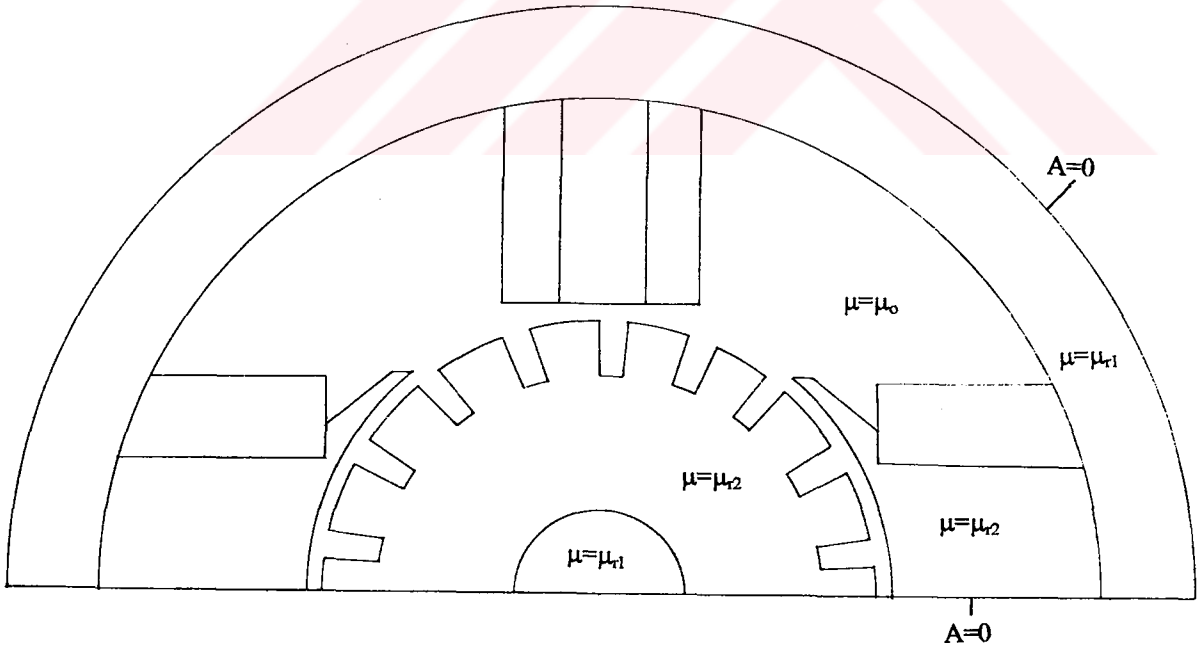
Şekil 6.1'de verilen dik kesite göre makinanın fiziksel tanımı Şekil 6.2'de gösterilmiştir. Makinanın simetrisinden dolayı analizde makinanın yarısı modelleneceğinden Şekil 6.2'de makinanın yarısı gösterilmiştir. Şekil 6.3'de makinanın sonlu elemanlar modelinde kullanılacak olan matematik modeli gösterilmiştir.

Endüi mili ile karkası oluşturan malzeme aynı seçilmiş ve rölatif permeabilitesi 1000 olarak seçilmiştir. Bu değer nonlineer çözümde de korunmaktadır. Lineer çözümde kutup ve endüi saçlarının rölatif permeabilitesi 3000 seçilmiş ve nonlineer çözümde bu değer herbir eleman için güncelleştirilmiştir.





Şekil 6.2. Seçilen doğru akım makinasının fiziksel tanımı



Şekil 6.3. Seçilen doğru akım makinasının matematik modeli

6.3.2. Sonlu eleman yaklaşımı

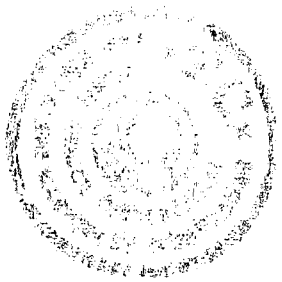
Sonlu eleman formülasyonunda ağ yapısında lineer üç düğümlü üçgen elemanlar kullanıldığından, Bölüm 2'de verilen yöntem uygulanmıştır. Analiz için yapılan kabuller aşağıdaki gibi

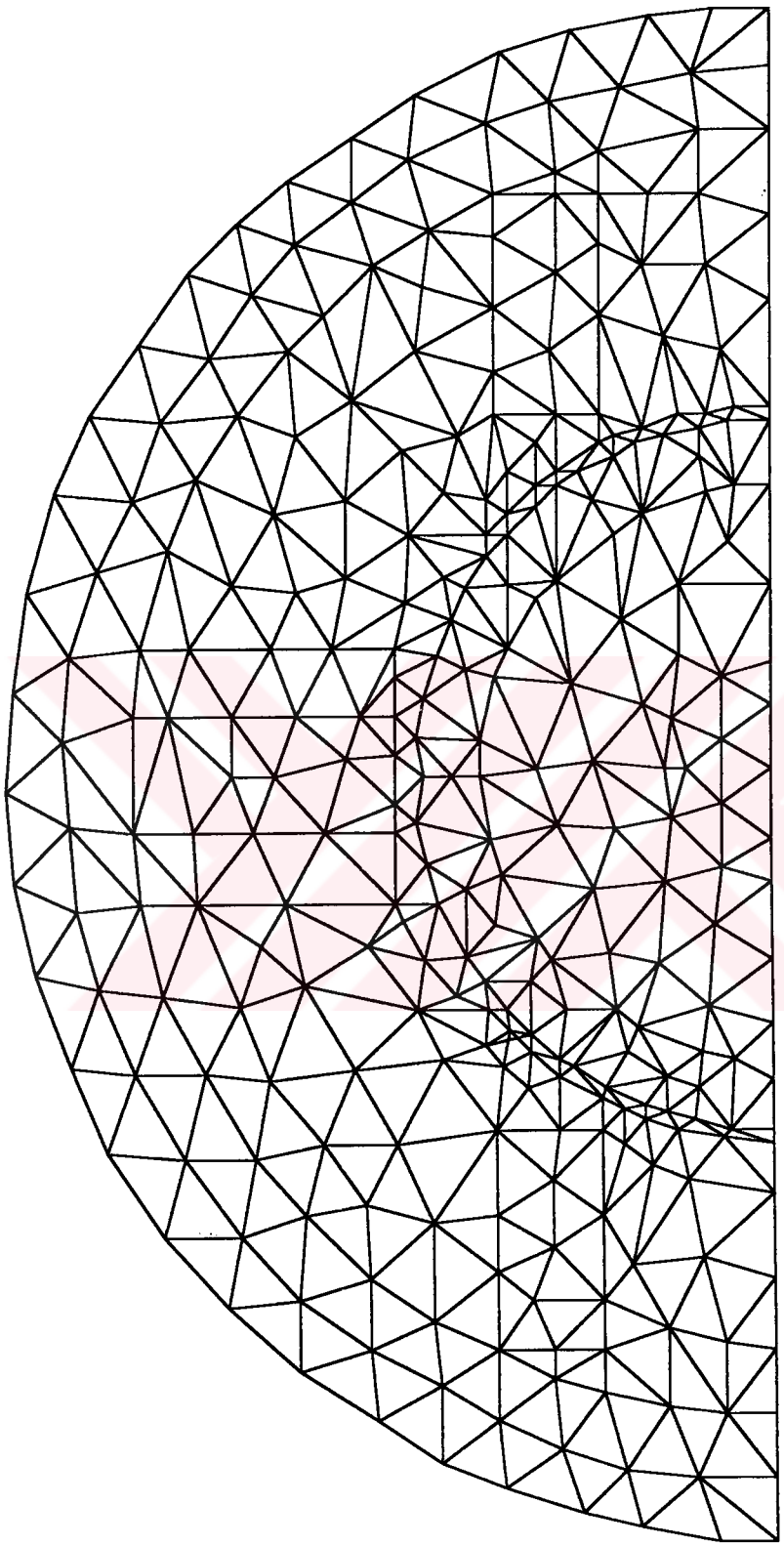
1. Makina z ekseni yönünde sonsuz uzundur.
2. Deplasman akımları ihmal edilmiştir.
3. Girdap akımları ihmal edilmiştir.
4. Seçilen elemanların lineerliğinden dolayı,üçgen elemanlar üzerinde her noktada akı yoğunluğu ve relüktivite sabit kabul edilmiştir.

Şekil 6.3'de görüldüğü gibi,doğru akım makinasının simetrisinden dolayı tüm sınırlarda manyetik vektör potansiyelin değeri sıfır olarak alınmıştır. Dolayısıyla tüm sınırlarda homojen Dirichlet sınır şartı uygulanmıştır.

6.3.3. Kullanılan ağ yapısı

Analiz için seçilen ağ yapısı Şekil 6.4'de verilmiştir. Hava aralığında enerjinin yoğun olması sebebiyle çözümün iyileştirilmesi için üçgen elemanlar mümkün olduğunca küçük seçilmiştir. Kararlı hal analizi yapıldığından dolayı hava aralığındaki üçgen elemanların konumu sabittir. Ağdaki toplam elemanların sayısı 553,düğüm sayısı 304'dür. Eleman sayısının düğüm sayısına oranı 1.82'dir. Bu oran sonlu eleman yaklaşımı için oldukça uygundur. Seçilen eleman sayısının arttırılması çözümüm iyileştirebilir ancak çözüm süresini arttıracağından bu değerde tutulmuştur.





Şekil 6.4. Seçilen doğru akım makinasının analizinde kullanılan ağ yapısı

6.4. Elde Edilen Alan Dağılımları

Sonlu eleman analizi ile elde edilen sonuçların grafiksel olarak değerlendirilebilmesi için elemanlar üzerinde eş vektör potansiyel noktalarının çizilmesi gereklidir. Sonuçta elde edilen dağılım aynı zamanda akı dağılımını verir. Makina üzerindeki bu dağılıma bakarak kaçak alanların durumuna göre geometri yeniden tasarlanabilir. Elde edilen alan dağılımlarına göre kaçak alan dağılımlarının çok az olduğu görülmektedir. Oluk için akı çizgilerine bakıldığında da bunların endüi devresi selfi üzerine çok fazla etkisinin olmayacağı görülmektedir.

6.4.1. Lineer çözümle elde edilen alan dağılımı

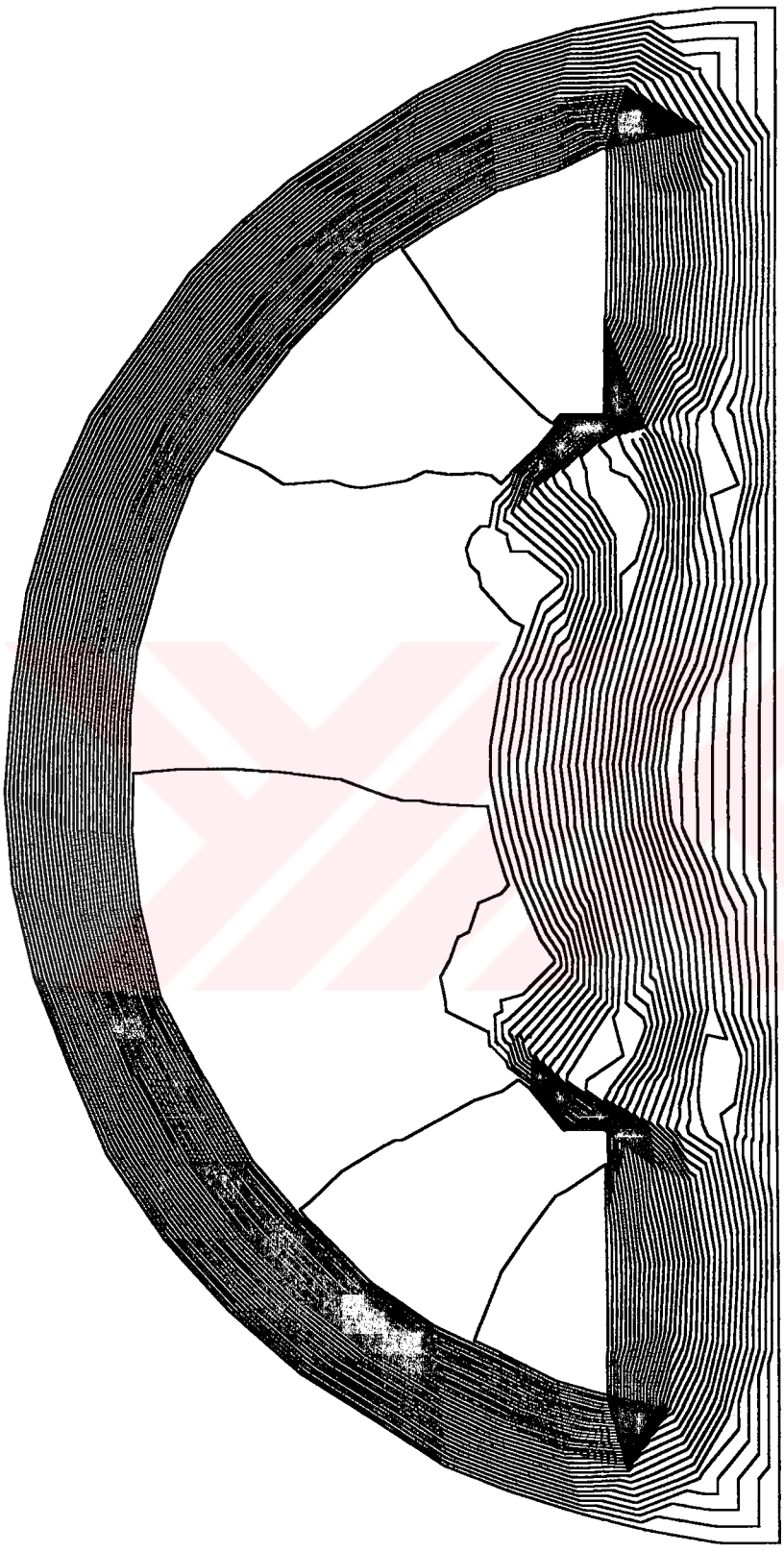
Lineer çözümde makinanın lineer bölgede çalıştığı kabul edilir. Dolayısıyla sonlu eleman eşitliğindeki relüktivite değeri manyetik vektör potansiyelinden bağımsız olur. Malzememiz için rölatif permeabilite değeri 3000 alınmıştır. Dolayısıyla saçın relüktivite değeri 265.25 m/H alınmıştır. Lineer çözümde elman üzerindeki akı yoğunluğunun değişik değerler almasına rağmen relüktivite değerinin sabit kalması çözümün doğruluğunu olumsuz yönde etkileyecektir. Şekil 6.5, 0.57 A'lik uyarma akımında 0.00055 Wb/m'lik eş potansiyel farkı için lineer çözümle elde edilen eş vektör potansiyel çizgilerini göstermektedir.

Çizelge 6.1 ise çeşitli uyarma akımlarına karşılık analitik ve sonlu elemanlar yöntemiyle hesaplanan hava aralığı akı yoğunluğu değerlerini göstermektedir.

Çizelge 6.1. Lineer analiz sonucunda elde edilen akı yoğunlukları.

Uyarma akımı (A)	Analitik çözüm B(T)	SEYile çözüm B(T)	Hata oranı
0.57	0.6	0.53	%11
1.5	1.63	1.43	%12
2.5	2.72	2.21	%18

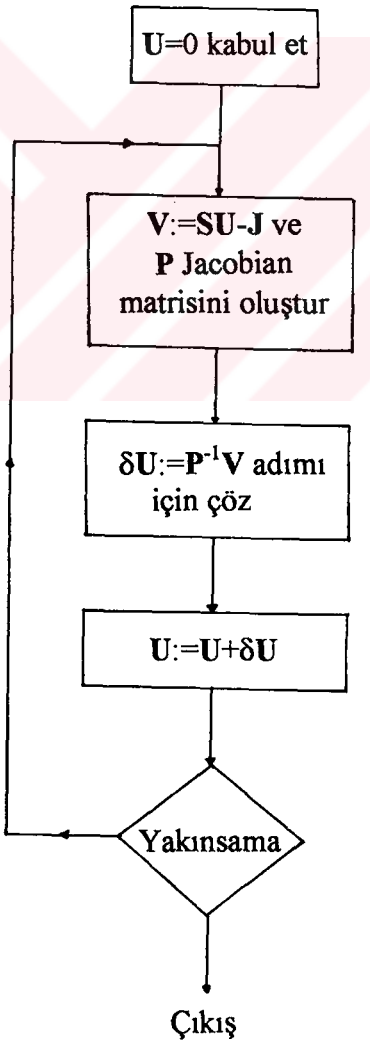
Hata oranının %10'dan büyük olması lineer çözümün yetersizliğini göstermektedir. Diğer uyarma akımlarındaki alan dağılımları benzer olduğundan değişimler verilmemiştir.

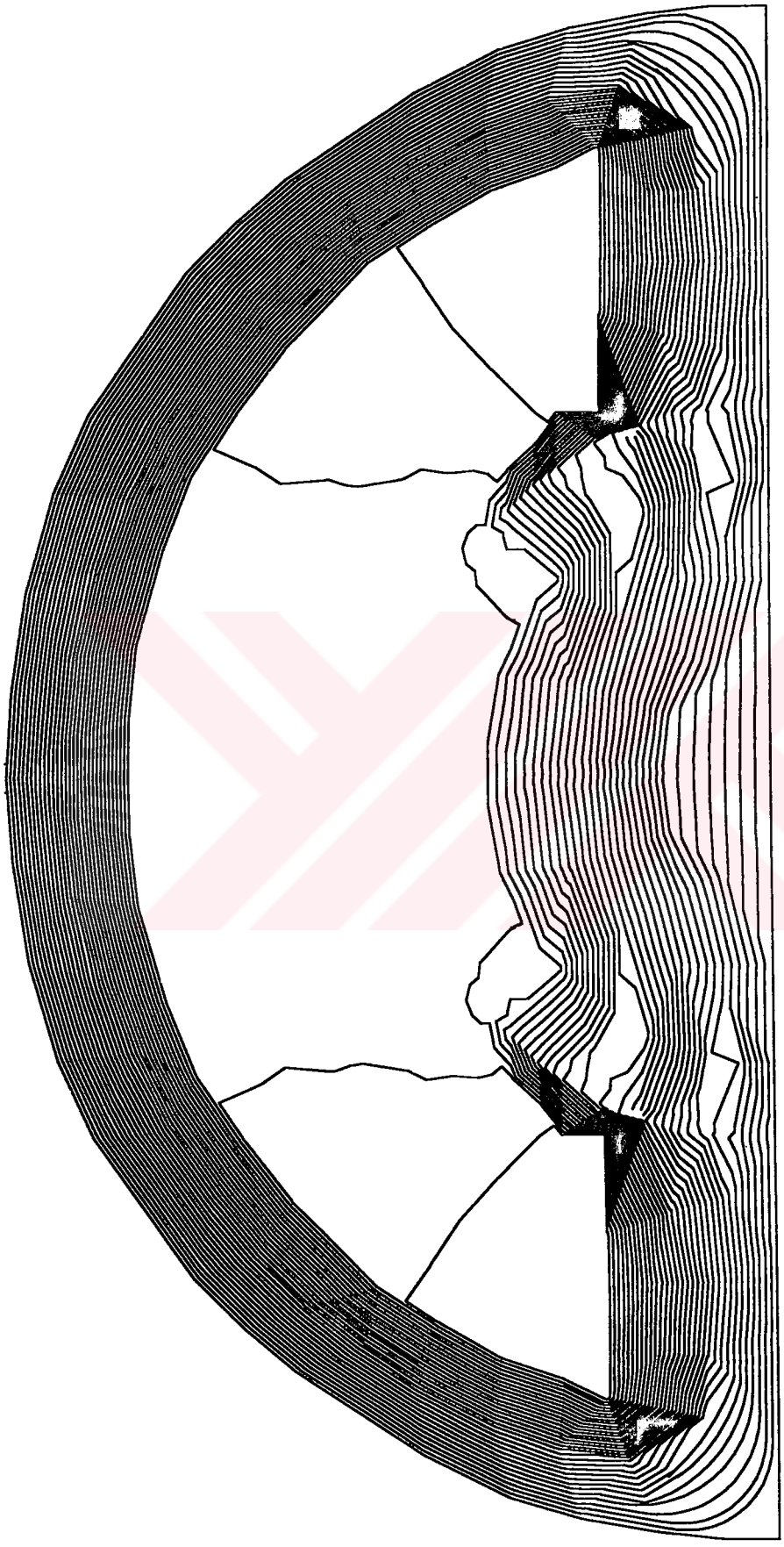


Şekil 6.5. Lineer çözümle elde edilen eş vektör potansiyel çizgileri

6.4.2. Nonlineer çözümle elde edilen alan dağılımı

Bu bölümde, Bölüm 3.1 ve 3.2'de açıklanan nonlineer eşitlik çözümleri yönteminin manyetostatik problemlere uygulanması esas alınarak, analiz edilen doğru akım makinasının alan çözümü elde edilmiştir. Doymanın göz önüne alınması sonlu eleman çözümünü iyileştirmiştir. Newton-Raphson yönteminin uygulanmasında,manyetik vektör potansiyel için başlangıç değerleri olarak lineer çözümle elde edilen değerler seçilmiştir. Bu durumda Newton-Raphson için iterasyon sayısı azalmıştır ancak lineer çözümde harcanan süre kayıp gibi gözükmekle beraber Newton-Raphson yöntemindeki güncelleştirme süresine eşdeğerdir. Newton-Raphson yönteminde tolerans 10^{-7} olarak seçilmiştir. Aşağıda Newton-Raphson yönteminin temel prensibini gösteren akış diyagramı verilmiştir.





Şekil 6.6. 2.5 A'lik uyarma akımında nonlineer çözümle elde edilen akı dağılımı

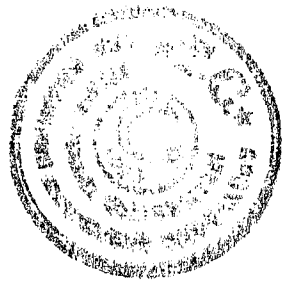
Nonlineer alan çözümünde, elemanlar lineer kaldığı halde her bir iterasyonda eleman üzerindeki relüktivite değeri güncelleştirilir. Yine elemanlar lineer kaldığından, her iterasyon sonrası bulunan akı yoğunluğu değeri elemanın her noktasında aynı kalmaktadır. Çizelge 6.2'de nonlineer analiz sonucunda elde edilen değerler ile analitik çözüm sonuçları verilmiştir.

Çizelge 6.2. Nonlineer analiz sonucunda elde edilen akı yoğunlukları.

Uyarma akımı (A)	Analitik çözüm B(T)	SEYile çözüm B(T)	Hata oranı
0.57	0.6	0.58	%3.3
1.5	1.63	1.57	%3.6
2.5	2.72	2.65	%2.5

Nonlineer analiz sonucunda elde edilen hata oranı %3 civarındadır. Doyma göz önüne alındığından, relüktivite değerleri sadece kutup ve endüi saçlarında kullanılan malzeme için güncelleştirilmiştir. Diğer malzemeler için relüktivite değerleri sabit, $d\upsilon/dB^2$ terimi ise sıfır alınmıştır. Elde edilen akı dağılımı Şekil 6.6'da gösterilmiştir.

Kullanılan malzemenin υ-B^2 eğrileri yapay sinir ağları ile modellenerek, sonlu elemanlar ile nonlineer analizde, gerekli olan relüktivite değerleri bu yöntemden elde edilmiştir. yapay sinir ağları eğitilmek suretiyle eğriyi öğrendikten sonra bir ağırlık dosyası meydana getirmektedir. Gerekli olan relüktivite değerleri yapay sinir ağları test modunda çalıştırılarak elde edilir. Dolayısıyla, eğri bir kere öğretildikten sonra, ağırlık dosyası saklandığından her zaman kullanıma hazırdır. Sonlu elemanlar yöntemi ile çözüme başlarken malzemeye ait ağırlık dosyası belirtilir ve gerekli değerler okunur. Yapay sinir ağları programı, sonlu elemanlar yöntemi işlemcisi programında bir alt program olarak çalışmaktadır. Program listeleri Ekler'de verilmiştir.



6.5. Değişik Hava Aralıkları ve Kutup Yüzey Şekilleri için Alan Dağılımları ve Akı Yoğunlukları

Bu bölümde değişik hava aralıkları ve kutup yüzey şekilleri için alan dağılımları ve akı yoğunlukları hesaplanarak analitik çözümle karşılaştırılmıştır. Hesaplamada nonlineer analiz kullanılmıştır. Hava aralığı 1 mm arttırılarak ve 1 mm azaltılarak akı yoğunlukları hesaplanmıştır. Elde edilen sonuçlar Çizelge 6.3 ve 6.4'de verilmiştir.

Çizelge 6.3. Hava aralığı 1 mm arttırıldığında akı yoğunluğu değerleri.

Uyarma akımı (A)	Analitik çözüm B(T)	SEYile çözüm B(T)	Hata oranı
0.57	0.46	0.43	%6.5
1.5	1.22	1.19	%2.4
2.5	2.04	1.98	%2.9

Çizelge 6.4. Hava aralığı 1 mm azaltıldığında akı yoğunluğu değerleri.

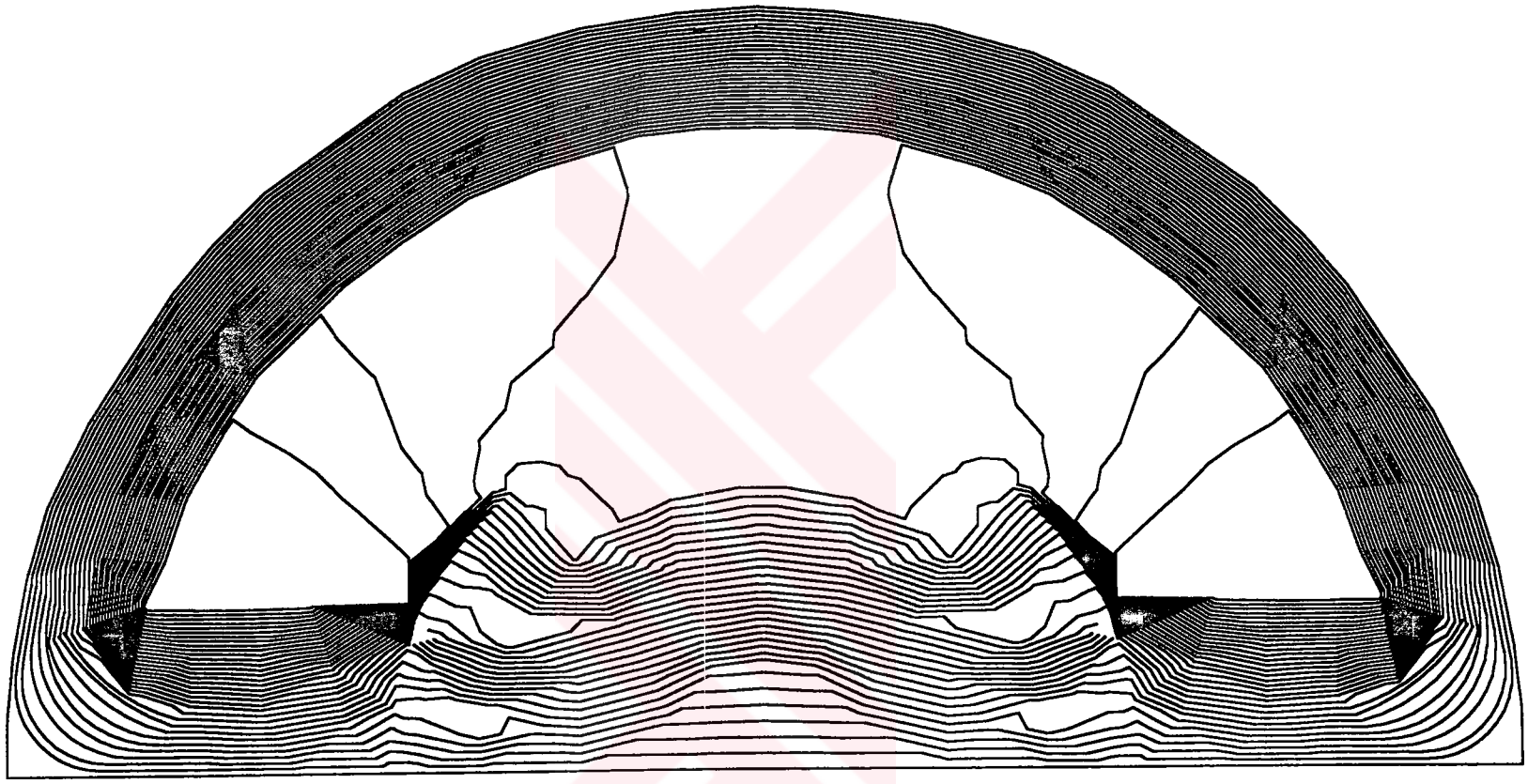
Uyarma akımı (A)	Analitik çözüm B(T)	SEYile çözüm B(T)	Hata oranı
0.57	0.93	0.85	%8.6
1.5	2.45	2.40	%2
2.5	4.08	3.96	%2.9

Daha sonra kutup yüzey şekli uç noktalarında 1.5 mm kapatılıp, orta noktalarda 1 mm açılarak değiştirilmiş ve yapılan analiz sonucunda elde edilen sonuçlar Çizelge 6.5'de verilmiştir.

Çizelge 6.5. Kutup yüzey şekli değiştirildiğinde elde edilen akı yoğunluğu.

Uyarma akımı (A)	Analitik çözüm B(T)	SEYile çözüm B(T)	Hata oranı
1.5	1.22	1.198	%1.8

Kutup yüzey şekli bu şekilde değiştirildiğinde kutup uçlarına doğru seçilen elemanlar daha da küçüldüğünden sonuca daha iyi bir yaklaşım sağlanmıştır. Elde edilen akı dağılımları Şekil 6.7'de gösterilmiştir.



Şekil 6.7. Kutup yüzey şekli değiştirildiğinde elde edilen akı dağılımı

7. SONUÇLAR VE TARTIŞMA

Yapay sinir ağları, sonlu elemanlar yönteminde direkt problem çözümünde ilk defa kullanılmıştır. İnvers elektromanyetik problemlerin çözümünde uygulama alanları bulunmasına rağmen manyetik alan problemlerinin direkt çözümünde kullanılmamıştı. Bu tezde, nonlineer alan çözümünde malzeme karakteristiklerinin modellenmesinde kullanılarak yapay sinir ağlarının sonlu elemanlar yöntemiyle birlikte kullanılmasının mümkün olduğu gösterilmiştir. Modelleme sonuçlarının kübik spline yöntemiyle karşılaştırılması sonucunda iyi bir yaklaşımın sağlandığı görülmüştür. Analiz için birleştirilen iki yöntemdeki hata miktarı toplam hataya yansımıştır. Sonuçlardaki hataların sebepleri aşağıdaki şekilde özetlenebilir:

1. Sonlu elemanlar yönteminde lineer üç düğümlü elemanlar kullanılmıştır. Bu elemanların enterpolasyon fonksiyonları birinci dereceden olduğundan, manyetik vektör potansiyel birinci dereceden bir fonksiyon ile tanımlanmıştır.

2. Sonlu elemanlar yöntemi için seçilen ağdaki eleman sayısı yeterli olmayabilir. Eleman sayısının arttırılması, eleman yüzeylerinin küçülmesi anlamına geleceğinden, hesaba olumlu etki yapması söz konusu olacaktır. Ancak bu durum çözüm süresini arttıracağından, seçilen eleman sayısı uygun bulunmuştur.

3. Nonlineer analizin yapılmış olması yukarıdaki iki etkenin getirdiği dezavantajları bir ölçüde ortadan kaldırmıştır.

4. Yapay sinir ağları ile modellemede tek gizli katman kullanılmıştır. Gizli katman sayısının artmasının modelleme sonucunu ne şekilde etkileyeceği araştırılmamıştır. Gizli katman sayısı ile birlikte diğer parametrelerin (momentum katsayısı, öğrenme katsayısı ve iterasyon sayısı) değişmesi öğrenmeyi büyük ölçüde etkilemektedir. Ancak bunların değerlerinin artmasıyla öğrenmenin iyileşmesi arasında lineer bir bağlantı yoktur.

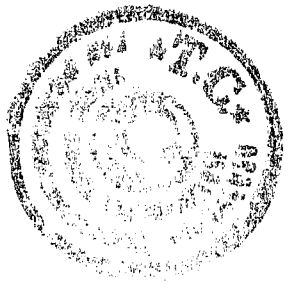
5.Doğru akım makinasının analizinde sadece boşta çalışma gözönüne alınmıştır. Mıknatıslanma eğrisinin modellenmesinde sonlu elemanlar yönteminin içinde yapay sinir

ağlarının kullanımı amaçlandığından yüklü çalışma analizi yapılmamıştır. Dolayısıyla endüi iletkenlerindeki akım sıfır alınmıştır. Boşta çalışmada küçük de olsa bir endüi akımı bulunacağından, bu akımın ihmali de sonuçlara yansımıştır.

6. Değişik kutup yüzey şekilleri için akı yoğunlukları hesaplandığında, kutup başlarında hava aralığının daraltılıp ortalarında genişletilmesinin, hava aralığında elde edilen akı yoğunluğu değerinin analitik çözüm sonucuna oldukça yaklaştığı görülmektedir.

Sonuç olarak , yapay sinir ağlarının sonlu elemanlar yönteminde kullanımı ile nonlineer analizde malzeme karakteristiğinin modellenmesinde yeni bir yaklaşım sunulmuştur. Yapay sinir ağları sonlu elemanlar yönteminde sadece test fazında çalıştığından çözüm süresi kısalarak analiz basite indirgenmiştir.

İlerideki çalışmalarda yüklü çalışmanın da göz önüne alınması mümkündür. Yüklü çalışma gözönüne alındığında, komütasyon kutuplarından da endüi akımı geçeceğinden bu kutupların da tasarımı mümkün olacaktır. Çalışma daha yüksek dereceden enterpolasyon fonksiyonuna sahip olan elemanlarla tekrarlanabilir.



KAYNAKLAR

Bazaraa, M. S., Sherali, H. D. ve Shetty, C. M., (1993), Nonlinear Programming Theory and Algorithms, John Wiley & Sons, Inc., New-York.

Berkol, A. N., (1974), Elektrik Makinalarının Hesabı, Dizerkonca Matbaası, İstanbul.

Boduroğlu, T., (1988), Elektrik Makinaları Dersleri, Doğru Akım Makinaları, Beta Basım Yayım Dağıtım A.Ş., İstanbul.

Beasley, M. D. R., (1979), "A Comparative Study of Three Methods for Computing Electric Fields", Proc. IEE. Vol.126, No.1, 126-134.

Brauer, J. R., (1975), "Simple Equations for the Magnetization and Reluctivity Curves of Steel", IEEE Trans On Magnetics, Vol. MAG-11, 81.

Brauer, J. R., (1982), "Finite Element Analysis of DC Motors and Step Motors", Proc. of Incremental Motion Control Symposium, Rosemont, IL, May.

Cardosa, J. R., (1994), "FEM Modelling of Grounded Systems with Unbounded Approach", IEEE Trans. On Magnetics, Vol.30, No.5, 2893-2896, September.

Chari, M. V. K., (1974), "Finite Element Solution of the Eddy-Current Problem in Magnetic Structures", IEEE Trans. On PAS., Vol. PAS 93, 62-72.

Chari, M. V. K. ve Silvester, P., (1971), "Analysis of Turboalternator Magnetic Fields by Finite Elements", IEEE Trans. On Power Apparatus and Systems, Vol. PAS 90, No.2, 454-464, March/April.

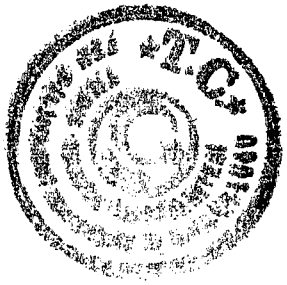
Chari, M. V. K. ve Silvester, P. P., (1971), "Finite Element Analysis of Magnetically Saturated DC Machines", IEEE Trans. On PAS., Vol PAS. 90, No.1, 2362-2972, Jan./ Feb.

Chari, M. V. K. ve Silvester, P. P., (1980), Finite Elements in Electrical and Magnetic Field Problems, John Wiley & Sons Ltd.

Chavanne, J., Meunier, G. ve Sabonnadiere, J.C., (1989), "Nonlinear Permenant Magnets Modeling with the Finite Element Method", IEEE Trans. On Magnetics, Vol. MAG 25, No.5, 3581-3583, September.

Chedid, R., (1994), "A Practical Introduction to the Finite Elements Analysis of Electromagnetic Problems", IEEE, Vol.31, 155-166.

Dedulle, J. M., Meunier, G., Foggia, A., Sabonnadiere, J. C. ve Shen, D., (1990), "Magnetic Fields in Nonlinear Anisotropic Grain-Oriented Iron Sheet", IEEE Trans. On Magnetics, Vol. MAG 26, No.2, 524-527, March



Elshafiey, I., Udpa, D. ve Udpa, S. S., (1994), "Application of Neural Networks to Inverse Problems in Electromagnetics", IEEE Trans On Magnetics, Vol.30, No.5, 3629-3632, September.

El-Sherbiny, M. K., (1973), "Representation of the Magnetization Characteristic by a Sum of Exponentials", IEEE Trans. On Magnetics, Vol. MAG 9, No.1, 60-61, March.

Enokizono, M., Kumoi, M. ve Kawano, S., (1994), "Finite Element Analysis of Anisotropic Magnetic Materials Taking Rotation Magnetization into Account", IEEE Trans. On Magnetics, Vol.30, No.5, 3387-3390, September.

Erdelyi, E. A., Ahamed, S. V. ve Burtness, R. D., (1965), "Flux Distribution in Saturated DC Machines at No Load", IEEE Winter Power Meeting Proc., 375-381.

Erdelyi, E. A. ve Fuchs, E., (1970), "Nonlinear Magnetic Field Analysis of DC Machines, Part I: Theoretical Fundamentals", IEEE Trans. On Power Apparatus and Systems, Vol. PAS. 89, No.7, 1546-1554, September / October.

Ergeneli, A., (1986), Elektromagnetik Alan Teorisi I, Magnetik Alan Teorisi, Matbaa Teknisyenleri Basımevi, İstanbul.

Ergeneli, A., (1988), Elektromagnetik Alan Teorisi II, Magnetik Alan Teorisi, Matbaa Teknisyenleri Basımevi, İstanbul.

Fink, D. G. ve Beaty, H. W., (1993), Standard Handbook for Electrical Engineers, 13th Ed., McHraw-Hill, Singapore.

Fitzgerald, A. E., Kingsley, C. Jr. ve Umans, S. D., (1988), Electric Machinery, 4th. ed., 1 st. metric ed., McGraw-Hill, U.S.A.

Foggia, A., Sabonnadiere, J. C. ve Silvester, P.P., (1975), "Finite Element Solution of Saturated Travelling Magnetic Field Problems", IEEE Trans. On PAS., Vol. PAS 94, No. 3, 1435-1444.

Forghani, B., Lowther, D. A., Silvester, P. P. ve Stone, G. O., (1983), "Newton -Raphson Finite Element Programs for Axisymmetric Vector Fields", IEEE Trans. On Magnetics, Vol. MAG 19, No.6, 2523-2526, November.

Gutierrez, Wang, J. ve Grandin, R. D., (1989), "Estimating Hidden Units for Two-Layer Perceptrons", Proceeding of the 1st International Conference on Artificial Neural Networks, London U. K., 120-124, October.

Jin, J., (1993), The Finite Element Method in Electromagnetics, 1st. ed., John Wiley & Sons, Inc., New York.

Kardeţtuncer, H. ve Norrie, D. H., (1988), Finite Element Handbook, 1st. ed., McGrow-Hill, Sinpapore.

Karlık, B., (1993), Çok Fonksiyonlu Protezler İçin Yapay Sinir Ağları Kullanılarak Miyoelektrik Kontrol, , Doktora Tezi , YTÜ Fen Bilimleri Enstitüsü.

Kraus, J. D., (1992), Electromagnetics, 4th. ed. McGraw-Hill, U.S.A.

Krause, P. C., (1987), Analysis of Electric Machinery, 2nd. ed., McGraw-Hill, Singapore.

Kreyszig, E., (1993), Advanced Engineering Mathematics, 7th. ed., John Willey & Sons, Inc., Canada.

Lippmann, (1987), "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, 4-22, April.

Malik, N. H., (1989), "A Review of the Charge Simulation Method and Its Applications", IEE Trans. On El. Ins., Vol.24, No.1, 3-20, February.

Mathews, J. H., (1992), Nümerical Methods for Mathematics, Science and Engineering, 2nd. ed., Prentice-Hall, Inc., U.S.A.

Minsky ve Papert, S., (1969), Perceptrons, Cambridge, MA: MIT Press, Introduction.

Naylor, C. J., (1990), Artificial Neural Networks Review, University of Nottingham, Version 1.1, December.

Pera, T., Ossart, F. ve Waeckerle, T., (1993), "Field Computation in Nonlinear Anisotropic Sheets Using Coenergy Model", IEEE Trans. On Magnetics, Vol. MAG 29, No.6, 2425-2427, November.

Ralph, J.W. ve Williamson, S., (1983), "Solution of Two-Dimensional Non-Linear Field Problems with Sinusoidal Excitation Sources using First-Order Finite-Elements", IEEE Trans. On Magnetics, Vol. MAG 19,2433-2436.

Reddy, J. N., (1984), An Introduction to the Finite Element Method, McGraw-Hill, U.S.A.

Ren, Z. ve Razek, A., (1994), "A Strong Coupled Model for Analysing Dynamic Behaviours of Non-Linear Electromechanical Systems", IEEE Trans. On Magnetics, Vol. 30, No.5, 3252-3255, September.

Sabonnadiere, J. C. ve Coulomb, J. L., (1987), Finite Element Methods in CAD, 2nd. ed., North Oxford Academic Publishers Ltd., London.

Sadiku, M. N., (1989), "A Simple Introduction to Finite Element Analysis of Electromagnetic Problems", IEEE Trans. On Education, Vol 32, No.2, 85-93, May.

Silvester, P. ve Chari, M. V. K., (1970), "Finite Element Solution of Saturable Magnetic Field Problems", IEEE Trans. On. Power Apparatus and Systems, Vol. PAS 89, No.7, 1642-1648, September/October.

Silvester, P. P., Cabayan, H. S. ve Browne, B., (1973), "Efficient Techniques for Finite-Element Analysis of Electric Machines", IEEE Trans. On PAS, Vol. PAS 92, 1274-1281.

Silvester, P. P. ve Ferrari, R. L., (1991), Finite Elements for Electrical Engineers, 2nd. ed., Cambridge University Press. Cambridge.

Silvester, P. P. ve Gupta, R. P., (1991), "Effective Computational Models for Anisotropic Soft B-H Curves", IEEE Trans. On Magnetics, Vol. MAG 27, No.5, 3804-3807.

Silvester, P. P. ve Ferrari, R. L., (1996), Finite Element for Electrical Engineers, 3rd ed., Cambridge University Press, Cambridge.

Tandon, S. C., Richter, E. ve Chari, M. V. K., (1980), "Finite Elements and Electrical Machine Design", IEEE Trans. On Magnetics, Vol. MAG 16, No.5,1020-1022, September.

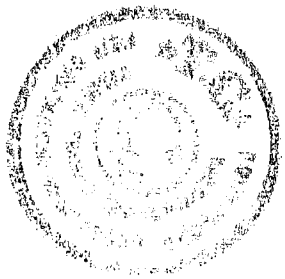
Türker, E. Ş. ve Can, E., (1997), Sayısal Analiz Yöntemleri Değişim Yayınları, Adapazarı.

Winrow ve Lehr, M. A., (1990), "Perceptron, Madaline and Back Propagation", Proceedings of the IEEE, Vol.78, No.9, 1417-1439, September.

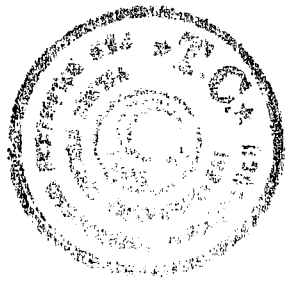
EK

PROGRAM modelling υ-B^2 curve;

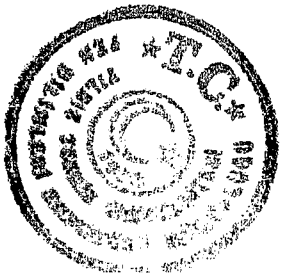
```
{$N+,E+}
uses crt;

CONST
Layers_Max    = 3;     { Maximum number of layers }
Nodes_Max     = 4;     { Maximum number of node in any layer }
{Outputs_Max   = 250;    { Number of input combinations }
Input_Nodes   = 1;     { Number of input layer nodes }
Output_Nodes  = 1;     { Number of output layer nodes }
Hidden_Nodes  = 4;     { Number of hidden layer nodes }
Epsilon       = 0.9;
Alpha         = 0.9;
Iterative_Max = 50000;   { Maximum iteration number }
Every         = 10000;   { After every iteration print results }

LABEL 10,20,30,40,y1,y2;
VAR
i, ii, j, count, tt, isylla, num : longint;
eps, alp, err, error, terr, terror, total_rate: single;
Layers, Inputs, Outputs : longint;
d1,te:INTEGER;
a1,b1,c1,a,b,c:real;
Node     : array [0..Layers_Max-1] of integer;
{matrix    : array [0..250-1, 0..(Input_Nodes+Output_Nodes-1)] of double;}
matrix_tmp: array [0..trunc(Iterative_Max/Every), 0..250-1,
            0..(Input_Nodes+Output_Nodes-1)] of single;
xd,yd,zd,cd,tc:array [0..250] of real;
des    : double;
rate   : array [0..250-1] of double;
out    : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
der    : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
theta  : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
dtheta : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
weight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dess   : array [0..250-1, 0..Nodes_Max-1] of double;
st1: string[10];
e,f,g,st2, st3, st4: string[12];
outkor,koor,outfile,dosya : text;
weightfile : TEXT;
```



```
wof : text;

PROCEDURE verimenu;
LABEL Y;
begin
clrscr;
TextAttr := yellow;
HighVideo;
gotoxy(25,12);writeln('name of the input file ?');
gotoxy(25,15);writeln('       ---------- ');
TextAttr := red;
Y:gotoxy(35,14);
READLN (g);
assign(dosya,g);
{$I-}
Reset(dosya);
{$I+}
if IOResult <> 0 then BEGIN
clrscr;gotoxy(25,12);Writeln('File not found. Try again');GOTO Y;END;
GOTOXY(25,16);WRITELN('Number of the input pattern');GOTOXY(35,18);
READLN(num);
for i:=0 to num-1 do
readln(dosya,xd[i],yd[i]);
close(dosya);
assign(outfile,g+'.out');rewrite(outfile);
assign(weightfile,g+'.bin');rewrite(weightfile);
end;

PROCEDURE Wrand;
{ Initializes node offsets & weights at the beginning of the program }
var
li, ni, ni_end, no, no_end : integer;
drand48 : real;
{ power: double;}
begin
randomize;
{  power:=1.0;
for li:=1 to 31 do
power:=power*2;
power:=power-1;
{ Randomize offsets }
for li:=1 to Layers_Max-1 do
begin
```

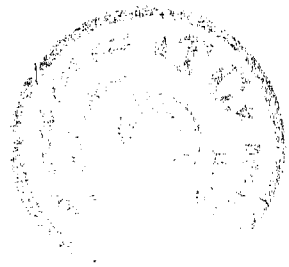


```
ni_end:=node[li];
for ni:=0 to ni_end-1 do
begin
drand48:=Random(30)/100;
theta[li,ni]:= drand48;
dtheta[li,ni]:=0;
end;
end;

{ Randomize weights }
for li:=1 to Layers_Max-1 do
begin
ni_end:=node[li];
no_end:=node[li-1];
for ni:=0 to ni_end-1 do
for no:=0 to no_end-1 do
begin
drand48:=Random(100)/100;
weight[li,ni,no]:=drand48;
dweight[li,ni,no]:=0;
end;
end;
end;  { procedure WRAND }

PROCEDURE Initial;
{ Reset to zero each output node before every iteration}
VAR
l, n, n_end: integer;
begin
for l:= 0 to Layers-1 do
begin
n_end:= node[l];
for n:= 0 to n_end-1 do
out[l,n]:= 0;
end;
end;    { procedure INITIAL }


FUNCTION sigmoid( xx: double ): double;
{ sigmoid function }
var
y: double;
begin
```



```
y:=1/(1+exp(-xx));
sigmoid:=y;
end;  { function SIGMOID }

PROCEDURE wforward;
{ wforward propagation }
var
li, ni, ni_end, no, no_end: integer;
x: double;
begin
for li:=1 to Layers-1 do
begin
ni_end:= node[li];
no_end:= node[li-1];
for ni:= 0 to ni_end-1 do
begin
x:= 0;
for no:= 0 to no_end-1 do
x:= x + weight[li,ni,no] * out[li-1,no];
out[li,ni]:= sigmoid( x-theta[li,ni] );
end;
end;
end;   { procedure WFORWARD }

FUNCTION back: double;
{ back propagation }
var
li, ni, ni_end, no, no_end: integer;
err : real;
e, x, d, w, y: double;
begin
err:=0;
for ni:=0 to outputs-1 do
begin
y:= out[2,ni];
des:= dess[isylla,ni];
e:= des - y;
der[2,ni]:= e*y*(1-y);
err:=err+ e*e;
end;
for li:=2 downto 1 do
begin
no_end:= node[li-1];
```

```
ni_end:= node[li];
for no:= 0 to no_end-1 do
begin
x:= 0;
y:= out[li-1,no];
for ni:= 0 to ni_end-1 do
begin
d:= der[li,ni];
w:= weight[li,ni,no];
x:= x + d*w;
end;
der[li-1,no]:= y*(1-y)*x;
end;
end;
back:= 0.5*err;
end; { function BACK }

PROCEDURE learning (alp, eps: double );
{ update offsets & weights }
var
li, lo, ni, ni_end, no, no_end: integer;
di, yo, ew, et: double;
begin
for li:= 1 to layers-1 do
begin
ni_end:= node[li];
for ni:= 0 to ni_end-1 do
begin
et:= der[li,ni];
dtheta[li,ni]:= -eps*et + alp*dtheta[li,ni];
theta[li,ni]:= theta[li,ni] + dtheta[li,ni];
end;
end;
for li:=1 to layers-1 do
begin
lo:= li-1;
ni_end:= node[li];
no_end:= node[lo];
for ni:= 0 to ni_end-1 do
begin
di:= der[li,ni];
for no:= 0 to no_end-1 do
begin
```

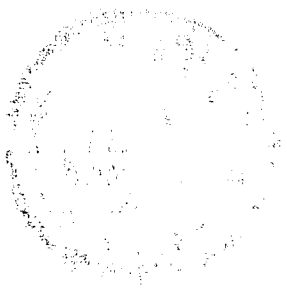


```
yo:= out[lo,no];
ew:= di*yo;
dweight[li,ni,no]:= eps*ew + alp*dweight[li,ni,no];
weight[li,ni,no]:= weight[li,ni,no] + dweight[li,ni,no];
end;
end;
end;
end;  { procedure LEARNING }


PROCEDURE make_matrix( it, isylla: integer );
var
i: integer;
begin
for i:= 0 to inputs-1 do
matrix_tmp[it,isylla,i]:= out[0,i];
for i:= 0 to outputs-1 do
matrix_tmp[it,isylla,inputs+i]:= out[Layers_Max-1,i];
end;  { procedure MAKE_MATRIX }

PROCEDURE write_weights;
var
lc,nc,nc_end,ic,ic_end: integer;
begin
rewrite(weightfile);
for lc:=1 to layers-1 do
begin
nc_end:=node[lc];
ic_end:=node[lc-1];
for nc:=0 to nc_end-1 do
for ic:=0 to ic_end-1 do
begin
writeln(weightfile,weight[lc,nc,ic],' ',theta[lc,nc]);
{       write(weightfile,theta[lc,nc]);}
end;
end;
end;

PROCEDURE read_weights;
var
lc,nc,nc_end,ic,ic_end: integer;
begin
reset(weightfile);
```

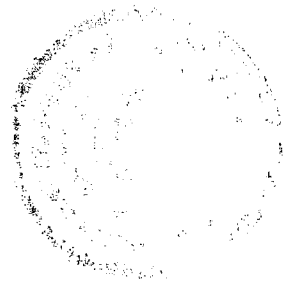
```
for lc:=1 to layers-1 do
begin
nc_end:=node[lc];
ic_end:=node[lc-1];
for nc:=0 to nc_end-1 do begin
for ic:=0 to ic_end-1 do
begin
read(weightfile,weight[lc,nc,ic],theta[lc,nc]);
{  read(weightfile,theta[lc,nc]);}
end;end;
end;
end ;

BEGIN { main }
node[2]:= Output_Nodes;
Outputs:= Output_Nodes;
node[1]:= Hidden_Nodes;
node[0]:= Input_Nodes;
Inputs:= Input_Nodes;
Layers:= Layers_Max;
eps:= Epsilon;
alp:= Alpha;
clrscr;
TextAttr := red;
40:clrscr;gotoxy(20,10);write('TRAINING (1) - TESTING (2) - EXIT (3) ? ');
TextAttr := yellow;
readln(tt);
if tt=2 then goto 10;
IF tt=3 THEN GOTO 30;
verimenu;



{ ********************
     INITIALIZE MATRIX
  ********************** }
{  for i:= 0 to num-1 do begin
for j:= 0 to ( inputs+Outputs-2 ) do
matrix[i,j]:= 0;
end;}
for ii:= 0 to ( Trunc(Iterative_Max/Every)-1 ) do begin
for i:= 0 to num-1 do begin
for j:= 0 to ( inputs+Outputs-2 ) do
```

```
matrix_tmp[ii,i,j]:=0;
end;end;


{ ************************
   DESIRED DATA READING
 ************************}

for i:=0 to num-1 do
dess[i,0]:=yd[i];

writeln('TRAINING ...');writeln;
wrand;
j:=0;{dcount:=trunc(iterative_max/every)+1;

{ ************************************
        LEARNING PHASE
************************************ }
while ( j < Iterative_Max ) do
begin
{dcount:=dcount-1;}
for count:= 0 to Every-1 do
begin
error:= 0;
for isylla:=0 to num-1 do
begin
initial;
out[0,0]:=xd[isylla];
{ yread( isylla );}
wforward;
err:= back;
learning( alp, eps );
error:= error+err;
end;
error:= error / outputs;
eps:= 0.5 * sqrt( error );
Inc( j );
TextAttr := green;
{writeln(j,'. iteration ',Iterative_Max-j,' iteration more ');}
end;

{ ********************************************
            TESTING PHASE
```

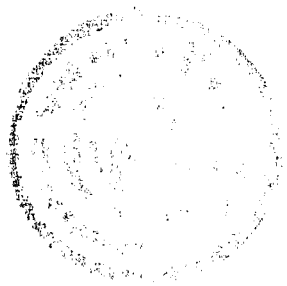
```
********************************************** }
terror:= 0;
for isylla:= 0 to num-1 do
begin
out[0,0]:=xd[isylla];
{ yread( isylla );}
wforward;
terr:= back;
terror:= terror + terr;
make_matrix( trunc( j/Every ) - 1, isylla );
end;
writeln (outfile, ' NO:   X        Y        Z       THC     GRADE ');
writeln (outfile, '-------------------------------------------------');
for i:= 0 to num-1 do
begin
str( i:3, st1 );
write(outfile,st1+' ');
for ii:= 0 to inputs-1 do
begin
str(10*matrix_tmp[trunc(j/Every)-1,i,ii]:6:2, st1 );
write(outfile,st1+' ');
end;
write( outfile,' ');
for ii:= 0 to outputs-1 do
begin
str(10000*matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:6:2,st1);
write( outfile, st1+' ');
end;
for ii:= 0 to outputs-1 do
begin
str(10000*dess[i,ii]:6:2,st1);
write(outfile,st1+' ');
end;
writeln(outfile,'');
end;
terror:= terror/outputs;
str ( j:4, st1 ); str ( eps:9:5, st2 );
str ( error:9:5, st3 ); str ( terror:9:5, st4 );
writeln( outfile, 'iteration=',st1,'    eps=',st2,
'    error_1=', st3, '    error_t=', st4 );
writeln( outfile );
{ }
total_rate:=0;
```



```
for isylla:= 0 to outputs-1 do
begin
rate[isylla]:= trunc(matrix_tmp[trunc(j/Every)-1,isylla,isylla]);
total_rate:= total_rate + rate[isylla];
end;
total_rate:= total_rate/outputs;
{ }
if ( j = Iterative_Max ) then
begin
writeln( outfile, 'learn & test finished at time' );
end;
write_weights;
end;
close( outfile );close(weightfile);
goto 40;
10:clrscr;
writeln('TESTING ...');writeln;
TextAttr := yellow;
HighVideo;
gotoxy(25,12);writeln('WRITE THE NAME OF THE WEIGHT FILE?');
TextAttr := red;
Y1:gotoxy(36,14);
READLN (g);
if g='' then goto y1;
assign(weightfile,g);
{$I-}
Reset(weightfile);
{$I+}
if IOResult <> 0 then BEGIN
clrscr;gotoxy(25,12);Writeln('File not found, TRY AGAIN');GOTO Y1;END;
read_weights;
20:write('X = ? ');readln(a1);
{write('Y = ? ');readln(b1);
write('Z = ? ');readln(c1);}
out[0,0]:=a1;{out[0,1]:=b1;out[0,2]:=c1;}
if (out[0,0]=0) then goto 40;
wforward;
writeln(' X          Y          Z        THC          GRADE ');
writeln('-------------------------------------------------------------');
for ii:=0 to inputs-1 do
begin
str(out[0,ii]:6:2,st1);
write(st1+'   ');
```

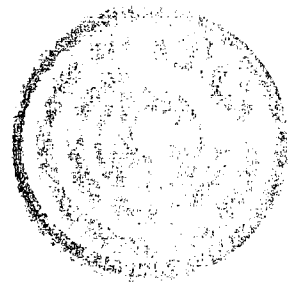


```
end;
write('');
for ii:=0 to outputs-1 do
begin
str(out[layers-1,ii]:4:6,st1);
write(st1+'');
end;
writeln;writeln;
goto 20;
goto 40;
30:end.
```

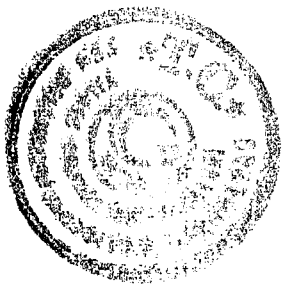


Sonlu elemanlar yöntemi için yazılan programın işlemci kısmı

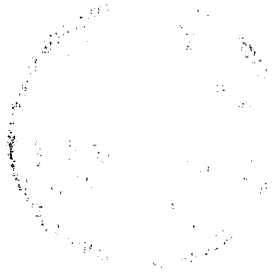
```
program SEY;
{$N+}
{*$M 65520,,0}
uses kutu,crt,graph,dos;
const mu0=4*pi*1E-07;sab=2500;

var
id1,id2,ji,i0,mi,j0,mj,mij,ij,jm1,ik,jk,jj,ii,j1,j2,j3,i1,kk,k:longint;
t1,b1,k2,i,k1,b4,b5,b6:longint;
nd1,b2,b3,t11,t22,t33,t44,t55,t66,et1,sk,w:integer;
REL,xd1,yd1,xd2,yd2,xd3,yd3,be1,ge1,be2,ge2,be3,ge3,y2,s,kl:real;
t2,t3,t4,t5,t6,t8,x2,x3,FY1,fy2,k11:longint;
fdenrt,fdenort,FDEN,f1,f2,v,e1,teta,rlp,rl1,rel1,at:real;
sum1,sum2,sum3,li,sli,cur,v1,v2,v3,v4,v5,v6,v7,v8,v9:real;
SUM:double;
GK,q1,e2,e3,e4,e5,e6,r1,r3:real;
NE,N1,NK,e,j,l,c,n,sds,neq:integer;
say,sinir,NN,FYUK:LONGINT;
BK,A11,B11,C11,D11:array [0..50] of DOUBLE;
NV:array[1..500] of integer;
AE,r2,grel:array[1..600] of real;
fdenk,YM:array[1..sab] of real;
cc,G,FF,cdt:STRING[10];
D,HH,H1,h2 :text;
dk,XD,YD,GE,BE,d1,DOSYA,RL,Q,EKM,mjac:file of real;
ND,ET,TD:file of integer;
ID,FY:file of longint;
dx,cds:text;

PROCEDURE menubasi;
begin
clrscr;
cer(1,1,79,25);
end;

procedure ses;
begin
Sound(220);
Delay(250);
NoSound;
end;
```



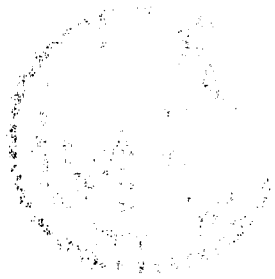
```
PROCEDURE verimenu;
BEGIN
menubasi;
TEXTCOLOR(14);
GOTOXY(20,10);WRITE('GIRIS DOSYASI ISMI                ?: ');
GOTOXY(20,13);WRITE('POTANSIYEL DOSYASINA VERILECEK ISIM ?: ');
GOTOXY(20,16);WRITE('AKI DOSYASINA VERILECEK ISIM       ?: ');
TEXTCOLOR(14);GOTOXY(58,10);READLN(g);GOTOXY(58,13);READLN(FF);G
OTOXY(58,16);READLN(cc);
end;

procedure filemake;
var
x1:real;
begin
assign(dosya,'katmat.dat');rewrite(dosya);
assign(dk,'katmat1.dat');rewrite(dk);
x1:=0.0;say:=0;
sinir:=NN*NN;
for say:=1 to sinir do begin
seek(dosya,say);write(dosya,x1);
seek(dk,say);write(dk,x1);
end;
close(dosya);close(dk);
end;

procedure gir;
label 5,10;
begin
assign(HH,g);reset(HH);
readln(HH,NN,NE,N1,sds);
assign(XD,'X.dat');rewrite(XD);
assign(YD,'Y.dat');rewrite(YD);
for i:=1 to NN do begin
readln(HH,xd1,yd1);
seek(XD,i);seek(YD,i);
write(XD,xd1);write(YD,yd1);
end;
assign(TD,'elmat.dat');rewrite(TD);
assign(ET,'etmat.dat');rewrite(ET);
for i:=1 to NE do begin
t1:=(i-1)*3+1;t2:=(i-1)*3+2;t3:=(i-1)*3+3;
```



```
readln(HH,T11,T22,T33,ET1);
seek(ET,i);write(ET,ET1);
seek(TD,t1);write(TD,T11);
seek(TD,t2);write(TD,T22);
seek(TD,t3);write(TD,T33);
end;
assign(ND,'ndmat.dat');rewrite(ND);
for i:=1 to N1 do begin
readln(HH,ND1);
seek(ND,i);write(ND,ND1);
end;
close(ND);
readln(HH,FYUK,RLP);
readln(HH,NK);
close(HH);
assign(RL,'relmat.dat');rewrite(RL);
s:=1.0;v:=1000;
for i:=1 to NE do begin
seek(ET,i);read(ET,et1);
if (ET1=1) or (ET1=3) then begin
seek(RL,i);write(RL,s);end;
if (ET1=4) then begin
seek(RL,i);write(RL,v);end;
if (ET1=2) then begin
seek(RL,i);write(RL,RLP);
end;
end;
assign(FY,'fyk.dat');rewrite(FY);
t1:=0;
for i:=1 to NE do begin
seek(FY,i);seek(ET,i);read(ET,et1);
if et1=1 then
write(FY,fyuk)
else
write(FY,t1) ;
end;
close(FY);

filemake;

assign(dosya,'katmat.dat');reset(dosya);
assign(BE,'bemat.dat');rewrite(BE);
assign(GE,'gemat.dat');rewrite(GE);
```

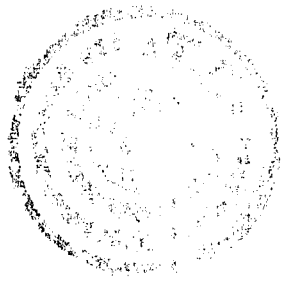


```
{assign(AE,'almat.dat');rewrite(AE);}
assign(FY,'fyk.dat');reset(FY);
assign(EKM,'ekm.dat');rewrite(EKM);
assign(Q,'rhs.dat');rewrite(Q);
t1:=0;t2:=0;t3:=0;
for n:=1 to NE do begin
t1:=(n-1)*3+1;t2:=(n-1)*3+2;t3:=(n-1)*3+3;
seek(TD,t1);read(TD,T11);
seek(TD,t2);read(TD,T22);
seek(TD,t3);read(TD,T33);
i:=T11;j:=T22;k:=T33;
seek(XD,i);read(XD,xd1);seek(XD,j);read(XD,xd2);seek(XD,k);read(XD,xd3);
seek(YD,i);read(YD,yd1);seek(YD,j);read(YD,yd2);seek(YD,k);read(YD,yd3);
AE[N]:=(ABS((Xd2*Yd3-Xd3*Yd2)+(Xd3*Yd1-Xd1*Yd3)+(Xd1*Yd2-d2*Yd1)))/2;
for l:=1 to 3 do begin
b1:=(n-1)*3+l;
seek(BE,b1);seek(GE,b1);
seek(XD,j);read(XD,xd2);seek(XD,k);read(XD,xd3);
seek(YD,j);read(YD,yd2);seek(YD,k);read(YD,yd3);
BE1:=Yd2-Yd3;
GE1:=Xd3-Xd2;
write(BE,BE1);write(GE,GE1);
c:=i;i:=j;j:=k;k:=c;
end;
seek(RL,n);read(RL,RL1);
REL:=1/(RL1*mu0);
seek(FY,n);read(FY,FY1);
for i:=1 to 3 do begin
t4:=(n-1)*3+i;seek(TD,t4);read(TD,t44);
q1:=(AE[N]/3)*FY1;
YM[t44]:=YM[t44]+q1;
seek(q,t4);write(q,q1);
end;
for i:=1 to 3 do begin
for j:=1 to 3 do begin
b2:=(n-1)*3+i;b3:=(n-1)*3+j;
seek(BE,b2);read(BE,BE2);seek(BE,b3);read(BE,BE3);
seek(GE,b2);read(GE,GE2);seek(GE,b3);read(GE,GE3);
S:=REL*(BE2*BE3+GE2*GE3)/(4*AE[N]);
KL:=(BE2*BE3+GE2*GE3)/(4*AE[N]);
w:=i+j;
if (i=1) or (j=1) then w:=w-1;
b2:=(n-1)*6+w;
```



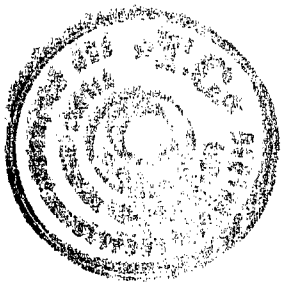
```
seek(EKM,b2);write(EKM,KL);
t5:=(n-1)*3+i;t6:=(n-1)*3+j;
seek(TD,t5);read(TD,t55);
seek(TD,t6);read(TD,t66);
x2:=(T55-1)*NN+T66;
seek(dosya,x2);
read(dosya,y2);
GK:=y2+S;
seek(dosya,x2);write(dosya,GK);
end;
end;
end;
close(TD);close(BE);
close(GE);
close(ET);
assign(ND,'ndmat.dat');reset(ND);
for i:=1 to N1 do begin
y2:=1E38;seek(ND,i);read(ND,nd1);
YM[ND1]:=0;
x2:=(ND1-1)*NN+ND1;
seek(dosya,x2);write(dosya,y2);
end;
close(dosya);
close(ND);erase(ND);
close( fy);erase(fy);
end;

procedure profil;
var
e11,t11,etc:integer;
ak:text;
label 101,350,351,352,353,452,454;
begin
assign(d1,'provek.dat');rewrite(d1);
assign(dosya,'katmat.dat');reset(dosya);
assign(ID,'prog.dat');rewrite(ID);
seek(dosya,1);seek(d1,1);
read(dosya,GK);write(d1,GK);
id1:=1;
seek(ID,1);write(ID,id1);
k:=1;
for j:=2 to NN do begin
for i:=1 to NN do begin
```

```
k1:=(j-1)*NN+i;
seek(dosya,k1);read(dosya,GK);
if (j>=i) and (GK<>0) then begin
for i:=i to j do begin
k2:=(j-1)*NN+i;
seek(dosya,k2);read(dosya,GK);
k:=k+1;
seek(d1,k);write(d1,GK);
end;end;end;
seek(ID,j);write(ID,k);
end;
close(dosya);erase(dosya);
clrscr;
for i:=2 to NN do begin
gotoxy(20,13);{writeln(i,'.','ilk döngü');}
seek(ID,i);read(ID,id1);
i0:=id1-i;
if i=1 then goto 350;
seek(ID,i-1);read(ID,id1);
mi:=id1-i0+1;
for j:=mi to i do begin
gotoxy(60,13); writeln(j,' ',i);
seek(ID,j);read(ID,id1);
SUM:=0;j0:=id1-j;mj:=1;
seek(ID,j-1);read(ID,id1);
if j>1 then mj:=id1-j0+1;
mij:=mi;
if mj>mi then mij:=mj;
ij:=i0+j;jm1:=j-1;
for k:=mij to jm1 do begin
if mij>jm1 then goto 351;
seek(ID,k);read(ID,id1);
ik:=i0+k;kk:=ID1;jk:=j0+k;
seek(d1,ik);read(d1,v1);
seek(d1,kk);read(d1,v2);
seek(d1,jk);read(d1,v3);
SUM:=SUM+V1*V2*V3;
351: end;
if j=i then goto 352;
seek(ID,j);read(ID,id1);
jj:=ID1;
seek(d1,ij);read(d1,v4);
seek(d1,jj);read(d1,v5);
```



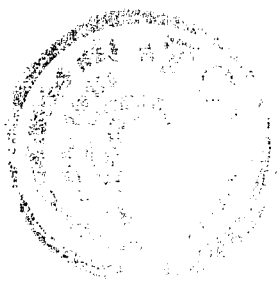
```
V4:=(V4-SUM)/V5;
seek(d1,ij);write(d1,v4);
goto 353;
352: ii:=i0+i;
seek(d1,ii);read(d1,v6);
V6:=V6-SUM;
seek(d1,ii);write(d1,v6);
353: end;
350: end;
for i:=2 to NN do begin
gotoxy(20,13);{writeln(i,'.','son döngü');}
SUM:=0;
seek(ID,i);read(ID,id1);
i0:=ID1-i;
seek(ID,i-1);read(ID,id1);
j1:=ID1-i0+1;
j2:=i-1;
if j1=i then goto 452;
for j:=j1 to j2 do begin
j0:=i0+j;
seek(d1,j0);read(d1,v7);
SUM:=SUM+V7*YM[j]
end;
452: YM[i]:=YM[i]-SUM;
end;
for i:=1 to NN do begin
seek(ID,i);read(ID,id1);
i1:=ID1;
seek(d1,i1);read(d1,v8);
YM[i]:=YM[i]/V8;
end;
for l:=2 to NN do begin
i:=NN-l+1;j1:=i+1;SUM:=0;
for j:=j1 to NN do begin
seek(ID,j);read(ID,id1);
seek(ID,j-1);read(ID,id2);
j3:=ID2-ID1+j+1;
if j3>i then goto 454;
seek(ID,j);read(ID,id1);
j0:=ID1-j+i;
seek(d1,j0);read(d1,v9);
SUM:=SUM+V9*YM[j];
454: end;
```

```
YM[i]:=YM[i]-SUM;
end;
close(ID);erase(ID);
close(d1);erase(d1);
assign(h2,ff);rewrite(h2);
for i:=1 to NN do begin
if abs(YM[i])<10E-30 then YM[i]:=0;
writeln(h2,i,' ',YM[i]);writeln(YM[i]);
end;
for i:=1 to NN do begin
if YM[i]=0 then goto 101
else NEQ:=NEQ+1;
NV[i]:=NEQ;
101: end;

close(h2);
assign(GE,'gemat.dat');reset(GE);
assign(BE,'bemat.dat');reset(BE);
assign(TD,'elmat.dat');reset(TD);
assign(ET,'etmat.dat');reset(ET);
assign(AK,cc);rewrite(AK);
{writeln(AK,'Eleman No',' ',' Akı Yoğunluğu (T)',' ','Theta(Derece)');}
{writeln(AK,'-------------------------------------------------------');}
j:=0;fdenrt:=0;
for e:=1 to NE do begin
seek(ET,e);read(ET,e11);
{if e11=2 then begin}
SUM1:=0;SUM2:=0;j:=j+1;
for i:=1 to 3 do begin
b1:=(e-1)*3+i;
seek(BE,b1);seek(GE,b1);
read(BE,be1);read(GE,ge1);
t1:=(e-1)*3+i;seek(TD,t1);read(TD,t11);
SUM1:=SUM1+BE1*YM[T11];
SUM2:=SUM2+GE1*YM[T11];
{if sum2<>0 then teta:=abs(abs(arctan(sum1/sum2)*180/pi)-90);}
end;
f1:=sqr(SUM1);f2:=sqr(SUM2);
FDEN:=sqrt(f1+f2)/(2*AE[E]);
writeln(AK,'  ',e:3,'          ',FDEN:6:4);
fdenrt:=fdenrt+fden;
end;
fdenort:=fdenrt/j;
```

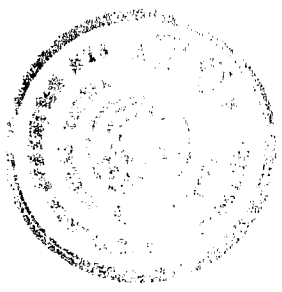


```
writeln(AK,'ortalama akı  yoğunluğu =',fdenort:7:5,'(T)');
li:=0;
{for e:=1 to NE do begin
sum3:=0;
seek(ET,e);read(ET,e11);
if (e11=12) or (e11=22) then begin
for i:=1 to 3 do begin
t1:=(e-1)*3+i;seek(TD,t1);read(TD,t11);
sum3:=sum3+YM[t11];
end;
seek(AE,e);read(AE,AE1);
li:=li+sum3*AE1;
end;
end;
cur:=0.443;
sli:=(1/(3*sqr(cur)))*li;
writeln(AK,' sargı  endüktansı  =',2*sli,'(H)');}
close(AK);
close(XD);erase(XD);
close(YD);erase(YD);
close(GE);close(BE);close(TD);close(ET);
assign(dx,'bnh.out');reset(Dx);
for i:=0 to NK do BEGIN
readln(Dx,BK[i],A11[i],B11[i],C11[i],D11[i]);END;
close(dx);
end;

procedure hes;
var
yl,s1:real;
label 1;
begin
assign(ET,'ETMAT.DAT');RESET(ET);
for e:=1 to NE do begin
seek(ET,e);read(ET,ET1);
for i:=1 to NK+1 do begin
if (ET1=2) and (BK[i]>=FDENK[e]) then begin
yl:=FDENK[e]-BK[i-1];
REL1:=A11[i-1]+B11[i-1]*yl+C11[i-1]*sqr(yl)+D11[i-1]*(sqr(yl)*yl);
seek(RL,e);write(RL,REL1);
s1:=yl+0.001;
r2[e]:=A11[i-1]+B11[i-1]*s1+C11[i-1]*sqr(s1)+D11[i-1]*(sqr(s1)*s1);
GREL[e]:=(r2[e]-REL1)/(0.001);goto 1;
```

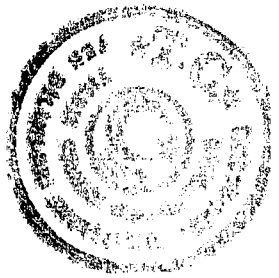
```
end;
end;
1:end;
CLOSE(ET);
end;

procedure nr;
var
SUMX:REAL;
U,ES:array[1..3] of real;
etc,w:integer;
f11,x1,gk1,v11,v12,v13:real;
snr,id1,ed1,k1:longint;
EX,ID:file of longint;
v1,rhs:file of real;
label 21,102,103,107,108,109,110,123,750,751,752,753,852,854;
begin
assign(EX,'EX.dat');rewrite(EX);
assign(ID,'ID.dat');rewrite(ID);
assign(V1,'v1.dat');rewrite(v1);
assign(RHS,'RHS1.dat');rewrite(RHS);
assign(mjac,'mjac.dat');rewrite(mjac);
assign(dk,'katmat1.dat');reset(dk);
k:=1;
seek(EX,1);write(EX,k);
k1:=0;
for i:=1 to NEQ do begin
for j:=1 to NEQ do begin
if (i-j)>0 then goto 107
else k1:=k1+1;
107: end;
k11:=k1+1;
SEEK(EX,i+1);write(EX,k11);
end;
CLOSE(EX);
assign(BE,'BEMAT.DAT');RESET(BE);
assign(GE,'GEMAT.DAT');RESET(GE);
assign(TD,'ELMAT.DAT');RESET(TD);
etc:=0;
writeln(' NR  Başlıyor ');
while etc<10 do
begin
for e:=1 to NE do begin
```



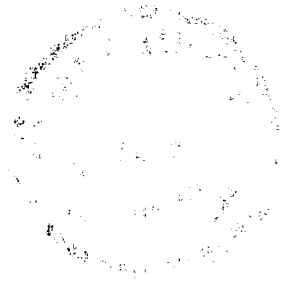
```
SUM1:=0;SUM2:=0;
for i:=1 to 3 do begin
b2:=(e-1)*3+i;
seek(BE,b2);read(BE,BE2);
seek(GE,b2);read(GE,GE2);
t1:=(e-1)*3+i;
seek(TD,t1);read(TD,T11);
SUM1:=SUM1+BE2*YM[T11];
SUM2:=SUM2+GE2*YM[T11];
end;
f11:=sqr(SUM1);f2:=sqr(SUM2);
FDEN:=sqrt(f11+f2)/(2*AE[E]);
FDENK[e]:=sqr(FDEN);
end;

hes;
v:=0.0;
for k:=1 to 100000 do begin
seek(RHS,k);write(rhs,v);
seek(MJAC,k);write(MJAC,v);
end;
ASSIGN(EX,'EX.DAT');RESET(EX);
for e:=1 to NE do begin
for j:=1 to 3 do begin
t1:=(e-1)*3+j;
seek(TD,t1);read(TD,t11);
u[j]:=YM[T11];
end;
b1:=(e-1)*6+1; seek(EKM,b1);read(EKM,e1);
b2:=(e-1)*6+2; seek(EKM,b2);read(EKM,e2);
b3:=(e-1)*6+3; seek(EKM,b3);read(EKM,e3);
b4:=(e-1)*6+4; seek(EKM,b4);read(EKM,e4);
b5:=(e-1)*6+5; seek(EKM,b5);read(EKM,e5);
b6:=(e-1)*6+6; seek(EKM,b6);read(EKM,e6);
es[1]:=E1*u[1]+E2*u[2]+E3*u[3];
es[2]:=E2*u[1]+E4*u[2]+E5*u[3];
es[3]:=E3*u[1]+E5*u[2]+E6*u[3];
for j:=1 to 3 do begin
t1:=(e-1)*3+j;
seek(TD,t1);read(TD,t11);
jj:=NV[T11];
if jj=0 then goto 102;
SEEK(rl,e);read(RL,rel1);
```



```
t4:=(e-1)*3+j;seek(q,t4);read(Q,q1);
seek(RHS,jj);read(RHS,r1);
R1:=R1+(rel1*es[j]-Q1);
seek(RHS,jj);write(RHS,r1);
for k:=1 to 3 do begin
t1:=(e-1)*3+k;
seek(TD,t1);read(TD,t11);
kk:=NV[T11];
if (kk=0) or ((jj-kk)>0) then goto 103;
seek(EX,jj);read(EX,ED1);
ji:=ED1+kk-jj;
w:=j+k;
if (j=1) or (k=1) then w:=w-1;
seek(RL,e);read(RL,RL1); b1:=(e-1)*6+w; seek(EKM,b1);read(EKM,e1);
seek(MJAC,ji);read(MJAC,v);
v:=v+(rl1*E1)+((2*AE[E])*GREL[e]*es[j]*es[k]);
seek(MJAC,ji);write(MJAC,v);
103:end;
102:end;
end;
close(EX);
for i:=1 to NEQ do begin
seek(RHS,i);read(RHS,r1);
r3:=-r1;seek(RHS,i);write(RHS,r3);
end;
k:=0;
for i:=1 to NEQ do begin
for j:=1 to NEQ do begin
if (i>j) then goto 108;
k:=k+1;
seek(MJAC,k);read(MJAC,v);writeln(i,' ',j,' ');
t4:=(i-1)*NEQ+j;writeln(t4);seek(dk,t4);write(dk,v);
108:end;
end;
k:=0;
for i:=1 to NEQ do begin
for j:=1 to NEQ do begin
if i<j then goto 110;
k:=k+1;
seek(MJAC,k);read(MJAC,v);
t4:=(i-1)*NEQ+j;seek(dk,t4);write(dk,v);
110:end;
end;
```

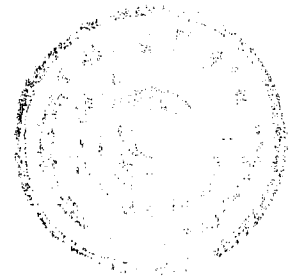


```
k:=1;
writeln('Sistem  çözülüyor ');
seek(dosya,1);read(dosya,GK1);
seek(ID,1);write(ID,k);
seek(v1,1);write(v1,GK1);k:=1;
for j:=2 to NEQ do begin
for i:=1 to NEQ do begin
t4:=(j-1)*NEQ+i;read(dosya,gk1);
if (j>=i) and (GK1<>0) then
for i:=i to j do begin
k:=k+1;
seek(v1,k);write(v1,gk1);
end;
end;
seek(ID,j);write(ID,k);
end;
close(dk);erase(dk);
jj:=0;
for i:=2 to NEQ do begin
seek(ID,i);read(ID,id1);
i0:=ID1-i;
if i=1 then goto 750;
seek(ID,i-1);read(ID,id1);
mi:=ID1-i0+1;
for j:=mi to i do begin
gotoxy(60,13); writeln(j,' ',i);
seek(ID,j);read(ID,id1);
SUMX:=0;j0:=ID1-j;mj:=1;
seek(ID,j-1);read(ID,id1);
if j>1 then mj:=ID1-j0+1;
mij:=mi;
if mj>mi then mij:=mj;
ij:=i0+j;jm1:=j-1;
for k:=mij to jm1 do begin
if mij>jm1 then goto 751;
seek(ID,k);read(ID,id1);
ik:=i0+k;kk:=ID1;jk:=j0+k;
seek(v1,ik);read(v1,v11);
seek(v1,kk);read(v1,v12);
seek(v1,jk);read(v1,v13);
SUMX:=SUMX+V11*V12*V13;
751: end;
if j=i then goto 752;
```



```
seek(ID,j);read(ID,id1);
jj:=ID1;seek(v1,i);read(v1,v11);seek(v1,jj);read(v1,v12);
V11:=(V11-SUMX)/V12;seek(v1,jj);read(v1,v11);
goto 753;
752: ii:=i0+i;seek(v1,ii);read(v1,v11);
V11:=V11-SUMX;seek(v1,ii);read(v1,v11);
753: end;
750: end;
for i:=2 to NEQ do begin
SUMX:=0;seek(ID,i);read(ID,id1);
seek(ID,i);read(ID,id1);
i0:=ID1-i;
seek(ID,i-1);read(ID,id1);
j1:=ID1-i0+1;
j2:=i-1;
if j1=i then goto 852;
for j:=j1 to j2 do begin
j0:=i0+j;seek(v1,j0);read(v1,v11);
seek(RHS,j);read(RHS,r1);
SUMX:=SUMX+V11*R1;
end;
852: seek(RHS,i);read(RHS,r1);R1:=R1-SUMX;seek(RHS,i);write(RHS,r1);
end;
for i:=1 to NEQ do begin
seek(ID,i);read(ID,id1);i1:=ID1;seek(v1,i1);read(v1,v11);
seek(RHS,i);read(RHS,r1);
R1:=R1/V11;seek(RHS,i);write(RHS,r1);
end;
for l:=2 to NEQ do begin
i:=NEQ-l+1;j1:=i+1;SUMX:=0;
for j:=j1 to NEQ do begin
seek(ID,j-1);read(ID,id1);seek(ID,j);read(ID,id2);
j3:=ID1-ID2+j+1;
if j3>i then goto 854;
seek(ID,j);read(ID,id1);
j0:=ID1-j+i; seek(v1,j0);read(v1,v11);
seek(RHS,j);write(RHS,r1);
SUMX:=SUMX+V11*R1;
854: end;
seek(RHS,i);read(RHS,r1);
R1:=R1-SUMX;
seek(RHS,i);write(RHS,r1);
end;
```

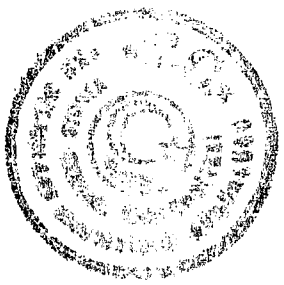
```
sk:=0;
for i:=1 to NN do begin
if YM[i]=0 then goto 123
else sk:=sk+1;seek(RHS,sk);read(RHS,r1);
YM[i]:=YM[i]+R1;
123:end;
seek(RHS,1);read(RHS,r1);
at:=abs(R1);
for j:=2 to NEQ do begin
seek(RHS,j);read(RHS,r1);
if (abs(R1))>at then at:=abs(R1);
end;
if at<0.0000001 then goto 21;
etc:=etc+1;
end;
21:close(rhs);erase(rhs);
END;

{ANA PROGRAMIN BASLANGICI}
begin
{baslik;}

verimenu;
gir;
profil;
nr;

close(RL);erase(RL);
close(TD);erase(TD);
close(ET);erase(ET);
close(mjac);erase(mjac);
writeln('SON ÇIKIŞ DOSYASI İSMİ');readln(cdt);
assign(cds,cdt);rewrite(cds);
for i:=1 to NN do
writeln(cds,YM[i]);
close(cds);
end.
```



Sonlu elemanlar yöntemi için yazılan programın ön ve son işlemci kısmı

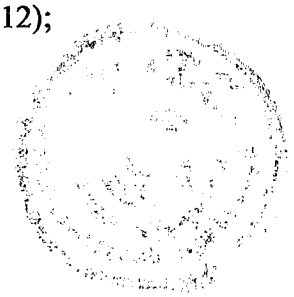
```
PROGRAM sey;
USES KUTU,CRT,GRAPH;
{$i readscrn.inc}
const
sab=2350;

var
sd,xxx,yyy,ps1,nk,n1,n2,n3,nc,ds,pds,es,i,bnd,bc,emag,a,xs,ys,rlp:integer;
ne,fy:longint;
ps,kon:real;
x1,x2,x3,y1,y2,y3:integer;
gdn:array[1..sab,1..6] of integer;
xd,yd:array[0..sab] of real;
h,v:array[1..3] of integer;
fi:array[1..3] of real;
g,cc,n:string[10];
d,e,D1:text;
kar:char;
ch,vh:string;
gd,gm:integer;
k,j,l,k1:integer;
XX,A1:ARRAY[0..30] OF REAL;
x,y:ARRAY[1..6] OF REAL;
et:array[1..sab] of integer;
bsd,nn,nb1,mm,m,np,ll,kk,c,q,sds1,sds,b1,b2:integer;
LABEL 3,4,5,111,23,100,200;

PROCEDURE menubasi;
begin
clrscr;
cer(1,1,79,25);
end;

procedure ses;
begin
Sound(220);       { Beep }
Delay(250);       { For 200 ms }
NoSound;          { Relief! }
end;

PROCEDURE ANAMENU;
```

```
BEGIN
textcolor(14);
gotoxy(20,6);write('GIRIS DOSYASI OLUSTURMA');
gotoxy(20,8);write('BOLMELEME   ');
gotoxy(20,10);write('AG CIZIMI  ');
gotoxy(20,12);write('ES POTANSIYEL CIZGILERIN CIZIMI');
gotoxy(20,14);write('CIKIS      ');
gotoxy(20,17);write('SECIM ICIN ', CHR(25) ,' ve ' , CHR(24) ,' TUSLARINI
KULLANIN');
inline($b4/$01/$b9/$00/$ff/$cd/$10);end;

procedure s;
begin
yyy:=6;
repeat
vh:='';for xxx:=28 to 55 do
begin
ch:=readscrn(xxx,yyy);
vh:=vh+ch;
end;
gotoxy(28,yyy);
write(vh,'*');
kar:=readkey;
case kar of
#80:begin
gotoxy(28,yyy);
write(vh);
if yyy=14 then
yyy:=6
else
yyy:=yyy+2;
writeln(' ');
end;
#72:begin
gotoxy(28,yyy);
write(vh);
if yyy=6 then
yyy:=14
else
yyy:=yyy-2;
writeln(' ');
end;
end;
```



```
until kar=#13;
inline($b4/$01/$b9/$10/$00/$cd/$10);
end;



PROCEDURE verimenu1;
BEGIN
menubasi;
TEXTCOLOR(14);
GOTOXY(15,5);WRITE('SONLU ELEMAN YONTEMI ICIN GIRIS DOSYASI
OLUSTURMA');
GOTOXY(20,8);WRITE('GIRIS DOSYASINA VERILECEK ISIM ?: ');
GOTOXY(20,11);WRITE('TOPLAM DUGUM SAYISI          ?: ');
GOTOXY(20,14);WRITE('TOPLAM ELEMAN SAYISI         ?: ');
GOTOXY(20,17);WRITE('POTANSIYELI BILINEN DUGUM SAY.?: ');
GOTOXY(20,20);WRITE('AGIN SINIRLARINDAKI DUGUM SAY.?: ');
TEXTCOLOR(14);GOTOXY(54,8);READLN(N);GOTOXY(54,11);READLN(DS);GGO
TOXY(54,14);READLN(ES);
GOTOXY(54,17);READLN(PDS);GOTOXY(54,20);READLN(SDS);
END;

PROCEDURE verimenu2;
BEGIN
menubasi;
TEXTCOLOR(14);
GOTOXY(22,7); WRITE('SONLU ELEMAN AGI BOLMELENECEK');
GOTOXY(20,10); WRITE('GIRIS DOSYASI ISMI          ?: ');
GOTOXY(20,13);WRITE('CIKIS DOSYASI ISMI           ?: ');
TEXTCOLOR(14);GOTOXY(54,10);READLN(g);GOTOXY(54,13);READLN(cc);
END;

PROCEDURE verimenu3;
BEGIN
menubasi;
TEXTCOLOR(14);
GOTOXY(22,9); WRITE('SONLU ELEMAN AGI CIZILECEK');
GOTOXY(20,12); WRITE('GIRIS DOSYASI ISMI          ?: ');
TEXTCOLOR(14);GOTOXY(54,12);READLN(g);
END;

PROCEDURE verimenu4;
BEGIN
```

```
menubasi;
TEXTCOLOR(14);
GOTOXY(20,9); WRITE('ES POTANSIYEL CIZGILER CIZILECEK');
GOTOXY(20,12); WRITE('GIRIS DOSYASI ISMI          ?: ');
GOTOXY(20,15);WRITE('POTANSIYEL DOSYASI ISMI      ?: ');
TEXTCOLOR(14);GOTOXY(54,12);READLN(g);GOTOXY(54,15);READLN(CC);
END;

procedure giris;
BEGIN
MENUBASI;
ASSIGN(D,N);
REWRITE(D);
WRITELN(D,DS,' ',ES,' ',PDS,' ',SDS);
CLRSCR;
MENUBASI;
k:=2;
FOR I:=1 TO DS DO
BEGIN
k:=k+1;
GOTOXY(20,k);WRITE(I);WRITE('. DUGUMUN KOORDINATLARI');
GOTOXY(48,k);READ(XD[I],YD[I]);
WRITELN(D,XD[I]:6:4,' ',YD[I]:6:4);
IF k=20 THEN BEGIN
MENUBASI;k:=2;
END;
END;
CLRSCR;
MENUBASI;
k:=2;
FOR I:=1 TO ES DO
BEGIN
k:=k+1;
GOTOXY(20,k);WRITE(I);WRITE('.ELEMANIN YERLESIMI ve TIPI');
GOTOXY(50,k);READ(GDN[I,1],GDN[I,2],GDN[I,3],ET[I]);
WRITELN(D,GDN[I,1],' ',GDN[I,2],' ',GDN[I,3],' ',ET[I]);
IF k=20 THEN BEGIN
MENUBASI;k:=2;
END;
END;
CLRSCR;
MENUBASI;
GOTOXY(28,12);WRITELN('SINIR SARTLARINI GIRIN');GOTOXY(50,12);
```

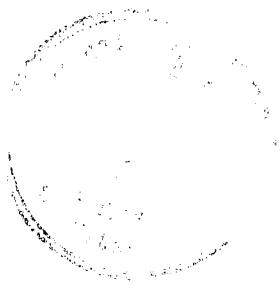


```
DELAY(1000);
MENUBASI;
k:=2;
FOR I:=1 TO PDS DO
BEGIN
k:=k+2;
GOTOXY(16,k);WRITE('POTANSIYELI BILINEN DUGUMUN NUMARASI');
GOTOXY(54,k);READ(NN);
GOTOXY(41,k+1);WRITE('POTANSIYELI');
GOTOXY(54,k+1);READ(BC);
WRITELN(D,NN,' ',BC);
IF k=20 THEN BEGIN
MENUBASI;k:=2;
END;
END;
menubasi;
K:=2;
ASSIGN(D1,'BH.DAT');REWRITE(D1);
GOTOXY(16,4);WRITE('SPLINE ICIN NOKTA SAYISINI GIRIN');
GOTOXY(50,4);READ(NK);WRITELN(D1,NK);K:=4;
FOR I:=0 TO NK DO BEGIN
K:=K+2;
GOTOXY(25,K);WRITE(I);WRITE('. NOKTANIN DEGERLERI?');
GOTOXY(48,K);READ(XX[I],A1[I]);WRITELN(D1,XX[I],' ',A1[I]);
IF K=20 THEN BEGIN
MENUBASI;K:=2;
END;
END;
CLOSE(D1);
MENUBASI;
GOTOXY(20,9);WRITE('AKIM YOGUNLUGUNU GIRIN');
GOTOXY(43,9);READ(FY);
GOTOXY(20,12);WRITE('DEMIR MAL. REL. PER. GIRIN');
GOTOXY(47,12);READ(RLP);
WRITELN(D,FY,' ',RLP);
close(D);
END;

procedure v1;
var
s:integer;
begin
for s:=1 to 3 do begin
```

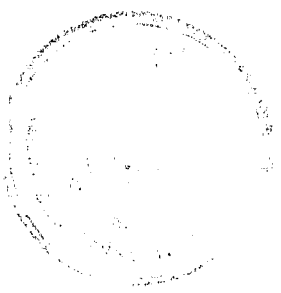
```
x[s]:=xd[gdn[i,s]];
y[s]:=yd[gdn[i,s]];
end;
x[4]:=x[1]+(x[2]-x[1])/2;
y[4]:=y[1]+(y[2]-y[1])/2;
x[5]:=x[2]+(x[3]-x[2])/2;
y[5]:=y[2]+(y[3]-y[2])/2;
x[6]:=x[1]+(x[3]-x[1])/2;
y[6]:=y[1]+(y[3]-y[1])/2;
end;

procedure refinement;
var
n11,n12,n13,n14,n15,say,sinir:longint;
x11,b,n21,n22,n23,n24,i1,i2,i3,i4,i5,j1,j2,j3,j4:integer;
PS,MF,NB:file of integer;
label 1,2,3,4,5,6,7,8,9,10;
begin
assign(d,cc);rewrite(d);
assign(e,g);reset(e);
readln(e,ds,es,pds,sds);
sd:=trunc((3*es+sds)/2);
np:=ds+sd;
ne:=4*es;
sds1:=2*sds;
for i:=1 to ds do
readln(e,xd[i],yd[i]);
for i:=1 to es do
readln(e,gdn[i,1],gdn[i,2],gdn[i,3],et[i]);
assign(MF,'mode.dat');rewrite(MF);
assign(PS,'ps.dat');rewrite(PS);
j:=0;
for i:=1 to np do begin
seek(MF,i);write(MF,j);
seek(PS,i);write(PS,j);
end;
x11:=1;
for i:=1 to pds do begin
readln(e,a);
seek(PS,a);write(PS,j);
seek(MF,a);write(MF,x11);
end;
readln(e,fy,rlp);
```

```
close(e);

assign(NB,'nb.dat');rewrite(NB);
sinir:=NE*4;
x11:=0;
for say:=0 to sinir do begin
seek(NB,say);write(NB,x11);
end;
close(NB);
c:=0;
assign(NB,'nb.dat');reset(NB);
for i:=1 to es do begin
for j:=1 to es do begin
n11:=(i-1)*4;seek(NB,n11);read(NB,n21);
if (j=i) or (n21=3) then goto 1;
for k:=1 to 3 do begin
for l:=1 to 3 do begin
nn:=gdn[i,k];mm:=gdn[j,l];
if nn=mm then c:=c+1;
end;
end;
if c=2 then begin
n12:=(i-1)*4;
seek(NB,n12);read(NB,n22);
n22:=n22+1;
seek(NB,n12);write(NB,n22);
n13:=(i-1)*4+n22;
seek(NB,n13);write(NB,j);
end;
c:=0;
1:end;
end;
q:=ds;
for i:=1 to es do begin
v1;
n14:=(i-1)*4;
seek(NB,n14);read(NB,n23);
for j:=4 to 6 do begin
for k:=1 to n23 do begin
for l:=4 to 6 do begin
n15:=(i-1)*4+k;seek(NB,n15);read(NB,n24);
kk:=n24;
ll:=gdn[kk,l];
```

```
if (x[j]=xd[ll]) and (y[j]=yd[ll]) and (ll<>0) then goto 2;
5:end;
end;
if m=0 then goto 3;
goto 4;
3:q:=q+1;
gdn[i,j]:=q;
xd[q]:=x[j];
yd[q]:=y[j];
goto 4;
2:gdn[i,j]:=ll;m:=m+1;goto 5;
4:m:=0;
end;
end;
for i:=1 to es do begin
seek(MF,gdn[i,1]);read(MF,i1);seek(PS,gdn[i,1]);read(PS,j1);
seek(MF,gdn[i,2]);read(MF,i2);seek(PS,gdn[i,2]);read(PS,j2);
seek(MF,gdn[i,3]);read(MF,i3);seek(PS,gdn[i,3]);read(PS,j3);
if (i1=1) and (i2=1) and (j1=j2) then goto 6;
if (i2=1) and (i3=1) and (j2=j3) then goto 7;
if (i3=1) and (i1=1) and (j3=j1) then goto 8;
goto 9;
6:b1:=gdn[i,1];b2:=gdn[i,4];
goto 10;
7:b1:=gdn[i,2];b2:=gdn[i,5];
goto 10;
8:b1:=gdn[i,3];b2:=gdn[i,6];
10:j:=1;seek(MF,b2);write(MF,j);
seek(PS,b1);read(PS,j4);
seek(PS,b2);write(PS,j4);
9:end;
nb1:=0;
for i:=1 to np do begin
seek(MF,i);read(MF,i4);
if i4=1 then nb1:=nb1+1;
end;
writeln(d,np,' ',ne,' ',nb1,' ',sds1);
for i:=1 to np do
writeln(d,xd[i],' ',yd[i]);
for i:=1 to es do begin
writeln(d,gdn[i,1],' ',gdn[i,4],' ',gdn[i,6],' ',et[i]);
writeln(d,gdn[i,4],' ',gdn[i,2],' ',gdn[i,5],' ',et[i]);
writeln(d,gdn[i,5],' ',gdn[i,3],' ',gdn[i,6],' ',et[i]);
```



```
writeln(d,gdn[i,4],' ',gdn[i,5],' ',gdn[i,6],' ',et[i]);
end;
for i:=1 to np do begin
seek(MF,i);read(MF,i5);
if i5=1 then writeln(d,i);
end;
writeln(d,fy,' ',rlp);
close(d);
close(NB);erase(NB);
close(MF);erase(MF);
close(PS);erase(PS);
end;

procedure agciz;
begin
clrscr;
assign(e,g);reset(e);
readln(e,ds,es,pds,nc);
for i:=1 to ds do
readln(e,xd[i],yd[i]);
for i:=1 to es do begin
readln(e,gdn[i,1],gdn[i,2],gdn[i,3],a);
end;
close(e);
clrscr;
gd:=detect;
initgraph(gd,gm,'c:\tp\bgi');
if graphresult<>grok then
halt(1);
setfillstyle(1,white);
bar(0,0,getmaxx,getmaxy);
setcolor(0);
rectangle(0,0,getmaxx,getmaxy);
setcolor(2);
emag:=3000;xs:=150;ys:=-80;
for i:=1 to es do begin
n1:=gdn[i,1];n2:=gdn[i,2];n3:=gdn[i,3];
x1:=round(xd[n1]*emag+xs);x2:=round(xd[n2]*emag+xs);x3:=round(xd[n3]*emag+xs);
y1:=round(yd[n1]*emag+ys);y2:=round(yd[n2]*emag+ys);y3:=round(yd[n3]*emag+ys);
line(x1,(300-y1),x2,(300-y2));
line(x2,(300-y2),x3,(300-y3));
line(x3,(300-y3),x1,(300-y1));
end;
```



```
setcolor(7);
outtextxy(250,420,'SONLU ELEMAN AGI');
readln;
closegraph;
end;

procedure potciz;
var psii:array[1..sab] of real;
Label 1;
begin
clrscr;
assign(e,g);reset(e);
assign(d,cc);reset(d);
readln(e,ds,es,pds,nc);
for i:=1 to ds do
readln(e,xd[i],yd[i]);
for i:=1 to es do
readln(e,gdn[i,1],gdn[i,2],gdn[i,3],a);
close(e);
for i:=1 to ds do
readln(d,i,psii[i]);
close(d);
kon:=psii[1];
for i:=2 to ds do begin
if psii[i]>kon then kon:=psii[i];
end;
clrscr;
gd:=detect;
initgraph(gd,gm,'c:\tp\bgi');
if graphresult<>grok then
halt(1);
setfillstyle(1,white);
bar(0,0,getmaxx,getmaxy);
setcolor(0);
rectangle(0,0,getmaxx,getmaxy);
setcolor(2);
emag:=3000;xs:=150;ys:=-80;
for i:=1 to es do begin
for j:=1 to 3 do begin
a:=gdn[i,j];h[j]:=round(emag*xd[a]+xs);v[j]:=round(emag*yd[a]+ys);fi[j]:=psii[a];
end;
for ps1:=0 to 100 do begin
ps:=ps1*kon*0.025;
```

```
if (ps=fi[1]) and (ps=fi[2]) then begin line(h[1],(300-v[1]),h[2],(300-v[2]));
goto 1;end;
if (ps=fi[2]) and (ps=fi[3]) then begin line(h[2],(300-v[2]),h[3],(300-v[3]));
goto 1;end;
if (ps=fi[3]) and (ps=fi[1]) then begin line(h[3],(300-v[3]),h[1],(300-v[1]));
goto 1;end;
if (((ps>=fi[1]) and (ps<=fi[2])) or ((ps<=fi[1]) and (ps>=fi[2])))
and (((ps>=fi[2]) and (ps<=fi[3])) or ((ps<=fi[2]) and (ps>=fi[3]))) then
begin
x1:=ROUND(((fi[1]-ps)*(h[1]-h[2])-(fi[1]-fi[2])*h[1])/(fi[2]-fi[1]));
y1:=ROUND(((fi[1]-ps)*(v[1]-v[2])-(fi[1]-fi[2])*v[1])/(fi[2]-fi[1]));
x2:=ROUND(((fi[2]-ps)*(h[2]-h[3])-(fi[2]-fi[3])*h[2])/(fi[3]-fi[2]));
y2:=ROUND(((fi[2]-ps)*(v[2]-v[3])-(fi[2]-fi[3])*v[2])/(fi[3]-fi[2]));
line(x1,(300-y1),x2,(300-y2));goto 1;
end;
if (((ps>=fi[2]) and (ps<=fi[3])) or ((ps<=fi[2]) and (ps>=fi[3])))
and (((ps>=fi[3]) and (ps<=fi[1])) or ((ps<=fi[3]) and (ps>=fi[1]))) then
begin
x1:=ROUND(((fi[2]-ps)*(h[2]-h[3])-(fi[2]-fi[3])*h[2])/(fi[3]-fi[2]));
y1:=ROUND(((fi[2]-ps)*(v[2]-v[3])-(fi[2]-fi[3])*v[2])/(fi[3]-fi[2]));
x2:=ROUND(((fi[3]-ps)*(h[3]-h[1])-(fi[3]-fi[1])*h[3])/(fi[1]-fi[3]));
y2:=ROUND(((fi[3]-ps)*(v[3]-v[1])-(fi[3]-fi[1])*v[3])/(fi[1]-fi[3]));
line(x1,(300-y1),x2,(300-y2));goto 1;
end;
if (((ps>=fi[3]) and (ps<=fi[1])) or ((ps<=fi[3]) and (ps>=fi[1])))
and (((ps>=fi[1]) and (ps<=fi[2])) or ((ps<=fi[1]) and (ps>=fi[2]))) then
begin
x1:=ROUND(((fi[3]-ps)*(h[3]-h[1])-(fi[3]-fi[1])*h[3])/(fi[1]-fi[3]));
y1:=ROUND(((fi[3]-ps)*(v[3]-v[1])-(fi[3]-fi[1])*v[3])/(fi[1]-fi[3]));
x2:=ROUND(((fi[1]-ps)*(h[1]-h[2])-(fi[1]-fi[2])*h[1])/(fi[2]-fi[1]));
y2:=ROUND(((fi[1]-ps)*(v[1]-v[2])-(fi[1]-fi[2])*v[1])/(fi[2]-fi[1]));
line(x1,(300-y1),x2,(300-y2));goto 1;
end;
1:end;
end;
setcolor(7);
outtextxy(200,420,'ES VEKTOR POTANSIYEL CIZGILERI');
readln;
closegraph;
end;
{ANA PROGRAMIN BA□LANGICI}
begin
23:menubasi;
```

```
anamenu;
s;
case yyy of
6:begin
verimenu1;
giris;
ses;
goto 23;
end;
8: begin
verimenu2;
refinement;
ses;
goto 23;
end;
10:begin
verimenu3;
agciz;
ses;
goto 23;
end;
12:begin
verimenu4;
potciz;
ses;
goto 23;
end;
14:CLRSCR;end;
end.
```

ÖZGEÇMİŞ

Doğum Tarihi	:	23.11.1967	
Doğum Yeri	:	İstanbul	
Lise	:	1981-1984	Beşiktaş Atatürk Lisesi
Lisans	:	1984-1988	Yıldız Teknik Üniversitesi (Birincilikle mezuniyet)
Yüksek Lisans	:	1989-1991	Y.T.Ü. Fen Bilimleri Enstitüsü Elektrik Müh. Anabilim Dalı
Doktora	:	1991-1998	Y.T.Ü. Fen Bilimleri Enstitüsü Elektrik Müh. Anabilim Dalı
Çalıştığı Kurum	:	1988-Devam ediyor	Y.T.Ü. Elektrik Müh. Bölümü Elektrik Makinaları Anabilim Dalı