

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ



77719

AŞIRI İNCE YAPI ETKİLEŞİMLERİ DİKKATE
ALINARAK YAPILAN
ESR SPEKTRUM SİMÜLATÖRÜ

Mevlüt BULUT

YÜKSEK LİSANS TEZİ
FİZİK ANABİLİM DALI

Tez Danışmanı
Prof.Dr. BEKİR AKTAŞ

GEBZE
1998



Bu tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu 'nun ~~10.8.2008~~ tarih ve ~~98.23~~ sayılı kararıyla oluşturulan jüri tarafından Fizik.....Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

JÜRİ

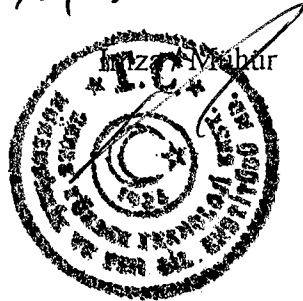
ÜYE
(Tez Danışmanı)

ÜYE

ÜYE

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 01.09.1998 tarih ve 98/24 sayılı kararı.



ÖNSÖZ

Teorik ve deneysel fizik çalışmalarında günümüzde geline nokta mevcut matematiksel araçlar kullanılarak yapılan analitik yaklaşımlar giderek daha karmaşık hale gelmekte ve bir türlü standartlaşamayan değişik gösterim teknikleri nedeniyle bu çalışmaların takibi de zorlaşmaktadır. Diğer yandan, yüksek işlem kapasiteli bilgisayarlar kullanılarak, gerçek problemlerin nümerik yöntemlerle çözülebilir hale gelmesi ve bu çözümlerden elde edilen sonuçların anlaşılması kolay görsel verilere dönüştürülebilmesi, nümerik çözüm sistemlerinin önemini artırmıştır. Bilgisayarların bilgi işleme ve aktarma yöntemlerinin gittikçe daha fazla insancillaşacağı (sesli iletişim, üç boyutlu algılama ve görüntüleme gibi...) düşünülürse, önümüzdeki yıllarda bu yöntemlerin öneminin daha da artacağı açıktır.

Benzer konularda nümerik çalışmalar dünyada da yapılmaktadır. Bu çalışmanın bu konuda ülkemizde yapılacak daha kapsamlı çalışmalara mütevazı bir katkı olması ümidiyle.

ÖZET

Bu çalışmanın amacı, EPR eğitiminde ve deneysel olarak elde edilen EPR spektrumlarının yorumlanması esnasında yardımcı olarak kullanılabilecek bir EPR spektrum simülatörü elde etmektir. Bunun için ilk aşamada Zeeman etkileşimine ilaveten, yalnızca aşırı ince yapı etkileşimleri dikkate alındı.

Simülatör yazılımı : etkileşimli kullanım araçlarının hazırlanmasında birtakım kolaylıklar sunan, nesne yönelimli Visual C++ programlama ortamında gerçekleştirildi. Program çalışır hale geldikten sonra, değişik spin durumları için elde edilen kuramsal spektrumlar, daha önce yapılmış ve ders kitaplarına geçmiş olan nümerik spectrumlarla karşılaştırılarak programın doğruluğu ve etkinliği test edildi. Uygulama olarak da; eldeki program kullanılarak, deneysel spektrumları literatürde bulunan ve aşırı ince etkileşimlerin başat etkiye sahip olduğu paramanyetik örnekler için, kuramsal spektrumlar elde edilerek deneysel olanlarla karşılaştırma yapıldı.

ABSTRACT

The purpose of this work was to realize an EPR spectrum simulator which can be used as a tool during the interpretation of experimental EPR spectra and/or during EPR study. In order to attain such a simulator, at first stage only the hyperfine interactions have been taken into account in addition to Zeeman interaction term.

The simulator software has been written using the Visual C++ which is a windows hosted development environment providing nice facilities to construct visual interface elements. After the spectrum program became working, its efficiency and flexibility have been tested by reproducing the previously obtained spectra for different hypothetical spin configurations. As a sample application of the software, the theoretical spectra for the paramagnetic species (for which hyperfine interactions are dominant in the formation of spectra and the experimental spectra of which are available in literature) have been obtained and checked against the experimental ones.

TEŞEKKÜR

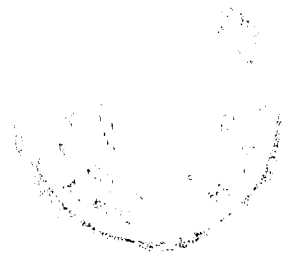
Daha öncesinde olduđu gibi bu çalışmam esnasında da her türlü bilgi ve tecrübesini benden esirgemeyen kıymetli hocam Prof. Dr. Bekir Aktaş'a, tezin yazımında emeđi geçen arkadaşım Fikret Yıldız'a, ayrıca geç vakitlere kadar çalışmamı sabır ve anlayışla karşılayan eşim Ayşegöl Bulut' a teşekkürü borç bilirim.





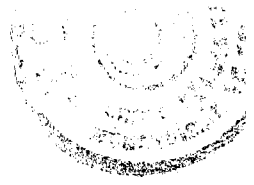
İÇİNDEKİLER

	Sayfa
ÖNSÖZ	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
İÇİNDEKİLER	v
ŞEKİL DİZİNİ	vi
SEMBOLLER VE KISALTMALAR	viii
1. GİRİŞ	1
2. MANYETİK REZONANSIN KURAMI	3
2.1. Genel Bilgiler	3
2.2. Zeeman Etkileşimleri	4
2.3. Manyetik Rezonans Geçişleri ve Deneysel Teknik	5
2.4. Aşırı İnce Yapı Etkileşimleri	7
2.5. Diğer Etkileşimler ve Enerjiye Katkı Mertebeleri	10
3. NÜMERİK MODELLEME	12
3.1 Algoritma	12
3.2 Program Kodu	14
3.3 Doğruluk Testleri	27
4. UYGULAMALAR	29
SONSÖZ	39
KAYNAKLAR	40
ÖZGEÇMİŞ	41



ŞEKİL DİZİNİ

Şekil 2.1.1 B dış manyetik alanı içerisindeki elektronun, toplam spini $I = \frac{1}{2}$ olan tek bir çekirdekle etkileşmesi sonucu ortaya çıkan enerji seviyeleri.	4
Şekil 2.2.1 B dış manyetik alanı içerisinde bulunan $\frac{1}{2}$ spinli bir parçacığın enerji diyagramı.	5
Şekil 2.3.1 Modern bir EPR spektrometresinin genel görünüşü.	6
Şekil 2.4.1 Klasik olarak düşünülen iki manyetik dipolün kuvvetli bir dış manyetik alan içerisindeki etkileşimleri.	8
Şekil 2.4.2 Çeşitli orbitallerin radyal değişimleri.	9
Şekil 3.1.1 Bilgilerin girildiği diyalog penceresinin görünümü.	13
Şekil 3.3.1 Bir elektronun spin quantum sayısı $\frac{1}{2}, 1, 3/2, 2, 5/2, 3, 7/2, 4$ olan çekirdeklerle yaptığı ince yapı etkileşmeleri sonucunda oluşan kuramsal spektrumlar.	28
Şekil 4.1 X ışınlarına maruz bırakılan CaF_2 de yerdeğiştiren safsızlık hidrojen atomlarının, oda sıcaklığında verdiği X-Band EPR spectrumu.	29
Şekil 4.2 173 kelvinde, tetrahidrofuran ve dimetiloksitan ile hazırlanan, 2:1 oranındaki çözelti içerisinde bulunan benzen anyon radikalinin X-band EPR spektrumu.	30
Şekil 4.3 İnce yapı etkileşimi açısından özdeş olmayan iki protona örnek olarak; 298 K de hazırlanan sulu çözeltideki glikolik asit radikalinin (HOCHCOOH) X-band EPR spektrumu.	31



Şekil 4.4 299 kelvindeki metanol içerisinde bulunan CH_2OH radikalinin X-band EPR spektrumu.	32
Şekil 4.5 195 K deki sıvı NH_3 'ün elektrolizi ile elde edilen 1,3-bütadiyen anyon radikalinin X-band EPR spektrumu.	33
Şekil 4.6 298 kelvinde bulunan dimetiloksitan 'daki naftalin anyon radikalinin X-band spektrumu. Deneyisel spektrumdaki herbir çizginin üzerinde görülen sayılar, bu çizginin karşı geldiği spin durumu için dejenerasyon sayısını ifade eder.	34
Şekil 4.7 295 K daki dimetiloksitan içerisinde bulunan antrasin anyon radikalinin X-band EPR spektrumunun düşük alan kısmını gösteren deneyisel ve kuramsal spektrumlar.	35
Şekil 4.8 295 K da benzen içerisine yerleştirilen bir nitronilnitroksit radikalinin X-band EPR spektrumu.	36
Şekil 4.9 a) Na^+ nın karşıt iyon olarak bulunduğu 297 K deki dimetiloksitan içerisinde bulunan parazin anyon radikalinin X-band EPR spektrumu; b) Karşıt iyon olarak Na^+ yerine K^+ kullanılarak elde edilen X-band EPR spektrumu.	37
Şekil 4.10 228 K deki tolüen çözeltisinde bulunan $(\text{p-NHCOCH}_3)_1\text{TPP}$ $\text{Co}(\text{NMeIm})\text{O}_2$ 'nin X-band EPR spektrumu. Burada TPP, tetrafenilporipirin 'i, MeIm ise metilimidazol 'ü göstermektedir.	38

SEMBOLLER VE KISALTMALAR

- h : Planc sabiti = $6.6260755 \text{ e-34 Js}$
- $\hbar$: $h/2\pi = 1.05457266 \text{ e-34 Js}$
- β : $\frac{|e|\hbar}{2m_e} = 9.2740154 \text{ e-24 JT}^{-1}$
- μ : Manyetik Moment
- γ : Magnetogyric Oran
- ℓ_i : Açısal Momentumun I yönündeki bileşeni
- B: Dış manyetik alan şiddeti
- B_y : Yerel manyetik alan şiddeti
- M_I : Çekirdek spininin z yönündeki açısal momentumunu temsil eden $\hat{I}_z$ operatörünün öz değeri
- a: Aşırı ince yapı etkileşim sabiti
- I: Çekirdek spin kuantum sayısı
- H: Hamiltoniyen
- g: Lande g çarpanı
- $\hat{S}_z$: z yönündeki açısal momentum bileşenini gösteren operatör.
- U: Enerji
- r_b : Bohr yarıçapı = $5.29177249 \text{ e-11 m}$
- ψ : Dalga fonksiyonu
- Z_n : Çekirdek yükü
- e: Elektronun elektrik yükü = $-1.60217733 \text{ e-19 C}$
- Q: Noktasal yük şiddeti
- X-band EPR: X ışınları bölgesinde sabit frekans mikrodalga üretici kullanılarak yapılan EPR
- mT: Mili tesla
- K: Kelvin

1. GİRİŞ

Uygun şartlar sağlandığında, yalnızca enerjileri belirli bir değerde olan parçacık veya quasi parçacıkların soğurulmasına yada yayımlanmasına neden olan fiziksel olaylar, spektroskopik inceleme tekniklerinin temelini oluştururlar. Örneğin : X-Ray spektroskopisi; kristal yapıların örgü sabitlerine bağlı olarak belirli dalga boylarındaki fotonları yansıtmasına, Magnetic Rezonans spektroskopisi ise sabit manyetik alan içerisine konan bir malzemenin nükleer, elektronik ve kimyasal özelliklerine bağlı olarak belirli frekanslardaki fotonları soğurmasına dayanan spektroskopik tekniklerdir. Genel olarak X-Ray (nötron, elektron... scattering), Magnetik Rezonans, Mass, akustik, optik spektroskopi ve bunların ikili hatta üçlü olarak birlikte kullanıldığı analiz sistemleri bilinmektedir. Spektroskopik tekniklerle rezonans noktaları ve bu noktalardaki rezonansların bağl şiddetleri ölçülür. Rezonans noktalarının bulunmasında mevcut teknolojilerin elverişliliğine bağlı olarak şu iki farklı yaklaşımdan biri tercih edilir. Ya belirli enerjilere sahip parçacıklar üretilip, ortam şartlarını oluşturan parametreler sistemli bir şekilde değiştirilerek rezonansın gözlemlendiği parametre değerleri aranır, ya da ortam şartları sabit tutulup mümkün olduğunca geniş bir enerji yelpazesinde parçacıklar üretilerek rezonansın meydana geldiği enerji değerleri araştırılır.

Magnetik rezonansın çıkış noktası, Stern-Gerlach [1] (1920'lerde) deneyinin sonucunda anlaşılan; bir manyetik alan içerisindeki elektronların yalnızca iki farklı yönelimden birine sahip olabileceği gerçeğidir. Elektronların sahip olduğu manyetik momentlerin dış manyetik alan doğrultusundaki bileşenlerinin değeri $+\hbar/2$ veya $-\hbar/2$ değerlerinden birini almak zorundadır. Bu nedenle bir manyetik alan içerisinde bulunan elektronlar (manyetik momentin, dış manyetik alan yönündeki bileşeninin alacağı değere bağlı olarak) iki farklı enerji seviyesinden birinde bulunurlar. Herbir enerji düzeyindeki elektronların sayısı, elektronlar fermiyon olduklarından dolayı "Boltzman dağılımı" na göre belirlenir. Elektronların, uygun enerjide foton salma veya soğurma yoluyla bu farklı seviyelerin birinden diğerine geçmeleri mümkündür. Dolayısıyla dışarıdan uygun enerjili fotonlar verilerek, elektronların alt enerji seviyesinden üst enerji seviyesine pompalanabileceği düşünülmüş ve bu olayı deneysel olarak ilk defa Rabi ve arkadaşları [2] gözlemiştir. Zavoisky [3] nin 1945 yılında $\text{CuCl}_2 \cdot 2\text{H}_2\text{O}$ örneğinde tesbit ettiği radyo frekansındaki soğurma çizgisi ise,

gözlenen ilk Elektron Paramanyetik Rezonans çizgisidir. Paramanyetik Rezonansın 1946 dan sonraki gelişimi, ikinci dünya savaşını takiben, tam teçhizatlı mikrodalga sistemlerinin her yerde rahatlıkla bulunabilir hale gelmesiyle bir hayli hız kazanmıştır. Örneğin: radar sistemlerinde kullanılan 9 GHz bölgesinde ki mikrodalga üreteçleri, parçalarının kolayca ve de ucuza bulunabilir olmasından dolayı, sabit frekans düzeneklerinde standart olarak kullanılır hale gelmişlerdir. Magnetic Rezonansın ilk dönemlerinin tarihi Ramsey [4] tarafından özetlenmiştir.

Magnetik rezonans teknikleri, diğer spektroskopi tekniklerine göre oldukça yeni olmalarına karşın; fizik, kimya, biyoloji, tıp gibi birbirinden farklı konularda yaygın kullanım alanları bulmuştur. Farklı magnetik rezonans tekniklerine kısaca göz atacak olursak:

EPR (veya ESR): (Elektron Paramagnetic Resonance) Malzemenin içerisindeki elektronların, harici ve yerel manyetik alanların etkisi altında elektromanyetik radyasyonla etkileşimlerini inceler.

NMR: (Nükleer Magnetic Resonance) Manyetik alan içerisindeki çekirdeklerin elektromanyetik radyasyonla etkileşimlerini inceler.

Çifte Rezonans Teknikleri:

ENDOR: (Electron-Nuclear Double Resonance) Doyuma götürülen NMR gücünün, aynı anda cereyan eden ESR geçişi üzerindeki etkisini inceler.

DNP : (Dynamic Nuclear Polarization) Doyuma götürülen ESR gücünün, aynı anda cereyan eden NMR geçişi üzerindeki etkisini inceler.

DoubleENDOR: Çift ENDOR uygulaması.

ELDOR: (Electron-Electron Double Resonance) Aynı malzeme üzerinde biri doyumaya götürülen ardışık iki ESR geçişini inceler.

NNMR : (Nuclear-Nuclear Double Resonance) Aynı malzeme üzerinde iki ardışık NMR geçişini inceler.

Burada yapılan nümerik çalışmanın amacı; deneysel olarak elde edilen EPR spektrumlarının yorumlanması esnasında ve/veya EPR eğitiminde yardımcı olarak kullanılabilecek bir EPR spektrum simülatörü elde etmektir. Böyle bir simülatör işlev olarak: İncelenmekte olan numuneden elde edilen deneysel spektrum ile, bu numune için öngörülen yapının vereceği kuramsal spektrumun karşılaştırılmasına imkan verir.

2. MANYETİK REZONANSIN KURAMI

2.1 Genel Bilgiler

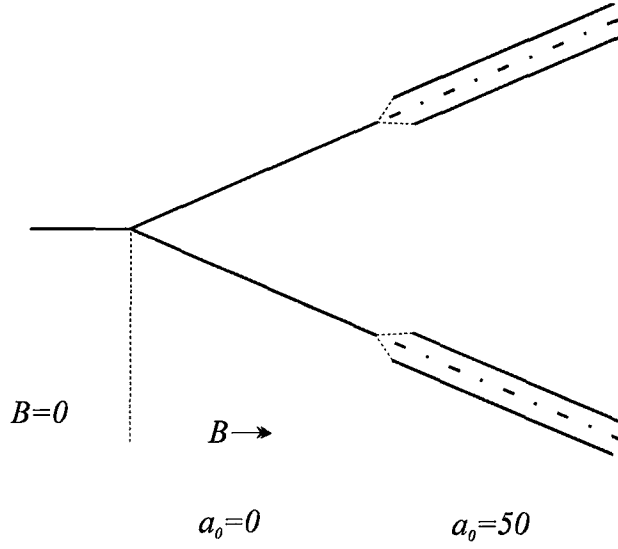
Şimdiye kadar yapılan deneysel fizik çalışmalarının hiç birinde manyetik monopol' e rastlanmamıştır. Bilinen bütün manyetik parçacıklar manyetik dipol davranışı gösterirler. Paramanyetik yapılarda manyetik rezonans incelenirken, dikkate alınması gereken ilk manyetik etkileşim, Zeeman etkileşimidir. İkincil olarak; elektronun, etrafında bulunan çekirdeklerin spin manyetik dipolleri ile yaptığı etkileşimdir ki, bu da aşırı ince yapı etkileşimi olarak bilinir. Bazı elementlerin çekirdekleri sıfır toplam spin 'e sahip olmakla beraber, elementlerin çoğunun çekirdek spinleri sıfırdan farklıdır veya çekirdek spinleri sıfırdan farklı olan izotopları vardır. I ile gösterilen, çekirdeklerin spin kuantum sayısı $1/2, 1, 3/2, 2, \dots$ gibi $1/2$ nin tam katları şeklinde pozitif değerler alır. I 'nın dış manyetik alan doğrultusundaki bileşeni olan M_I ise, $2I + 1$ farklı değerden birini alır. Elektronların olduğu gibi çekirdeklerin de spin açısal momentlerine ilişkin

$$\mu_i = \gamma \cdot \ell_i \quad (2.1)$$

şeklinde bir manyetik momentleri vardır. Dolayısıyla dış manyetik alan doğrultusuna göre iki farklı yönelimden birinde bulunan elektron, (aynı doğrultuya göre $2I + 1$ adet olası yönelimden birinde bulunan) "çekirdek manyetik momentin" den dolayı, dış alana ilaveten yerel bir manyetik alan görür. Bu durumda elektronun gördüğü toplam manyetik alan:

$$B' = B + aM_I + \dots + a^q/B^{q-1} \quad (2.2)$$

olur. Burada a , aşırı ince yapı etkileşim parametresini göstermek üzere aM_I , elektronun hissettiği yerel manyetik alanı, B ise dış manyetik alanı göstermektedir. Yerel katkıların dış manyetik alana göre dikkate değer bir büyüklüğe sahip olduğu durumlarda, elektronun enerji seviyeleri hesaplanırken ikinci ve daha yüksek mertebeden pertürbasyon terimlerinde hesaba katılması gerekir. Bu gibi durumlarda elektronun hissedeceği yerel alan değerinin ifadesinde, " a^q/B^{q-1} " şeklinde a' ya göre ikinci ve daha yüksek mertebeden terimler de bulunacaktır.



Şekil 2.1.1 B dış manyetik alanı içerisindeki elektronun, toplam spini $I = 1/2$ olan tek bir çekirdekle etkileşmesi sonucu ortaya çıkan enerji seviyeleri.

Çekirdeklerin neden olduğu aşırı ince yapı etkileşimleri, elektron Zeeman enerji seviyelerinin herbirini ikiye yarar. B dış manyetik alanı içerisindeki elektronun, toplam spini $I = 1/2$ olan tek bir çekirdekle etkileşmesi sonucu ortaya çıkan enerji seviyeleri şekil 2.1.1 de görülmektedir.

2.2 Zeeman Etkileşimi:

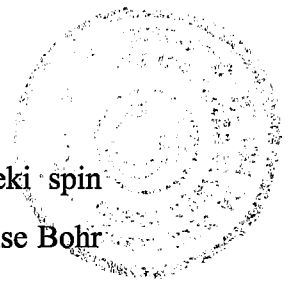
Zeeman 1896'da sodyum alevi ile spektrum elde ederken, magnetik alan içerisine konulan sodyum alevinde spektrum genişlemesi gözlemiş, daha sonraki ayrıntılı incelemelerle spektrum çizgilerinde genişleme şeklinde gözüken olayın, gerçekte bazı enerji seviyelerinin yarılmasından ibaret olduğu gösterilmiştir.

Magnetik momenti olan bir parçacık (elektron veya çekirdek) dış magnetik alan içerisine konulduğunda, Zeeman Hamiltoniyeni

$$\mathbf{H} = - \mathbf{B} \cdot \boldsymbol{\mu} \quad (2.3)$$

Dış alan doğrultusu z eksenini olarak alınırsa Hamiltoniyen,

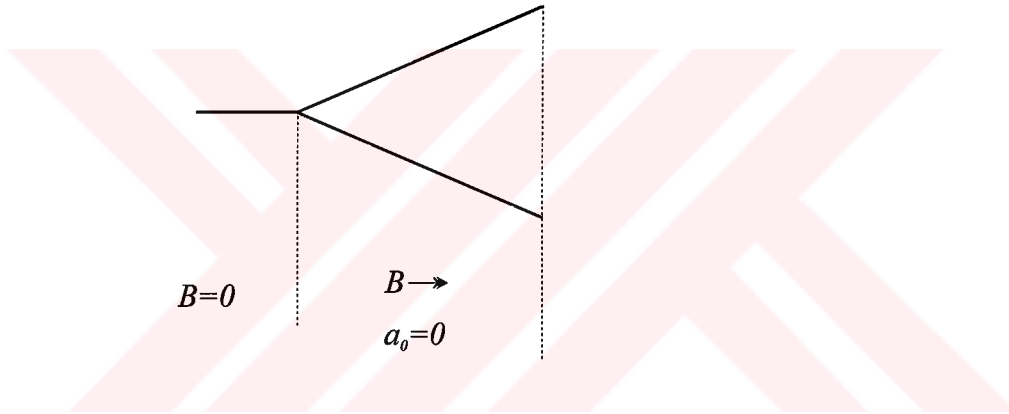
$$\mathbf{H} = - B \mu_z \quad (2.4)$$



şeklinde yazılabilir. Elektron magnetik momentinin $\mu_{ez} = -g\beta_e S_z$ biçimindeki spin açısal momentum bağıntısı kullanılırsa: g, elektron için Lande çarpanını, β_e ise Bohr magnetonunu göstermek üzere Hamiltoniyen,

$$H = g\beta_e B S_z \quad (2.5)$$

olur. Elektron spin açısal momentumunun z bileşeni için kuantum sayısı $S_z = \pm \frac{1}{2}$ olduğundan, magnetik alan içerisindeki elektronun enerjisi $\pm \frac{1}{2}g\beta_e B$ değerlerinden birini alabilir. Magnetik alan içerisine konulan $\frac{1}{2}$ spinli bir parçacığın enerji şeması şekil 2.2.1 de gösterilmiştir. 1990 yılında, magnetik ve elektrik alanların uygun bir konfigürasyonu ile uzayda hapsedilen tek bir elektronun verdiği EPR sinyali gözlemlenmiştir [5].

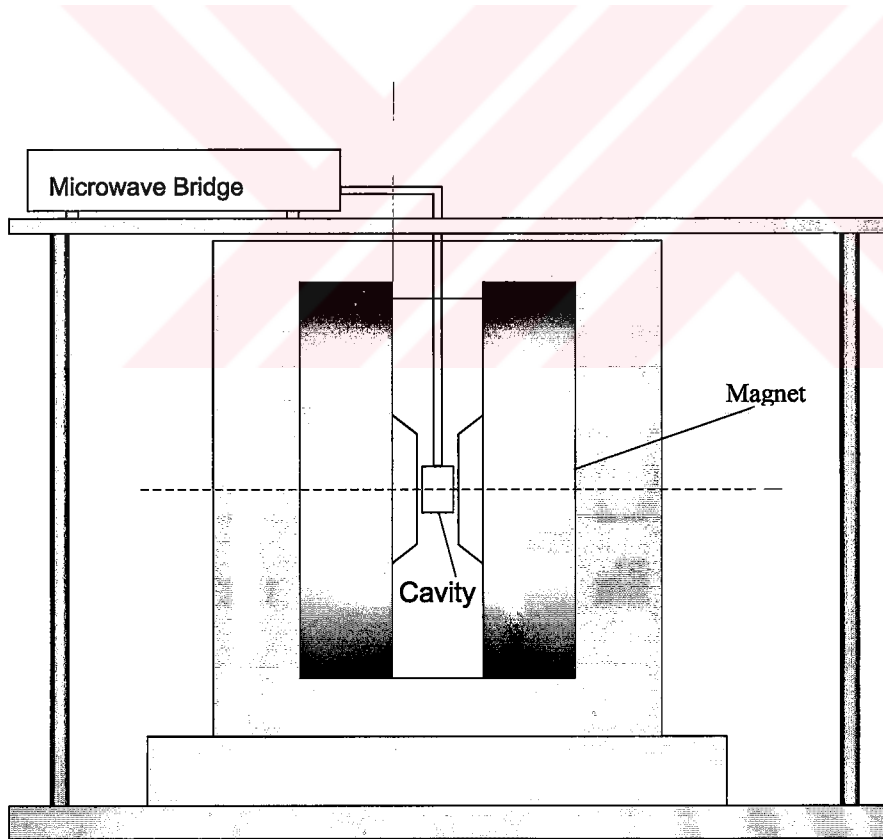


Şekil 2.2.1 B dış manyetik alanı içerisinde bulunan $\frac{1}{2}$ spinli bir parçacığın enerji diyagramı.

2.3 Manyetik Rezonans Geçişleri ve Deneysel Teknik

Elektronların, içinde bulundukları manyetik alanın değerine bağlı olarak, uygun enerjili fotonlar soğurmak suretiyle düşük enerjili spin yöneliminden yüksek enerjili spin yönelimine geçebileceği anlaşıldıktan sonra, bu olayın kontrollü olarak gerçekleştirilebilmesi ve geçiş noktalarındaki parametre değerlerinin net bir şekilde tesbit edilebilmesi için çeşitli deneysel teknikler geliştirilmiştir. En basit haliyle bir manyetik spektrometre, mikrodalga kaynağı, mıknatıs, örnek ve dedektörden oluşur. Daha önceden de bahsedildiği gibi, ikinci dünya savaşından sonraki gelişmelerle, sabit frekans mikrodalga üreteçleri hazır bir şekilde kolayca ve de uygun fiyata

bulunabilir hale gelmişler, bu nedenle de manyetik rezonans deney setlerinde standart bir kullanım alanı bulmuşlardır. Günümüzde kullanılan modern EPR düzeneklerinde, elektromanyetik ışıyım kaynağı ve dedektör birlikte “Microwave Bridge” denilen bir kutu içerisinde bulunurlar. Bir deney düzeneğinde birden fazla Microwave Bridge bulunabilir. Sabit frekans mikrodalga ışıyımı altında rezonansın gözleneceğı elektronik enerji seviyelerinin ayarlanması için kullanılan mıknatıs ise, 0’den birkaç tesla (süperiletkenlerin kullanımıyla onlarca tesla) ‘ya kadar statik manyetik alan oluşturabilen bir elektromıknatıstır. Numune, örnekten gelen zayıf sinyallerin büyütülmesine yardımcı olan metal bir kutu (microwave cavity) içerisine yerleştirilir. Ayrıca iyi bir spektrum elde etmek amacıyla aksamaların koordinasyonu ve elde edilen verilerin incelenmesi için, sinyal işleme, elektronik kontrol devresi ve bilgisayardan oluşan bir konsol ünitesi de bulunabilir. Modern bir EPR spektrometresinin genel görünüşü şekil 2.3.1 de verilmektedir.



Şekil 2.3.1: Modern bir EPR spektrometresinin genel görünüşü

EPR deney setlerinde kullanıcı tarafından kontrol ve kumanda edilebilen parametrelerin belli başlı olanlarını sayacak olursak:

Mikrodalga kaynak frekansının ince ayarı, Bios Level, Attenuation: Sabit frekans mikrodalga şiddeti, Receiver Gain: Dedektör kazancı, Tarama alanının alt ve üst sınırları, Modülasyon genliği ve modülasyon frekansı.

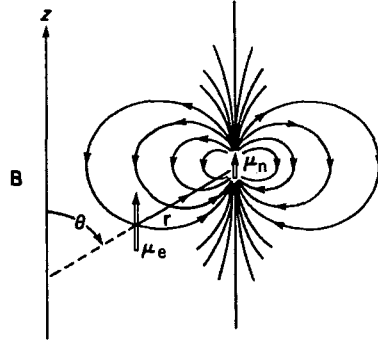
Modern EPR spektrometrelerinde, gürültülerin elimine edilmesi, ve spektrometre hassasiyetinin artırılması maksadıyla; numunenin soğurduğu mikrodalga şiddeti yerine, mikrodalga soğurma şiddetinin manyetik alana göre türevi ölçülür. Dolayısıyla, elde edilen manyetik alan-soğurma spektrumları, gerçekte soğurma eğrisinin manyetik alana göre birinci türevidir.

2.4 Aşırı İnce Yapı Etkileşimleri

z-eksenine paralel olarak uygulanan yeterince büyük, statik bir manyetik alan içerisinde bulunan elektron ve çekirdek manyetik dipolleri, klasik olarak etkileşecek olsalardı ilk konumları ne olursa olsun derhal dış manyetik alanla aynı doğrultuyu gösterecek şekilde yönlenmeye çalışacak ve bu durumda aralarındaki dipol - dipol etkileşiminin enerjisi de yaklaşık olarak

$$U_{dipolar} = -\frac{\mu_0}{4\pi} \frac{3 \cos^2 \theta - 1}{r^3} \mu_{nz} \mu_{ez} = -B_y \mu_{ez} \quad (2.6)$$

şeklinde olacaktı. Şekil 2.4.1 de şematik olarak gösterilen bu klasik sistemde elektronun, çekirdeğin varlığından dolayı hissedeceği yerel manyetik alan B_y , denklem 2.6 dan da görüldüğü gibi θ 'nın ve r 'nin anlık değerlerine bağlıdır ve r sıfıra yaklaşırken, etkileşim enerjisi sonsuza gitmektedir. Boyutları çok küçük olan bu parçacıkların üst üste gelme olasılıkları son derece az olduğundan, bu ıraksama çoğunlukla ciddi bir soruna neden olmaz. Bu çelişkili durumun matematiksel ayrıntıları Skinner ve Weil [6] tarafından irdelenmiştir.

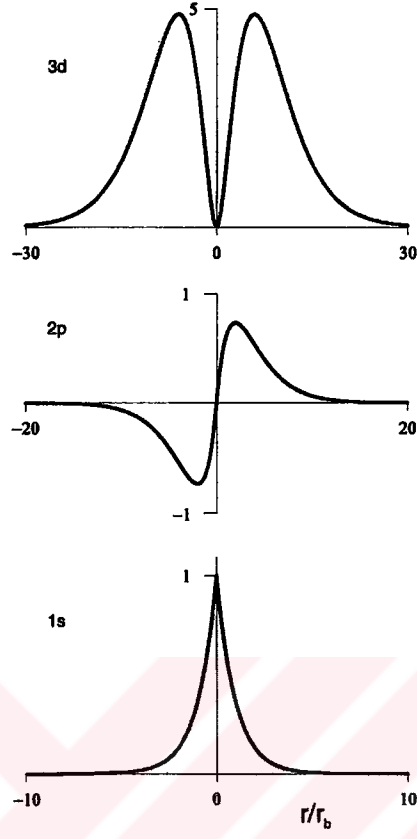


Şekil 2.4.1 Klasik olarak düşünülen iki manyetik dipolün kuvvetli bir dış manyetik alan içerisindeki etkileşimleri.

Elektron uzayda sürekli olarak bir yerde durmadığı için etkileşim enerjisi, elektronun olasılık dağılım fonksiyonu üzerinden ortalama alınarak hesaplanmalıdır. Eğer olasılık dağılımı θ dan bağımsız ise (s orbitallerinde olduğu gibi), $\cos^2\theta$ nın küre üzerindeki beklenen değeri denklem 2.6 da kullanılarak, yerel alanın r ' ye göre değişim ifadesi elde edilir.

$$\langle \cos^2 \theta \rangle = \frac{\int_0^{2\pi} \int_0^\pi \cos^2 \theta \cdot \sin \theta \cdot d\theta \cdot d\phi}{\int_0^{2\pi} \int_0^\pi \sin \theta \cdot d\theta \cdot d\phi} = \frac{1}{3} \quad (2.7)$$

$\langle \cos^2 \theta \rangle = 1/3$ olduğundan, denklem 2.6 deki B_y sıfır çıkar. Dolayısıyla, küresel simetrik 1s orbitalinde eşlenmemiş bir elektronu olan hidrojen atomunun gösterdiği aşırı ince yapı etkileşimlerinin kökeni, klasik olarak düşünülen uzaktan-uzağa dipolar etkileşim olamaz. Hidrojen atomundaki aşırı ince yapı etkileşimlerinin gerçek nedeni, 1s orbitalinin radyal bağımlılığı incelenerek anlaşılabilir. Şekil 2.4.2 (non-relativistik çözüm) den de görüldüğü gibi 1s orbitalinin, matematiksel bir nokta olarak alınan çekirdek üzerindeki yoğunluğu sıfırdan farklıdır. Bu noktada meydana gelen temas etkileşimi, genel olarak s orbitallerinde bulunan eşlenmemiş elektronların neden olduğu aşırı ince yapı etkileşimlerinin temelini oluşturur.



Şekil 2.4.2 Çeşitli orbitallerin radyal değişimleri.

S orbitallerinin küresel simetrilerinden dolayı, bu durumda meydana gelen aşırı ince yapı etkileşimleri de izotropiktir. Fermi [7], yeterince büyük dış manyetik alan ($B \parallel z$) içerisindeki bir elektronun, çekirdekle yapmış olduğu izotropik etkileşim enerjisinin yaklaşık olarak

$$U_{iso} = -\frac{2\mu_0}{3} |\psi(0)|^2 \mu_{ez} \mu_{nz} \quad (2.8)$$

ile verilebileceğini göstermiştir. Burada $\psi(0)$, elektron dalga fonksiyonunun çekirdek üzerindeki değerini göstermektedir.

2.5 Diğer Etkileşimler ve Enerjiye Katkı Mertebeleri

Sipini sıfırdan farklı olan çekirdek ve eşlenmemiş elektron barındıran bir atomun veya radikalın enerjisi Hamiltoniyen operatörünün terimleriyle ifade edilebilir.

$$H = H_{el} + H_{CF} + H_{LS} + H_{SS} + H_{Ze} + H_{HF} + H_{Zn} + H_{II} + H_Q \quad (2.9)$$

Bu Hamiltoniyendeki ilk üç terim, atomik Hamiltoniyeni, kalan altı terim de spin hamiltoniyenini oluşturur. Bu terimleri sırasıyla ele alacak olursak:

H_{el} -Elektronik Hamiltoniyen: her bir elektrona ait kinetik enerji, potansiyel enerji ve diğer elektronlardan gelen itme potansiyel enerjisinin toplamından oluşur.

$$H_{el} = \sum_i \frac{P_i^2}{2m} - \sum_{i,j} \frac{Z_n e^2}{r_{ni}} + \sum_{i>j} \frac{e^2}{r_{i,j}} \quad (2.10)$$

n, çekirdekleri gösteren, i ve j ise elektronları gösteren indislerdir. Elektronik enerji dalga boyu türünden ifade edilecek olursa: $10^{-4} - 10^{-5}$ cm civarındadır. Bu enerji aralığı spektrumun görünen ışık bölgesini kapsamaktadır. Zeeman enerjisiyle kıyaslandığında, ondan kat kat büyüktür

H_{CF} -Kristal alan Hamiltoniyeni : İyonik bileşiklerdeki iyonların elektrostatik yüklerinden ve covalent bileşiklerdeki kimyasal bağlardan dolayı var olan bir enerjidir. H_{el} 'nin ötelenmesine veya yarılmasına neden olur. Temelde Q_i iyonik yükleri, i ise elektronları göstermek üzere

$$V = \sum_{i,j} \frac{Q_i}{r_{i,j}} \quad (2.11)$$

şeklindeki elektrik potansiyelden kaynaklanan bir Stark etkisidir. Bu potansiyel noktasal yük yaklaşımı altında ve yalnızca en yakın komşu ligantlar dikkate alınarak yazılmıştır. Daha gerçekçi bir potansiyel, yük dağılımları üzerinden integraller içerir.

H_{LS} -Spin_Orbit Etkileşim Hamiltoniyeni: λ , spin orbit eşlenim parametresini, $\vec{L}$ ve $\vec{S}$ de sırayla orbital ve spin açısal momentlerini göstermek üzere:

$$H_{LS} = \lambda \vec{L} \cdot \vec{S} \quad (2.12)$$

şeklindedir. Spin_Orbit enerjisi H_{LS} , 10^{-2} cm mertebesinde ve ilk geçiş serisi elementlerinin tipik kristal alan yarılmalarından çok daha küçük, rare earth'ler için olan kristal alan yarılmalarından ise büyüktür.

H_{SS} -Spin_Spin Etkileşim Hamiltoniyeni:

$$H_{SS} = D \left[S_z^2 - \frac{1}{3} s(s+1) \right] + E(S_x^2 - S_y^2) \quad (2.13)$$

şeklindedir. Spektrumun güçlü bir açısal bağımlılık gösterdiği durumlarda Elektronik Zeeman enerjisiyle hemen hemen aynı mertebededir ($0 - 10^{-1}$ cm). Eksenel simetrisinin bulunduğu durumlarda $E=0$ dır.

H_{Ze} -Elektronik Zeeman Enerjisi:

$$H_{Ze} = \beta \vec{H} \cdot (\vec{L} + 2\vec{S}) = \beta \vec{S} \cdot \vec{g} \cdot \vec{H} \quad (2.14)$$

3.3 cm (x bandı - 10^{-10} Hz) mertebesinde ve bazan anizotropiktir.

H_{Zn} -Nükleer Zeeman Enerjisi :

$$H_{Zn} = - \sum_i g_{ni} \beta_n \vec{H} \cdot \vec{I}_i \quad (2.15)$$

Elektronik Zeeman enerjisinin binde biri mertebesinde ve genellikle izotropiktir.

H_{HF} -Aşırı İnce Yapı Enerjisi:

$$H_{HF} = \vec{S} \sum_i \vec{A}_i \cdot \vec{I}_i \quad (2.16)$$

H_{II} -Nükleer Spin-Spin Etkileşim Hamiltoniyeni:

$$H_{II} = \sum_{i>j} \vec{I}_i \cdot \vec{J}_{ij} \cdot \vec{I}_j \quad (2.17)$$

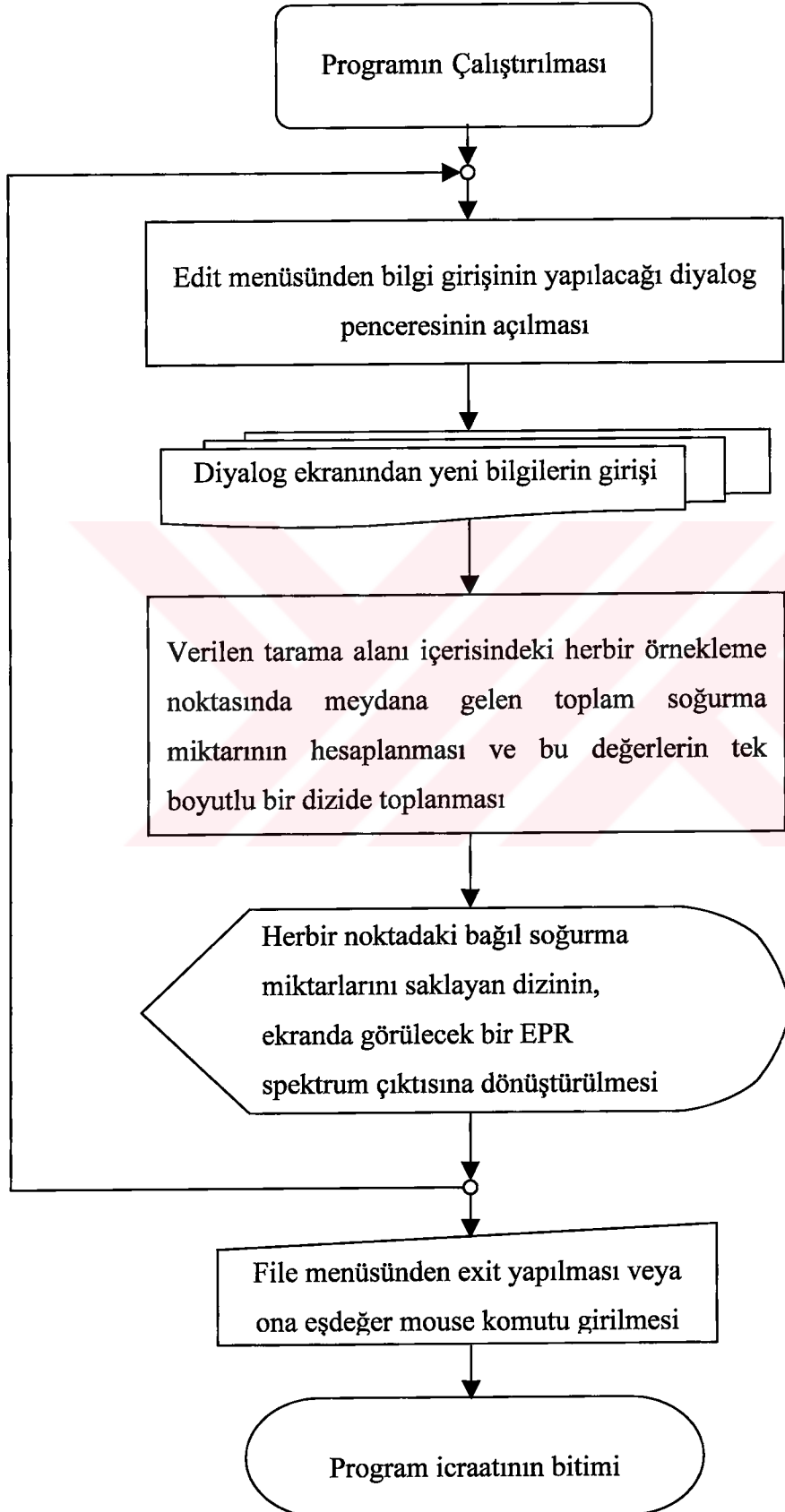
H_Q -Quadrupolar Enerji : Q quadrupol momenti, $\partial^2 V / \partial z^2$ de alan gradyentini göstermek üzere

$$H_Q = \frac{e^2 Q}{4I(2I-1)} \left(\frac{\partial^2 V}{\partial z^2} \right) [3I_z^2 - I(I+1) + \eta(I_x^2 - I_y^2)] \quad (2.18)$$

Burada η , asimetri parametresini göstermektedir ve $\eta = \frac{\partial^2 V / \partial x^2 - \partial^2 V / \partial y^2}{\partial^2 V / \partial z^2}$ şeklindedir. Eksenel simetrisinin var olduğu durumlarda $\eta=0$ dır.

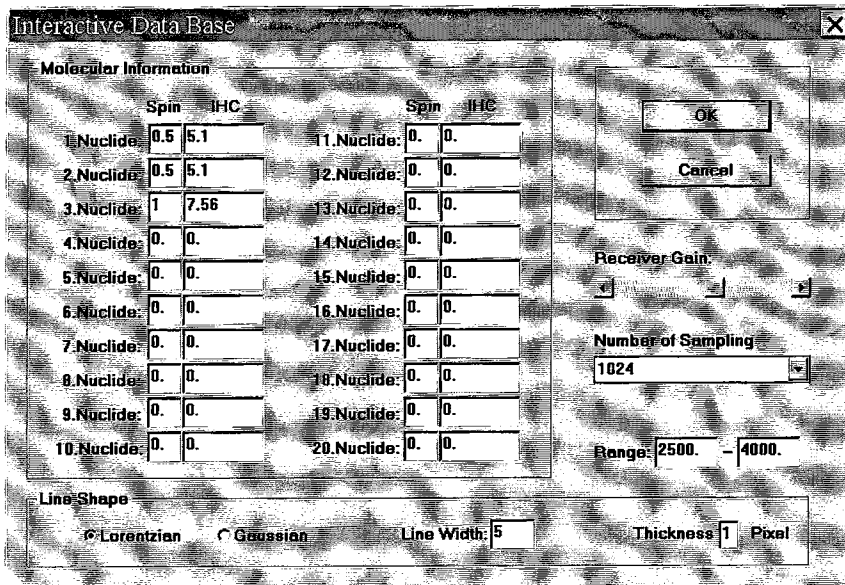
3. NÜMERİK MODELLEME

3.1 Algoritma



Yukarıda algoritması verilen programın, Visual C++ 2.0 kullanılarak hazırlanışı:

Visual C++ editörü açılarak, file menüsünden 'new' seçilir. Açılan diyalog penceresinden 'Project' seçeneği onaylanarak, yeni proje için bir isim girilmesini isteyen ikinci diyaloga geçilir. Burada isim olarak 'spectra' yazılıp 'create' tuşu onaylanır. Daha sonra gelen diyalog kutularında hali hazırda işaretli bulunan seçeneklerin hepsi kabul edilerek, en son diyalog penceresine gelinir. Burada 'finish' düğmesi tıklanarak, minimum konfigürasyonda bir windows uygulaması için gerekli olan tüm alt dosyalar oluşturulur. Yeni oluşturulan dosyaların listesi editör ekranında görülecektir. Bu dosyalardan spectra.rc isimli olanı çift tıklanarak, programın kullandığı windows öğelerine ait grafik dosyaların listesi açılır. Menü maddesinin alt öğelerinden olan IDR_spectrtype çift tıklanırsa menülerin özelleştirilmesine izin veren grafik tabanlı bir editör açılacaktır. Edit menüsünde default olarak bulunan, fakat ilk etapta kullanılmayacak olan parçalar çıkarılıp, yeni bir öğe olarak 'Interactive Menu' eklenir. Grafik editörden save yapılarak çıkılır. Resource menüsünden New seçilir ve açılan pencerede 'dialog' işaretlenerek OK tuşuna basılırsa, diyalog pencerelerinin özelleştirilebildiği yeni bir editör açılır. Burada başlangıçta yalnızca OK ve Cancel tuşlarını içeren boş bir diyalog çerçevesi görülecektir. Bu diyalog çerçevesi, yeni diyalog parçaları ve yazılar oluşturmak için, editörün sunduğu imkanlardan faydalanılarak şekil 3.1.1 deki hale getirilir.



Şekil 3.1.1 Bilgilerin girildiği diyalog penceresinin görünümü.

Daha sonra buradaki diyalog alanları birtakım deęiřkenlerle irtibatlandırılmak üzere ClassWizard açılır. Bu diyalogun çalışması için gerekli olan C++ kodu, “spectrum” ismiyle, CDialog Class' a mensup bir nesne tanımlanarak oluşturulur. Burada verilen program kodlarıyla uyumluluk göz önünde bulundurulmak suretiyle, yeni deęiřken isimleri diyalog alanlarla ilişkilendirilir. Burada dökümleri verilen .cpp uzantılı dosyaların hepsi tek tek açılarak, gerekli ekleme ve deęiřiklikler yapılır. Son olarak program derlenerek çalışır hale getirilir.



3.2 Spectra.exe'nin Program Kodu

```
// mainfrm.cpp : implementation of the CMainFrame class

#include "stdafx.h"
#include "Spectra.h"
#include "mainfrm.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
////////////////////////////////////
// CMainFrame
IMPLEMENT_DYNAMIC(CMainFrame, CMDIFrameWnd)
BEGIN_MESSAGE_MAP(CMainFrame, CMDIFrameWnd)
   //{{AFX_MSG_MAP(CMainFrame)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        // DO NOT EDIT what you see in these blocks of generated code !
        ON_WM_CREATE()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// arrays of IDs used to initialize control bars
// toolbar buttons - IDs are command buttons
static UINT BASED_CODE buttons[] =
{
    // same order as in the bitmap 'toolbar.bmp'
    ID_FILE_NEW,
    ID_FILE_OPEN,
    ID_FILE_SAVE,
    ID_SEPARATOR,
    ID_EDIT_CUT,
    ID_EDIT_COPY,
    ID_EDIT_PASTE,
    ID_SEPARATOR,
    ID_FILE_PRINT,
    ID_APP_ABOUT,
};
static UINT BASED_CODE SpectraButtons[] =
{
    // same order as in the bitmap 'SpectraToolBar.bmp'
    ID_EDIT_INTERACTIVEMENU,
};
static UINT BASED_CODE indicators[] =
{
    ID_SEPARATOR, // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};
////////////////////////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here
}

CMainFrame::~CMainFrame()
```

```

{
}
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CMDIFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadBitmap(IDR_MAINFRAME) ||
        !m_wndToolBar.SetButtons(buttons,
            sizeof(buttons)/sizeof(UINT)))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }
    //////////////////////////////////////
    if (!m_wndSpectraToolBar.Create(this) ||
        !m_wndSpectraToolBar.LoadBitmap(IDR_SPECTRUM) ||
        !m_wndSpectraToolBar.SetButtons(SpectraButtons,
            sizeof(SpectraButtons)/sizeof(UINT)))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }
    //////////////////////////////////////

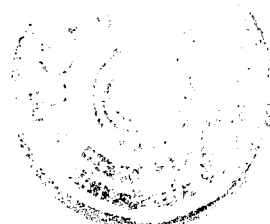
    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
            sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;    // fail to create
    }

    // TODO: Delete these three lines if you don't want the toolbar to
    // be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY | CBRS_FLOAT_MULTI);
    m_wndSpectraToolBar.EnableDocking(CBRS_ALIGN_ANY | CBRS_FLOAT_MULTI);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);
    DockControlBar(&m_wndSpectraToolBar);

    // TODO: Remove this if you don't want tool tips
    m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY);
    return 0;
}
////////////////////////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CMDIFrameWnd::AssertValid();
}
void CMainFrame::Dump(CDumpContext& dc) const
{
    CMDIFrameWnd::Dump(dc);
}
#endif // _DEBUG

```



```
// Spectdoc.cpp : implementation of the CSpectraDoc class
//
#include "stdafx.h"
#include "Spectra.h"
#include "Spectdoc.h"
#include "Spectrum.h"

#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CSpectraDoc

IMPLEMENT_DYNCREATE(CSpectraDoc, CDocument)

BEGIN_MESSAGE_MAP(CSpectraDoc, CDocument)
   //{{AFX_MSG_MAP(CSpectraDoc)
    ON_COMMAND(ID_EDIT_INTERACTIVEMENU, OnEditInteractivemenu)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CSpectraDoc construction/destruction

CSpectraDoc::CSpectraDoc()
{
    // TODO: add one-time construction code here
}

CSpectraDoc::~CSpectraDoc() {}
BOOL CSpectraDoc::OnNewDocument()
{
    if (!CDocument::OnNewDocument())
        return FALSE;

    // TODO: add reinitialization code here
    // (SDI documents will reuse this document)

    int counter = 0;
    while (20 - counter) molecule.nuclei[counter].m_SpinData = 0.;
    molecule.nuclei[counter].m_HyperfineData = 0.; counter++;}

    m_nCurrent_PosData = 30;
    m_Line_WidthData = 15;
    m_Line_ThicknessData = 1;
    m_Lorentz_or_GaussData = 0;
    m_Low_End_of_RangeData = 2500;
    m_High_End_of_RangeData = 4000.0;

    return TRUE;
}

////////////////////////////////////
// CSpectraDoc serialization
void CSpectraDoc::Serialize(CArchive& ar)
{
    if (ar.IsStoring())
    {
        // TODO: add storing code here
    }
    else
    {
        // TODO: add loading code here
    }
}
```

```

}
////////////////////////////////////
// CSpectraDoc diagnostics

#ifdef _DEBUG
void CSpectraDoc::AssertValid() const
{
    CDocument::AssertValid();
}
void CSpectraDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////
// CSpectraDoc commands

void CSpectraDoc::OnEditInteractivemenu()
{
    CSpectrumDlg dlg;
    dlg.m_Line_Width      = m_Line_WidthData;
    dlg.m_nCurrent_Pos    = m_nCurrent_PosData;
    dlg.m_Line_Thickness  = m_Line_ThicknessData;
    dlg.m_Lorentz_or_Gauss = m_Lorentz_or_GaussData;
    dlg.m_Low_End_of_Range = m_Low_End_of_RangeData;
    dlg.m_High_End_of_Range = m_High_End_of_RangeData;

    dlg.m_Spin_1          = molecule.nuclei[ 0].m_SpinData;
    dlg.m_Spin_2          = molecule.nuclei[ 1].m_SpinData;
    dlg.m_Spin_3          = molecule.nuclei[ 2].m_SpinData;
    dlg.m_Spin_4          = molecule.nuclei[ 3].m_SpinData;
    dlg.m_Spin_5          = molecule.nuclei[ 4].m_SpinData;
    dlg.m_Spin_6          = molecule.nuclei[ 5].m_SpinData;
    dlg.m_Spin_7          = molecule.nuclei[ 6].m_SpinData;
    dlg.m_Spin_8          = molecule.nuclei[ 7].m_SpinData;
    dlg.m_Spin_9          = molecule.nuclei[ 8].m_SpinData;
    dlg.m_Spin_10         = molecule.nuclei[ 9].m_SpinData;
    dlg.m_Spin_11         = molecule.nuclei[10].m_SpinData;
    dlg.m_Spin_12         = molecule.nuclei[11].m_SpinData;
    dlg.m_Spin_13         = molecule.nuclei[12].m_SpinData;
    dlg.m_Spin_14         = molecule.nuclei[13].m_SpinData;
    dlg.m_Spin_15         = molecule.nuclei[14].m_SpinData;
    dlg.m_Spin_16         = molecule.nuclei[15].m_SpinData;
    dlg.m_Spin_17         = molecule.nuclei[16].m_SpinData;
    dlg.m_Spin_18         = molecule.nuclei[17].m_SpinData;
    dlg.m_Spin_19         = molecule.nuclei[18].m_SpinData;
    dlg.m_Spin_20         = molecule.nuclei[19].m_SpinData;
    dlg.m_Hyperfine_1     = molecule.nuclei[ 0].m_HyperfineData;
    dlg.m_Hyperfine_2     = molecule.nuclei[ 1].m_HyperfineData;
    dlg.m_Hyperfine_3     = molecule.nuclei[ 2].m_HyperfineData;
    dlg.m_Hyperfine_4     = molecule.nuclei[ 3].m_HyperfineData;
    dlg.m_Hyperfine_5     = molecule.nuclei[ 4].m_HyperfineData;
    dlg.m_Hyperfine_6     = molecule.nuclei[ 5].m_HyperfineData;
    dlg.m_Hyperfine_7     = molecule.nuclei[ 6].m_HyperfineData;
    dlg.m_Hyperfine_8     = molecule.nuclei[ 7].m_HyperfineData;
    dlg.m_Hyperfine_9     = molecule.nuclei[ 8].m_HyperfineData;
    dlg.m_Hyperfine_10    = molecule.nuclei[ 9].m_HyperfineData;
    dlg.m_Hyperfine_11    = molecule.nuclei[10].m_HyperfineData;
    dlg.m_Hyperfine_12    = molecule.nuclei[11].m_HyperfineData;
    dlg.m_Hyperfine_13    = molecule.nuclei[12].m_HyperfineData;
    dlg.m_Hyperfine_14    = molecule.nuclei[13].m_HyperfineData;
    dlg.m_Hyperfine_15    = molecule.nuclei[14].m_HyperfineData;
    dlg.m_Hyperfine_16    = molecule.nuclei[15].m_HyperfineData;

```

```

dlg.m_Hyperfine_17 = molecule.nuclei[16].m_HyperfineData;
dlg.m_Hyperfine_18 = molecule.nuclei[17].m_HyperfineData;
dlg.m_Hyperfine_19 = molecule.nuclei[18].m_HyperfineData;
dlg.m_Hyperfine_20 = molecule.nuclei[19].m_HyperfineData;

```

```

if (dlg.DoModal() != IDOK) return;

```

```

molecule.nuclei[ 0].m_SpinData = dlg.m_Spin_1;
molecule.nuclei[ 1].m_SpinData = dlg.m_Spin_2;
molecule.nuclei[ 2].m_SpinData = dlg.m_Spin_3;
molecule.nuclei[ 3].m_SpinData = dlg.m_Spin_4;
molecule.nuclei[ 4].m_SpinData = dlg.m_Spin_5;
molecule.nuclei[ 5].m_SpinData = dlg.m_Spin_6;
molecule.nuclei[ 6].m_SpinData = dlg.m_Spin_7;
molecule.nuclei[ 7].m_SpinData = dlg.m_Spin_8;
molecule.nuclei[ 8].m_SpinData = dlg.m_Spin_9;
molecule.nuclei[ 9].m_SpinData = dlg.m_Spin_10;
molecule.nuclei[10].m_SpinData = dlg.m_Spin_11;
molecule.nuclei[11].m_SpinData = dlg.m_Spin_12;
molecule.nuclei[12].m_SpinData = dlg.m_Spin_13;
molecule.nuclei[13].m_SpinData = dlg.m_Spin_14;
molecule.nuclei[14].m_SpinData = dlg.m_Spin_15;
molecule.nuclei[15].m_SpinData = dlg.m_Spin_16;
molecule.nuclei[16].m_SpinData = dlg.m_Spin_17;
molecule.nuclei[17].m_SpinData = dlg.m_Spin_18;
molecule.nuclei[18].m_SpinData = dlg.m_Spin_19;
molecule.nuclei[19].m_SpinData = dlg.m_Spin_20;

```

```

molecule.nuclei[ 0].m_HyperfineData = dlg.m_Hyperfine_1;
molecule.nuclei[ 1].m_HyperfineData = dlg.m_Hyperfine_2;
molecule.nuclei[ 2].m_HyperfineData = dlg.m_Hyperfine_3;
molecule.nuclei[ 3].m_HyperfineData = dlg.m_Hyperfine_4;
molecule.nuclei[ 4].m_HyperfineData = dlg.m_Hyperfine_5;
molecule.nuclei[ 5].m_HyperfineData = dlg.m_Hyperfine_6;
molecule.nuclei[ 6].m_HyperfineData = dlg.m_Hyperfine_7;
molecule.nuclei[ 7].m_HyperfineData = dlg.m_Hyperfine_8;
molecule.nuclei[ 8].m_HyperfineData = dlg.m_Hyperfine_9;
molecule.nuclei[ 9].m_HyperfineData = dlg.m_Hyperfine_10;
molecule.nuclei[10].m_HyperfineData = dlg.m_Hyperfine_11;
molecule.nuclei[11].m_HyperfineData = dlg.m_Hyperfine_12;
molecule.nuclei[12].m_HyperfineData = dlg.m_Hyperfine_13;
molecule.nuclei[13].m_HyperfineData = dlg.m_Hyperfine_14;
molecule.nuclei[14].m_HyperfineData = dlg.m_Hyperfine_15;
molecule.nuclei[15].m_HyperfineData = dlg.m_Hyperfine_16;
molecule.nuclei[16].m_HyperfineData = dlg.m_Hyperfine_17;
molecule.nuclei[17].m_HyperfineData = dlg.m_Hyperfine_18;
molecule.nuclei[18].m_HyperfineData = dlg.m_Hyperfine_19;
molecule.nuclei[19].m_HyperfineData = dlg.m_Hyperfine_20;

```

```

m_Low_End_of_RangeData = dlg.m_Low_End_of_Range;
m_High_End_of_RangeData = dlg.m_High_End_of_Range;
m_Line_ThicknessData = dlg.m_Line_Thickness;
m_Lorentz_or_GaussData = dlg.m_Lorentz_or_Gauss;
m_Line_WidthData = dlg.m_Line_Width;
m_nCurrent_PosData = dlg.m_nCurrent_Pos;

```

```

UpdateAllViews(NULL);

```

```

}

```

```

// Spectra.cpp : Defines the class behaviors for the application.
//
#include "stdafx.h"
#include "Spectra.h"
#include "mainfrm.h"
#include "Spectdoc.h"
#include "Spectvw.h"
#include "Windows.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////
// CSpectraApp
BEGIN_MESSAGE_MAP(CSpectraApp, CWinApp)
   //{{AFX_MSG_MAP(CSpectraApp)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        // DO NOT EDIT what you see in these blocks of generated code!
    }}AFX_MSG_MAP
    // Standard file based document commands
    ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
    ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
    // Standard print setup command
    ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////
// CSpectraApp construction
CSpectraApp::CSpectraApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////
// The one and only CSpectraApp object
CSpectraApp theApp;

/////////////////////////////////////////////////////////////////
// CSpectraApp initialization

BOOL CSpectraApp::InitInstance()
{
    // Standard initialization
    // If you are not using these features and wish to reduce the size
    // of your final executable, you should remove from the following
    // the specific initialization routines you do not need.
    Enable3dControls();

    //LoadStdProfileSettings(); // Load standard INI file options (including MRU)
    // Register the application's document templates. Document templates
    // serve as the connection between documents, frame windows and views.
    CMultiDocTemplate* pDocTemplate;
    pDocTemplate = new CMultiDocTemplate(
        IDR_SPECTRTYPE,
        RUNTIME_CLASS(CSpectraDoc),
        RUNTIME_CLASS(CMDIChildWnd), // standard MDI child frame
        RUNTIME_CLASS(CSpectraView));
    AddDocTemplate(pDocTemplate);

    // create main MDI Frame window
    CMainFrame* pMainFrame = new CMainFrame;

```

```

        if (!pMainFrame->LoadFrame(IDR_MAINFRAME))                return FALSE;
        m_pMainWnd = pMainFrame;
        // Enable DDE Execute open
        EnableShellOpen();
        RegisterShellFileTypes();

        // simple command line parsing
        if (m_lpCmdLine[0] == '\\0')
        {
            // create a new (empty) document
            OnFileNew();
        }
        else
        {
            // open an existing document
            OpenDocumentFile(m_lpCmdLine);
        }
        // Enable drag/drop open
        m_pMainWnd->DragAcceptFiles();
        // The main window has been initialized, so show and update it.
        pMainFrame->ShowWindow(m_nCmdShow);
        pMainFrame->UpdateWindow();

        return TRUE;
    }
    //////////////////////////////////////
    // CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();
    // Dialog Data
   //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA
    // Implementation
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
   //{{AFX_MSG(CAboutDlg)
        // No message handlers
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
// App command to run the dialog
void CSpectraApp::OnAppAbout()
{
    CAboutDlg aboutDlg;    aboutDlg.DoModal();
}

```

// **Spectvw.cpp** : implementation of the CSpectraView class

#include "stdafx.h"
#include "Spectra.h"

#include "Spectdoc.h"
#include "Spectvw.h"
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CSpectraView

IMPLEMENT_DYNCREATE(CSpectraView, CView)

BEGIN_MESSAGE_MAP(CSpectraView, CView)
//{{AFX_MSG_MAP(CSpectraView)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()

////////////////////////////////////
// CSpectraView construction/destruction

CSpectraView::CSpectraView()
{ // TODO: add construction code here }
CSpectraView::~CSpectraView() {}

////////////////////////////////////
// CSpectraView drawing

void CSpectraView::OnDraw(CDC* pDC)
{ CSpectraDoc* pDoc = GetDocument();
ASSERT_VALID(pDoc);

// TODO: add draw code for native data here
int N_of_SumplingPoints = 1050, N_of_Resonance_Points = 1, temp_i;
int n[10], N_of_SpinStates[10], H0, k = 0, i = 0, j = 0, jj = 0;
double TotalSplittingEffect = 0, YmaxPrime, YPrimeL;
double hBar = 1.0e-34, DHpp, EksiOtuziki = -32, Three = 3, Four = 4, xx, yy;
double spectraMatris[1050], hPlanc = 6.626e-34, frekans = 9.1e9, g = 2.0023193;
double Resonance_Point[100000], ScaleFactor = 1, BetaElectron = 9.2740154e-24, DeltaH;

DHpp = double(pDoc->m_Line_WidthData);
YmaxPrime = double(pDoc->m_nCurrent_PosData);
DeltaH = (pDoc->m_High_End_of_RangeData - pDoc->m_Low_End_of_RangeData)/(N_of_SumplingPoints - 1);

//Create and Select The Necessary Pen Style
CPen pen;

```

CPen eraser;
pen.CreatePen( PS_SOLID, pDoc->m_Line_ThicknessData, RGB(0, 0, 0) );
eraser.CreatePen( PS_SOLID, pDoc->m_Line_ThicknessData, RGB(255, 255, 255) );
pDC->SelectObject(&pen);
//Draw the coordinate axes
pDC->MoveTo( 50, 10);           //Move to upper point of y axes
(*pDC).LineTo(50, 760);        //Draw y axes
(*pDC).LineTo(1100, 760);       //Draw H axes
//Initialize The Matrices
while (N_of_SumplingPoints - i) {spectraMatris[i] = 0;          i++; }

//Prepare the spectrum data for the possible N Points
k = 0;
N_of_Resonance_Points = 1;           //Default Zeeman Resonance Point.

while (10 - k) {N_of_SpinStates[k] = int(2*pDoc->molecule.nuclei[k].m_SpinData + 1);
                N_of_Resonance_Points = N_of_Resonance_Points *
N_of_SpinStates[k];          k++;}

j = 0;
n[0] = 0; while (N_of_SpinStates[ 0] - n[0]) {
n[1] = 0;      while (N_of_SpinStates[ 1] - n[1]) {
n[2] = 0;      while (N_of_SpinStates[ 2] - n[2]) {
n[3] = 0; while (N_of_SpinStates[ 3] - n[3]) {
n[4] = 0; while (N_of_SpinStates[ 4] - n[4]) {
n[5] = 0; while (N_of_SpinStates[ 5] - n[5]) {
n[6] = 0; while (N_of_SpinStates[ 6] - n[6]) {
n[7] = 0; while (N_of_SpinStates[ 7] - n[7]) {
n[8] = 0;      while (N_of_SpinStates[ 8] - n[8]) {
n[9] = 0; while (N_of_SpinStates[ 9] - n[9]) {

jj = 0;
TotalSplittingEffect = 0;

while (10 - jj)
{TotalSplittingEffect += (pDoc->molecule.nuclei[jj].m_SpinData-n[jj])*pDoc-
>molecule.nuclei[jj].m_HyperfineData*10; jj++;}
Resonance_Point[jj] = 10000 * hPlanc*freaks/ (g*BetaElectron) - (TotalSplittingEffect) ;
in Gauss units
j++;
    n[ 9]++;}
        n[ 8]++;}
            n[ 7]++;}
                n[ 6]++;}
                    n[ 5]++;}
                        n[ 4]++;}
                            n[ 3]++;}
                                n[ 2]++;}
                                    n[ 1]++;}
                                        n[ 0]++;}

//Draw the spectrum

pDC->MoveTo( 50, 380); //Move to the orijin.

i = 1;
while (N_of_SumplingPoints - i) {j = 0;          while (N_of_Resonance_Points - j)

    H0 = int((Resonance_Point[j] - pDoc->m_Low_End_of_RangeData)/DeltaH);

```

```

        xx = (i-H0)*DeltaH/DHpp; yy = 3 + 4*xx*xx; YPrimeL = -
32*YmaxPrime*xx/yy/yy;

spectraMatris[i] -= YPrimeL;    j++;}
if (fabs(ScaleFactor * spectraMatris[i]) < YmaxPrime ) (*pDC).LineTo(50 + i, int(ScaleFactor *
spectraMatris[i] + 380));
    else    {
                pDC->MoveTo( 50, 380); //Move to the orijin again.
                pDC->SelectObject(&eraser);
                temp_i = 0;
                while (i - temp_i) {(*pDC).LineTo(50 + temp_i, int(ScaleFactor *
spectraMatris[temp_i] + 380)); temp_i++; }
                pDC->MoveTo( 50, 380); //Move to the orijin again.
                pDC->SelectObject(&pen);
                ScaleFactor = ScaleFactor / 2 ;
                temp_i = 0;
                while (i+1 - temp_i) {(*pDC).LineTo(50 + temp_i, int(ScaleFactor
* spectraMatris[temp_i] + 380)); temp_i++; }
            }
i++; }
}
////////////////////////////////////
// CSpectraView printing

BOOL CSpectraView::OnPreparePrinting(CPrintInfo* pInfo)
{
    // default preparation
    return DoPreparePrinting(pInfo);
}

void CSpectraView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add extra initialization before printing
}

void CSpectraView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add cleanup after printing
}
////////////////////////////////////
// CSpectraView diagnostics

#ifdef _DEBUG
void CSpectraView::AssertValid() const
{
    CView::AssertValid();
}

void CSpectraView::Dump(CDumpContext& dc) const
{
    CView::Dump(dc);
}

CSpectraDoc* CSpectraView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CSpectraDoc)));
    return (CSpectraDoc*)m_pDocument;
}
#endif // _DEBUG

```

// CMainFrame message handlers// **spectrum.cpp** : implementation file

```
#include "stdafx.h"
#include "Spectra.h"
#include "spectrum.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif

// CSpectrumDlg dialog
CSpectrumDlg::CSpectrumDlg(CWnd* pParent /*=NULL*/)
: CDialog(CSpectrumDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CSpectrumDlg)
    m_Line_Thickness = 3;
    m_Lorentz_or_Gauss = 1;
    m_Spin_2 = 0.0;
    m_Spin_3 = 0.0;
    m_Spin_4 = 0.0;
    m_Spin_5 = 0.0;
    m_Spin_6 = 0.0;
    m_Spin_7 = 0.0;
    m_Spin_8 = 0.0;
    m_Spin_9 = 0.0;
    m_Spin_10 = 0.0;
    m_Spin_11 = 0.0;
    m_Spin_12 = 0.0;
    m_Spin_13 = 0.0;
    m_Spin_14 = 0.0;
    m_Spin_15 = 0.0;
    m_Spin_16 = 0.0;
    m_Spin_17 = 0.0;
    m_Spin_18 = 0.0;
    m_Spin_19 = 0.0;
    m_Spin_20 = 0.0;
    m_Hyperfine_1 = 0.0;
    m_Hyperfine_2 = 0.0;
    m_Hyperfine_3 = 0.0;
    m_Hyperfine_4 = 0.0;
    m_Hyperfine_5 = 0.0;
    m_Hyperfine_6 = 0.0;
    m_Hyperfine_7 = 0.0;
    m_Hyperfine_8 = 0.0;
    m_Hyperfine_9 = 0.0;
    m_Hyperfine_10 = 0.0;
    m_Hyperfine_11 = 0.0;
    m_Hyperfine_12 = 0.0;
    m_Hyperfine_13 = 0.0;
    m_Hyperfine_14 = 0.0;
    m_Hyperfine_15 = 0.0;
    m_Hyperfine_16 = 0.0;
    m_Hyperfine_17 = 0.0;
    m_Hyperfine_18 = 0.0;
    m_Hyperfine_19 = 0.0;
    m_Hyperfine_20 = 0.0;
    m_Low_End_of_Range = 0.0;
    m_High_End_of_Range = 0.0;
    m_Line_Width = 0;
    m_nCurrent_Pos = 30;
    //}}AFX_DATA_INIT
```

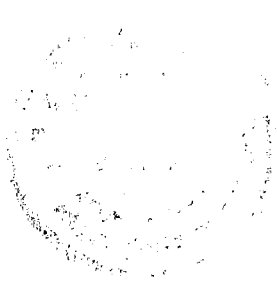
```

}
void CSpectrumDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSpectrumDlg)
    DDX_Scroll(pDX, IDC_SCROLLBAR1, m_nCurrent_Pos);
    DDX_Control(pDX, IDC_COMBO1, m_Number_of_Points);
    DDX_Radio(pDX, IDC_RADIO1, m_Lorentz_or_Gauss);
    DDX_Control(pDX, IDC_SCROLLBAR1, m_Receiver_Gain);
    //}}AFX_DATA_MAP
    m_Receiver_Gain.SetScrollRange(0, 100);
    m_Receiver_Gain.SetScrollPos(m_nCurrent_Pos, TRUE);

    DDX_Text(pDX, IDC_EDIT1, m_Spin_1);
    DDV_MinMaxDouble(pDX, m_Spin_1, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT2, m_Spin_2);
    DDV_MinMaxDouble(pDX, m_Spin_2, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT3, m_Spin_3);
    DDV_MinMaxDouble(pDX, m_Spin_3, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT4, m_Spin_4);
    DDV_MinMaxDouble(pDX, m_Spin_4, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT5, m_Spin_5);
    DDV_MinMaxDouble(pDX, m_Spin_5, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT6, m_Spin_6);
    DDV_MinMaxDouble(pDX, m_Spin_6, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT7, m_Spin_7);
    DDV_MinMaxDouble(pDX, m_Spin_7, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT8, m_Spin_8);
    DDV_MinMaxDouble(pDX, m_Spin_8, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT9, m_Spin_9);
    DDV_MinMaxDouble(pDX, m_Spin_9, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT10, m_Spin_10);
    DDV_MinMaxDouble(pDX, m_Spin_10, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT11, m_Spin_11);
    DDV_MinMaxDouble(pDX, m_Spin_11, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT12, m_Spin_12);
    DDV_MinMaxDouble(pDX, m_Spin_12, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT13, m_Spin_13);
    DDV_MinMaxDouble(pDX, m_Spin_13, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT14, m_Spin_14);
    DDV_MinMaxDouble(pDX, m_Spin_14, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT15, m_Spin_15);
    DDV_MinMaxDouble(pDX, m_Spin_15, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT16, m_Spin_16);
    DDV_MinMaxDouble(pDX, m_Spin_16, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT17, m_Spin_17);
    DDV_MinMaxDouble(pDX, m_Spin_17, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT18, m_Spin_18);
    DDV_MinMaxDouble(pDX, m_Spin_18, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT19, m_Spin_19);
    DDV_MinMaxDouble(pDX, m_Spin_19, 0.0, 4.5);
    DDX_Text(pDX, IDC_EDIT20, m_Spin_20);
    DDV_MinMaxDouble(pDX, m_Spin_20, 0.0, 4.5);

    DDX_Text(pDX, IDC_EDIT21, m_Hyperfine_1);
    DDV_MinMaxDouble(pDX, m_Hyperfine_1, -3000.0, 3000.0);
    DDX_Text(pDX, IDC_EDIT22, m_Hyperfine_2);
    DDV_MinMaxDouble(pDX, m_Hyperfine_2, -3000.0, 3000.0);
    DDX_Text(pDX, IDC_EDIT23, m_Hyperfine_3);
    DDV_MinMaxDouble(pDX, m_Hyperfine_3, -3000.0, 3000.0);
    DDX_Text(pDX, IDC_EDIT24, m_Hyperfine_4);

```



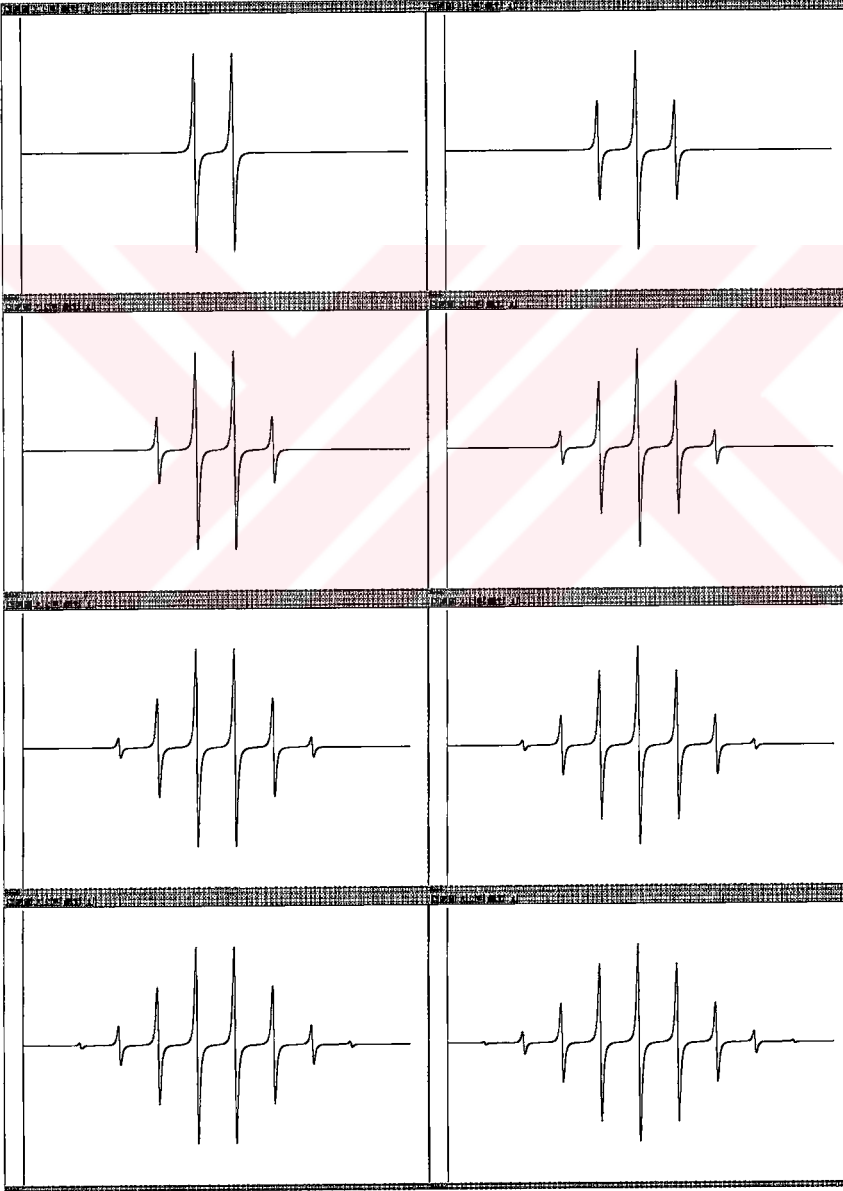
```
DDV_MinMaxDouble(pDX, m_Hyperfine_4, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT25, m_Hyperfine_5);
DDV_MinMaxDouble(pDX, m_Hyperfine_5, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT26, m_Hyperfine_6);
DDV_MinMaxDouble(pDX, m_Hyperfine_6, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT27, m_Hyperfine_7);
DDV_MinMaxDouble(pDX, m_Hyperfine_7, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT28, m_Hyperfine_8);
DDV_MinMaxDouble(pDX, m_Hyperfine_8, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT29, m_Hyperfine_9);
DDV_MinMaxDouble(pDX, m_Hyperfine_9, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT30, m_Hyperfine_10);
DDV_MinMaxDouble(pDX, m_Hyperfine_10, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT31, m_Hyperfine_11);
DDV_MinMaxDouble(pDX, m_Hyperfine_11, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT32, m_Hyperfine_12);
DDV_MinMaxDouble(pDX, m_Hyperfine_12, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT33, m_Hyperfine_13);
DDV_MinMaxDouble(pDX, m_Hyperfine_13, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT34, m_Hyperfine_14);
DDV_MinMaxDouble(pDX, m_Hyperfine_14, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT35, m_Hyperfine_15);
DDV_MinMaxDouble(pDX, m_Hyperfine_15, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT36, m_Hyperfine_16);
DDV_MinMaxDouble(pDX, m_Hyperfine_16, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT37, m_Hyperfine_17);
DDV_MinMaxDouble(pDX, m_Hyperfine_17, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT38, m_Hyperfine_18);
DDV_MinMaxDouble(pDX, m_Hyperfine_18, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT39, m_Hyperfine_19);
DDV_MinMaxDouble(pDX, m_Hyperfine_19, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT40, m_Hyperfine_20);
DDV_MinMaxDouble(pDX, m_Hyperfine_20, -3000.0, 3000.0);
DDX_Text(pDX, IDC_EDIT41, m_Line_Thickness);
DDV_MinMaxInt(pDX, m_Line_Thickness, 1, 9);
DDX_Text(pDX, IDC_EDIT42, m_Low_End_of_Range);
DDV_MinMaxDouble(pDX, m_Low_End_of_Range, 0.0, 20000.0);
DDX_Text(pDX, IDC_EDIT43, m_High_End_of_Range);
DDV_MinMaxDouble(pDX, m_High_End_of_Range, 0.0, 20000.0);
DDX_Text(pDX, IDC_EDIT44, m_Line_Width);
DDV_MinMaxInt(pDX, m_Line_Width, 1, 2999);
}
BEGIN_MESSAGE_MAP(CSpectrumDlg, CDialog)
    //{AFX_MSG_MAP(CSpectrumDlg)
    ON_WM_HSCROLL()
    //}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////
// CSpectrumDlg message handlers

void CSpectrumDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    // TODO: Add your message handler code here and/or call default
    //if (SB_THUMBTRACK == nSBCode) m_Receiver_Gain.SetScrollPos(nPos, TRUE);

    switch( nSBCode )
    {
    case SB_THUMBTRACK:                m_Receiver_Gain.SetScrollPos(nPos, TRUE);
    case SB_THUMBPOSITION:            m_Receiver_Gain.SetScrollPos(nPos, TRUE);
    default:                          CDialog::OnHScroll(nSBCode, nPos, pScrollBar);
    }
}
```

3.3 Doğruluk Testleri

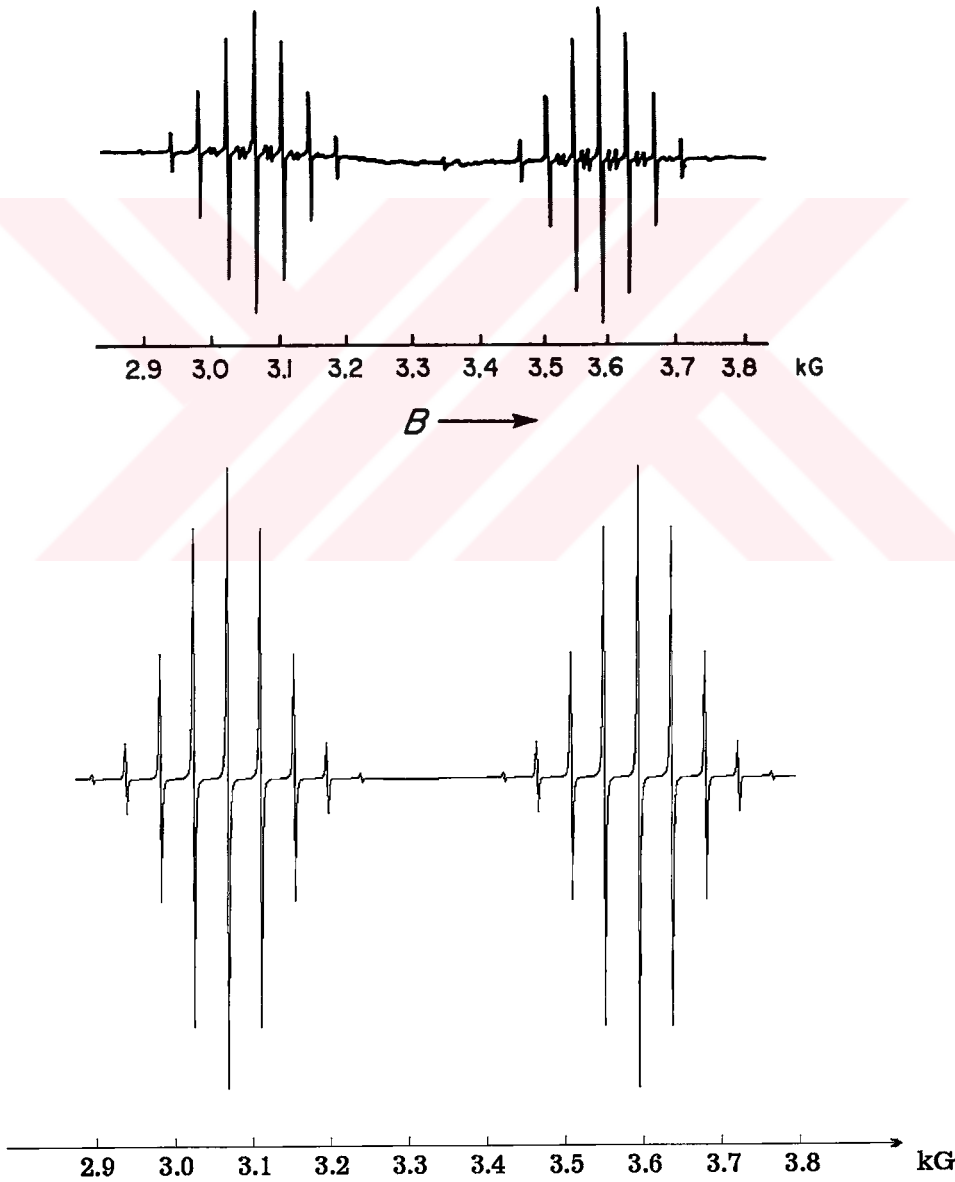
Şekil 3.3.1 de görülen spectrumlar, Spectra.exe programı kullanılarak elde edilmiştir. Çekirdek spin değerleri soldan sağa ve aşağıya doğru $\frac{1}{2}$ lik adımlarla artmaktadır. Daha önceden yapılmış benzeri nümerik çalışmalar neticesinde de bir elektronun değişik spin değerlerine sahip hipotetik çekirdeklerle yaptığı ince yapı etkileşmeleri sonucu ortaya çıkacak teorik spektrumlar elde edilmiş fakat burada elde edilenlerle hemen hemen aynı oldukları için alıntılanmalarına gerek duyulmamıştır.[8]



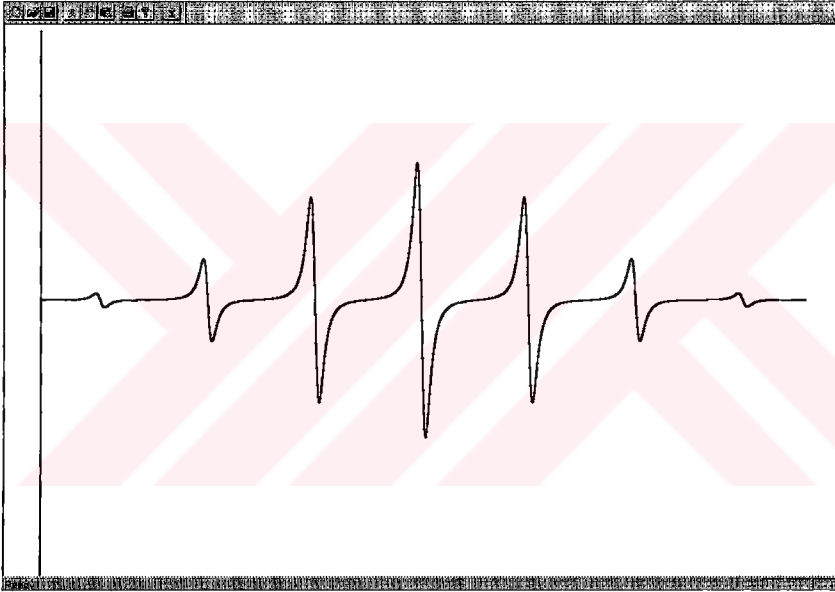
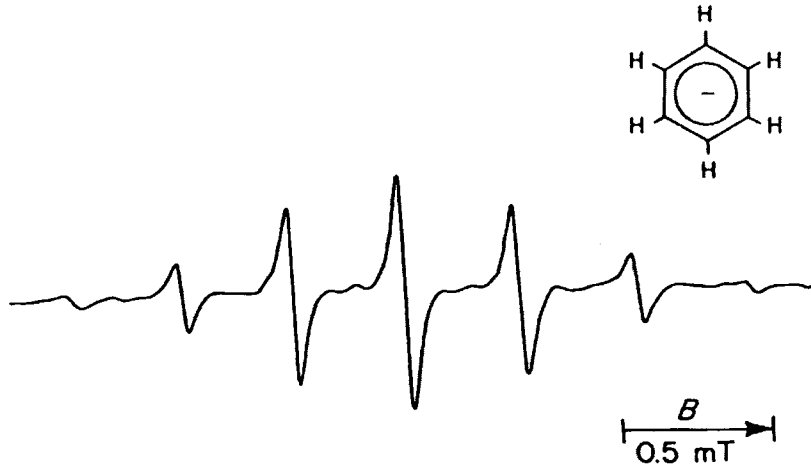
Şekil 3.3.1 Bir elektronun spin quantum sayısı $\frac{1}{2}$, 1, $\frac{3}{2}$, 2, $\frac{5}{2}$, 3, $\frac{7}{2}$, 4 olan çekirdeklerle yaptığı ince yapı etkileşmeleri sonucu oluşan kuramsal spektrumlar.

4. UYGULAMALAR: Kuramsal Spektrumların Üretilmesi ve Deneysel Spektrumlarla Karşılaştırma

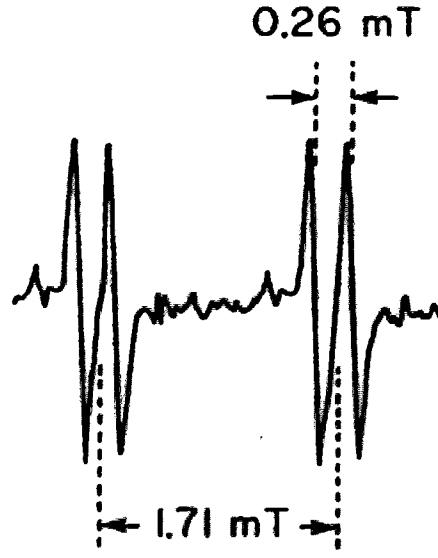
Bu kısımda eldeki program kullanılarak, deneysel spektrumları literatürde bulunan ve aşırı ince etkileşimlerin başat etkiye sahip olduğu paramanyetik örnekler için, kuramsal spektrumlar elde edilerek deneysel olanlarla karşılaştırma yapıldı. Aşağıda görülen her bir şekilde ilk önce örneğin deneysel spectrumu, arkasından simülasyon programı ile elde edilen kuramsal spektrum verilmiştir. Şekillerden de görüleceği üzere, bu tür örnekler için gerçeğe yeterince yakın spektrumlar elde edilebilmektedir.



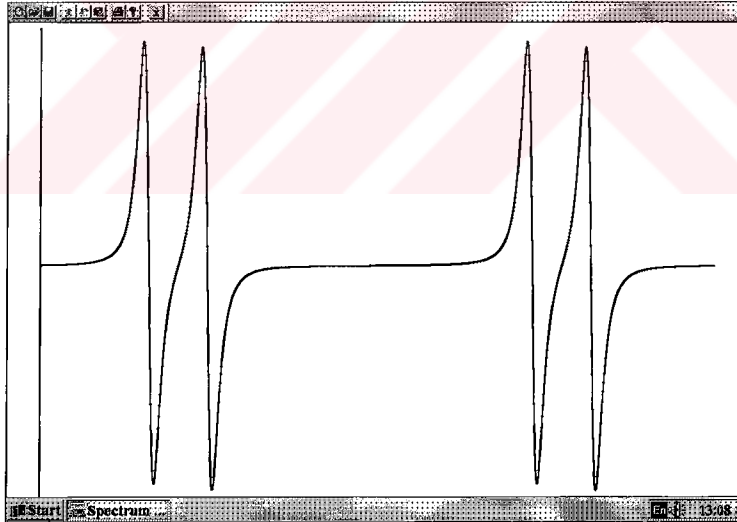
Şekil 4.1 X ışınlarına maruz bırakılan CaF_2 de yerdeğıştiren safsızlık hidrojen atomlarının, oda sıcaklığında verdiği X-Band EPR spectrumu.[9]



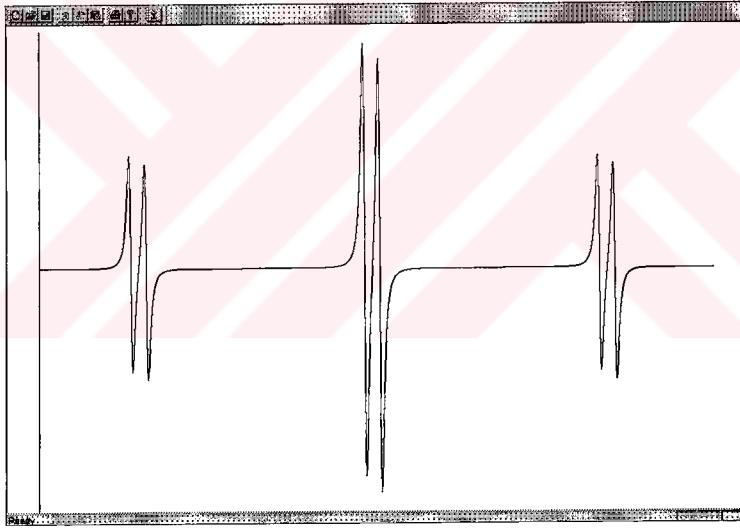
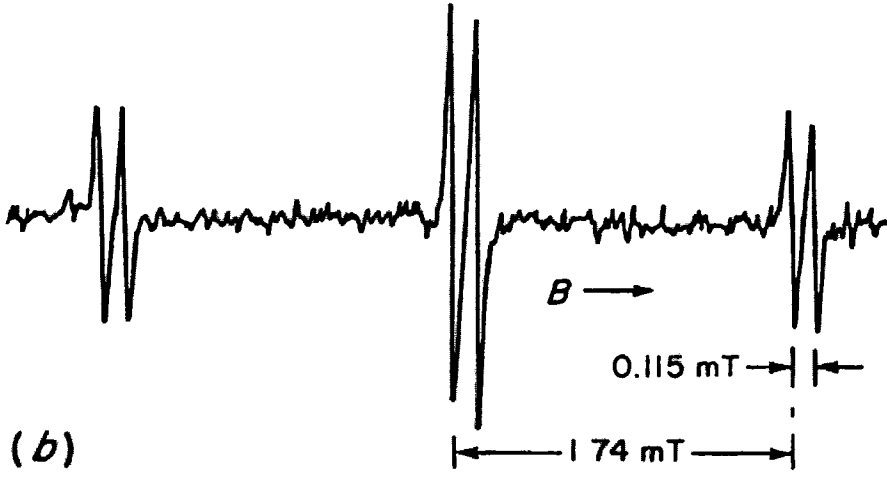
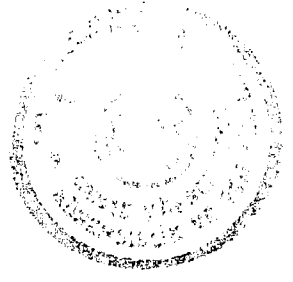
Şekil 4.2 173 kelvinde, tetrahidrofuran ve dimetiloksitan ile hazırlanan, 2:1 oranındaki çözelti içerisinde bulunan benzen anyon radikalinin X-band EPR spektrumu [10]



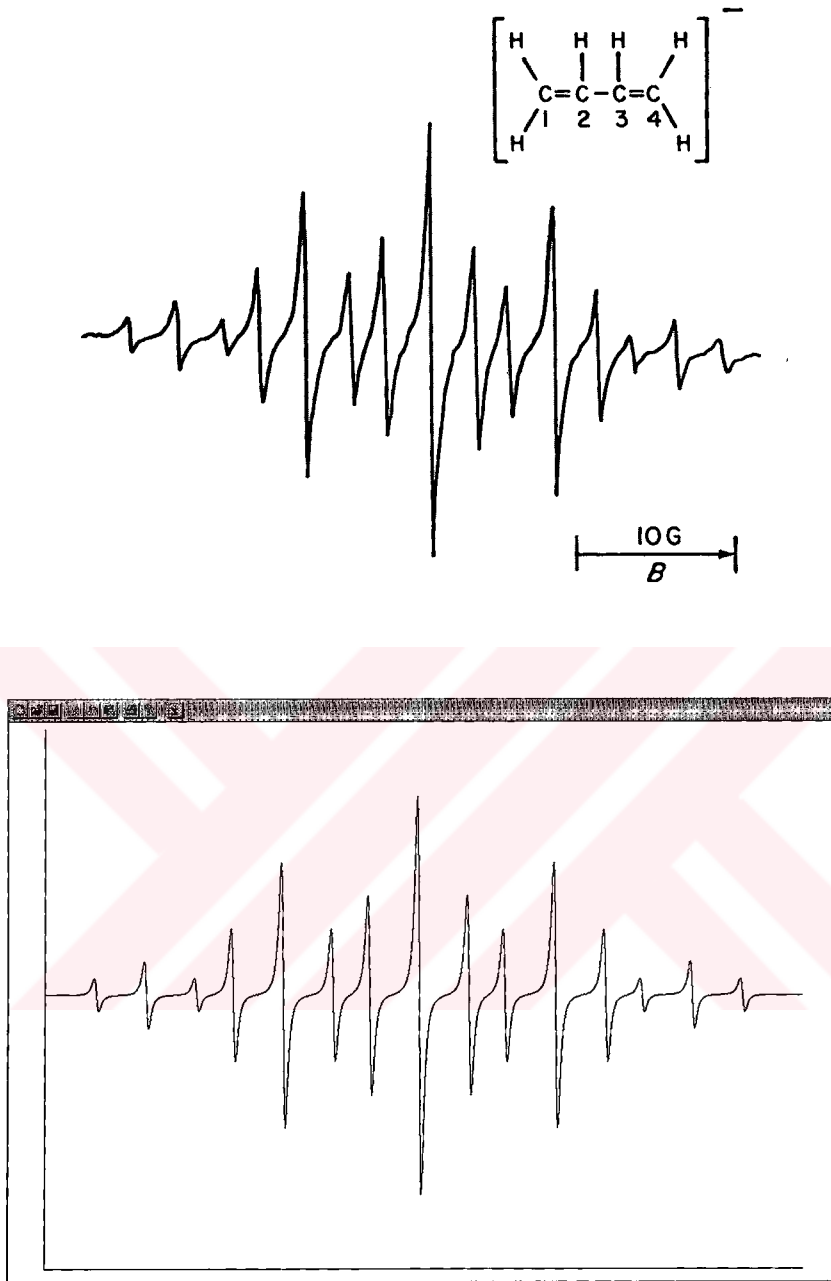
$B \rightarrow$



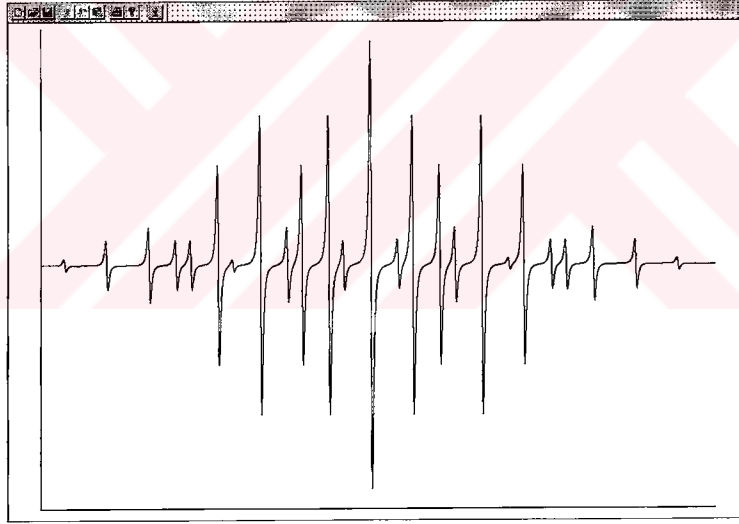
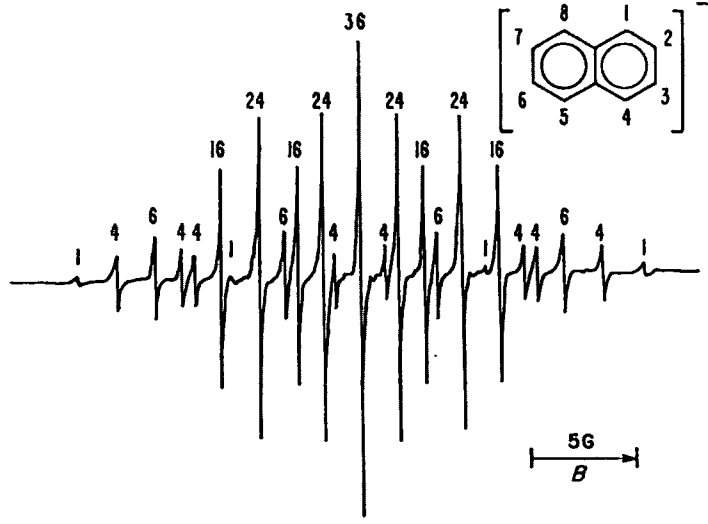
Şekil 4.3 İnce yapı etkileşimi açısından özdeş olmayan iki protona örnek olarak; 298 K de hazırlanan sulu çözeltideki glikolik asit radikalının (HOCHCOOH) X-band EPR spektrumu. Büyük yarıлма CH 'ın protonundan, küçük yarıлма ise en yakın OH 'ın protonundan kaynaklanmaktadır.[11]



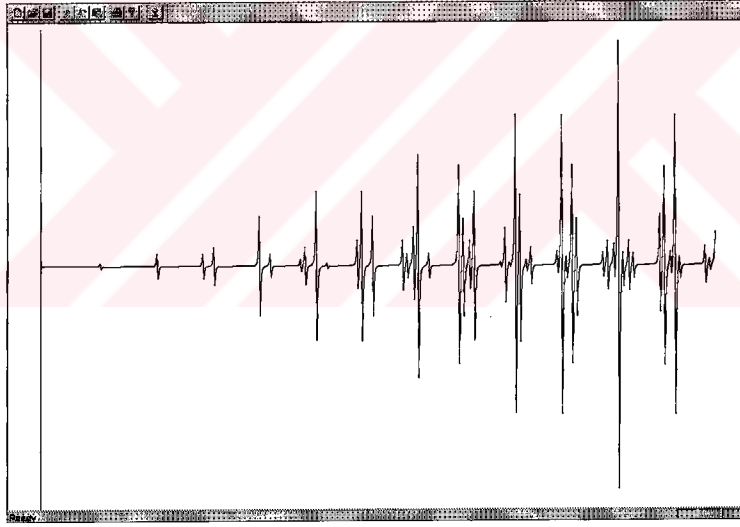
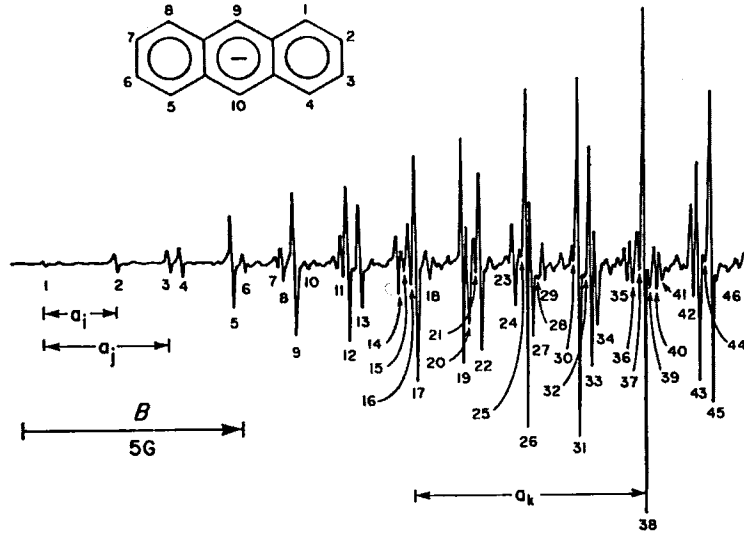
Şekil 4.4 299 K deki metanol içerisinde bulunan CH_2OH radikalinin X-band EPR spektrumu. [12]



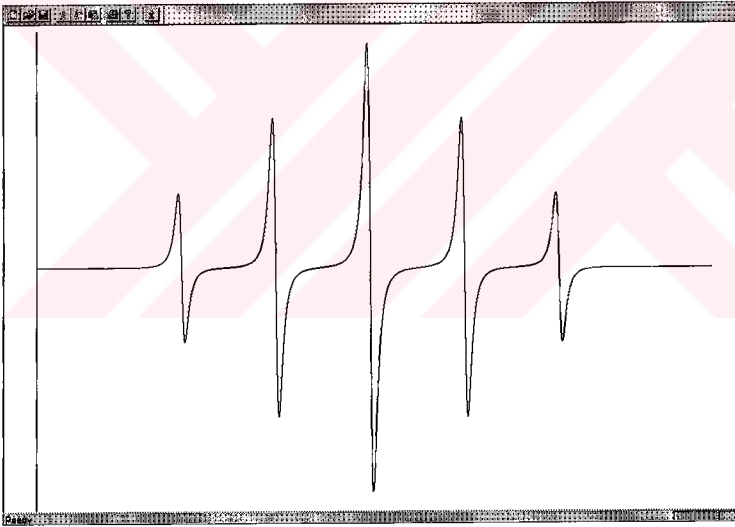
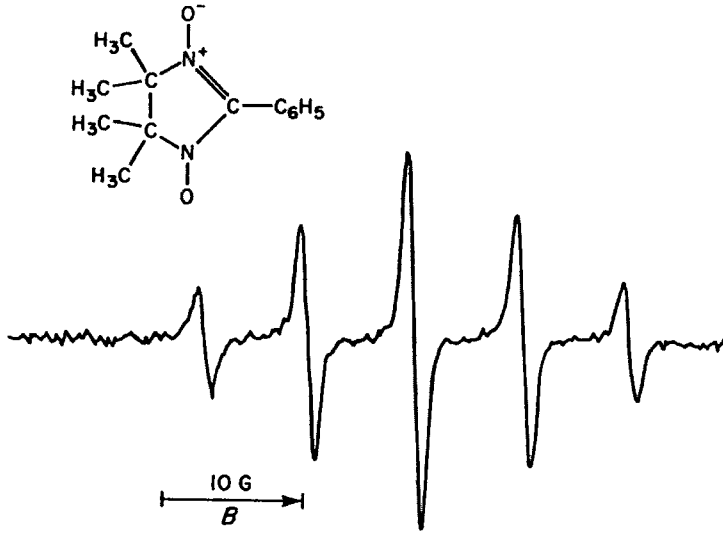
Şekil 4.5 195 K deki sıvı NH_3 'ün elektrolizi ile elde edilen 1,3-bütadiyen anyon radikalinin X-band EPR spektrumu. [13]



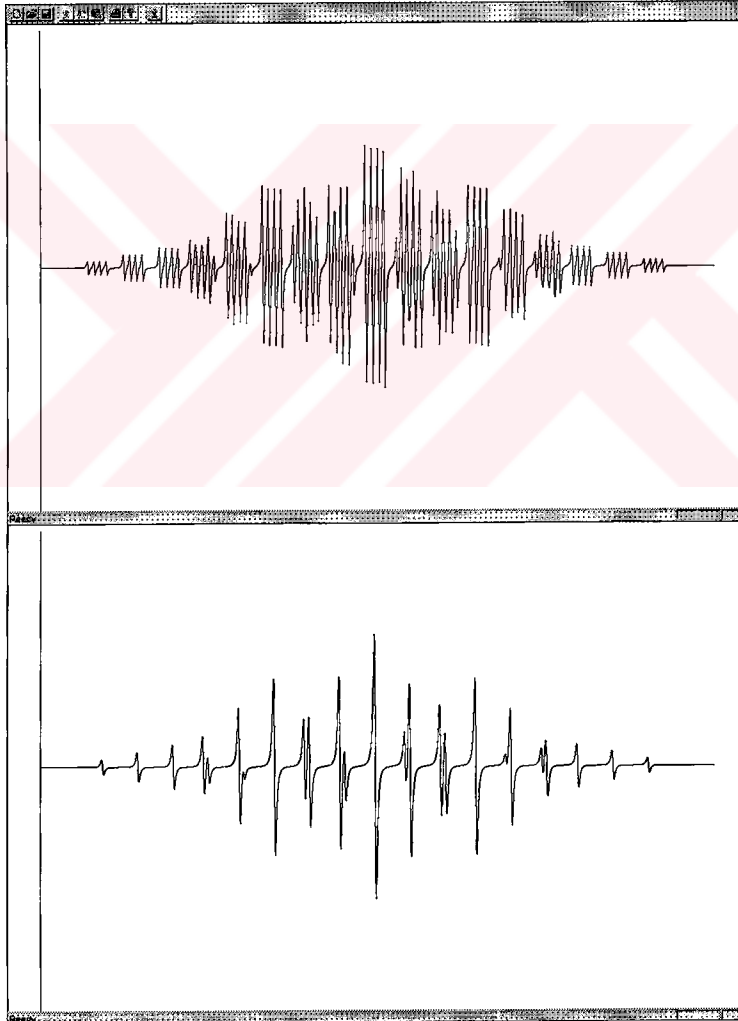
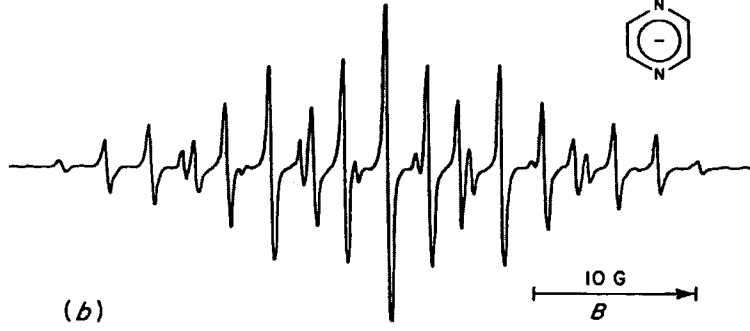
Şekil 4.6 298 K de dimetiloksitan'daki naftalin anyon radikalının X-band spektrumu. Deneyisel spektrumda herbir çizginin üzerinde bulunan sayılar, bu çizginin karşı geldiği spin durumu için dejenerasyon sayısını ifade etmektedir. Bu sayılar nispi olarak birinci dereceden türev genlikleridir.[14]



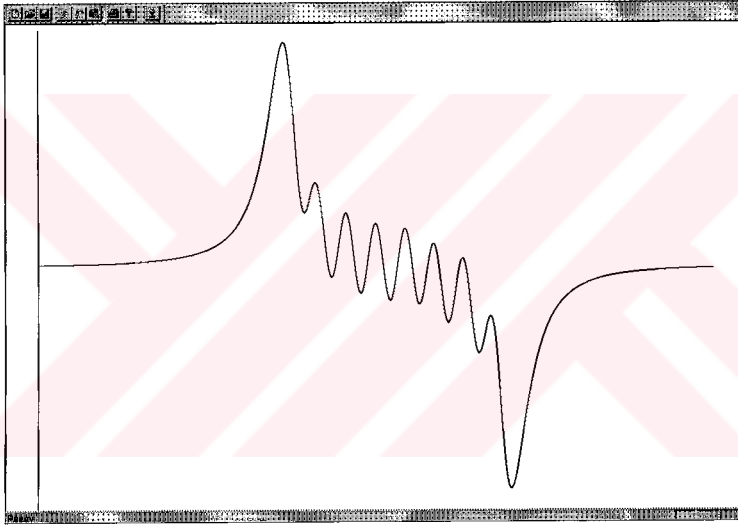
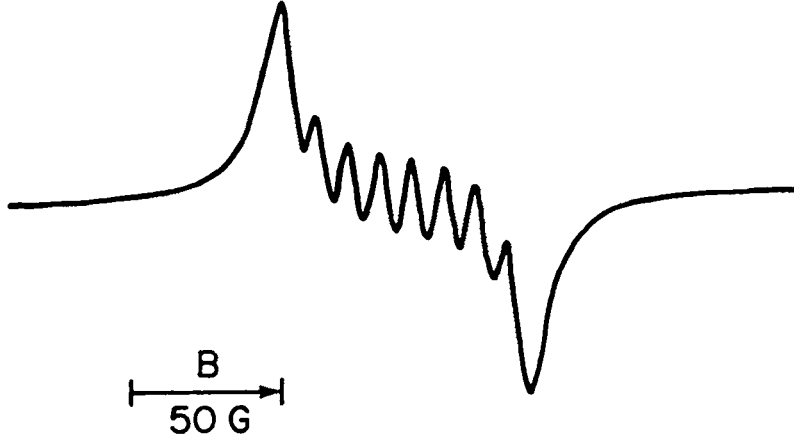
Şekil 4.7 295 K'daki dimetiloksitan içerisinde bulunan antrasin anyon radikalinin X-band EPR spektrumunun düşük alan kısmı. Deneysel olan spektrumda protonlardan kaynaklanan ince yapı çizgileri numaralandırılmış, ^{13}C yarılmalarından kaynaklanan çizgiler ise numaralandırılmamıştır. Üç farklı protondan yarıлма sabitleri de şekil üzerinde gösterilmiştir. [15]



Şekil 4.8 295 K da benzen içerisinde yerleştirilen bir nitronitroksit radikalının X-band EPR spektrumu. Burada iki özdeş azot atomundan kaynaklanan yarılmalar gözükmemektedir. Diğer hidrojen atomlarının hiçbirisi, farkedilir büyüklükte bir ince yapı etkileşimi vermemiştir.[16]



Şekil 4.9 Na^+ nın karşıt iyon olarak bulunduğu 297 K deki dimetiloksitan içerisinde bulunan parazin anyon radikalinin X-band EPR spektrumu; Burada ^{23}Na 'ün neden olduğu dörtlü yarılmalar farkedilmektedir.[17] b) Karşıt iyon olarak Na^+ yerine K^+ kullanıldığı zaman, dörtlü yarılmalar ayırd edilemez duruma gelmektedir. Çünkü K^+ çekirdeğinin manyetik momenti, Na^+ unkinden yaklaşık dört kat daha küçüktür.[18]



Şekil 4.10 228 K deki tolüen çözeltisindeki $(p\text{-NHCOCH}_3)_1\text{TPP Co(NMeIm)O}_2$ 'nin X-band EPR spektrumu. Burada TPP, tetrafenilporipirin 'i, MeIm ise metilimidazol 'ü göstermektedir.[19]

SONSÖZ

EPR spektrumu alınan bir örnek içerisindeki eşlenmemiş elektronların, etraflarında bulunan sıfırdan farklı manyetik momente sahip çekirdeklerle yaptıkları etkileşimlerin bolluk miktarı, numune içerisindeki eşlenmemiş elektronların sayısı ile doğru orantılıdır. Bununla birlikte dikkate alınması gereken bir kaç nokta daha vardır. Eşlenmemiş elektronun etkileştiği çekirdeklerin tümü veya bir kısmı, elektronun bağlı bulunduğu ligant içerisinde değilse ve bunların elektronun bulunduğu ligant ile yaptıkları komşulukların sayısı ve /veya yakınlıkları değişebiliyorsa, bu durumda numuneden elde edilecek spektrum; (kesikli veya sürekli değişim gösteren) muhtemel komşuluk durumlarının herbiri için o durumun nisbi bolluğu ile orantılı yarı eğri genişliklerine sahip alt rezonans eğrilerinin süperpozisyonu şeklinde olacaktır. Benzeri bir durum, eşlenmemiş elektronların birden çok tipte (farklı orbitallerde veya farklı ligantlarda bulunan) olduğu zaman ortaya çıkar. Simülasyon programı bu tür durumları da kapsayacak şekilde geliştirilmek istenirse, herbir eşlenmemiş elektron cinsi için nisbi bolluk miktarının ve aşırı ince yapı parametrelerinin ayrı ayrı girilmesi gerekir. Ayrıca yukarıda bahsedilen tek tip eşlenmemiş elektronun farklı nisbi bolluktaki çekirdeklerle etkileştiği durum da gözönüne alınacak olursa, herbir eşlenmemiş elektronun etkileştiği çekirdeklerin nisbi bolluk miktarlarının da işlenmesi gerekir.

Aşırı ince yapı etkileşimlerini temel alan daha etkin bir simülasyonun, taşınması gereken bir başka özellik; yerel alanların dış manyetik alana göre, dikkate değer bir büyüklüğe ulaştığı durumlarda rezonans noktalarını hesaplarken, ikinci ve daha yüksek mertebeden pertürbasyon terimlerini de dikkate almasıdır.

Burada yapılan programa dışarıdan girilen ince yapı sabitlerini, kuantum kimyasal nümerik hesap teknikleriyle çalışarak kendi içerisinde hesaplayacak ayrı bir modülün eklenmesi, program için bir diğer gelişim yönü olabilir.

Genel Spin hamiltoniyeninde (denklem 2.9) Zeeman ve Aşırı İnce Yapı hamiltoniyenlerinden başka, dört etkileşim hamiltoniyeni daha bulunduğu göz önüne alınırsa, bütün etkileşimlerin nümerik olarak modellenmesinin, hiç de kolay olmadığı görülür. Bu gerçekten yola çıkarak, ‘analitik ve nümerik yöntemler beraberce kullanılmalı ve hangisiyle nereye kadar gidileceğine performansın optimizasyonu açısından karar verilmelidir’ diyebiliriz.

KAYNAKLAR

- 1- W. Gerlach and O. Stern, Z. Phys., **8**, 110 (1921); *ibid.*, **9**, 349, 353 (1922).
- 2- G. Breit I. I. Rabi, Phys. Rev., **38** 2082 (1931).
- 3- E. Zavoisky, J. Phys. U.S.S.R., **9**, 211, 245 (1945); *ibid.*, **10**, 170 (1946).
- 4- N. F. Ramsey, Bull. Magn. Reson., **7**, 94 (1985).
- 5- H. Dehmelt, Am. J. Phys., **58**, 17 (1990).
- 6- R. Skinner and J. A. Weil, Am. J. Phys., **57**, 777 (1989).
- 7- E. Fermi, Z. Phys., **60**, 320 (1930).
- 8- John A. Weil, James R. Bolton, John E. Wertz, Electron Paramagnetic Resonance. Sayfa, **59**.
- 9- J. L. Hall and R. T. Schumacher, Phys. Rev., **127**, 1892 (1962).
- 10- James R. Bolton, Mol. Phys., **6**, 219 (1963).
- 11- John A. Weil, James R. Bolton, John E. Wertz, E. P. Resonance. Sayfa, **60**.
- 12- R. Livingston and H. Zeldes, J. Chem. Phys., **44**, 1245 (1966).
- 13- D. H. Levy and R. J. Myers, J. Chem. Phys., **41**, 1062 (1964).
- 14- John A. Weil, James R. Bolton, John E. Wertz, E. P. Resonance. Sayfa, **64**.
- 15- James R. Bolton and G. K. Fraenkel, J. Chem. Phys., **40**, 3307 (1964).
- 16- J. H. Osiecki and E. F. Ullman, J. Am. Chem. Soc., **90**, 1078 (1968).
- 17- J. dos Santos-Veiga and A. F. Neiva-Correia, Mol. Phys., **9**, 395 (1965).
- 18- A. Carrington and J. dos Santos-Veiga, Mol. Phys., **5**, 21 (1962).
- 19- F. A. Walker and J. Bowen, J. Am. Chem. Soc., **107**, 7632 (1985).

ÖZGEÇMİŞ

1965 yılında Sivas'ta doğdu. 1982 de Sivas Lisesinden mezun olduktan sonra, İ.T.Ü. Uçak ve Uzay Bilimleri, Uçak Mühendisliği bölümünde yüksek öğrenimine başladı. Dört yıl sonra bu bölümden uçak mühendisi ünvanı ile mezun oldu. Mezuniyetten sonra Eskişehir Hava İkmal ve Bakım Merkez Komutanlığı'nda dokuz ay süreyle F-4 uçaklarının sivil bakım mühendisi olarak çalıştı. Buradan aralık 1987 de askerlik görevini yerine getirmek üzere ayrıldı. Askerlik görevini Deniz Kuvvetleri'ne bağlı Cengiz Topel Deniz Hava Üs Komutanlığında mütercim asteğmen olarak tamamlayıp, mart 1989 da terhis oldu. Boğaziçi Üniversitesinde aldığı 6 aylık hardware kursundan sonra, bir bilgisayar şirketinde 1 yıl teknik servis elemanı olarak çalıştı. Müteakip beş yıl aile mesleği olan konfeksiyonla ilgilendikten sonra, 1996 yılında GYTE fizik bölümünde yüksek lisans öğrenimine başladı. Halen aynı bölümde araştırma görevlisi olarak çalışmaktadır.