

T.C.
DİCLE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
(FİZİK ANABİLİM DALI)

BLOCH DENKLEMLERİNİN FORTRAN'DA NÜMERİK YÖNTEMLERLE ÇÖZÜMÜ

T.C. YÜKSEK ÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

NİL ERTEKİN BİNBAY

YÜKSEK LİSANS TEZİ

109508


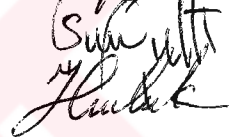

109508

DİYARBAKIR
Eylül-2001

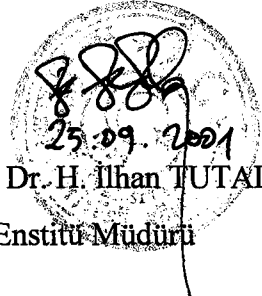
T.C.
DİCLE ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü Müdürlüğüne
DİYARBAKIR

Bu çalışma, Jürimiz tarafından Fizik Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

Jüri Üyesinin Ünvanı	Adı Soyadı	İmza
Başkan: Prof. Dr.	Ali YILMAZ	
Üye: Doç. Dr.	Şemsettin OSMANOĞLU	
Üye: Yrd. Doç. Dr.	Hatice BUDAK	

Yukarıdaki bilgilerin doğruluğunu onaylarım.


Prof. Dr. H. İlhan TUTALAR
Enstitü Müdürü

TEŐEKKÜR

Çalıőmalarım boyunca benden desteęini esirgemeyen danıőman Hocam Sayın Yrd.Doç. Dr. Hatice BUDAK' a teőekkürlerimi sunarım. Ayrıca tezimin hazırlanması sürecinde bilgisayar konusundaki bilgilerini benimle paylaőan sevgili eőim Veysel BİNBA Y'a teőekkür ederim.

İÇİNDEKİLER

Sayfa No

AMAÇi

ÖZETii

SUMMARY.....iii

BÖLÜM 1.

1.1.GİRİŞ.....1

BÖLÜM 2.

TEMEL NMR (NÜKLEER MAGNETİK REZONANS) TEORİSİ

2.1. Magnetik Rezonansın Çalışma Alanı3

2.2. Yalıtılmış Spinlerin Hareketi – Klasik İnceleme3

2.3. Yalıtık Olmayan Spin Sisteminde Hareket Denklemi6

2.3.1. Bloch Denklemleri8

2.3.1.1. Bloch Denklemlerinin Çözümleri10

2.3.1.2. Döner Koordinat Sisteminde Durum17

2.3.2. Durulma Zamanları24

BÖLÜM 3.

MATERYAL ve METOT

3.1. Adi Diferansiyel Denklemler İçin Başlangıç Değer Problemi	27
3.1.1 Problem	27
3.1.2. Çözüm İçin Varlık ve Teklik Koşulları	28
3.1.3. Nümerik Yöntemlerin Prensipleri	29
3.1.4. Tek Adımlı Metotlar	32
3.1.4.1. Euler – Cauchy Poligon Metodu.....	32
3.1.4.2. Düzeltilmiş Euler – Cauchy Metodu.....	33
3.1.4.3. Taylor Serisi Metodu	34
3.1.4.4. Heun'un Tahmin Edici – Düzeltici Metodu	35
3.1.4.5. Runge –Kutta Metotları	36
3.1.4.5.1 Runge-Kutta Metotlarının Yapısı	36
3.1.4.5.2. Klasik Runge-Kutta Metodu ($m=4$)	37
3.1.4.5.3. Runge-Kutta Metotları İçin Liste	39
3.1.4.5.4. İç İç Gömülmüş Runge-Kutta Prosedürleri.....	44
3.1.4.6. Tek Adımlı Metotların Tutarlılığı ve Uyumu	45
3.1.4.7. Hata Hesabı ve Adım Aralığı Kontrolü	46
3.1.4.7.1. Hata Hesabı	46
3.1.4.7.1.1. Euler-Cauchy Metodu	47
3.1.4.7.1.2. Heun'un Metodu ve Düzeltilmiş Euler Cauchy Metodu	48
3.1.4.7.1.3. Klasik Runge Kutta Metodu	48
3.1.4.7.2. Otomatik Adım Aralığı Kontrolü; Başlangıç Değer Problemi için Uyarlamalı Metotlar	48
3.1.4.7.2.1. Adım Aralığını Kontrol Etmek için Metot A	48

3.1.4.7.2.2. Adım Aralığını Kontrol Etmek için Metot B	49
3.1.5. Çok Adımlı Metotlar	50
3.1.5.1. Çok Adımlı Metotların Prensibi	50
3.1.5.2. Adams-Bashforth Metodu	52
3.1.5.3. Adams-Moulton Tahmin Edici-Düzeltilici Metodu	55
3.1.6. Kararlılık	58
3.1.7. Uygun Metot Seçimi	59
3.2. FORTRAN (FORMula TRANslation) Programlama Dili	61
3.3. Bloch Denklemlerinin Nümerik Çözümünü Yapan FORTRAN Programı	63
BÖLÜM 4.	
SONUÇLAR ve TARTIŞMA	65
EK-1 : BLOCH Denklemlerinin Çözümünü Veren FORTRAN Programı	68
KAYNAKLAR	79
TABLO LİSTESİ	80
ŞEKİL LİSTESİ	81
ÖZGEÇMİŞ	82

AMAÇ

Bilindiği üzere günümüzde bilim ve teknoloji birbirlerinin gelişimlerini karşılıklı ivmelendirerek ve tamamen iç içe geçmiş bir şekilde baş döndürücü ilerlemelerini sürdürmektedirler. Teknolojinin en çarpıcı ve hayatın her alanında olduğu gibi bilimsel alanda da en yaygın kullanılan ürünü tartışmasız günden güne gelişen bilgisayarlardır. Günümüzde artık neredeyse tüm teorik çalışmalar hatta deneylerin çoğu, sanal ortamlarda simülasyonlarla gerçekleştirilip; teorilerin dayattığı ve analitik yöntemlerle içinden çıkılması imkansız gibi görünen karmaşık denklemlerin bir çoğu, güçlü ve hızlı bilgisayarların varlığıyla anlam bulmuş olan nümerik yöntemler sayesinde etkin bir şekilde çözülebilmektedir.

Bu çalışmanın amacı daha önce analitik yöntemlerle çözülememiş bir problemi çözmek ya da nümerik yöntem literatürüne yeni bir yöntem katmaya çalışmaktan çok, gelecekte daha üst düzey çalışmalarda kaçınılmaz olarak karşılaşılabilecek olan bu yöntemlerin NMR (Nükleer Magnetik Rezonans)'ın temel denklemleri olan Bloch Denklemleri gibi fiziksel bir probleme nasıl uygulanacağı, temel prensipleri ve amaca yönelik üretilen algoritmaların bilgisayarda nasıl işleneceği konularını pekiştirip, bu çerçevede üst düzey çalışmalara bir platform oluşturabilmektir.

ÖZET

Bu çalışmada NMR'ın temel denklemlerinden olan " Bloch Denklemleri " nin nümerik yöntemlerle çözümü yapılmıştır. Bunun için önce probleme uygun nümerik algoritmalar ve yöntemler incelenmiş ve tanıtılmıştır. Bloch denklemlerinin çözümü problemi, adi diferansiyel denklemler için bir başlangıç değer problemidir. Bu probleme ilişkin tek adımlı ve çok adımlı nümerik yöntemler ile tahmin - düzeltme algoritmaları incelenmiştir.

Problemi çözmek için 4. mertebeden Runge – Kutta metodu (klasik Runge – Kutta metodu) kullanılmıştır. Aynı zamanda çözümü veren algoritma kullanılarak FORTRAN programlama dilinde Bloch denklemlerinin çözümünü yapan bir bilgisayar programı yapılmıştır.

Ayrıca Bloch denklemlerinin analitik çözümleri yapılmış ve sonuçlar tartışılmıştır.

SUMMARY

In this study, the Bloch Equations have been solved by using numerical algorithms. These equations are known as fundamental equations for NMR theory.

To do this, numerical algorithms have been discussed firstly. The problem of solving the Bloch equations is an initial value problem for ordinary differential equations. For solving this problem numerically one step methods, multi step methods and predictor – corrector algorithms have been considered and compared.

The fourth order Runge – Kutta method (which is also known as classical Runge – Kutta algorithm) have been used for solving Bloch equations numerically. And at last a computer programme has been written in FORTRAN language which solves Bloch equations by using 4. order Runge – Kutta method.

Bloch equations have also been solved by analytically and results have been discussed.

BÖLÜM 1.

1.1. GİRİŞ

Nükleer Magnetik Rezonans (NMR), kuvvetli bir magnetik alana yerleştirilen atomik çekirdeklerin karakteristik bir frekansta ışınım soğurmaları esasına dayanır. NMR olayı, ilk olarak 1946 yılında birbirinden bağımsız iki grup tarafından birkaç gün ara ile gözlemlendi. Stanford Üniversitesinde Bloch liderliğindeki grup (Bloch, Hansen, Packard) parafin, Harvard Üniversitesinde ise Purcell liderliğindeki grup (Purcell, Torrey ve Pound) suda NMR'ı gözlediler. Bu gözlemler Physical Review dergisinin aynı sayısında yayımlandı ve 1952 yılında Bloch ve Purcell'e Nobel ödülü kazandırdı. O zamandan beri NMR fizik, kimya, biyoloji, tıp ve malzeme bilimleri gibi bir çok dalda büyük etki yaptı. NMR spektrometreleri sıvı ve katıların yapısal, analitik ve dinamik incelemelerinin yapıldığı bir çok araştırma laboratuvarının vazgeçilmez aletlerinden biri oldu.

Başlangıçta sadece çekirdeklerin magnetik momentlerini saptamak için kullanılan bir teknik olan NMR'ın gelişme süreci içerisinde önemli bir adım 1951 yılında Arnold ve arkadaşları tarafından etilalkolün spektrumunun üç çizgiden (metil, metilen, hidroksil) oluştuğunun gözlenmesi oldu. Farklı kimyasal çevredeki çekirdeklerin farklı frekanslarda rezonansa gelmeleri nedeni ile kimyasal kayma diye adlandırılan bu parametre bundan sonra organik moleküllerin yapılarının araştırılmasında yaygın olarak kullanılmaya başlandı. 1953 yılında ilk ticari NMR spektrometresi yapıldı. Proton NMR bundan sonra organik kimyanın vazgeçilmez bir yöntemi oldu. Sıvılarda hidrojen bağlarının oluşumu, iyon değiş tokuşu, fonksiyonel grupların kimyasal reaksiyonları ve spektrumda sıcaklığa bağlı değişiklikler sonucunda moleküler hareketlerin karakterlerinin ortaya çıkarılması ve tabii ki yapı analizi bu alandaki çalışmalardan bazılarıdır.

NMR'da son yirmi yılda bilgisayar teknolojisindeki gelişmeler, süperiletken magnetlerin yapımı, sürekli dalga tekniklerinden puls tekniklerine geçilmesi ve Fourier Transform tekniklerinin geliştirilmesi ile devrim sayılabilecek gelişmeler oldu. Aynı zamanda geliştirilen mini puls spektrometreleri ile NMR, endüstride de büyük bir uygulama alanı buldu. NMR'ın son yıllardaki en önemli uygulamalarından biri de şüphesiz NMR görüntülemenin (MRG) bulunması oldu. İlk iki boyutlu NMR görüntüsü 1973 yılında Lauterbur tarafından elde edildi. Görüntüsü elde edilecek cisme uzaysal çözücülük elde etmek için farklı doğrultularda alan gradyenti uygulanarak birden fazla farklı, tek boyutlu projeksiyonlar elde edildi ve bunlar bir bilgisayar tarafından cismin iki boyutlu görüntüsünü

vermek üzere birleştirildi. MRG iki ve üç boyutlu görüntü elde etmek için geliştirilen bir çok yöntemle bu gün tıpta tanı alanında X ışınları bilgisayar tomografisi (XCT) yanındaki yerini aldı.(1,2,3,4,5)

Nümerik analiz yöntemlerinin temelleri ise, özellikle çözümü zorlaştıran fonksiyonlara yaklaşımlar yaparak çözülebilir hale getirmeye çalışan (seri açılımları gibi) yöntemlerle eş zamana dayanır. Ancak ivmeli gelişimini ilk transistörlerin üretilmesi ve ardından gitgide hızlanan ve küçülen işlemcileri ile bilgisayarların geliştiği dönemde sergileyen analiz metodlarıdır. Nümerik analiz ve bilgisayarların kullanımı, her alanda olduğu gibi NMR da da kaçınılmaz bir yer edinmeye başlamıştır. Günümüzde bir çok teori gerçek koşullara sahip bir laboratuvaradan önce sanal ortamlarda sümülasyonlar kullanılarak test edilmektedir. Nümerik yöntemlerle bir probleme çözüm aramak araştırmacılar için kesinlikten ve genellikten ödün vermek anlamına gelse de; uygulamalar göstermiştir ki, yöntemlerin sonuçları oldukça yeterli olabilmektedir.



BÖLÜM 2.

TEMEL NMR (NÜKLEER MAGNETİK REZONANS) TEORİSİ

2.1. Magnetik Rezonansın Çalışma Alanı

Magnetik rezonans, magnetik momentleri ve açısal momentumları olan magnetik sistemlerde gözlenebilecek bir olaydır. Rezonans sözcüğü, söz konusu bir magnetik sistemin doğal frekansına uyumlu olma durumunu anlatır ve bu durum durgun bir dış magnetik alanda magnetik momentin jirokopik presesyon hareketine karşılık gelir. Atomik spektrumun belirgin frekanslar arasındaki benzerliği ve magnetik rezonans frekanslarının tipik olarak çekirdek için radyo frekans bölgesine, elektron spinleri için ise mikrodalga frekansı bölgesine düşmesi nedeni ile çoğu zaman “ radyo frekans spektroskopisi ” veya “mikrodalga spektroskopisi” terimleri kullanılır.

Rezonans yönteminin üstünlüğü, ilgilenilen örnekte, bütünle karşılaştırıldığında çok zayıf olabilen bir katkıyı seçip ayırmayı mümkün kılmasıdır. Bunun yanında rezonans, kesin, oldukça ayrıntılı ve başka yollarla elde edilemeyen tipte magnetik bilgi toplanmasını da mümkün kılar. Magnetik rezonansın fizikle kaynaşmasının nedenlerinden biri de, atomik düzeydeki süreçler hakkında bilgi verebilmesidir. (6)

2.2. Yalıtılmış Spinlerin Hareketi-Klasik İnceleme

Temel kuramın incelenmesine, zamanla değişebileceğini de varsayabileceğimiz bir \vec{H} dış magnetik alanı içinde bir spinin hareketini klasik olarak tanımlamakla başlıyoruz . \vec{H} , $\vec{\mu}$ magnetik momenti üzerine $\vec{\mu} \times \vec{H}$ kadar bir tork etki ettirecektir. Eğer yataklara oturtulmuş ve isteğine göre dönebilen herhangi bir adi çubuk mıknatıs bir magnetik alan uygularsak mıknatıs \vec{H} boyunca yönelmeye çalışır. \vec{H} zamana göre sabit ve yataklar sürtünmesiz olursa mıknatıs denge durumu etrafında titreşimler yapar. Eğer yataklar sürtünmeli olursa mıknatıs yataklara enerji vereceğinden titreşimler söner ve en sonunda \vec{H} boyunca yönelir.

Mıknatıs aynı zamanda bir açısal momentuma da sahip olduğu zaman durum değişiktir ve o zaman bir jirokop gibi davranır. İlerde görüleceği gibi yatakların sürtünmesiz olması halinde moment \vec{H} ile sabit açı yaparak (\vec{H} 'nin zamana göre sabit olması koşuluyla)

\vec{H} etrafında presesyon hareketi yapar. Enerjinin, potansiyel ve kinetik enerji arasında ileri geri gidip gelmesi olmaz. Bununla birlikte yataklar sürtünmeye sahip olursa mıknatıs en sonunda durgun \vec{H} alanına paralel olur. Görüleceği üzere sürtünme, T_1 gibi durulma sürecine karşılık gelir. Mıknatısın hareket denklemi, torku \vec{J} açısal momentumundaki değişikliğe eşitleyerek bulunur.

$$\frac{d\vec{J}}{dt} = \vec{\mu} \times \vec{H} \quad (2.1)$$

$\vec{\mu} = \gamma \vec{J}$ olduğundan \vec{J} 'yi aradan çıkarabilir ve

$$\frac{d\vec{\mu}}{dt} = \vec{\mu} \times (\gamma \vec{H}) \quad (2.2)$$

elde ederiz.

\vec{H} 'nin zamana bağlı olup olmadığına bakmaksızın geçerli olan bu eşitlik bize herhangi bir anda $\vec{\mu}$ 'deki değişmelerin hem $\vec{\mu}$ ve hem de \vec{H} ya dik olduğunu söylemektedir.

Denklem (2.2)'nin çözümüne diferansiyel denklemlerin standart yöntemleri ile ve \vec{H} 'nin varsayılan zamana bağılıklarının çeşitli şekilleri halinde devam edilebilir. Ancak biz dönen koordinat sistemine geçmeyi tercih edeceğiz.

Zamanın fonksiyonu olan bir $\vec{F}(t)$ vektörü düşünelim, bu vektörü bir dik koordinat sisteminin eksenleri boyunca $F_x(t), F_y(t), F_z(t)$ bileşenleri cinsinden yazabiliriz. Eksen sistemine karşılık gelen birim vektörleri $\hat{i}, \hat{j}, \hat{k}$ cinsinden

$$\vec{F} = iF_x + jF_y + kF_z \quad (2.3)$$

yazarız. Yaygın olarak \hat{i}, \hat{j} ve \hat{k} zamana göre sabit düşünülür, fakat biz daha genel olmayı seçeceğiz. Uzunlukları sabit olan bu vektörler en çok dönme hareketi yapabilirler. Bunların Ω ani açısal hızıyla döndüklerini varsayalım. O zaman

$$\frac{d\hat{i}}{dt} = \vec{\Omega} \times \hat{i} \quad (2.4)$$

olduğundan \vec{F} 'nin zamana göre türevi

$$\frac{d\vec{F}}{dt} = \hat{i} \frac{dF_x}{dt} + F_x \frac{d\hat{i}}{dt} + \hat{j} \frac{dF_y}{dt} + F_y \frac{d\hat{j}}{dt} + \hat{k} \frac{dF_z}{dt} + F_z \frac{d\hat{k}}{dt} \quad (2.5)$$

$$= \hat{i} \frac{dF_x}{dt} + \hat{j} \frac{dF_y}{dt} + \hat{k} \frac{dF_z}{dt} + \vec{\Omega} \times (\hat{i}F_x + \hat{j}F_y + \hat{k}F_z)$$

$$= \frac{\delta \vec{F}}{\delta t} + \vec{\Omega} \times \vec{F}$$

dir, burada $\frac{\delta \vec{F}}{\delta t}$ gösterimi \vec{F} 'nin $\hat{i}, \hat{j}, \hat{k}$ koordinat sistemine göre zamanla değişme miktarını

temsil etmek için kullanılmıştır. Örneğin $\frac{\delta \vec{F}}{\delta t} = 0$ olduğu zaman \vec{F} 'nin \hat{i}, \hat{j} ve \hat{k} boyunca bileşenleri zamanla değişmez.

Denklem (2.5)'i kullanarak $\vec{\mu}$ 'nün hareket denklemini keyfi $\vec{\Omega}$ açısal hızıyla dönen bir koordinat sistemi cinsinden yeniden

$$\frac{\delta \vec{\mu}}{\delta t} + \vec{\Omega} \times \vec{\mu} = \vec{\mu} \times (\gamma \vec{H}) \quad (2.6)$$

veya

$$\frac{\delta \vec{\mu}}{\delta t} = \vec{\mu} \times (\gamma \vec{H} + \vec{\Omega}) \quad (2.7)$$

şeklinde yazabiliriz. Denklem (2.7), $\vec{\mu}$ 'nün dönen koordinat sistemindeki hareketinin laboratuvar sistemindeki aynı denklemi, gerçek magnetik alan yerine etkin bir \vec{H}_{et} magnetik alanı

$$\vec{H}_{et} = \vec{H} + \frac{\vec{\Omega}}{\gamma}$$

yerleştirmek koşuluyla sağladığını söyler.

Durgun $\vec{H}_0 = \hat{k}H_0$ alanında $\vec{\mu}$ nün hareketini, $\vec{\Omega}$ 'yı $\vec{H}_e = 0$ olacak şekilde seçersek derhal çözebiliriz. Yani $\vec{\Omega} = -\gamma H_0 \hat{k}$ alırız. Bu görelî çerçevede $\delta\vec{\mu} / \delta t = 0$ olduğundan $\vec{\mu}$, \hat{i}, \hat{j} ve \hat{k} 'ya göre sabit kalır. Bundan ötürü laboratuvara göre hareket, bir eksen sisteminde sabit bir vektörün eksenlerle birlikte $\vec{\Omega} = -\gamma H_0 \hat{k}$ hızı ile dönmesidir. Başka bir deyişle $\vec{\mu}$ laboratuvara göre $\vec{\Omega} = -\gamma H_0 \hat{k}$ hızı ile döner. Açısal frekans γH_0 "Larmor frekansı" diye adlandırılır. Böylece magnetik rezonans olması için gereken, açısal frekansın klasik presesyon frekansı $\vec{\Omega}$ 'nın büyüklüğü ile aynı olması gerçeğine ulaşmış olduk.

Ayrıca toplam mıknatıslanma vektörü

$$\vec{M} = \sum \vec{\mu} \quad (2.9)$$

olduğundan (2.2) denklemi

$$\frac{d\vec{M}}{dt} = \vec{M} \times (\gamma \vec{H}) \quad (2.10)$$

şeklinde yazılabilir. (6,7)

2.3. Yalıtık Olmayan Spin Sisteminde Hareket Denklemi

Yalıtılmış spinlerin sabit bir magnetik alandaki hareketini inceledikten sonra, şimdi de birbirlerinden yalıtılmamış spinlerden oluşan bir spin sistemini göz önüne alalım. Bu sistemde yine $\vec{H}_0 = \hat{k} H_0$ olacak şekilde z doğrultusunda uygulanmış bir sabit magnetik alan olduğunu düşünelim. Ayrıca dönen koordinat sisteminin koordinatlarını da x', y', z' ile gösterelim.

Spin sistemindeki magnetik momentler, z' doğrultusunda uygulanan değişmez \vec{H}_0 alanı etkisi ile o şekilde yönelme kazanırlar ki, net mıknatıslanma vektörünün yalnızca z' doğrultusunda bir bileşeni olur. Bunu M_z ile gösterelim (şekil 2.1.a). O halde $(x' - y')$ düzleminde herhangi bir bileşen yoktur.

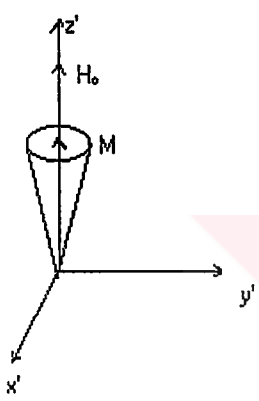
Sisteme belli bir süre için x' doğrultusunda bir \vec{H}_1 alanı uygulayalım. Bu alanı $\vec{H}_1 = \hat{i}H$ şeklinde gösterebiliriz. Ayrıca koordinat sisteminin $\Omega = -\omega \hat{k}$ hızıyla döndüğünü varsayarsak bu durumda (2.8) denklemindeki \vec{H}_e için

$$\vec{H}_{et} = \hat{i}H_1 + (H_0 - \frac{W}{\gamma})\hat{k} \quad (2.11)$$

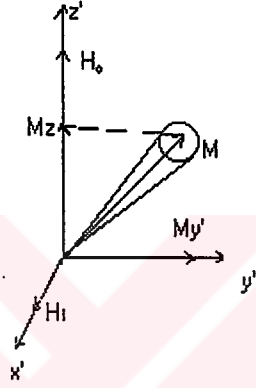
yazabiliriz.

Bu durumda magnetik momentler o şekilde bir yönelme gösterirler ki, mıknatıslanma vektörü y' doğrultusunda bir M_y' bileşenine sahip olur. (şekil 2.1.b)

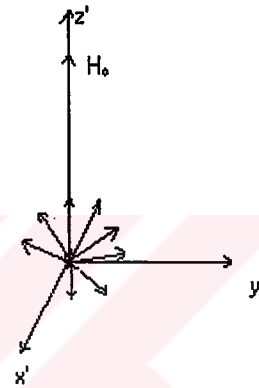
Şimdi sistem üzerinden \vec{H}_1 alanını kaldıralım ve magnetik momentlerin nasıl bir davranış içinde olduklarını arayalım.



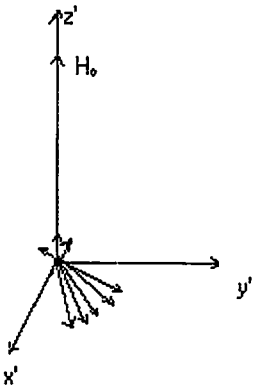
Şekil 2.1.a



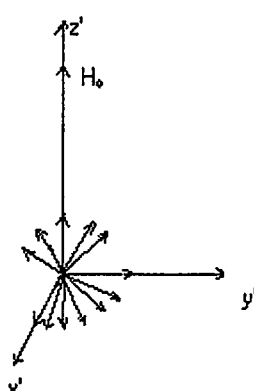
Şekil 2.1.b



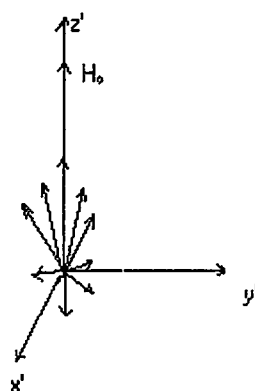
Şekil 2.1.c



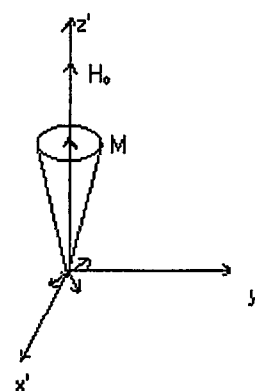
Şekil 2.1.d



Şekil 2.1.e



Şekil 2.1.f



Şekil 2.1.g

Magnetik dipoller yalıtık olmadıkları için kendi aralarında enerji alış-verişi içinde olacaklar ve $(x' - y')$ düzlemi içinde bir dağılım göstereceklerdir. Ayrıca, dipoller üzerine uygulanan dış magnetik alan yeter derecede homojen değilse magnetik momentlerin her biri birbirinden farklı magnetik alan göreceklerdir ve bunun sonucu olarak da farklı frekanslarda Larmor dönü hareketi yapacaklardır. İşte bu iki etki nedeni ile \vec{H}_1 alanı kalkar kalkmaz mıknatıslanma vektörünün M_y bileşeninde bir azalma oluşacaktır. Yani, spin – spin etkileşmesi denen bu etkileşme sonucu, M_y zamanla sifira yaklaşır (şekil 2.1.c,d,e,f).

Öte yandan magnetik dipoller yalıtık olmadıkları için örgü ile etkileşme içindedirler. Enerjilerinin bir kesimini örgüye aktararak yeniden ısı dengeye ulaşma eğilimi gösterirler. Spin – örgü etkileşmesi adını verdiğimiz bu etkileşme, mıknatıslanma vektörünün z' doğrultusundaki bileşeninin yeniden M_0 ısı denge değerine yaklaşmasını sağlar. Bir süre sonra M_z yeniden M_0 değerine ulaşırken M_y bileşeni de sifira yaklaşır (şekil 2.1.g).

O halde mıknatıslanma vektörünün hareketini incelerken durulma süreçlerinin de gözönünde tutulması gerekmektedir. Bu amaçla yapılan ilk çalışma Bloch'dan gelmiştir. (4,8,6)

2.3.1 Bloch Denklemleri

Yukarıda açıklandığı gibi mıknatıslanma vektörünün z doğrultusundaki bileşeninin ısı denge durumundaki değerine ulaşması, spin sistemi ile örgü sisteminin etkileşmesi sonucu ortaya çıkmaktadır. Oysa mıknatıslanma vektörünün z doğrultusundaki bileşeni, taban enerji düzeyindeki spin fazlalığı ile orantılıdır. Bu nedenle mıknatıslanma vektörünün z doğrultusundaki M_z bileşeninin zamana göre değişimi,

$$\frac{dM_z}{dt} = \frac{M_z - M_0}{T_1} \quad (2.12)$$

yazılabilir. Burada T_1 parametresine boyuna ya da spin-örgü durulma zamanı adı verilir. Bağlantıya göre herhangi bir nedenle ısı denge durumundaki değeri olan M_0 'a üstel fonksiyonla yaklaşır. Öte yandan mıknatıslanma vektörünün $\omega_0 = \gamma H_0$ frekansı ile dönen M_x ve M_y bileşenleri, H_0 'daki homojensizlik nedeni ile farklı dönü hareketinden ötürü,

zamanla sifira yaklaşırlar. M_x ve M_y 'deki sifira yaklaşma (2.12)' deki gibi üstel bir fonksiyon şeklinde varsayılırsa,

$$\frac{dM_x}{dt} = -\frac{M_x}{T_2}$$

$$\frac{dM_y}{dt} = -\frac{M_y}{T_2} \quad (2.13)$$

yazılabilir.

Buradaki T_2 parametresine enine ya da spin-spin durulma zamanı adı verilir. O halde yalıtık olmayan bir spin sistemini niteleyen hareket denklemi, yalıtık spinler için bulunan (2.10) denklemi ile durulma süreçlerini içeren (2.12) ve (2.13) denklemlerinin toplamı şeklindedir, yani xyz- Laboratuvar sisteminde yalıtık olmayan bir spin sisteminin hareket denklemi:

$$\frac{d\vec{M}}{dt} = \gamma (\vec{M} \times \vec{H}) - \frac{M_x \hat{i} + M_y \hat{j}}{T_2} - \frac{M_z - M_0}{T_1} \hat{k} \quad (2.14)$$

dır. Bu denklemi laboratuvar sistemindeki bileşenleri cinsinden yazarsak,

$$\frac{dM_x}{dt} = \gamma (\vec{M} \times \vec{H})_x - \frac{M_x}{T_2} \quad (2.15.a)$$

$$\frac{dM_y}{dt} = \gamma (\vec{M} \times \vec{H})_y - \frac{M_y}{T_2} \quad (2.15.b)$$

$$\frac{dM_z}{dt} = \gamma (\vec{M} \times \vec{H})_z - \frac{M_z - M_0}{T_1} \quad (2.15.c)$$

elde edilir. Bu denklemlere Bloch Denklemleri denir. Burada \vec{H} magnetik alanı, sistemin toplam olarak gördüğü magnetik alandır ve değişmez dış magnetik alan ile dairesel ya da çizgisel olarak kutuplanmış RF ya da MD alanının toplamından oluşmaktadır. Yani Bloch denklemlerindeki magnetik alan değişken bir alandır. (4,8,6)

2.3.1.1. Bloch Denklemlerinin Çözümleri

Dönen magnetik alanımız, laboratuvar koordinat sisteminde

$$\vec{H} = \hat{i} H_1 \cos \omega t + \hat{j} H_1 \sin \omega t \quad \text{şeklindedir.} \quad (2.16)$$

Bu alanın keza, $H_x = 2H_1 \cos \omega t$ şeklinde titreşim yapan, bir alanın uygun şekilde dönen bir alan bileşeninden meydana geldiğini de düşünebiliriz ve ters yönde dönen bileşenide

$$\hat{i} H_1 \cos \omega t - \hat{j} H_1 \sin \omega t \quad \text{olur.}$$

Magnetizasyonun x - bileşeni olan M_x 'in; H_x ile eş fazlı zıt ve zıt fazlı bileşenleri vardır. Kompleks sayılar, çözümde kolaylık sağladıkları için H_x 'i, $2H_1 e^{i\omega t}$ nin gerçel kısmı olarak düşünelim ve M_x 'i de

$$M_x = \chi 2H_1 e^{i\omega t}$$

nin gerçel kısmı olarak alalım. Bundan sonra, kompleks alınganlığı

$$\chi = \chi' - i\chi'' \quad (2.17)$$

olarak tanımlayalım. O halde;

$$\text{Re} M_x = M_x = \chi' \cdot 2H_1 \cos \omega t + \chi'' 2H_1 \sin \omega t \quad \text{olur.} \quad (2.18)$$

(2.16) daki dönen alanı H_0 alanına ilave edilmiş olarak, Bloch Denklemlerini tekrar yazalım;

$$\frac{dM_x}{dt} = \gamma(\vec{M} \times \vec{H})_x - \frac{M_x}{T_2} = \gamma(M_y H_z - H_y M_z) - \frac{M_x}{T_2} \quad (2.19.a)$$

$$\frac{dM_y}{dt} = \gamma(\vec{M} \times \vec{H})_y - \frac{M_y}{T_2} = \gamma(M_z H_x - H_z M_x) - \frac{M_y}{T_2} \quad (2.19.b)$$

$$\frac{dM_z}{dt} = \gamma(\vec{M} \times \vec{H})_z + \frac{M_0 - M_z}{T_1} = \gamma(M_x H_y - H_x M_y) + \frac{M_0 - M_z}{T_1} \quad (2.19.c)$$

veya $H_z = H_0$, $H_y = H_1 \sin \omega t$, $H_x = H_1 \cos \omega t$ yazarak ;

$$\frac{dM_x}{dt} = \gamma (H_0 M_y - M_z H_1 \sin \omega t) - \frac{M_x}{T_2} \quad (2.20.a)$$

$$\frac{dM_y}{dt} = \gamma (M_z H_1 \cos \omega t - H_0 M_x) - \frac{M_y}{T_2} \quad (2.20.b)$$

$$\frac{dM_z}{dt} = \gamma (M_x H_1 \sin \omega t - M_y H_1 \cos \omega t) + \frac{M_0 - M_z}{T_1} \quad (2.20.c)$$

buluruz.

Şimdi , bu denklemleri çözmek için;

$$M_+ = M_x + iM_y \quad \text{ve} \quad M_- = M_x - iM_y$$

yazalım ve

$$\frac{dM_+}{dt} = \frac{d(M_x + iM_y)}{dt}$$

eşitliğinin değerini arayalım :

$$\frac{dM_+}{dt} = \left[\gamma (H_0 M_y - M_z H_1 \sin \omega t) - \frac{M_x}{T_2} \right] + i \left[\gamma (M_z H_1 \cos \omega t - H_0 M_x) - \frac{M_y}{T_2} \right]$$

$$= \left[\gamma H_0 M_y - \gamma M_z H_1 \sin \omega t - \frac{M_x}{T_2} \right] + \gamma i (M_z H_1 \cos \omega t) - i \gamma H_0 M_x - \frac{M_y}{T_2}$$

$$= \gamma H_0 M_y - \gamma M_z H_1 \sin \omega t - \frac{M_x}{T_2} + i \gamma M_z H_1 \cos \omega t - i \gamma H_0 M_x - i \frac{M_y}{T_2}$$

$$= \gamma H_0 (M_y - i M_x) - \gamma M_z H_1 (\sin \omega t - i \cos \omega t) - \frac{M_x}{T_2} - i \frac{M_y}{T_2}$$

$$= i \gamma H_0 \frac{(M_y - i M_x)}{i} + \gamma M_z H_1 (i \cos \omega t - \sin \omega t) - \frac{M_x + i M_y}{T_2}$$

$$= i \gamma H_0 (-i M_y - M_x) + i \gamma M_z H_1 (\cos \omega t - \frac{1}{i} \sin \omega t) - \frac{1}{T_2} M_+$$

$$= -i \gamma H_0 (M_x + i M_y) + i \gamma M_z H_1 (\cos \omega t + i \sin \omega t) - \frac{1}{T_2} M_+$$

$$= -i \gamma H_0 M_+ + i \gamma M_z H_1 e^{i \omega t} - \frac{1}{T_2} M_+$$

$$\frac{dM_+}{dt} = \gamma (i M_z H_1 e^{i \omega t} - i H_0 M_+) - \frac{1}{T_2} M_+ \quad (2.21)$$

$$\frac{dM_-}{dt} = \gamma (-i M_z H_1 e^{-i \omega t} + i H_0 M_-) - \frac{1}{T_2} M_- \quad (2.22)$$

bulunur. $\frac{dM_+}{dt}$ ve $\frac{dM_-}{dt}$ denklemlerinin herbirindeki zamana bağlı

terimler $M_{\pm} = e^{\pm i \omega t} N_{\pm}$ yazmakla yok edilebilirler :

$$\frac{dM_+}{dt} = \frac{d}{dt} e^{i\omega t} N_+ = i\omega e^{i\omega t} N_+ \quad (2.23)$$

$$i\omega e^{i\omega t} N_+ = \gamma (iM_z H_1 e^{i\omega t} - iH_0 e^{i\omega t} N_+) - \frac{1}{T_2} e^{i\omega t} N_+$$

$$i\omega N_+ = i\gamma (M_z H_1 - H_0 N_+) - \frac{1}{T_2} N_+ \quad (2.24)$$

bulunur. Ve buradan N_{\pm} için ;

$$\pm i\omega N_{\pm} = i\gamma (\pm M_z H_1 \pm H_0 N_{\pm}) - \frac{1}{T_2} N_{\pm} \quad (2.25)$$

elde edilir.

Şimdi $\frac{dM_z}{dt} = 0$ şartını koyalım ve buna uygun bir çözüm arayalım. Bunun için,

$(M_x \sin \omega t - M_y \cos \omega t)$ nin M_+ ve M_- cinsinden eşdeğerini bulalım :

$$\begin{aligned} M_+ &= M_x + iM_y \\ M_- &= M_x - iM_y \end{aligned} \Rightarrow \begin{aligned} M_x &= \frac{1}{2}(M_+ + M_-) \\ M_y &= \frac{1}{2i}(M_+ - M_-) \end{aligned} \quad (2.26)$$

o halde ;

$$\begin{aligned} M_x \sin \omega t - M_y \cos \omega t &= \frac{1}{2}(M_+ + M_-) \sin \omega t - \frac{1}{2i}(M_+ - M_-) \cos \omega t \\ &= \left(\frac{1}{2}\right)M_+ \sin \omega t + \left(\frac{1}{2}\right)M_- \sin \omega t - \left(\frac{1}{2i}\right)M_+ \cos \omega t + \left(\frac{1}{2i}\right)M_- \cos \omega t \\ &= \left(\frac{1}{2i}\right)M_- \cos \omega t + \left(\frac{1}{2}\right)M_+ \sin \omega t - \left(\frac{1}{2i}\right)M_+ \cos \omega t + \left(\frac{1}{2}\right)M_- \sin \omega t \\ &= \left(\frac{1}{2i}\right)M_- (\cos \omega t + i \sin \omega t) - \left(\frac{M_+}{2i}\right)(\cos \omega t - i \sin \omega t) \end{aligned}$$

$$\begin{aligned}
&= \frac{M_-}{2i} e^{i\omega t} - \frac{M_+}{2i} e^{-i\omega t} \\
&= \frac{1}{2i} (M_- e^{i\omega t} - M_+ e^{-i\omega t})
\end{aligned} \tag{2.27}$$

$\frac{dM_z}{dt} = 0$ yazdığımız zaman, Bloch denklemlerinin üçüncüsü için;

$$\begin{aligned}
\frac{d_z}{dt} &= 0 \\
\frac{M_z - M_0}{T_1} &= \gamma H_1 (M_x \sin \omega t - M_y \cos \omega t) \\
&= \frac{\gamma H_1}{2i} (M_- e^{i\omega t} - M_+ e^{-i\omega t})
\end{aligned} \tag{2.28}$$

$$= \frac{\gamma H_1}{2i} (N_- - N_+)$$

yazılabilir. N_- ve N_+ nın değerlerini $\pm i\omega N_{\pm}$ den bulursak (mesela N_+ için)

$$i\omega N_+ = i\gamma M_z H_1 - i\gamma H_0 N_+ - \frac{N_+}{T_2}$$

$$i\gamma M_z H_1 = (i\omega + i\gamma H_0 + \frac{1}{T_2}) N_+$$

$$N_+ = \frac{i\gamma H_1 M_z}{i(\omega + \gamma H_0) + \frac{1}{T_2}} = \frac{\gamma H_1 M_z}{(\omega + \gamma H_0) - \frac{i}{T_2}} \tag{2.29}$$

$$N_- = \frac{-i\gamma H_1 M_z}{-i\omega - i\gamma H_0 + \frac{1}{T_2}} = \frac{\gamma H_1 M_z}{(\omega + \gamma H_0) + \frac{i}{T_2}} \tag{2.30}$$

olur. N_+ ve N_- birbirinin kompleks eşlenikleridirler. Şimdi $\gamma H_0 = -\omega_0$ yazarak

$$\frac{(M_z - M_0)}{T_1} \text{ değerini bulalım ;}$$

$$\frac{(M_z - M_0)}{T_1} = \frac{\gamma H_1}{2i} (N_- - N_+) = \frac{\gamma H_1}{2i} \left(\frac{\gamma H_1 M_z}{(w - w_0) + i/T_2} - \frac{\gamma H_1 M_z}{(w - w_0) - i/T_2} \right)$$

$$= \frac{\gamma^2 H_1^2 M_z}{2i} \left(\frac{w - w_0 - i/T_2 - w + w_0 - i/T_2}{(w - w_0)^2 + 1/T_2^2} \right)$$

$$= \frac{\gamma^2 H_1^2 M_z}{2i} \left(\frac{-2i/T_2}{(w - w_0)^2 + 1/T_2^2} \right)$$

$$= -\gamma^2 H_1^2 M_z \frac{1}{T_2 [(w - w_0)^2 + 1/T_2^2]}$$

$$\frac{M_z - M_0}{T_1} = -\gamma^2 H_1^2 M_z \frac{T_2}{1 + T_2^2 (w - w_0)^2} \quad (2.31)$$

$$M_z - M_0 = -\gamma^2 H_1^2 M_z \frac{T_1 T_2}{1 + T_2^2 (w - w_0)^2}$$

$$M_z = M_0 \frac{1 + T_2^2 (w - w_0)^2}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2}$$

ve $M_0 = \chi_0 H_0$ yazarak ;

$$M_z = \chi_0 H_0 \frac{1 + T_2^2 (w - w_0)^2}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} \quad (2.32)$$

Şimdi ; $M_x = (1/2)(M_+ + M_-) = (1/2)(N_+ e^{iwt} + N_- e^{-iwt})$ değerini bulalım;

$$M_x = \frac{1}{2} \left(\frac{\gamma H_1 M_z}{(w - w_0) - i/T_2} e^{iwt} + \frac{\gamma H_1 M_z}{(w - w_0) + i/T_2} e^{-iwt} \right)$$

$$= \frac{1}{2} \gamma H_1 M_z \left(\frac{e^{iwt}}{(w - w_0) - i/T_2} + \frac{e^{-iwt}}{(w - w_0) + i/T_2} \right)$$

$$= \frac{1}{2} \gamma H_1 M_z \left(\frac{\cos wt + i \sin wt}{(w - w_0) - i/T_2} + \frac{\cos wt - i \sin wt}{(w - w_0) + i/T_2} \right)$$

[$\Delta w = w - w_0$ yazarak ve $-\sin wt = \sin(-wt)$]

$$= \frac{1}{2} \gamma H_1 M_z \frac{2 \Delta w \cos wt - \frac{2}{T_2} \sin wt}{(w - w_0) + \frac{1}{T_2}}$$

$$= \frac{1}{2} \gamma H_1 \chi_0 H_0 \left(\frac{1 + T_2^2 (w - w_0)^2}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} \right) \left(\frac{2 \Delta w \cos wt + \frac{2}{T_2} \sin wt}{\frac{1}{T_2} (1 + T_2^2 (w - w_0)^2)} \right)$$

$$= \frac{1}{2} \chi_0 \gamma H_0 T_2 \frac{T_2 \Delta w 2 H_1 \cos wt + 2 H_1 \sin wt}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2}$$

$$M_x = \frac{1}{2} \chi_0 (\gamma H_0) T_2 \frac{T_2 (w - w_0) 2H_1 \cos wt + 2H_1 \sin wt}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} \quad (2.33)$$

bulunmuş olur. Bu sonucu (2.18) ile karşılaştırırsak ($\gamma H_0 = w_0$) :

$$M_x = \frac{1}{2} \chi_0 w_0 T_2 \frac{T_2 (w - w_0)}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} 2H_1 \cos wt +$$

$$\frac{1}{2} \chi_0 w_0 T_2 \frac{1}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} 2H_1 \sin wt \quad (2.34)$$

yazabiliriz. Benzer şekilde M_y için de

$$M_y = \frac{1}{2} \chi_0 (\gamma H_0) T_2 \frac{-T_2 (w - w_0) 2H_1 \sin wt + 2H_1 \cos wt}{1 + T_2^2 (w - w_0)^2 + \gamma^2 H_1^2 T_1 T_2} \quad (2.35)$$

elde edilebilir. Böylece M_z , M_x ve M_y için topluca (2.32), (2.33) ve (2.35) denklemleri ile ifade edilen çözümler bulunmuş olur. (7,6,3)

2.3.1.2. Dönen Koordinat Sisteminde Durum

Bloch denklemlerinin ($x'y'z'$) döner koordinatlardaki karşılığı:

$$\frac{dM_{x'}}{dt} = \gamma (\vec{M}' \times \vec{H}_{et})_{x'} - \frac{M_{x'}}{T_2} \quad (2.36.a)$$

$$\frac{dM_{y'}}{dt} = \gamma (\vec{M}' \times \vec{H}_{et})_{y'} - \frac{M_{y'}}{T_2} \quad (2.36.b)$$

$$\frac{dM_{z'}}{dt} = \gamma (\vec{M}' \times \vec{H}_{et})_{z'} - \frac{M_{z'} - M_0}{T_1} \quad (2.36.c)$$

şeklindedir. Burada \vec{H}_{et} alanı, (2.11) bağıntısında verildiği gibi,

$$\vec{H}_{et} = H_1 \hat{i} + (H_0 - \frac{w}{\gamma}) \hat{k}$$

ya da

$$\vec{H}_{et} = \frac{1}{\gamma} \left\{ (\gamma H_0 - w) \hat{k} + \gamma H_1 \hat{i} \right\}$$

$$\Delta w = \gamma H_0 - w = w_0 - w ; w_1 = \gamma H_1 \quad (2.37)$$

kısaltmaları yapılırsa,

$$\vec{H}_{et} = \frac{1}{\gamma} (\Delta w \hat{k} + w_1 \hat{i}) \quad (2.38)$$

bulunur. Bu bağıntı (2.36) eşitliklerinde yerine konursa,

$$\frac{dM_{x'}}{dt} = \left\{ \vec{M}' \times (\Delta w \hat{k} + w_1 \hat{i}) \right\}_{x'} - \frac{M_{x'}}{T_2} \quad (2.39.a)$$

$$\frac{dM_{y'}}{dt} = \left\{ \vec{M}' \times (\Delta w \hat{k} + w_1 \hat{i}) \right\}_{y'} - \frac{M_{y'}}{T_2} \quad (2.39.b)$$

$$\frac{dM_{z'}}{dt} = \left\{ \vec{M}' \times (\Delta w \hat{k} + w_1 \hat{i}) \right\}_{z'} - \frac{M_{z'} - M_0}{T_1} \quad (2.39.c)$$

elde edilir. Burada gerekli matematiksel işlemler yapılırsa,

$$\frac{dM_{x'}}{dt} = \Delta w M_{y'} - \frac{M_{x'}}{T_2} \quad (2.40.a)$$

$$\frac{dM_{y'}}{dt} = -\Delta w M_{x'} + w_1 M_{z'} - \frac{M_{y'}}{T_2} \quad (2.40.b)$$

$$\frac{dM_{z'}}{dt} = -w_1 M_{y'} - \frac{M_{z'} - M_0}{T_1} \quad (2.40.c)$$

bulunur. Bu diferansiyel denklem takımının analitik çözümü ancak bazı özel varsayımlarla bulunabilir. Bloch'un Yavaş Geçiş adını verdiği bu durumda; spin sisteminin döner koordinatlarda kararlı duruma ulaştığı varsayılmıştır. Yani, mıknatıslanma vektörünün kutuplanması ve z' doğrultusundan sapması çok yavaştır. Buna göre,

$$\frac{dM_{x'}}{dt} = \frac{dM_{y'}}{dt} = \frac{dM_{z'}}{dt} = 0 \quad \text{dır.} \quad (2.41)$$

O halde yukarıdaki bağıntı, yavaş geçiş ve dolayısıyla kararlı durum koşulunda,

$$\begin{aligned} -\frac{1}{T_2} M_{x'} + \Delta w M_{y'} &= 0 \\ -\Delta w M_{x'} - \frac{1}{T_2} M_{y'} + w_1 M_{z'} &= 0 \end{aligned} \quad (2.42)$$

$$-w_1 M_{y'} - \frac{1}{T_1} M_{z'} = -\frac{1}{T_1} M_0$$

şeklinde yazılabilir. Bu bağıntıların çözümünden,

$$M_{x'} = \Delta w M_0 w_1 \frac{T_2^2}{A}$$

$$M_{y'} = w_1 M_0 \frac{T_2}{A} \quad (2.43)$$

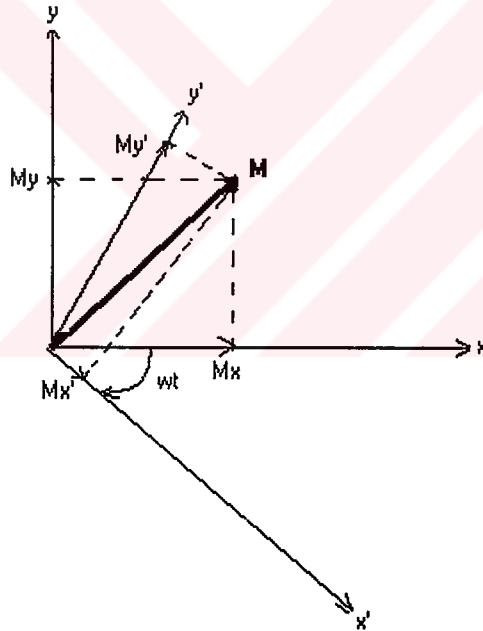
$$M_z = M_o (1 + \Delta w^2 T_2^2) / A$$

bulunur. Burada,

$$A = \{1 + \Delta w^2 T_2^2 + w_1^2 T_1 T_2\} \quad (2.44)$$

kısaltması yapılmıştır. Böylece mıknatıslanma vektörünün döner koordinatlardaki bileşenleri bulunmuş olur. Bu sonuçların, laboratuvar sistemindeki sonuçlarla aynı olduğunu gösterebiliriz.

Şimdi yeniden (xyz) Laboratuvar sistemine geçerek mıknatıslanma vektörünün M_x, M_y bileşenlerini bulmaya çalışalım. $x'y'z'$ sistemi, Laboratuvar sistemine göre saat yönünde dönsün (şekil 2.2).



Şekil 2.2

Döner koordinat sistemi

O halde :

$$M_x = M_x \cos wt + M_y \sin wt \quad (2.45.a)$$

$$M_y = -M_x \sin wt + M_y \cos wt \quad (2.45.b)$$

yazılır. (2.43)' deki değerler (2.45)' de yerine konur ve w_1 için (2.37) ve A için de (2.44)'deki değerler alınır, mıknatıslanma vektörünün laboratuvar sistemindeki bileşenleri daha önce de elde edilen , alışılmış şekilleri ile bulunur:

$$M_x = \frac{1}{2} \gamma M_0 T_2 \frac{T_2 \Delta w 2H_1 \cos wt + 2H_1 \sin wt}{\{1 + \Delta w^2 T_2^2 + \gamma^2 H_1^2 T_1 T_2\}} \quad (2.46)$$

$$M_y = \frac{1}{2} \gamma M_0 T_2 \frac{(-T_2 \Delta w 2H_1 \sin wt + 2H_1 \cos wt)}{\{1 + \Delta w^2 T_2^2 + \gamma^2 H_1^2 T_1 T_2\}} \quad (2.47)$$

$$M_z = M_0 \frac{1 + \Delta w^2 T_2^2}{\{1 + \Delta w^2 T_2^2 + \gamma^2 H_1^2 T_1 T_2\}} \quad (2.48)$$

Bu denklemler daha önce elde edilen (2.32), (2.33) ve (2.35) denklemleri ile aynıdır. Görülüyor ki, mıknatıslanma vektörünün z doğrultusundaki bileşeni M_z değişmez kalırken M_x ve M_y bileşenleri w açısal hızı ile z ya da H_0 doğrultusuna dik bir düzlem içinde dönmektedirler. Yani, M_x ve M_y bileşenlerinin zamana bağlı olması, bir makarada algılanabilecek bir voltaj oluşturabilmeleri anlamına gelir.

Öte yandan (2.43) ile verilen bağıntıların Bloch tarafından ilk kez türetilen ve kaynaklarda sık sık kullanılan karşılıkları,

$$M_x' = u = \frac{w_1 \Delta w M_0 T_2^2}{\{1 + \Delta w^2 T_2^2 + \gamma^2 H_1^2 T_1 T_2\}} \quad (2.49.a)$$

$$M_y' = -v = \frac{w_1 M_0 T_2}{\{1 + \Delta w^2 T_2^2 + \gamma^2 H_1^2 T_1 T_2\}} \quad (2.49.b)$$

şeklindedir.

Magnetik rezonansda deneysel olarak U ve V 'den herhangi biri ya da bu ikisinin çeşitli bileşimleri gözlenir. Eğer,

$$\Delta w = \frac{1}{T_2} \left\{ 1 + \gamma^2 H_1^2 T_1 T_2 \right\}^{\frac{1}{2}} \quad (2.50)$$

olacak şekilde seçilirse U en büyük değerine ulaşır, bu değer :

$$U_{\max} = \frac{\gamma H_1 T_2 M_0}{2 \left\{ 1 + \gamma^2 H_1^2 T_1 T_2 \right\}^{\frac{1}{2}}} = M_{x'\max} \quad (2.51)$$

olacaktır. Görülüyor ki bu değer H_1 değeri ile artmaktadır. Ayrıca ,

$$\gamma^2 H_1^2 T_1 T_2 \gg 1 \quad (2.52)$$

seçilirse,

$$U_{\max.} = \frac{M_0}{2} \left\{ \frac{T_2}{T_1} \right\}^{\frac{1}{2}} = M_{x'\max} \quad (2.53)$$

olacaktır. Öte yandan $\Delta w = 0$ için $|v|$ maksimum değer alır ve bu değer,

$$|v|_{\max.} = \frac{\gamma H_1 T_2 M_0}{1 + \gamma^2 H_1^2 T_1 T_2} = M_{y'\max} \quad (2.54)$$

olacaktır. U_{\max} 'un aksine, burada $|v|_{\max}$, H_1 alanı arttıkça artmaz ve,

$$\gamma^2 H_1^2 T_1 T_2 = 1 \quad (2.55)$$

seçilirse,

$$|v_{\max}| = \frac{M_0}{2} \left\{ \frac{T_2}{T_1} \right\}^{1/2} = M_{y_{\max}} \quad (2.56)$$

bulunur.

Bu sonuçlara göre, döner koordinat sisteminde mıknatıslanma vektörünün H_0 alanına dik bileşenleri olan M_x ve M_y aynı maksimum değere ulaşırlar. Ancak birinde maksimum değer,

$$H_1 \gg \frac{1}{\gamma} \sqrt{T_1 T_2}$$

koşuluna uyarken ötekinde,

$$H_1 = \frac{1}{\gamma} \sqrt{T_1 T_2}$$

koşuluna uyar.

M_x ve M_y bileşenlerinin yanında M_z bileşeni de rezonans durumunda ve

$$\gamma^2 H_1^2 T_1 T_2 \ll 1$$

koşulunda (yani küçük H_1 değerlerinde) M_0 'dan daha küçük değerler alır. Ancak H_1 arttıkça M_z azalır ve sonunda sifira yaklaşır. (8,4,5,6,)

2.3.2. Durulma Zamanları T_1 ve T_2

T_1 ve T_2 zamanları, döner koordinatlarda (2.40) bağıntıları ile verilen Bloch denklemlerinden yararlanarak tanımlanabilir.

Eğer sisteme bir 90° -Atması uygulanır ve rezonans durumunda 90° -Atması kesilirse (2.40) denklemleri,

$$\frac{dM_{x'}}{dt} = -\frac{M_{x'}}{T_2} \quad (2.57)$$

$$\frac{dM_{y'}}{dt} = -\frac{M_{y'}}{T_2} \quad (2.58)$$

$$\frac{dM_{z'}}{dt} = -\frac{M_{z'} - M_0}{T_1} \quad (2.59)$$

biçimini alırlar. 90° -Atmasının kesildiği anı $t=0$ olarak alalım.

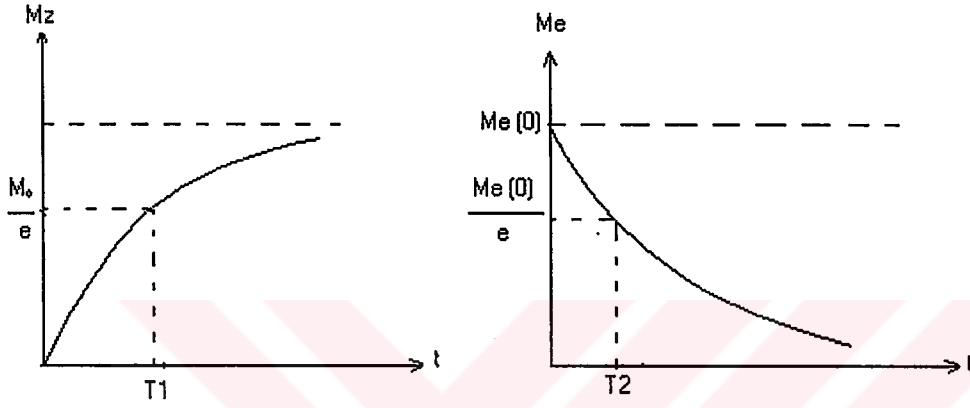
Eğer, $t=0$ 'da $M_z = 0$ seçilirse (2.59)'ün çözümü,

$$M_z = M_0(1 - e^{-t/T_1}) \quad (2.60)$$

olacaktır.

Bu bağıntı, spin-örgü durulma zamanına özdeş olan boyuna durulma zamanı T_1 'i tanımlamaya yeter. Yani, 90° -Atması uygulandıktan sonra mıknatıslanma tümüyle ($x'y'$) düzleminde yer aldığı için $M_z = 0$ 'dır. $t=0$ 'da 90° -Atması kesildikten sonra M_z 'nin yeniden M_0/e değerine ulaşması için geçen süreye T_1 boyuna durulma zamanı denir.

Bu şekilde 90° -Atması nedeni ile sıfıra düşen M_z 'nin, 90° -Atması kesildikten sonra yeniden M_0 denge durumundaki değerine ulaşması şekil 2.3'de görülmektedir. Bu eğri öte yandan, RF ya da MD kaynağından soğurulan enerjinin örgüye aktarılmasını belirlemektedir.



Şekil 2.3

90° atması kesildikten sonra M_z ve M_e nin durumu.

Ayrıca, $t = 0$ 'da yani, 90° -Atmasının kesildiği anda, mıknatıslanma vektörünün x' ve y' doğrultularındaki değerleri $M_{x'}(0)$ ve $M_{y'}(0)$ ise (2.57) ve(2.58) bağıntılarının çözümü;

$$M_{x'} = M_{x'}(0)e^{-t/T_1} \quad (2.61)$$

$$M_{y'} = M_{y'}(0)e^{-t/T_1} \quad (2.62)$$

olacaktır. O halde döner koordinat sisteminde toplam enine mıknatıslanmanın bir t anındaki değeri,

$$M_e = M_e(0)e^{-t/T_1} \quad (2.63)$$

olacaktır. Burada,

$$M_e(0) = \left\{ M_{x'}^2(0) + M_{y'}^2(0) \right\}^{1/2} \quad \text{dir.}$$

Spin-spin durulma zamanına özdeş olan enine durulma zamanı T_2 , 90° -Atması kesildikten sonra M_e 'nin $\frac{M_e(0)}{e}$ değerine düşmesi için geçen süre olarak tanımlanır.

Böylece döner koordinatlarda spin sisteminin rezonans durumunda iken, üzerine uygulanan RF ya da MD alanı kesildikten sonra mıknatıslanma vektörünün davranışını incelemiştir. Mıknatıslanma vektörünün aynı koşullar altında Laboratuvar sistemindeki davranışı da incelenebilir. Örneğin $t=0$ 'da, M_e Laboratuvar sisteminde x eksenini boyunca seçilmiş ise herhangi bir t anında enine bileşenlerin değeri, (2.61) ve (2.62) bağıntıları yardımı ile,

$$M_x = M_e(0)e^{-t/T_2} \cos \omega_0 t \quad (2.64)$$

$$M_y = M_e(0)e^{-t/T_2} \sin \omega_0 t \quad (2.65)$$

ve boyuna bileşen de (2.60) bağıntısında verildiği gibi,

$$M_z = M_0(1 - e^{-t/T_2}) \quad (2.66)$$

olacaktır.

Demek ki, durulma süreçlerinin etkisinde kalan magnetik sistemin Laboratuvar koordinatlarındaki hareketi, periyodik bir dönü hareketi yerine sönümlü bir harekettir. (5,8)

BÖLÜM 3.

MATERYAL ve METOT

3.1. ADİ DİFERANSİYEL DENKLEMLER İÇİN BAŞLANGIÇ DEĞER PROBLEMİ

3.1.1 Problem

Elimizde n adet 1.mertebeden tek bağımsız değişkenli adi diferansiyel denklem olduğunu düşünelim

$$Y_r, \quad r = 0, 1, 2, \dots, n$$

Bu durumda,

$$\underline{y} = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \cdot \\ \cdot \\ y_n(x) \end{bmatrix} \quad \text{ve} \quad \underline{f} = \begin{bmatrix} f_1(x, y_1, y_2, \dots, y_n) \\ f_2(x, y_1, y_2, \dots, y_n) \\ \cdot \\ \cdot \\ f_n(x, y_1, y_2, \dots, y_n) \end{bmatrix}$$

dersek $x \in [x_0, \beta]$, $\underline{y}(x_0) = \underline{y}_0$ olmak üzere

$$\underline{y}' = \underline{f}(x, \underline{y}) \quad (3.1)$$

denkleminin çözümü “başlangıç değer problemi” olarak bilinir. Burada $[x_0, \beta]$ aralığı “integrasyon aralığı” olarak adlandırılır. (9,10,11,12)

3.1.2. Çözüm İçin Varlık ve Teklik Koşulları

Aşağıdaki koşullar (3.1) denkleminin tekil bir çözümünün bulunabilmesi için sağlanması gerekli koşullardır.

(i) $(x, y_1, y_2, \dots, y_n)$ parametre uzayındaki bir D bölgesinde $f_r, r = 1, 2, \dots, n$ fonksiyonlarının sürekli olması;

(ii) f_r fonksiyonlarının, her $(\underline{x}, \underline{y}), (\underline{x}, \underline{y}^*) \in D$ için aşağıdaki Lipschitz şartını sağlaması.

$$\left\| \underline{f}(\underline{x}, \underline{y}) - \underline{f}(\underline{x}, \underline{y}^*) \right\| \leq L \left\| \underline{y} - \underline{y}^* \right\| \quad ; \quad L = \max_{\substack{1 \leq r, k \leq n \\ (\underline{x}, \underline{y}) \in D}} \left| \frac{\partial f_r}{\partial y_k} \right|$$

Bu durumda bir $\tilde{D} \subset D$ bölgesinin varlığı söz konusu olur ki bu bölgede (3.1) denklemini $\underline{y}(x_0) = \underline{y}_0$ başlangıç değeri için bir kesin çözüme sahiptir. n mertebeden bir diferansiyel denkleme ilişkin bir başlangıç değer problemi; n tane 1. mertebeden denklemleri içeren bir başlangıç değer problemine aşağıdaki şekilde dönüştürülebilir.

$$\begin{aligned} y^{(n)}(x) &= f(x, y, y', y'', \dots, y^{(n-1)}) \\ y(x_0) &= y_0, \quad y'(x_0) = y_0', \dots, y^{(n-1)}(x_0) = y_0^{(n-1)} \end{aligned} \quad (3.2)$$

olmak üzere

$y^k(x) \equiv y_{k+1}(x), k = 0, 1, 2, \dots, n-1$ şeklinde bir dönüşümle (3.2) sistemi n adet 1. mertebeden adi diferansiyel denklemleri içeren bir sisteme dönüştürülebilir. Burada başlangıç değerleri de

$$y^k(x_0) = y_{k+1}(x_0), \quad k = 0, 1, 2, \dots, n-1 \quad \text{olmuş olur. (9,13)}$$

3.1.3. Nümerik Yöntemlerin Prensipleri

Nümerik yöntemlerde genel olarak $[x_0, \beta]$ aralığı aşağıdaki şekilde parçalara bölünür;

$$x_0 < x_1 < x_2 < \dots < x_N = \beta$$

Bu bölüntüde herhangi iki ardışık değer arasındaki mesafeye, bu ardışık değerlere ilişkin 'yerel adım aralığı' denir. Yerel adım aralıkları ,

$$h_i \equiv x_{i+1} - x_i > 0 \quad , \quad i = 0, 1, 2, \dots, N-1 \quad \text{şeklinde ifade edilir.}$$

Bazı yöntemlerde bu adım aralığı tüm integrasyon aralığı boyunca sabittir ancak bu genel bir durum değildir. Nitekim bazı yöntemlerde adım aralığı çözümü istenen fonksiyonun türevlerinin aralığın o bölgesindeki özelliklerine göre belirlenir.

Bir diferansiyel denklemin analitik çözümü yapıldığı zaman (eğer yapılabiliyorsa !), çözüm olan fonksiyonun integrasyon aralığı boyunca her noktadaki değerini tanımlayabilen bir ifade elde edilmiş olur ki, bu da sonsuz sayıda değer verir.

Ancak nümerik yöntemler için durum farklıdır. Çözümü nümerik olarak ifade edilen bir fonksiyonun ancak belli sayıdaki değeri yöntem tarafından hesaplanır ve bu sayı asla sonsuz olamaz. Ancak yine de hemen her zaman pratik bir uygulamada yeterince iyi bir şekilde işe yarayacak kadar çok sayıda ve analitik çözümün verdiği sonuçlara çok yakın değerler elde edilebilir.

Probleme sadece bu açıdan yaklaşırsa, nümerik yöntemlerle çözüm yapmanın aslında analitik yöntemle yapılan çözüme göre bir adım geriye gitmek, belli şeylerden (sonsuz sayıda ya da her noktadaki değerden ve değerlerin kesinliğinden) fedakarlık etmek anlamına geldiği ve dolayısıyla dezavantajlı olup bu yöntemlerle uğraşmak yerine çözümü zor olan problemlerin gerçek çözümüne yönelik çalışmalara ağırlık verilmesi gerektiği düşünülebilir. Ancak gerçekte durum öyle değildir.

Başlangıcından bu güne fiziğin evrimi ve kavramların dayandığı ya da ortaya attığı denklemlerin gelişimi gözden geçirilirse, geline noktada artık birçok gerçek problem için analitik çözümlerden neredeyse bahsetmenin bile imkansız olduğu, nümerik yöntemlerin kaçınılmaz olarak kullanılmak zorunda oldukları gözlenebilir. Hele bütün bunlarla beraber gelişen teknolojinin sunduğu en önemli avantaj olan bilgisayarların bu yöntemlere yatkınlığı

ve günden güne gelişen üstün yeteneklerini (simülasyon gibi) kullanma avantajı düşünülürse, nümerik yöntemlerin aslında ilk bakışta algılandığından çok daha işe yarar ve yavaş yavaş kaçınılmaz hale gelmeye başladıkları anlaşılır.

Neticede nümerik yöntemlerin yaptığı şey parçalara böldüğü $[x_0, \beta]$ integrasyon aralığının her parça sınırı için $\{x_i\}$ çözüm fonksiyonunun $(y(x_i) = y_i)$ yaklaşık bir değerini $(Y(x_i) = Y_i)$ bulmaktır. Dolayısıyla N parçaya bölünmüş bir integrasyon aralığı için nümerik yöntem, çözüm fonksiyonun N adet yaklaşık değerini bulur.

$$Y(x_i) = Y_i \approx y(x_i) = y_i$$

Şimdi (3.1) ifadesini tek bir diferansiyel denklem için ele alalım ve çözümünü arayalım.

$$y'(x) = f(x, y(x))$$

$$\int_{x_i}^{x_{i+1}} y'(x) dx = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx$$

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx$$

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx \quad ; \quad i = 0, 1, 2, \dots, N-1 \quad (3.3)$$

Bütün nümerik yöntemler, birbirlerinden yalnızca (3.3) denklemindeki integral ifadesine yaklaşmakta seçtikleri metodla ayrılırlar. Bu metodlar da üç sınıfa ayrılabilirler;

- i) Tek adımlı metodlar
- ii) Çok adımlı metodlar
- iii) Tahmin algoritmaları

Tek adımlı metodlar Y_{i+1} yaklaşık değerini hesaplamak için yalnızca önceki değer olan " Y_i " yi kullanırlar. Bu yüzden bunlara tek adımlı metodlar denmiştir.

Çok adımlı metodlar ise adından da anlaşılacağı üzere Y_{i+1} yaklaşık değerini hesaplamak için yalnızca önceki Y_i değerini değil $Y_i, Y_{i-1}, Y_{i-2}, \dots, Y_{i-n}$; $n(i)$ gibi bir den çok daha önce hesaplanmış değer kullanırlar.

Tahmin algoritmaları, başlangıç değer problemlerini çözmek için " Romberg dörtlü metodu" nu kullanırlar.

Genelde " tahminci – düzeltici " olarak anılan özelleşmiş metodlar da vardır. Bunlar "tahmin etme " ve " düzeltme " gibi iki ana adımdan oluşurlar. İlk adımda (tahmin etme) $Y_{i+1}^{(0)}$ gibi bir yaklaşık değer tek ya da çok adımlı bir metodla bulunur. Sonraki adımda ise $Y_{i+1}^{(1)}, Y_{i+1}^{(2)}, \dots$ gibi düzeltme terimleri eklenerek hata azaltılır.

$Y(x_{i+1}) \approx y(x_{i+1})$ ve $Y(x_i) = y(x_i)$ kabul edilmek üzere $\varepsilon_{i+1} \equiv y(x_{i+1}) - Y(x_{i+1})$ ifadesi ile tanımlanan " ε_{i+1} " , x_{i+1} noktasındaki "yerel hata " olarak adlandırılır. Bu hata x_i den x_{i+1} ' e kadar (3.3) eşitliğindeki integrale yaklaşırken yapılan hatanın ölçüsüdür. $O(h_i^q)$ yerel hatanın mertebesini ifade eder.

$e_{i+1} \equiv y(x_{i+1}) - Y(x_{i+1})$ ifadesi ile tanımlanan " e_{i+1} " ise (x_{i+1} noktasındaki) "global hata " olarak adlandırılır. Ancak global hata yerel hata gibi sadece x_i ' den x_{i+1} ' e kadar olan hatayı değil daha öncesini kapsayan tüm hataları hesaba katan bir ölçüdür. $O(h_i^q)$ global hatanın mertebesini gösterir (9,13).

3.1.4. Tek Adımlı Metodlar

3.1.4.1. Euler- Cauchy Poligon Metodu

Euler –Cauchy metodunda x_{i+1} noktasındaki yaklaşık Y_{i+1} değerini hesaplamak için tek adımlı metodların hepsinde olduğu gibi sadece bir önceki Y_i yaklaşık değeri kullanılır. Metodun önerdiği algoritma;

$$Y_{i+1} = Y_i + h_i f(x_i, Y_i) \quad , \quad i = 1, 2, \dots, N-1 \quad (3.4)$$

şeklindedir. Burada h_i yerel adım aralığıdır. Metodun dayandığı mantık şöyle özetlenebilir;

Çözüm fonksiyonunun eğrisi üzerindeki hesaplanmış bir değeri temsil eden bir nokta $p_i(x_i, Y_i)$ olsun. Buradan yola çıkılarak hesaplanacak olan $p_{i+1}(x_{i+1}, Y_{i+1})$ noktası ile bu noktayı birleştiren doğru ; “ p_{i+1} ” noktası “ p_i ” noktasına yaklaştırıldıkça, eğriye p_i noktasında çizilen teğet ile çakışma eğilimindedir. Noktaların yeterince yakın olduğu varsayılan bir durum için bu doğru teğet gibi kabul edilirse ve bu teğetin eğiminin de

$$\text{eğim} = \frac{\Delta Y}{\Delta x} = \frac{Y_{i+1} - Y_i}{x_{i+1} - x_i}$$

olduğu ve ayrıca herhangi bir eğriye bir noktasından çizilen teğetin eğiminin de fonksiyonun o noktadaki türev değerine $f'(x_i, Y_i)$ eşit olduğu hatırlanırsa

$$\frac{Y_{i+1} - Y_i}{x_{i+1} - x_i} = f'(x_i)$$

$$\Rightarrow Y_{i+1} - Y_i = (x_{i+1} - x_i) f'(x_i, Y_i)$$

$$\Rightarrow Y_{i+1} = Y_i + (x_{i+1} - x_i) f'(x_i, Y_i)$$

$$\Rightarrow Y_{i+1} = Y_i + h_i f'(x_i, Y_i)$$

yazılabilir. Son yazılan ifade zaten Euler - Cauchy metodunun önerdiği ifadedir. Bu methoda, h_i ne kadar küçük yapılabiliyorsa gerçek değerlere o kadar yaklaşmış olunur. (9,13)

3.1.4.2. Düzeltilmiş Euler – Cauchy Metodu

Bu metod , standart Euler – Cauchy metodundan çok farklı değildir. Yaklaşımındaki hatayı azaltabilmek için

$$Y_{i+1} = Y_i + h_i f(x_i, Y_i)$$

denklemindeki $f(x_i, Y_i)$ değeri yerine, daha iyi sonuç verecek olan $f(x_i + \frac{h_i}{2}, Y_{i+\frac{1}{2}})$ değerini kullanır. Bu değer daha iyi sonuç verir, çünkü yalnızca x_i ve x_{i+1} noktaları ve buradaki yaklaşık değerler değil; aralarında bulunan $(x_i + \frac{h_i}{2})$ noktasındaki değer de hesaba katılmıştır. Bu methodda $f(x_i + \frac{h_i}{2}, Y_{i+\frac{1}{2}})$ ifadesindeki $Y_{i+\frac{1}{2}}$ değeri yine standart Euler – Cauchy metodu ile aşağıdaki şekilde bulunur.

$$Y_{i+\frac{1}{2}} = Y_i + \frac{h_i}{2} f(x_i, Y_i)$$

Sonuçta düzeltilmiş Euler – Cauchy metodunun algoritması

$$\begin{aligned} Y_{i+1} &= Y_i + h_i f(x_i + \frac{h_i}{2}, Y_{i+\frac{1}{2}}) \\ &= Y_i + h_i f(x_i + \frac{h_i}{2}, Y_i + \frac{h_i}{2} f(x_i, Y_i)) \end{aligned} \quad (3.5)$$

şeklinde olur. (9,13)

3.1.4.3. Taylor Serisi Metodu

Nümerik metodların, bir diferansiyel denkleme çözüm olarak, bir aralık boyunca ardışık olarak h_i aralıkları ile sıralanmış N adet x_i noktası için çözüm fonksiyonunun bu noktalara karşılık gelen yaklaşık değerlerini bulduğunu belirtmiştik. Bu yaklaşık değerler de " Y_i "lerdir. " Y_{i+1} " değeri ile kastedilen şey aslında

$$Y_{i+1} = Y(x_{i+1}) = Y(x_i + h_i)$$

şeklindedir.

Öte yandan Taylor serisinin

$$y(x + h) = y(x) + hy'(x) + \frac{h^2}{2!} y''(x) + \frac{h^3}{3!} y'''(x) + \dots$$

şeklinde olduğu hatırlanırsa

$$Y_{i+1} = Y(x_i + h_i) = Y(x_i) + h_i Y'(x_i) + \frac{h_i^2}{2!} Y''(x_i) + \frac{h_i^3}{3!} Y'''(x_i) + \dots$$

yazılabilir.

Ayrıca ,

$$Y(x_i) = Y_i$$

$$Y'(x_i) = f(x_i, Y_i)$$

$$Y''(x_i) = f''(x_i, Y_i)$$

$$Y'''(x_i) = f'''(x_i, Y_i)$$

.

yazılabileceği göz önüne alınarak;

$$Y_{i+1} = Y_i + h_i f(x_i, Y_i) + \frac{h_i^2}{2!} f'(x_i, Y_i) + \frac{h_i^3}{3!} f''(x_i, Y_i) + \dots \quad (3.6)$$

yazılabilir. Bu ifade ,hesaplanmış bir "Y_i" değerinden yola çıkılarak bir sonraki "Y_{i+1}" değerinin nasıl hesaplanabileceğini anlatır ki , bu da Taylor serisi metodunun algoritmasıdır. Yalnız burada gözden kaçmaması gereken nokta, metodu uygulayabilmek için

$$f(x, y) = y'$$

fonksiyonunun sürekli yüksek mertebeden türevlerinin,

$$(y'', y''', y^{(4)}, \dots)$$

var olmasının gerektiğidir. Bu metodun sadece ilk iki terim göz önüne alınarak uygulanması, standart Euler – Cauchy metoduna karşılık gelir. (9,13)

3.1.4.4. Heun'un Tahmin Edici- Düzeltici Metodu

Bu metod “ tahmin etme ” ve “düzeltme ” adımlarından oluşur. Tahmin etme adımında yaklaşık "Y_{i+1}" değeri Euler – Cauchy metodu ile

$$Y_{i+1}^{(0)} = Y_i + h_i f(x_i, Y_i)$$

şeklinde hesaplanır. Daha sonra , düzeltme adımında "Y_{i+1}" değerine aşağıdaki iterasyonla adım adım yaklaşılır;

$$Y_{i+1}^{(r+1)} = Y_i + \frac{h_i}{2} (f(x_i, Y_i) + f(x_{i+1}, Y_{i+1}^{(r)})), \quad r = 0, 1, 2, \dots$$

Denemeler göstermiştir ki yeterince küçük h_i adım aralığı için bir ya da iki iterasyon ile yeterince doğruya yakın değer elde edilebilmektedir. (9)

3.1.4.5. Runge – Kutta Metodları

3.1.4.5.1. Runge- Kutta Metodlarının Yapısı

Bu metodlar tek adımlı metodların en önemlileridirler “m” inci mertebeden Runge-Kutta metodunun genel yapısı

$$Y_{i+1} = Y_i + h_i \sum_{j=1}^m A_j k_j(x_i, Y_i, h_i),$$

$$k_1(x, Y, h) \equiv f(x, Y), \quad (3.7)$$

$$k_j(x, Y, h) \equiv f(x + a_j h, Y + h \sum_{s=1}^{j-1} b_{js} k_s(x, Y, h)),$$

$$j = 2, 3, \dots, m$$

şeklindedir.

“m” inci mertebeden Runge – Kutta metodu için $m \leq 4$ ise yerel hata oranı $q_l = m+1$ ve global hata oranı $q_g = m$ dir. $m > 4$ olursa global hata oranı $q_g < m$ olmaya başlar. Bu durum daha açık olarak aşağıdaki tabloda görülmektedir.

m	1	2	3	4	5	6	7	8	9
q_g	1	2	3	4	4	5	6	6	7

Tablo 1.

Runge-Kutta metodları için metodun mertebesi ve hata oranı arasındaki ilişki.

$m=1$ için Runge-Kutta metodu Euler-Cauchy metodu ile eşdeğer olur. $m=2$ olduğu zaman da Heun’un metoduna ve düzeltilmiş Euler – Cauchy metoduna karşılık gelir. Yani bunlar 2. mertebeden Runge – Kutta metodlarıdır. $m=4$ olduğu zaman ise ,en çok kullanılan ve “klasik Runge –Kutta metodu ” olarak bilinen metod elde edilir. (9,13,14,11)

3.1.4.5.2. Klasik Runge – Kutta Metodu(m=4)

Metodun Algoritması

$$Y_{i+1} = Y_i + h_i \left[\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right]$$

$$k_1 = f(x_i, Y_i) \quad (3.8)$$

$$k_2 = f\left(x_i + \left(\frac{h_i}{2}\right), Y_i + h_i\left(\frac{k_1}{2}\right)\right)$$

$$k_3 = f\left(x_i + \left(\frac{h_i}{2}\right), Y_i + h_i\left(\frac{k_2}{2}\right)\right)$$

$$k_4 = f(x_i + h_i, Y_i + h_i k_3)$$

şeklindedir.

Bu metot için hesaplama şeması diğer sayfadaki tablodan takip edilebilir.

i	x	Y	$k_j(x_i, Y_i, h_i), j=1,2,3,4$	$K^{(i)}$
0	x_0	y_0	$k_1 = f(x_0, y_0)$	k_1
	$x_0+h_0/2$	$y_0+h_0(k_1/2)$	$k_2 = f(x_0 + h_0/2, y_0 + h_0(k_1/2))$	$2k_2$
	$x_0+h_0/2$	$y_0+h_0(k_2/2)$	$k_3 = f(x_0 + h_0/2, y_0 + h_0(k_2/2))$	$2k_3$
	x_0+h_0	$y_0+h_0k_3$	$k_4 = f(x_0 + h_0, y_0 + h_0k_3)$	k_4
	$x_1 = x_0 + h_0$	$Y_1 = y_0 + h_0k^{(0)}$		$k^{(0)} = (1/6)\Sigma$
1	x_1	Y_1	$k_1 = f(x_1, Y_1)$	k_1
	$x_1+h_1/2$	$Y_1+h_1(k_1/2)$	$k_2 = f(x_1 + h_1/2, Y_1 + h_1(k_1/2))$	$2k_2$
	$x_1+h_1/2$	$Y_1+h_1(k_2/2)$	$k_3 = f(x_1 + h_1/2, Y_1 + h_1(k_2/2))$	$2k_3$
	x_1+h_1	$Y_1+h_1k_3$	$k_4 = f(x_1 + h_1, Y_1 + h_1k_3)$	k_4
	$x_2 = x_1 + h_1$	$Y_2 = y_1 + h_1k^{(1)}$		$k^{(1)} = (1/6)\Sigma$
2	x_2	Y_2	$k_1 = f(x_2, Y_2)$	k_1

Tablo 2.

Klasik Runge-kutta metodu için hesaplama şeması.

Klasik Runge –Kutta metodu hesaplama zamanı açısından (bilgisayar için) tasarruflu bir yöntem değildir. Her adım için kişi, 4 adet fonksiyonel değeri hesaplamak zorundadır. Ayrıca “m” ne kadar büyük olursa, hesaplama zamanı o kadar artacaktır. Buna rağmen klasik Runge – Kutta metodu harcanan zaman ve buna karşılık elde edilen doğruluk derecesi bakımından diğer metotlara oranla tercih edilir durumdadır. (9,13,12,11)

3.1.4.5.3. Runge – Kutta Metodları için Liste

“m” inci mertebeden Runge – Kutta metodu için formülasyon

$$Y_{i+1} = Y_i + h_i (A_1 k_1 + A_2 k_2 + \dots + A_m k_m) \quad (3.9)$$

şeklindedir.

Burada,

$$k_1(x_i, Y_i, h_i) = f(x_i, Y_i)$$

$$k_2(x_i, Y_i, h_i) = f(x_i + a_2 h_i, Y_i + h_i b_{21} k_1) \quad (3.10)$$

$$k_3(x_i, Y_i, h_i) = f(x_i + a_3 h_i, Y_i + h_i (b_{31} k_1 + b_{32} k_2))$$

-
-
-

$$k_m(x_i, Y_i, h_i) = f(x_i + a_m h_i, Y_i + h_i (b_{m1} k_1 + b_{m2} k_2 + \dots + b_{m,m-1} k_{m-1}))$$

dir.

Aşağıdaki listede A_j, a_j ve b_{js} katsayıları verilmiştir.

m	J	A_j	a_j	$b_{j,s} , s = 1,2, \dots , m-1$		q_g	Metodun adı
1	1	1	0			1	Euler-Cauchy
2	1	0	0			2	Düzeltilmiş Euler-Cauchy
	2	1	1/2	1/2			
2	1	1/2	0			2	Heun'un metodu
	2	1/2	1	1			
3	1	1/6	0			3	3. mertebeden Runge-Kutta I
	2	2/3	1/2	1/2			
	3	1/6	1	-1	2		
3	1	1/4	0			3	3. mertebeden Runge-Kutta II
	2	0	1/3	1/3			
	3	3/4	2/3	0	2/3		

5	1	$\frac{25}{216}$	0						4	4. mertebeden Runge-Kutta Fehlberg
	2	0	$\frac{1}{4}$	$\frac{1}{4}$						
	3	$\frac{1408}{2565}$	$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$					
	4	$\frac{2197}{4104}$	$\frac{12}{13}$	$\frac{1932}{2197}$	$-(\frac{7200}{2197})$	$\frac{7296}{2197}$				
	5	$-\frac{1}{5}$	1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-(\frac{845}{4104})$			
6	1	$\frac{16}{135}$	0						5	5. mertebeden Runge-Kutta Fehlberg
	2	0	$\frac{1}{4}$	$\frac{1}{4}$						
	3	$\frac{6656}{12825}$	$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$					
	4	$\frac{28561}{56430}$	$\frac{12}{13}$	$\frac{1932}{2197}$	$-(\frac{7200}{2197})$	$\frac{7296}{2197}$				
	5	$-(\frac{9}{50})$	1	$\frac{439}{216}$	-8	$\frac{3680}{519}$	$-(\frac{845}{4104})$			
	6	$\frac{2}{55}$	$\frac{1}{2}$	$-(\frac{8}{27})$	2	$-(\frac{3544}{2565})$	$\frac{1859}{4104}$	$-(\frac{11}{40})$		
6	1	$\frac{23}{192}$	0						5	Kutta-Nyström
	2	0	$\frac{1}{3}$	$\frac{1}{3}$						
	3	$\frac{125}{192}$	$\frac{2}{5}$	$\frac{4}{25}$	$\frac{6}{25}$					
	4	0	1	$\frac{1}{4}$	$-(\frac{12}{4})$	$\frac{15}{4}$				
	5	$-(\frac{81}{192})$	$\frac{2}{3}$	$\frac{6}{81}$	$\frac{90}{81}$	$-(\frac{50}{81})$	$\frac{8}{81}$			
	6	$\frac{125}{192}$	$\frac{4}{5}$	$\frac{6}{75}$	$\frac{36}{75}$	$\frac{10}{75}$	$\frac{8}{75}$	0		

6	1	31/384	0							5	Runge-Kutta Fehlberg I
	2	0	1/6	1/6							
	3	1125/2816	4/15	4/75	16/75						
	4	9/32	2/3	5/6	-(8/3)	5/2					
	5	125/768	4/5	-(8/5)	144/25	-4	16/25				
	6	5/66	1	361/320	-(18/5)	407/128	-(11/80)	55/128			
7	1	11/120	0							6	Butcher
	2	0	1/3	1/3							
	3	27/40	2/3	0	2/3						
	4	27/40	1/3	1/12	1/3	-(1/12)					
	5	-(4/15)	1/2	-(1/16)	9/8	-(3/16)	-(3/8)				
	6	-(4/15)	1/2	0	9/8	-(8/3)	-(3/4)	1/2			
	7	11/120	1	9/44	-(9/11)	63/44	18/11	0	-(16/11)		
.
.
.

Tablo 3.

Runge-Kutta metodları için A_j , a_j , ve b_{js} katsayıları. (9)

3.1.4.5.4. İç İçe Gömülmüş Runge – Kutta Prosedürleri

Bu yöntem “P” inci ve “q” uncu mertebeden (genel olarak $q=p+1$ olur) iki adet Runge – Kutta metodunun, aynı denklem için aynı noktalarda iç içe kullanılması esasına dayanır. Aynı nokta için elde edilen iki yaklaşık değer adım aralığının efektif kontrollü için kullanılabilir. “P” inci mertebeden yaklaşık değeri “Y” ve “q” uncu mertebeden yaklaşık değeri \bar{Y} ile gösterirsek; iç içe gömülmüş Runge – Kutta prosedürünün algoritması aşağıdaki gibi ifade edilebilir. (9,13,11)

$$Y_{i+1} = Y_i + h_i \sum_{j=1}^p A_j k_j$$

$$\bar{Y}_{i+1} = Y_i + h_i \sum_{j=1}^q A_j k_j$$

$$k_1 = f(x_i, Y_i)$$

$$k_2 = f(x_i + a_2 h_i, Y_i + h_i b_2, k_1)$$

.

.

.

$$k_j = f(x_i + a_j h_i, Y_i + \sum_{s=1}^{j-1} h_i b_{js} k_s) , j=3,4,\dots,q$$

3.1.4.6. Tek Adımlı Metodların Tutarlılığı ve Uyumu

Başlangıç değer problemini çözmeye çalışan tek adımlı metodların algoritmalarını göz önüne alırsak, her prosedürün aşağıdaki şekilde ifade edilebileceğini görebiliriz ;

$$Y_{i+1} = Y_i + h_i \Phi(x_i, Y_i, h_i) \quad , \quad i = 0, 1, 2, \dots, N-1 \quad (3.11)$$

Herhangi bir tek adımlı metod Φ fonksiyonu ile tek olarak belirlenir. Mesela düzeltilmiş Euler – Cauchy metodu için

$$\Phi(x_i, Y_i, h_i) = f\left(x_i + \frac{h_i}{2}, Y_i + \frac{h_i}{2} f(x_i, Y_i)\right)$$

şeklindedir.

“ m ” inci dereceden Runge – Kutta metodu için ise Φ ;

$$\Phi(x_i, Y_i, h_i) = \sum_{j=1}^m A_j k_j(x_i, Y_i, h_i)$$

şeklindedir.

Tek adımlı bir metod için tanımlanmış bir $\Phi(x, Y, h)$ fonksiyonunu göz önüne alalım.

Ayrıca,

$$y'(x) = f(x, y) \quad , \quad y(x_0) = y_0$$

başlangıç değer probleminin analitik çözümü $y(x)$ olsun. Bu durumda

$$\tau_i \equiv \frac{1}{h_i} (y(x_{i+1}) - y(x_i)) - \Phi(x_i, y(x_i), h_i) \quad (3.12)$$

şeklinde tanımladığımız değere x_i noktasındaki “kesme hatası ” denir. Ayrıca herhangi bir tek adımlı metod için kesme hatalarının ağırlıklı toplamı

$$\sum_{i=0}^{N-1} h_i \|\tau_i\|$$

maximum adım aralığı h_{\max} sıfıra doğru giderken, sıfıra doğru küçülüyorsa, yani

$$h_{\max} \rightarrow 0 \Rightarrow \left\{ \sum_{i=0}^{N-1} h_i \|\tau_i\| \right\} \rightarrow 0 \quad \text{ise,}$$

bu tek adımlı metot “ tutarlıdır ” denir. Ya da başka bir deyişle

$$h_{\max} \rightarrow 0 \Rightarrow \max_{0 \leq i \leq N-1} \{\|\tau_i\|\} \rightarrow 0$$

ise, bu metot tutarlıdır denir.

Şu ana kadar incelediğimiz bütün tek adımlı metotların tutarlı olduğu gösterilebilir. Bir metot için tutarlılık oranı $o(h_{\max}^q)$ global hata oranına eşittir. (9,13)

3.1.4.7. Hata Hesabı ve Adım Aralığı Kontrolü

3.1.4.7.1. Hata Hesabı

$Y_h(x)$ ve $Y_{\bar{h}}(x)$, $x \in [x_0, \beta]$ noktasında "Y" nin

farklı adım aralıkları (h ve \bar{h}) seçilerek global hata oranı q_g olan bir metotla hesaplanmış iki yaklaşık değeri olsunlar. Bu durumda global prosedürel hatayı aşağıdaki şekilde hesaplayabiliriz ;

$$e_h \equiv y(x) - Y_h(x) \approx \frac{Y_h(x) - Y_{\bar{h}}(x)}{\left(\frac{\bar{h}}{h}\right)^{q_g} - 1} = e_h^* \quad (3.13)$$

$$Y_h^* \equiv Y_h(x) + e_h^* = \frac{\left(\frac{\bar{h}}{h}\right)^{q_s} Y_h(x) - Y_{\bar{h}}(x)}{\left(\frac{\bar{h}}{h}\right)^{q_s} - 1} \quad (3.14)$$

değeri $Y_h(x)$ değerine oranla daha yakın (düzeltilmiş) bir değer olur. Şimdi $\bar{h} = 2h$ için durumu irdeleyelim ;

$$e_h(x) \approx \frac{Y_h(x) - Y_{2h}(x)}{2^{q_s} - 1} = e_h^* \quad (3.15)$$

$$Y_h^*(x) = \frac{2^{q_s} Y_h(x) - Y_{2h}(x)}{2^{q_s} - 1}$$

olur.

Mesela biz aşağıdaki metodlar için aşağıdaki şekilde düzeltilmiş değerleri $\bar{h} = 2h$ için elde edebiliriz. (9,13)

3.1.4.7.1.1. Euler –Cauchy Metodu

$$e_h^{EC}(x) \approx Y_h^{EC}(x) - Y_{2h}^{EC}(x) \quad (3.16)$$

$$Y_h^*(x) = 2Y_h^{EC}(x) - Y_{2h}^{EC}(x)$$

3.1.4.7.1.2. Heun'un Metodu ve Düzeltmiş Euler-Cauchy Metodu

$$e_h^H(x) \approx \frac{1}{3}(Y_h^H(x) - Y_{2h}^H(x)) \quad (3.17)$$

$$Y_h^*(x) = \frac{1}{3}(4Y_h^H(x) - Y_{2h}^H(x))$$

3.1.4.7.1.3. Klasik Runge -Kutta Metodu

$$e_h^{RK}(x) \approx \frac{1}{15}(Y_h^{RK}(x) - Y_{2h}^{RK}(x)) \quad (3.18)$$

$$Y_h^*(x) = \frac{1}{15}(16Y_h^{RK}(x) - Y_{2h}^{RK}(x))$$

3.1.4.7.2. Otomatik Adım Aralığı Kontrolü ; Başlangıç Değer Problemi İçin Uyarlamalı Metotlar

Genel olarak, sabit adım aralığı ile hesap yapmak avantajlı bir yöntem değildir. Çözüm fonksiyonunun yerel özelliklerine göre adım aralığını yerel olarak tayin etmek en uygun ve önerilen yöntemdir. Mesela kişi çözüm fonksiyonunda küçük değişimlerin olduğu bölgelerde bağıl olarak büyük adım aralığı ile çalışabilir ve aynı şekilde çözüm fonksiyonunda büyük değişimlerin yaşandığı bölgelerde de nispeten küçük adım aralığı ile çalışılmalıdır. Hata hesabı kullanılarak adım aralığını otomatik olarak tayin etmek mümkündür. Bunun için olası durumlar ve bu durumlara ilişkin metotlara aşağıda değinilmiştir.

3.1.4.7.2.1. Adım Aralığını Kontrol Etmek İçin Metot A

(3.13) denklemini ile tanımlanan hata hesabı kullanılarak şu algoritma izlenebilir; adım aralığı “ h” olan her iki adım sonrasında, yani x_{i-1} den önce x_i 'den x_{i+1} 'e adım aralığı $2h$ alınarak tek adımlık bir hesap daha yapılır. Eğer bu iki değer kullanılarak hesaplanan hata istenen sınırın yeterince altında kalıyorsa ; bir sonraki adım için adım aralığı yükseltilebilir. Eğer hesaplanan hata daha büyükse, son hesaplanan yarım aralık, tekrar ve bu kez daha düşük bir adım aralığı ile hesaplanabilir. (9)

3.1.4.7.2.2. Adım Aralığını Kontrol Etmek İçin Metot B:

Tanımlayıcı fonksiyonları Φ ve $\bar{\Phi}$ olan iki tane tek adımlı ve global hata oranları q_g ile en az q_g olan metodumuz olsun. Bu durumda, Y_i yaklaşık değerinden yola çıkılarak (x_{i+h}) noktasındaki iki yaklaşık değer Y ve \bar{Y} iki ayrı metotla hesaplanabilir. Hata hesabının durumuna göre istenirse "h" adım aralığı kabul edilir ya da işleme daha küçük bir adım aralığı ile devam edilir.

Metot B için örnek bir algoritma ;

i) Global hata oranları q_g ve q_g olan iki tane tek adımlı metot seçilmiş olsun. Bu metotlara ilişkin tanımlayıcı fonksiyonlar da Φ ve $\bar{\Phi}$ olsun. Ayrıca " Y_i " değeri problemin gerçek çözümü olan " y "nin " x_i " noktasındaki, "h" adım aralığı kullanılarak hesaplanmış olsun. Bu durumda " x_{i+h} " noktasındaki yaklaşık değer iki ayrı metot tarafından aşağıdaki şekilde hesaplanır;

$$Y = Y_i + h\Phi(x_i, Y_i, h) \quad , \quad \bar{Y} = Y_i + h\bar{\Phi}(x_i, Y_i, h)$$

ve öngörülen bir hata sınırı $\varepsilon > 0$ için aşağıdaki değer hesaplanır.

$$s = \left(\frac{h\varepsilon}{\|Y - \bar{Y}\|} \right)^{1/q_g}$$

ii) Eğer $s \geq 1$ olabiliyorsa \bar{Y} değeri " x_{i+h} " noktasındaki Y_{i+1} değeri olarak kabul edilir ve yeni adım aralığı

$$h = \min\{2; s\}.h$$

olarak seçilerek sonraki adım için tekrar "i"ye dönülerek işlem yapılır.

Eğer $s < 1$ oluyorsa yeni adım aralığı

$$h = \max\left\{\frac{1}{2}; s\right\}h$$

alınarak "i" adımı tekrarlanır. (Önceki adım tekrar edilir bu durumda, sonraki adıma geçilmez). Bu algoritmayla adım aralığı otomatik olarak kontrol edilmiş olur. (9)

3.1.5. Çok Adımlı Metotlar

3.1.5.1. Çok Adımlı Metotların Prensibi

Çok adımlı metotlar bir x_{i+1} noktasındaki Y_{i+1} yaklaşık değerini hesaplamak için, "s+1" adet daha önceki adımlarda hesaplanmış $Y_{i-s}, Y_{i-s+1}, \dots, Y_{i-1}, Y_i$ değerlerini kullanırlar. Yani tek adımlı metotlardaki gibi Y_{i+1} değerini hesaplamak için sadece bir önceki " Y_i " değeri ile yetinmezler.

Bu noktada problemimizi yeniden hatırlamakta fayda var. Problemimiz, yani başlangıç değer problemi ; $\underline{y}(x_{-s}) = \underline{y}_{-s}$ başlangıç koşulu için

$$\underline{y}'(x) = \underline{f}(x, \underline{y}) = \underline{f}(x, \underline{y}(x)) = \underline{f}(x, y_1, y_2, \dots, y_n) \quad (3.19)$$

denkleminin $[x_{-s}, \beta]$ aralığındaki çözümünü aramaktan ibarettir. Analitik yöntemlerde bu çözüm (yani $\underline{y}(x)$ fonksiyonu) $[x_{-s}, \beta]$ aralığında integral alınarak bulunur. Ancak nümerik yöntemle yaptığımız $[x_{-s}, \beta]$ aralığını

$$x_{-s} < x_{-s+1} < \dots < x_{N-s} = \beta$$

şeklinde ve yerel adım aralığı

$$h_i = x_{i+1} - x_i > 0, \quad i = -s, -s+1, \dots, N-s, \quad N > s$$

olacak şekilde parçalara bölmektir. Daha sonra da bölüntünün sınırlarının her biri (x_i 'ler) için $y(x_i)$ nin yaklaşık değeri olan Y_i 'yi hesaplamaktır. Tek adımlı yöntemlerde " Y_i " yi hesaplamak için Y_{i-1} değerinin bilinmesi yetmekte idi. Ama çok adımlı yöntemlerde Y_i değerini hesaplamak için

$x_{-s}, x_{-s+1}, \dots, x_{-1}, x_0$ noktalarındaki $s+1$ adet

$y(x_i)$, $i = -s, -s+1, \dots, 0$

değerinin bilinmesi gerekmektedir. Bu değerler, ya problemin başlangıç koşulu olarak verilmiş olmalıdır ya da tek adımlı bir yöntem (mesela klasik Runge – Kutta metodu) kullanılarak yaklaşık olarak hesaplanmalıdır. Ancak bundan sonra çok adımlı hesaplamaya geçilebilir. Bu değerlerin tümüne başlangıç değerleri diyelim ve bundan böyle bu başlangıç değerlerini

$(x_i, f(x_i, y_i)) = (x_i, f_i)$, $i = -s, -s+1, \dots, 0$

olarak ifade edelim.

Tek denklem için irdelemeyi sürdürürsek problemimizi

$$\begin{aligned} y'(x) &= f(x, y) \\ y(x_0) &= y_0 \end{aligned} \tag{3.20}$$

şeklinde ifade edebiliriz. Şimdi parçaladığımız aralık boyunca bir x_i noktasından bir sonraki x_{i+1} noktasına kadar olan kısım üzerinde duralım. Bu aralık boyunca gerçek çözüm

$$\int_{x_i}^{x_{i+1}} y'(x) dx = \int_{x_i}^{x_{i+1}} f(x, y) dx \tag{3.21}$$

integrali ile verilir.

Ayrıca temel integral teoreminden

$$\int_{x_i}^{x_{i+1}} y'(x) dx = y(x_{i+1}) - y(x_i) = y_{i+1} - y_i \quad (3.22)$$

yazılabilir.

$$\begin{aligned} \Rightarrow \int_{x_i}^{x_{i+1}} y'(x) dx &= \int_{x_i}^{x_{i+1}} f(x, y) dx \\ y_{i+1} - y_i &= \int_{x_i}^{x_{i+1}} f(x, y) dx \\ y_{i+1} &= y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx \end{aligned} \quad (3.23)$$

elde edilir.

Şimdi bu ifadeye yaklaşmaya çalışan çok adımlı metotlardan birkaçını inceleyelim. (9,13)

3.1.5.2 Adams-Bashforth Metodu

Bu metod, yaklaşım için aşağıdaki formülü önerir ;

$$Y_{i+1} = Y_i + af_i + bf_{i-1} + cf_{i-2} + \dots \quad (3.24)$$

burada $f_i = f(x_i, y_i)$ dir. Daha önce de değinildiği gibi bu metodu kullanmak için $f_i = f_{i-1}, f_{i-2}, \dots$ değerleri bilinmeli ya da bilinmiyorsa bir tek adımlı metot yardımıyla hesaplanmalıdır. Eşit adım aralığı için örneğin 5.mertebeden Adams-Bashforth formülü

$$Y_{i+1} = Y_i + \frac{h}{720} [1901f_i - 2774f_{i-1} + 2616f_{i-2} - 1274f_{i-3} + 251f_{i-4}]$$

şeklindedir.

Şimdi de bu formüllerde kullanılan a, b, c, d, katsayılarının nasıl hesaplandığını inceleyelim. Bunun için

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx \quad (3.25)$$

denklemindeki integrale aşağıdaki şekilde yaklaşalım.

$$\int_{x_i}^{x_{i+1}} f(x, y(x)) dx \approx h [af_n + bf_{n-1} + cf_{n-2} + df_{n-3} + ef_{n-4}] \quad (3.26)$$

işleme devam ederken $x_i = 0$ ve $h = 1$ almamız a, b, c, d, e katsayılarının hesabı açısından genellemede herhangi bir kayıba sebep olmaz. Şimdi katsayıları hesaplamak için aşağıdaki polinomları baz alalım ;

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= x(x+1) \\ P_3(x) &= x(x+1)(x+2) \\ P_4(x) &= x(x+1)(x+2)(x+3) \end{aligned} \quad (3.27)$$

5. mertebeden Adams-Boshforth metodunu incelediğimiz için 5 adet polinom aldık .Şimdi bu polinomları tek tek

$$\int_0^1 P_n(x) dx = aP_n(0) + bP_n(-1) + cP_n(-2) + dP_n(-3) + eP_n(-4) \quad (3.28)$$

denkleminde yerine koyalım. Bu durumda katsayılar için sonraki sayfanın başındaki 5 denklemini elde ederiz.

$$\begin{aligned}
a + b + c + d + e &= 1 \\
-b - 2c - 3d - 4e &= \frac{1}{2} \\
2c + 6d + 12e &= \frac{5}{6} \\
-6d - 24e &= \frac{9}{4} \\
24e &= \frac{251}{30}
\end{aligned} \tag{3.29}$$

5 bilinmeyen katsayı bu 5 denklem yardımı ile çözülür ve böylece 5. mertebe Adams-Bashforth formülünün katsayıları elde edilir. Bu katsayıları elde etmek için kullanılan bu yöntem 5. mertebeye özgü bir yöntem değildir. Herhangi bir mertebeden Adams-Bashforth formülünün katsayıları aynı prosedürle, polinom sayısı uygun seçilerek bulunur. Adams-Bashforth metodunun değişik mertebeleri için birkaç formül aşağıda verilmiştir. (9,13)

4.Mertebe

$$\begin{aligned}
Y_{i+1} &= Y_i + \frac{h}{24} [55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}] \\
i &= 0, 1, 2, \dots, n-4
\end{aligned} \tag{3.30}$$

5.Mertebe

$$\begin{aligned}
Y_{i+1} &= Y_i + \frac{h}{720} [1901f_i - 2774f_{i-1} + 2616f_{i-2} - 1274f_{i-3} + 251f_{i-4}] \\
i &= 0, 1, 2, \dots, n-5
\end{aligned} \tag{3.31}$$

6.Mertebe

$$\begin{aligned}
Y_{i+1} &= Y_i + \frac{h}{1440} [4277f_i - 7923f_{i-1} + 9982f_{i-2} - 7298f_{i-3} + 2877f_{i-4} - 475f_{i-5}] \\
i &= 0, 1, 2, \dots, n-6
\end{aligned} \tag{3.32}$$

7.Mertebe

$$Y_{i+1} = Y_i + \frac{h}{60480} [198721f_i - 447288f_{i-1} + 705549f_{i-2} - 688256f_{i-3} + 407139f_{i-4} - 1343472f_{i-5} + 19087f_{i-6}] \quad (3.33)$$

$$i = 0, 1, 2, \dots, n-7$$

3.1.5.3. Adams-Moulton Tahmin Edici Düzeltici Metodu

Bu metod Adams Bashforth metodunu bir adım öteye taşır ve yaptığı manipülasyon ile Adams-Bashforth metodu ile hesaplanan bir Y_{i+1} yaklaşık değerini daha iyiye doğru geliştirir; yani “düzeltir”. Metodun Adams-Bashforth metodunda yaptığı manipülasyon, formülasyonu aşağıdaki şekilde değiştirmektedir.

$$Y_{i+1} = Y_i + af_{i+1} + bf_i + cf_{i-1} + df_{i-2} + \dots \quad (3.34)$$

Burada manipülasyon denklemdaki "f" değerlerinin “Adams – Bashforth ” metodunda olduğu gibi "f_i" den başlamayıp "f_{i+1}" değerinden başlamasıdır. Ancak bu durum bir terim eklemekten daha karmaşık bir sonuca yol açmaktadır, çünkü "f_{i+1}" ya da başka bir deyişle "f(x_{i+1}, Y_{i+1})" terimini hesaplamak için Y_{i+1} değerine ihtiyaç vardır. Yani bu önceki ve hesaplanmış bir değer değildir ve bu da denklemin sağ tarafında da, sol tarafında da hesaplanması amaçlanan “Y_{i+1}” değerinin içerilmesi demektir.

Bu problem Adams-Moulton metodu tarafından şu şekilde aşılmıştır. $f_{i+1} = f(x_{i+1}, Y_{i+1})$ değerini hesaplamak için gerekli Y_{i+1} değeri, yaklaşık olarak önce Adams-Bashforth metodu ile hesaplanır. Hesaplanan bu yaklaşık değeri Y_{i+1}^{*} ile gösterelim. Bu aşamaya kadar olan kısım “tahmin etme” adımıdır.

Daha sonra Y_{i+1}^{*} değeri $f_{i+1} = f(x_{i+1}, Y_{i+1})$ ifadesinde yerine konularak f_{i+1} terimi hesaplanır. Sonra bu terim başlangıçta verilen Adams-Moulton formülünde yerine konularak Y_{i+1}⁽¹⁾ değeri hesaplanır ki bu adım da düzeltme adımıdır. Ancak düzeltme adımı burada durmaz ve Y_{i+1}⁽¹⁾ değerini kullanarak yeniden yeni bir f_{i+1} değeri hesaplar. Ardından bu değeri

yine Adams-Moulton formülünde kullanarak $Y_{i+1}^{(2)}$ değerini hesaplar. Aslında bu düzeltme işlemi bu şekilde daha da ileriye götürülebilir; ancak bu kadarı genelde yeterlidir ve son hesaplanan değer olan $Y_{i+1}^{(2)}$ değeri x_{i+1} noktasında yaklaşılmaya çalışılan gerçek y_{i+1} değerine yeterince yakındır. Değişik mertebelerden birkaç Adams-Moulton formülü aşağıda verilmiştir.

5. mertebe:

$$Y_{i+1} = Y_i + \frac{h}{720} [251f_{i+1} + 646f_i - 264f_{i-1} + 106f_{i-2} - 19f_{i-3}] \quad (3.35)$$

6. mertebe:

$$Y_{i+1} = Y_i + \frac{h}{1440} [475f_{i+1} + 1427f_i - 798f_{i-1} + 482f_{i-2} - 173f_{i-3} + 27f_{i-4}] \quad (3.36)$$

7. mertebe:

$$Y_{i+1} = Y_i + \frac{h}{60480} [19087f_{i+1} + 65112f_i - 46461f_{i-1} + 37504f_{i-2} - 20211f_{i-3} + 6312f_{i-4} - 863f_{i-5}] \quad (3.37)$$

Adams-Moulton metodu için algoritmayı daha açık ifade eden bir hesaplama şeması sonraki sayfada verilmiştir. (9)

	i	x_i	$Y_i = Y(x_i)$	$f_i = f(x_i)$
Başlangıç değerleri	-3	x_{-3}	$Y_{-3} = y_{-3}$	f_{-3}
	-2	x_{-2}	Y_{-2}	f_{-2}
	-1	x_{-1}	Y_{-1}	f_{-1}
	0	x_0	Y_0	f_0
Adams-Bashforth metodu ile Tahmin	1	x_1	$Y_1^{(0)}$	$f(x_1, Y_1^{(0)})$
Adams-Moulton metodu ile düzeltme	1	x_1	$Y_1^{(1)}$	$f(x_1, Y_1^{(1)})$
	1	x_1	$Y_1^{(2)} \Rightarrow Y_1$	$f(x_1, Y_1)$
Adams-Bashforth metodu ile tahmin	2	x_2	$Y_2^{(0)}$	$f(x_2, Y_2^{(0)})$
Adams-Moulton metodu ile düzeltme	2	x_2	$Y_2^{(1)}$	$f(x_2, Y_2^{(1)})$
	2	x_2	$Y_2^{(2)} \Rightarrow Y_2$	$f(x_2, Y_2^{(1)})$
.
.

Tablo 4.

Adams Moulton metodu için hesaplama şeması

3.1.6. Kararlılık

Şu ana kadar incelediğimiz yöntemlerin hepsi başlangıç değer problemine nümerik çözüm olarak $[x_0, \beta]$ integral aralığının $x_0 < x_1 < x_2 < \dots < x_n = \beta$ bölüntüsünün her ayrık $x_i (0 \leq i \leq n)$ noktası için $y(x_i) = y_i$ gerçek çözüm değerine yaklaşan $Y(x_i) = Y_i$ yaklaşık değerini önermektedirler. Bu yöntemlerin hepsi için, $h_{\max} \rightarrow 0$ durumunda $Y_i \rightarrow y_i$ olduğu ispatlanabilir. Bu durum kararlılık için olması gereken bir kriter sayılabilsede bile; yine de herhangi bir yöntemin kararlılığını tek başına garanti etmez. Kararlılık üzerine daha detaylı bir inceleme yapmalıyız.

İşe kararlılığı tanımlamakla başlayabiliriz; eğer hesaplamamızın herhangi bir adımına eşlik eden hata, ardı sıra devam eden adımlarda artış göstermiyorsa, söz konusu algoritma “kararlıdır” denir. Eğer hata sınırsız olarak büyüyorsa bu durumda da algoritma kararsızdır denir.

Kararsızlık, çözülmeye çalışılan denklemin kendisinden ya da seçilen algoritmadan kaynaklanabilir. Birinci durumda kararsızlık diferansiyel denklemin tanımladığı fiziksel özellikten kaynaklanır. İkinci durumda ise kararsızlıktan, daha uygun bir algoritma seçilerek sakınılabılır.

Diferansiyel denklemin kararlılığı:

$$y'(x) = f(x, y) \quad , \quad y(x_0) = y_0 \quad (3.38)$$

başlangıç değer probleminin çözümü olan fonksiyon y olsun. U ise y ile aynı diferansiyel denklemini sağlayan ve y ye çok yakın bir fonksiyon olsun. U için başlangıç koşulunun y ye göre çok az değişik olduğunu varsayalım. (Bu durum, mesela y_0 in nümerik bir yöntemle yaklaşık olarak hesaplanmasından kaynaklanabilir). Bu durumda U yu

$$U(x) = y(x) + \varepsilon \eta(x)$$

şeklinde ifade edebiliriz. Buradaki η “hata fonksiyonu” adını alır, ε ise $0 < \varepsilon \ll 1$ olacak şekilde bir parametredir. η aşağıdaki “diferansiyel varyasyonel denklem”i sağlar;

$$\eta' = f_y \eta$$

$f_y = C$ =sabit varsayımı altında bu denklemin çözümü

$$\eta(x) = \eta_0 e^{c(x-x_0)} \quad , \quad \eta(x_0) = \eta_0$$

şeklinde. Eğer $c < 0$ ise $\eta(x)$, x arttıkça azalır. Bu durumda diferansiyel denklem kararlıdır. Aksi halde kararsızdır. Kararlı bir diferansiyel denklem için farklı başlangıç koşullarına ilişkin çözümler arasındaki fark x 'in artışıyla birlikte azalır; dolayısıyla mesela herhangi bir noktada nümerik metodun yaptığı hata ilerleyen adımlarda azalma eğiliminde olur. (13,9,10)

3.1.7. Uygun Metot Seçimi

Şu ana kadar irdelenen hiçbir metodun her problem için diğer tüm metotlara oranla her açıdan iyi olduğunu ya da optimal olduğunu söylemek mümkün değildir. Ayrıca burada, problemi girdiğiniz zaman size en optimal metodu kesin olarak verebilecek bir formül ya da bir algoritma veya bir kesin kriter de yoktur. Her metodun kendine göre avantajları ve dezavantajları vardır ve kişi bunları irdeleyip, her problemin kendi yapısına uygun metodu ,o probleme özel olarak tayin etmelidir. Bu durumda bütün yapılacak olan yöntemlerin avantajlarını ve dezavantajlarını sıralamaya çalışmaktır. Şimdi bunu yapmaya çalışalım ;

Tek Adımlı Metotlar:

Bu metotları temsilen klasik Runge –Kutta metodu göz önüne alınmıştır.

Avantajları:

Ayrı bir başlangıç metoduna ihtiyaç duymaz kendi kendine yeter. Başlamak için, sabit yerel hata oranı $O(h^5)$ vardır, işlenmesi kolaydır, otomatik adım aralığı kontrolü kolaylıkla uygulanabilir.

Dezavantajları

Her adım 4 adet fonksiyonel değerin hesaplanmasını gerektirir.

Ekstrapolasyon Metodu:

Avantajları

Başlamak için kendi kendine yeterlidir, hata oranı sabit değildir.

Dezavantajları

Her adım için dikkate değer bir hesaplama zamanı gerektirir.

Çok Adımlı Metodlar:

Avantajları

Genel olarak, her adım yalnızca iki ya da üç fonksiyonel hesaplamaya ihtiyaç duyar, keyfi düzeyde yüksek mertebeden denklemlere kolaylıkla uygulanabilir.

Dezavantajları

Başlangıç için kendine yeterli değildir. Başlangıç değerlerinin herhangi başka bir metodla hesaplanmasına ya da bu değerlerin verilmesine ihtiyaç duyar.

Adım aralığı kontrolü mümkün olmasına rağmen çoğu zaman adım aralığının değişmesi başlangıç değerlerinin yeniden hesaplamasını gerektirir. (9,10)

3.2. FORTRAN (FORmula TRANslation) Programlama Dili

Bu programlama dili 1950 lerin sonunda IBM tarafından geliştirildi. Bu dil, programcılarının hesaplamaları matematiksel ifadeler şeklinde tanımlayabilmelerini sağlayan ilk ana programlama dilidir.

Mikroişlemcinin programlama komutları ile

```
LOAD
A
ADD
B
MULTIPLY C
STORE
D
```

şeklinde kalabalık dil talimatları kullanılarak yazılması gereken bir işlem zincirini, FORTRAN programcıları

$$D = (A + B) * C$$

şeklinde, orjinal matematiksel ifadeye oldukça paralel bir şekilde ifade edebilme olanağına kavuşmuşlardır (burada * sembolü çarpma işlemine karşılık gelir).

Bilgisayarlar en temel anlamda mikroişlemcilerinin elektronik devre konfigürasyonları ile tanıyabilir hale geldiği, ortalama birkaç yüz adet, makine dilinde (0 lar ve 1 ler kullanılarak) yazılabilen komutlar ile idare edilirler ve bunun ötesine geçebilmek mümkün değildir. Mesela yukarıdaki ilk ifadeler mikroişlemci tarafından uygulanabilir olan yükle (LOAD), topla (ADD) gibi komutlardır. Ancak tabii ki yukarıda yazıldıkları halleri ile mikroişlemci tarafından algılanamazlar; bu komutların her birinin ikili tabanda 0 ve 1 'lerle ifade edilen sayısal karşılıkları vardır. Ancak bilgisayara kendi seçtiği işlem zincirlerini uygulamak isteyen her kişi (yani programcılar) bu karmaşık, hataya açık ve denetlemesi oldukça zor makine dili komutları ve programları ile uğraşmak zorunda kalmamalıdır. İşte programlama dillerinin hepsinin oluşturulmasını sağlayan neden budur.

Her programlama dili, kabaca 'insan' ile 'işlemci' arasında bir tercüman olarak görev yapar. Genellikle İngilizce ve matematik terimleri kullanılarak yazılan ifadeler bütünü olan program, programlama dilinin derleyicisi tarafından işlemcinin algılayabileceği makine dili programa dönüştürülür. Yani aslında programlama dilinin kendisi, bu dönüştürme işlemini yapan bir programdır.

FORTRAN'ın bir çok versiyonu çıkmıştır. Bunların en önemlileri FORTRAN IV (USASI Standart FORTRAN, 1966) ve FORTRAN 77 (ANSI FORTRAN , 1977) ' dir. Aşağıda basit bir FORTRAN IV programı görülmektedir.

```

C      1-100 arasi sayilarin toplamini
C      bulan Fortran IV programi
      INTEGER I, TOPLAM
      TOPLAM=0
      DO 30 I=1,100
30     TOPLAM=TOPLAM+I
      WRITE(6,20) TOPLAM
20     FORMAT(1X,I5)
      STOP
      END

```

Yorumlar sütun 1 de ki C ile başlar ve bu satırlar derleyici tarafından derlenmezler. Aslında programın parçası da değildirler. Sadece, programcının program ile uğraşırken işini kolaylaştırabilecek ipuçlarını ya da programı tanıttıcı bilgileri yazabilmesi için kullanılacak bölgelerdir.

Diğer tüm ifadeler 7'inci sütun ya da daha ötesinden başlarlar. Bunun yanısıra 2-5 no'lu sütunlarda bazı sayılar bulunabilir. Sütun 6 da ise boşluk bırakılmamış herhangi bir satır, bir üst satırdaki ifadenin devamı olarak algılanır.

Örnekteki FORMAT ifadesi, çıktının tek bir boşluktan ve onu takip eden 5 karakterlik yer işgal eden bir tam sayıdan oluştuğunu belirtir. Boşluk gerçekte basılmaz. WRITE ifadesindeki 6 sayısı da yazıcıyı ifade eder (eski IBM bilgisayar mimarisinden bir kalıntı).

FORTRAN değişkenlerinin isimleri 6 basamağa kadar harf ya da sayıdan oluşabilir; ilk karakter bir harf olmalıdır. Değişkenler INTEGER (tamsayı), REAL (ondalık sayı),

DOUBLE PRECISION (ondalık sayı ancak daha fazla bitle daha hassas değer tutmak için), COMPLEX (karmaşık sayı), LOGICAL (mantıksal Boolean) ya da (FORTRAN 77 'de) CHARACTER (karakter) olarak tanımlanabilirler. Eğer bir değişken tanımlanmamışsa , adı, I, J, K, L, M ya da N ile başladığı takdirde bir tamsayı; aksi takdirde kesirli sayı olarak kabul edilir.

FORTRAN IV ile karşılaştırıldığında FORTRAN 77 'nin daha karmaşık döngü ve blok IF-THEN-ELSE komutlarına sahip olduğunu ve karakter işlemeye yönelik daha fazla komutunun bulunduğunu görebiliriz. (15)

3.3. Bloch Denklemlerinin Nümerik Çözümünü Yapan Fortran Programı

Program oluşturulurken, Bloch denklemlerini çözmek için nümerik yöntem olarak, klasik Runge – Kutta metodu olarak da bilinen 4. mertebeden Runge – Kutta metodu kullanılmıştır.

Program, bir ana (çağırıcı) program ve bir alt programdan oluşur. Ana program, metodun gerektirdiği 4 adet k_1, k_2, k_3, k_4 değerini hesaplamak için her seferinde alt programı çağırır. Alt program standart bir işlem zincirini belli parametrelere uygular. Ana program alt programı her çağırışından önce alt programın kullanacağı parametrelere ilişkin gerekli düzenlemeyi yapar ve alt programın işleyişi bittikten sonra hesaplanan değerlerin uygun şekilde işler.

Bilindiği üzere Bloch denklemleri 3 adet diferansiyel denklemden oluşur. Program, 3 denklemleri bağımsız olarak çözmeye çalışan özdeş işlem zincirlerinin eş zamanlı yürütülmesi esasına dayanarak, kullandığı nümerik yöntemi üçlü bir denklem takımına uygulamıştır. Dolayısıyla, çözümün her adımı için programın hesaplamaya ihtiyaç duyduğu parametre sayısı 4 den (k_1, k_2, k_3, k_4) 12'ye yükselmiştir. Bu nedenle hesaplama zamanı üç katına çıkmıştır.

Mıknatıslanma vektörünün üç bileşenine ilişkin denklemlerin her biri için ayrı bir adım aralığı kullanılmamıştır. Bunun yerine, üç denklem için gereken farklı adım aralıklarının en küçüğü üç denklem için ortak adım aralığı olarak alınmıştır.

Program yapılırken, FORTRAN 77'de olup FORTRAN IV'te olmayan IF-THEN-ELSE yapıları oldukça işe yaramış ve sıklıkla kullanılmıştır.

Programın başında özellikle adım aralığının değeri üzerinde durulmuş ve programın kendi içinde gerekli adım aralığı hesaplanmıştır. Daha sonra bu adım aralığı kullanılarak

program 1000000 adım işleyecek şekilde tasarlanmıştır. Bu noktadan sonra program kullanıcıya başlangıçtan itibaren hangi ana kadar hesaplamaların yapıldığını ve bu zamanın T_1 ve T_2 'nin % kaçına kadar bir dilim olduğunu bildirir. Daha sonra bu periyot boyunca hesaplanmış yaklaşık değerleri görmek isteyip istemediğini sorar. Eğer istenirse bu zaman dilimi içerisinde herhangi bir an yazılarak, mıknatıslanma vektörünün bileşenlerinin o ana ilişkin değerleri (yani Bloch denklemlerinin çözümleri) görülebilir. Ancak kullanıcı bu zaman dilimindeki değerlerle ilgilenmek istemeyip; bir sonraki periyot için hesaplamaların devam etmesini sağlayabilir. Bu yöndeki isteği her aşamada program tarafından kullanıcıya sorulur. Kullanıcı bunu istediğinde, önceki periyodun hesaplanan son değerleri ve bunlara ilişkin zaman başlangıç alınarak aynı adım aralığı ile yeni bir 1000000 değer için daha hesaplama yapılır. Ve tüm işlemler aynı şekilde bu ve daha sonraki periyotlar için devam eder.

Program durdurulmak istendiğinde sonuçların görülmek istendiği zaman değeri olarak negatif bir sayı girilir. Program başlangıç anı olarak $t=0$ anını kullanmaktadır.

Herhangi bir periyoda ilişkin hesap yapıldığında, yalnızca o periyoda ilişkin değerler görüntülenebilir, öncesine dönmek mümkün değildir. Gerekli durumlarda program durdurulup aynı değerler girilerek yeniden çalıştırılmalıdır.

BÖLÜM 4.

SONUÇLAR VE TARTIŞMA

Bloch denklemlerini çözmek için nümerik yöntem olarak 'klasik Runge-Kutta metodu' olarak da bilinen 4. mertebeden Runge-Kutta metodu kullanılmıştır. Bu metot Bloch denklemleri gibi başlangıç değer problemleri için oldukça iyi sonuç vermesinin yanında, kolay programlanabilir olmasının da getirdiği avantaj düşünülmüş ve seçilmiştir. Bu metot, yazılan programda da görüldüğü gibi, tek bir altprogram kullanılarak programlanabilmektedir. Bir çok problemin çözümünde oldukça popüler olduğu söylenebilecek bu metot, bizim problemimiz olan 'Bloch denklemleri' için de uygun metot olarak düşünülmüştür.

Bu metodun işleneceği programlama dili olarak da FORTRAN 77 seçilmiştir. Çünkü en temel ve bilimsel altyapıya en uygun programlama dili (daha çok matematikçilerin, fizikçilerin ve mühendislerin kullandığı dil) FORTRAN programlama dilidir.

Günümüzde daha gelişkin görsel özelliklere (grafik arabirimleri, üç boyutlu çizimler gibi) sahip programlama dilleri bulunmasına rağmen, programcıya en az esneklik tanınması itibarı ile kullanımı en zor programlama dili olan FORTRAN, programcıyı bilgisayar kuralları ile diğer dillere oranla daha çok kuşatıp, bilgisayarın sınırlarını en çok hissettiren programlama dili olma özelliğini taşır. Bu özelliği nedeni ile programcıların (özellikle yoğun matematik ve hesaplamalarla ilgilenen programcıların) başlangıç olarak FORTRAN'ı seçmeleri tavsiye edilir. FORTRAN'ı kullanmayı öğrendikten sonra diğer dilleri kullanmaya başlayan bir programcı, bilgisayarın çalışma prensiplerini ve programlama mantığını en iyi şekilde kavrama olasılığını arttırır. Tüm bu nedenler göz önüne alınarak Bloch denklemlerinin çözümünü veren programı yazmak için FORTRAN 77 programlama dili seçilmiştir. Bu seçimin oluşturacağı altyapının üzerine, ileride daha gelişmiş çeşitli simülasyon programları oluşturmak mümkün olabilecektir.

Bloch denklemlerinin çözümü için seçilen 4. mertebeden Runge-Kutta algoritması Ek1 deki programın tam dökümünde görüldüğü şekilde FORTRAN 77 programlama dilinde işlenmiştir.

Temel çözüm algoritmasına, yani klasik Runge-Kutta metoduna ek olarak, metodun ihtiyaç duyduğu uygun adım aralığını, programa başlangıçta girilen değerlere uygun olacak şekilde tespit eden bir algoritma geliştirilip programa eklenmiştir. Bu algoritmanın özü, adım aralığı küçüldükçe doğruluk derecesinin her zaman artacağı düşüncesine dayanır. Ancak,

kolayca tahmin edilebileceği gibi adım aralığını keyfi olarak istediğimiz kadar küçültmek mümkün değildir. Çünkü eğer bu yapılırsa hesaplama zamanı tahammül edilemeyecek kadar artar ve buna karşılık elde edilen sonuçların doğruluk derecesi dikkate degecek ölçüde artmayabilir. Bu nedenle hatayı yeterince (istenilen oranda) küçülten ancak hesaplama zamanı yönünden de tasarruflu davranan optimal bir adım aralığı tespit etmek programın kalitesi açısından zorunlu hale gelmektedir.

Oluşturulan programda adım aralığının tespiti için kullanılan algoritma şu şekildedir; önce belli bir adım aralığı değeri için tek bir adım atılarak ilk yaklaşık değer hesaplanır. Sonra başa dönülüp adım aralığı yarıya düşürülerek aynı nokta için yaklaşık değer, bu kez iki adım atılarak hesaplanır. İki farklı adım aralığıyla aynı nokta için hesaplanan bu iki değer kıyaslanır. Eğer adım aralığını yarıya düşürmekle oluşan fark belli bir oranın (bizim yaptığımız program için % 0.1) altında kalıyor ise bu durumda adım aralığının yarıya düşmemiş değeri (tek adımda hesap yapmak için kullanılan) uygun bir değerdir ve bu adım aralığını daha da küçültmenin bir anlamı yoktur. Ancak eğer fark oranın üstünde ise bu kez adım aralığı yarıya düşürülür ve başa dönülüp aynı prosedür bu kez yarım aralığına uygulanır. Bu işlem zinciri, aradaki fark istenilen oranın altına düşünceye kadar adım aralığı küçültülerek sürer. Sonuçta uygun adım aralığı, gereğinden fazla küçültülmeden bu yolla tespit edilir.

Bloch denklemlerinin analitik ve nümerik yolla yapılan çözümlerini irdeleyerek her iki temel yöntem için birtakım genel yargılara varabiliriz. Şöyle ki; Bloch denklemlerini analitik yolla çözmek istediğimizde, sonuca özel bir varsayım altında ulaşabilmekteyiz. Bu varsayım, spin sisteminin döner koordinatlarda kararlı duruma ulaştığı yani mıknatıslanma vektörünün kutuplanmasının ve z' doğrultusundan sapmasının çok yavaş olduğu varsayımdır. Buna göre denklemler;

$$\frac{dM_{x'}}{dt} = \frac{dM_{y'}}{dt} = \frac{dM_{z'}}{dt} = 0 \quad \text{yazılarak çözülebilmektedir.}$$

Nümerik yöntemle çözüme gidebilmek için bu varsayımlara ihtiyaç duyulmamaktadır. Dolayısıyla analitik çözüm için geçerli varsayımın sağlanamadığı aralıklarda analitik çözüm sonuç vermez iken, nümerik yöntemle yapılan çözüm hala etkinliğini sürdürmekte ve sonuç verebilmektedir.

Ayrıca çözümü çok fazla zaman alabilecek, çok zor olabilecek ya da çok fazla değer ve parametre içeren bir denklem grubunun temsil ettiği bir deneyi laboratuarda, ancak sınırlı sayıda değerle ve uzun süren ölçümlerle gerçekleştirebilirken; söz konusu denklemleri bir nümerik yöntem aracılığıyla bilgisayara işleyerek bir deney simülasyon programı oluşturulduğunda, araştırmacı tarafından denetlenmesi imkansız gibi görünen bir veri yığını, bilgisayarın sunduğu olanakları kullanarak çok değişik açılardan aynı anda denetlemek ve işlemek mümkün olabilmektedir.

Bütün bunlara ek olarak, gerçek bir deneyde belki de asla ulaşamayacağımız bazı parametre değerlerini (örneğin: 100 Tesla'lık bir magnetik alan değeri ya da 0.94 c hızında giden bir gemi gibi) simülasyon programı sayesinde test etmek ve sonuçları gözlemek oldukça kolaydır.

NMR spektrometresi ile yapılan bir deneyde magnetik sistem üzerine uygulanan RF (Radyo Frekans) alanı 90^0 - 180^0 atması uygulanıp kesildikten sonra mıknatıslanma vektörünün M_x , M_y ve M_z bileşenleri incelenmektedir. Standart bir NMR deneyinde ortalama 10-15 parametre girilebilirken; aynı deneyi simüle eden bir programda böyle bir sınır sözkonusu değildir.

Ancak nümerik yöntem kullanmanın belli bazı dezavantajları da çözüm boyunca kendini hissettirmiştir. Bunlardan ilki, çözümün doğruluk derecesinin küçük de olsa nümerik yöntemin kullanılmasına paralel azalmasıdır. İkincisi ise, birinci sıkıntıyı gidermek için uygulanan en etkili yöntem olan adım aralığının küçültülmesi durumunda, hesaplanma zamanının oldukça uzayabilir olmasıdır.

Sonuç olarak, Bloch denklemleri gibi analitik çözümü yapılabilen denklemler için aynı şiddette olmasa bile; çözümü çok zor yapılabilen hatta hiç yapılamayan problemler için nümerik yöntemlerin ve bilgisayar destekli çözümün göz ardı edilemez bir alternatif olduğu söylenebilir.

Bu çalışmada, sonuç olarak, geniş çaplı bir simülasyon programı oluşturmaya doğru gidebilecek bir yolun altyapısı hazırlanmış, ayrıca henüz çözülememiş bir takım problemlere nümerik yaklaşımlarla çözüm aranabilecek bir platform sağlanabileceği düşünülmüştür..

READ(*,*)X(1),Y(1),Z(1)

TN=0

H=0.001

TTN=TN

XMAX=X(1)

XMIN=X(1)

YMAX=Y(1)

YMIN=Y(1)

ZMAX=Z(1)

ZMIN=Z(1)

WRITE(*,*)'JiROMAGNETİK ORANI GİRİNİZ : G'

READ(*,*)G

WRITE(*,*)'SABİT VE DEĞİSKEN MAGNETİK ALAN SİDDETLERİNİ

GİRİNİZ : Bo, Brf'

READ(*,*)BZ,BX

WRITE(*,*)'RELAKSASYON SABİTLERİNİ GİRİNİZ : T1,T2'

READ(*,*)T1,T2

Z0=Z(1)

15 H=H/2

DO 30 N=1,2

XN=X(N)

YN=Y(N)

ZN=Z(N)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK1=XK

YK1=YK

ZK1=ZK

TN=TN+H/2

XN=X(N)+H*(XK1/2)

YN=Y(N)+H*(YK1/2)

ZN=Z(N)+H*(ZK1/2)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK2=XK

YK2=YK

ZK2=ZK

XN=X(N)+H*(XK2/2)

YN=Y(N)+H*(YK2/2)

ZN=Z(N)+H*(ZK2/2)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK3=XK

YK3=YK

ZK3=ZK

TN=TN+H/2

XN=X(N)+H*XK3

YN=Y(N)+H*YK3

ZN=Z(N)+H*ZK3

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK4=XK

YK4=YK

ZK4=ZK

X(N+1)=X(N)+(H*(XK1+2*XK2+2*XK3+XK4))/6

Y(N+1)=Y(N)+(H*(YK1+2*YK2+2*YK3+YK4))/6

Z(N+1)=Z(N)+(H*(ZK1+2*ZK2+2*ZK3+ZK4))/6

30 CONTINUE

C*****

H=2*H

N=1

TN=TTN

XN=X(N)

YN=Y(N)

ZN=Z(N)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK1=XK

YK1=YK

ZK1=ZK

TN=TN+H/2

XN=X(N)+H*(XK1)/2

YN=Y(N)+H*(YK1)/2

ZN=Z(N)+H*(ZK1)/2

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK2=XK

YK2=YK

ZK2=ZK

XN=X(N)+H*(XK2/2)

YN=Y(N)+H*(YK2/2)

ZN=Z(N)+H*(ZK2/2)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK3=XK

YK3=YK

ZK3=ZK

TN=TN+H/2

XN=X(N)+H*XK3

YN=Y(N)+H*YK3

ZN=Z(N)+H*ZK3

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK4=XK

YK4=YK

ZK4=ZK

X(N+1)=X(N)+(H*(XK1+2*XK2+2*XK3+XK4))/6

Y(N+1)=Y(N)+(H*(YK1+2*YK2+2*YK3+YK4))/6

Z(N+1)=Z(N)+(H*(ZK1+2*ZK2+2*ZK3+ZK4))/6

C*****

C* adım aralığı tespiti *******

C*****

DX=X(3)-X(N+1)

DY=Y(3)-Y(N+1)

DZ=Z(3)-Z(N+1)

DDX=(DX/X(3))*100

DDY=(DY/Y(3))*100

DDZ=(DZ/Z(3))*100

F=0.1

IF(DDX.GT.F.OR.DDY.GT.F.OR.DDZ.GT.F)THEN

H=H/2

GO TO 15

ENDIF

C*****

WRITE(*,*)'ADIM ARALIGI',H

TN=TTN

55 DO 35 N=1,999999

XN=X(N)

YN=Y(N)

ZN=Z(N)

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

XK1=XK

YK1=YK

ZK1=ZK

$$TN=TN+H/2$$

$$XN=X(N)+H*(XK1/2)$$

$$YN=Y(N)+H*(YK1/2)$$

$$ZN=Z(N)+H*(ZK1/2)$$

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

$$XK2=XK$$

$$YK2=YK$$

$$ZK2=ZK$$

$$XN=X(N)+H*(XK2/2)$$

$$YN=Y(N)+H*(YK2/2)$$

$$ZN=Z(N)+H*(ZK2/2)$$

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

$$XK3=XK$$

$$YK3=YK$$

$$ZK3=ZK$$

$$TN=TN+H/2$$

$$XN=X(N)+H*XK3$$

$$YN=Y(N)+H*YK3$$

$$ZN=Z(N)+H*ZK3$$

CALL FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

$$XK4=XK$$

$$YK4=YK$$

$$ZK4=ZK$$

$$X(N+1)=X(N)+(H*(XK1+2*XK2+2*XK3+XK4))/6$$

$$Y(N+1)=Y(N)+(H*(YK1+2*YK2+2*YK3+YK4))/6$$

35 $Z(N+1)=Z(N)+(H*(ZK1+2*ZK2+2*ZK3+ZK4))/6$

C*****

C***** Minimum ve maksimum değerlerin tespiti ***

C*****

DO 45 KL=1,1000000

IF(X(KL).GT.XMAX)THEN

 XMAX=X(KL)

ELSE IF(X(KL).LT.XMIN)THEN

 XMIN=X(KL)

ENDIF

IF(Y(KL).GT.YMAX)THEN

 YMAX=Y(KL)

ELSE IF(Y(KL).LT.YMIN)THEN

 YMIN=Y(KL)

ENDIF

IF(Z(KL).GT.ZMAX)THEN

 ZMAX=Z(KL)

ELSE IF(Z(KL).LT.ZMIN)THEN

 ZMIN=Z(KL)

ENDIF

45 CONTINUE

C*****

TIME=TTN+H*1000000

WRITE(*,101)TTN,TIME

**101 FORMAT(/,,'program ',E14.8,2X,'ile',2X,E14.8,' nci saniyeler arasindaki
yaklasik**

+degerleri hesapladı',/)

T1YZD=100*(TIME/T1)

T2YZD=100*(TIME/T2)

WRITE(*,104)T1YZD,T2YZD

**104 FORMAT(/,'T1 in %',F18.6,' sine ve T2 nin %',F18.6,' sine kadar hesaplama
yapildi.',/)**

WRITE(*,105)

**105 FORMAT(/,'Bu zaman dilimindeki degerleri gormeden hesaplamanin bir periyot
daha devam et**

+mesini ister misiniz?(E/H)')

READ(*,*)CHA

IF(CH.A.EQ.'E')THEN

X(1)=X(1000000)

Y(1)=Y(1000000)

Z(1)=Z(1000000)

TTN=TN

GO TO 55

ENDIF

**WRITE(*,*)'Magnetizasyon vektorunun bileşenlerinin su ana kadar hesaplanan
minimum ve max**

+imum degerleri asagidaki gibidir:'

WRITE(*,102)XMAX,XMIN,YMAX,YMIN,ZMAX,ZMIN

102

```

    FORMAT(/,'MXmax=',F14.6,5X,'MXmin=',F14.6,/,/, 'MYmax=',F14.6,5X,'MYmi
n=',F14.6,/,/, 'MZmax
+=',F14.6,5X,'MZmin=',F14.6,/,/)

```

```

40  WRITE(*,107)TTN,TN

```

```

107  FORMAT(/,'Hesaplama araligi [ ',F16.12,' ',F16.12,' ] seklindedir.',/)

```

```

    WRITE(*,*)'Bu araliktaki bir yerde magnetizasyon degerlerini gormek icin
zamani girmeniz
+ yeterlidir.'
```

```

    READ(*,*)TTIME

```

```

    IF(TTIME.LT.0)GO TO 20

```

```

    IF(TTIME.LT.TTN.OR.TTIME.GT.TN)THEN

```

```

    WRITE(*,*)'Girdiginiz zaman degeri hesap yapilan zaman diliminin disinda
kalmaktadir.Arali
+gin icinde bir deger girin.'
```

```

    GO TO 40

```

```

    ENDIF

```

```

    JKL=(TTIME-TTN)/H

```

```

    WRITE(*,100)TTIME,X(JKL),TTIME,Y(JKL),TTIME,Z(JKL)

```

100

```

    FORMAT(5X,'Mx(',E14.9,')=',F20.10,/,5X,'My(',E14.9,')=',F20.10,/,5X,'Mz(',E14.
9,')=',F20.10)

```

```

    WRITE(*,106)

```

```

106  FORMAT(/,'Bir sonraki periyot icin hesaplamanin devam etmesini ister
misiniz?(E/H)')

```

```

    READ(*,*)CHA

```

```

    IF(CH.A.EQ.'E')THEN

```

X(1)=X(1000000)

Y(1)=Y(1000000)

Z(1)=Z(1000000)

TTN=TN

GO TO 55

ENDIF

GO TO 40

20 STOP

END

C***** altprogram *****

SUBROUTINE FNIL(H,TN,G,BZ,BX,T1,T2,XN,YN,ZN,XK,YK,ZK)

S=SIN(G*BZ*TN)

C=COS(G*BZ*TN)

XK=G*(YN*BZ-ZN*BX*sin(G*BZ*TN))-XN/T2

YK=G*(ZN*BX*cos(G*BZ*TN)-XN*BZ)-YN/T2

ZK=G*(XN*BX*sin(G*BZ*TN)-YN*BX*cos(G*BZ*TN))+(Z0-ZN)/T1

RETURN

END

KAYNAKÇA

- 1) TABAK, F. , 1994. NMR'ın Temelleri, Hacettepe Üniversitesi, ANKARA
- 2) HENNEL, W. J. and KLINOWSKI, J. , 1993. Çekirdek Magnetik Rezonansın Temelleri, (Çeviren; Semiha BAHÇELİ). ANKARA
- 3) ABRAGAM, A. , 1961. The Principles of Nuclear Magnetism, Oxford University, LONDON
- 4) Bloch, F., HANSEN, W.W., PACKARD, M., "Nuclear Induction", Phys. Rev. ,69; 127, (1946)
- 5) PINES, D., SLICHTER, C.P., "Relaxation Times in Magnetic Resonance", Phys. Rev., 100;1014-1020, (1955)
- 6) SLICHTER, C.P. , 1996. Principles of Magnetik Resonance, New York, U.S.A.
- 7) ORAL, B. , 1972, Magnetik Rezonans, Hacettepe Üniversitesi, ANKARA.
- 8) APAYDIN, F. , 1996. Magnetik Rezonans, Hacettepe Üniversitesi, ANKARA.
- 9) MÜLLGES, G.E. and UHLİG, F. ,1996. Numerical Algoritms with Fortran, Berlin, GERMANY
- 10) HULL, T.E. , ENRIGHT, W. H. , FELLE, B. M. , SEDGWICK, A. E. , 1972, "Comparing numerical methods for ordinary differential equations.", Journal of Numerical Analysis of the Society for Industrial and Applied Mathematics, 9; 603-637
- 11) JACKSON, K. R. , ENRIGHT, W. H. , HULL, T. E. , 1978, "A Theoretical Criterion for Comparing Runge-Kutta Formulas." Journal of Numerical Analysis of the Society for Industrial and Applied Mathematics." , 15; 618-641
- 12) AYDIN, M. , GÜNDÜZ, G. ,KURYEL, B. , 1987. Diferansiyel Denklemler ve Uygulamaları, Ege Üniversitesi, İZMİR
- 13) KINCALD, D. ,CHENEY, W. , 1991. Numerical Analysis, The University of Texas at Austin, California, U.S.A.
- 14) SCHMID, W.E. , SPITZ, G. , LÖSCH, W. ,1988. Theoretical Physics on the Personal Computer, Berlin, GERMANY. (Translator: A. H. Armstrong).
- 15) DOWNING, D.A. , COVINGTON, M. A. , COVINGTON, M. M. ,1996. Dictionary of Computer and Internet Terms, New York, U.S.A.

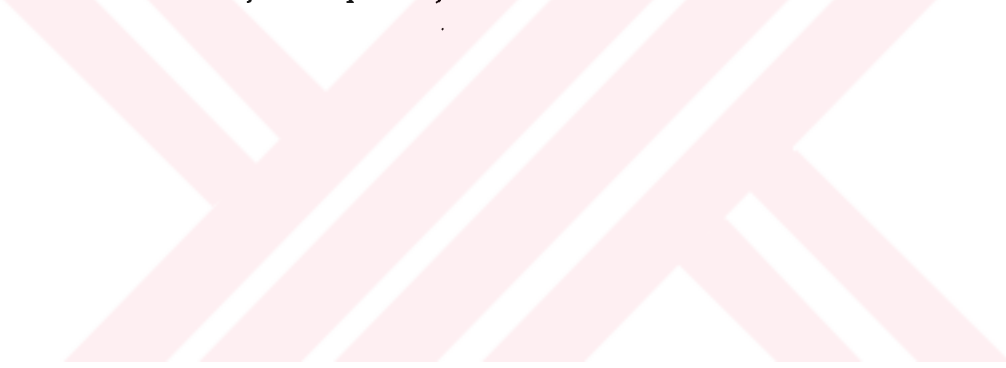
TABLO LİSTESİ

Tablo 1. Runge-Kutta Metodları için metodun mertebesi ve hata oranı arasındaki ilişki.

Tablo 2. Klasik Runge-Kutta Metodu için hesaplama şeması.

Tablo 3. Runge-Kutta Metodları için A_j , a_j , b_j katsayıları.

Tablo 4. Adams-Moulton Metodu için hesaplama şeması.



ŞEKİL LİSTESİ

Şekil 2.1. Mıknatıslanma vektörünün durumu.

Şekil 2.2. Döner koordinat sistemi.

Şekil 2.3. 90° atması kesildikten sonra M_z ve M_e nin durumu



ÖZ GEÇMİŞ

1972 yılında Diyarbakır'da doğdum. İlk öğrenimimi İstanbul'da İskenderpaşa İlkokulunda, orta ve lise öğrenimimi ise yine İstanbul'da Kadıköy Erenköy Kız Lisesinde tamamladım. 1991 yılında Dicle Üniversitesi Fen – Edebiyat Fakültesi Fizik Bölümünü kazandım. 1995 yılında mezun olduktan sonra 1996 yılında Dicle Üniversitesi Fen – Edebiyat Fakültesi Fizik Bölümünde Yüksek Enerji ve Plazma Fiziği Anabilim dalında Araştırma Görevlisi olarak göreve başladım. 1998 – 1999 öğretim yılında Dicle Üniversitesi Fen Bilimleri Enstitüsü Fizik Anabilim Dalında Yüksek Lisans Eğitimine başladım. Halen Dicle Üniversitesi Fen-Edebiyat Fakültesi Fizik Bölümünde Araştırma Görevlisi olarak çalışmaktayım.

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMAN YAYIN MERKEZİ**