ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

Tez YAZARI

SONLU DOĞURAYLI DEĞİŞMELİ MONOİDLER VE UYGULAMALARI

MATEMATİK ANABİLİM DALI

ADANA, 2005

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SONLU DOĞURAYLI DEĞİŞMELİ MONOİDLER VE UYGULAMALARI

Tez YAZARI

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

Bu tez ..... tarihinde aşağıdaki jüri üyeleri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

İmza.....	İmza.....	İmza.....
Prof.Dr. Danışman ÜYE	Doç.Dr. Birinci ÜYE	Yrd.Doç.Dr. İkinci ÜYE
DANIŞMAN	ÜYE	ÜYE

Bu tez Enstitümüz Matematik Anabilim Dalında hazırlanmıştır.
Kod No:

Prof.Dr. Aziz ERTUNÇ
Enstitü Müdürü
İmza ve Mühür

Bu Çalışma Ç.Ü. Bilimsel Araştırma Projeleri Birimi Tarafından Desteklenmiştir.

Proje No: FEF2003YL??

Sevgili Anne ve Babama

ÖZ

YÜKSEK LİSANS TEZİ

SONLU DOĞURAYLI DEĞİŞMELİ MONOİDLER VE UYGULAMALARI

Tez YAZARI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

Danışman: Prof.Dr. Danışman ÜYE
Yıl: 2005, Sayfa: 60
Jüri: Prof.Dr. Danışman ÜYE
Doç.Dr. Birinci ÜYE
Yrd.Doç.Dr. İkinci ÜYE

Bu tezde abelyen monoid teorisinin klasik sonuçları nı n bir derlemesi yapılmıştır. Ayrıca bu derlemede yer alan temel algoritmaları n program olarak hayata geçirilebilmeleri için MS Visual Basic dilinde bir nesnel yapı önerilmiştir. Bu derlemede yer alan baş lıca teorem ve algoritmalar şunlardır:

M, $\mathbb{Z}^n$ nin invaryant faktörleri $d_1, \ldots, d_r$ olan bir alt grubu ise

$$\mathbb{Z}^n/M \simeq \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^{n-r}$$

olduğu ve M nin bir doğuray kümesinden veya denklemlerinden bir bazının nasıl hesaplandığı gösterilmiştir.

Sonlu doğuraylı monoidlerin sadeleşmeli olma, burulmasız olma, indirgenmiş olma ve sonlu olma gibi özellikleri ve buna bağlı bazı sonuçlar incelenmiştir. Grillet'in, sonlu doğuraylı monoidler üzerindeki bir teoremine de değinilmiştir.

Minkowski-Farkas lemma ve bunun sonlu doğuraylı monoidler üzerindeki uygulamala- rından bahsedilmiştir. Bu lemma kullanılarak, $\mathbb{Q}^n$ nin bir alt uzayının; eğer varsa, belirli özellikteki negatif olmayan elemanlarını veya kuvvetli pozitif elemanlarını bulacak algoritmalar ile $\mathbb{N}^n/\sim_M$ nin bir grup olup olmadığına veya bir afin yarıgrubu olup olmadığına karar veren algoritmalar verilmiştir. $U(\mathbb{N}^n/\sim_M)$ nin hesaplanması için yine bir algoritma verilmiştir. Rasyonel katsayılı bir homojen lineer denklem sisteminin, tüm koordinatları negatif olmayan tamsayı olacak ş ekilde bir aşikar olmayan çözüme sahip olup olmadığının kontrolü için bir algoritma verilmiştir. $\mathbb{N}^n$ nin iki sonlu doğuraylı alt monoidinin kesişiminin aşikar olup olmadığının belirlenmesi, eğer aşikar değilse kesi şimindeki bir elemanın bulunması için bir metod verilmiştir.

Anahtar Kelimeler: Sonlu doğuraylı değişmeli monodiler, sadeleşmeli olma, burulmasız olma, indirgenmiş olma, kuvvetli pozitif eleman.

ABSTRACT

MSc THESIS

FINITELY GENERATED COMMUTATIVE MONOIDS AND ITS APPLICATIONS

Tez YAZARI

DEPARTMENT OF MATHEMATICS
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF ÇUKUROVA

Supervisor: Prof.Dr. Danışman ÜYE
Year: 2005, Pages: 60
Jury: Prof.Dr. Danışman ÜYE
Assoc.Prof.Dr. Birinci ÜYE
Assist.Prof.Dr. İkinci ÜYE

In this thesis we give an exposition of the fundamental results in abelian monoid theory. In addition to this we give a description of some classes in MS Visual Basic that can be uses to implement the algorithms which take place in this exposition. Following is a list of some important theorems and algorithms in this thesis.

If M is a subgroup of $\mathbb{Z}^n$ with invariant factors $d_1, \ldots, d_r$ then it is shown that

$$\mathbb{Z}^n/M \simeq \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^{n-r}.$$

Furthermore, we give an algorithm to compute a basis of a subgroup of $\mathbb{Z}^n$ from one of its systems of generators or its defining equations.

We discuss the properties of finitely generated monoids such as of being cancellative, torsion free, reduced and finite and give some results. We mention a result by Grillet on finitely generated monoids.

Minkowski-Farkas lemma and some of its consequences on finitely generated monoids are given. Especially we apply this lemma to decide when a subspace of $\mathbb{Q}^n$ has a non negative or strongly positive elements, if any. We give algorithms to find out whether $\mathbb{N}^n/\sim_M$ is a group or an affine semigroup. We describe an algorithm to calculate $U(\mathbb{N}^n/\sim_M)$. An algorithmic method deciding whether a linear homogeneous system of equations has a nontrivial nonnegative solution and another algorithmic method deciding whether two submonoids of $\mathbb{N}^n$ have nontrivial intersection and if so then finding an element in intersection is given.

Key Words: Finitely generated commutative monoids, being cancellative, being torsion free, being reduced, strongly positive element.

ÖNSÖZ

Önsöz buraya yazılacak.

TEŞEKKÜR

Teşekkür buraya girilecek.

İÇİNDEKİLER **SAYFA**

ÇİZELGELER DİZİNİ **SAYFA**

ŞEKİLLER DİZİNİ **SAYFA**

SİMGELER VE KISALTMALAR DİZİNİ

Γ	gama fonksiyonu
γ	yüzey enerjisi
$\mathbf{F}$	kuvvet
T_{ij}	stres bileşenleri
$T^{\nu}_{\alpha\beta}$	tensör bileşenleri

1. TEMEL TANIMLAR VE SONUÇLAR

Gruplarda bölüm grubunu elde etmek için onun bir normal altgrubuna, halkalarda ise onun bir idealine gerek vardı r. Monoidlerde ise bu gereksinimi kongruanslar karşılar. Gerekli bütün tanımları yaptıktan sonra dikkatimizi sonlu doğuraylı monoidler ü zerinde yoğunlaştı racağız ve her sonlu doğuraylı monoidin $\mathbb{N}^n$ nin bir bölümü olduğu durumlarda bir temsil vereceğiz.

Bu tezde $\mathbb{Z}$ ile tamsayılar ve $\mathbb{N}$ ile negatif olmayan tamsayıların kümesi gösterilecektir.

Denklik bağıntıları ile ilgili temel bilgileri burada kı saca hatırlayalım. X herhangi bir küme ise $X \times X$ nin σ gibi bir alt kümesine X üzerinde bir *bağıntı* denir. $(a,b) \in \sigma$ yerine $a\sigma b$ gösterimini de kullanacağız.

X üzerindeki bir σ bağıntısına; her $a \in X$ iç in $a\sigma a$ oluyorsa *yansımalı*, her $a,b \in X$ için $a\sigma b$ iken $b\sigma a$ oluyorsa *simetrik*, her $a,b,c \in X$ iç in $a\sigma b$ ve $b\sigma c$ iken $a\sigma c$ oluyorsa *geçiş meli* denir. Eğer X üzerindeki bir σ bağıntısı yansımalı, simetrik ve geçişken ise σ ya bir *denklik bağıntısı* denir. σ bir denklik bağınıtı sı ve $a \in X$ ise $[a]_\sigma = \{b \in X \mid a\sigma b\}$ ya a nın *denklik sınıfı* denir. Yanlış anlaşılmaya yol açmayan durumlarda $[a]_\sigma$ yerine $[a]$ kullanacağız. Bilindiği gibi $a,b \in X$ ise $[a] \cap [b] \neq \emptyset$ olması için gerek ve yeter koşul $[a] = [b]$ olmasıdır. O halde denklik sınıfları X in bir parçalanışını verir. Herhangi iki elemanın denk olması için gerek ve yeter koş ul elemanların bu parçalanışın aynı kümesi içinde olmasıdır.

Tanım 1.1 S boştan farklı bir küme ve $+$, S üzerinde birleşme özelliğini sağlayan bir ikili işlem ise $(S,+)$ ikilisine bir *yarıgrup* denir.

Tanım 1.2 Bir $(S,+)$ yarıgrubunda her $a \in S$ için $a+0 = 0+a = a$ olacak şekilde bir $0 \in S$ varsa $(S,+)$ ikilisine bir *monoid* denir.

Not 1.3 Aksi belirtilmedikçe bu tezdeki bütün monoidleri ve yarıgrupları değişmeli olarak alacağız.

Tanım 1.4 Bir S monoidinde her $a,b,c \in S$ için $a+c = b+c$ olduğunda $a = b$ oluyorsa S monoidi *sadeleşmelidir* denir.

Tanım 1.5 Bir $(S,+)$ monoidinde her $a \in S$ için $a+b = 0$ olacak şekilde bir $b \in S$ varsa $(S,+)$ ikilisine bir *grup* denir.

Eğer bir monoidde bir elemanının tersi varsa tektir. Bu yüzden bir a elemanının tersini $-a$ ile göstereceğiz. Bir grubun daima sadeleşmeli bir monoid olacağı açıktır.

Tanım 1.6 S bir monoid ve $H \subseteq S$ olsun. $0 \in H$ ve her $a, b \in H$ için $a+b \in H$ ise H ye S nin bir *alt monoidi* denir.

Verilen bir S monoidinin bütün alt monoidlerinin kesişiminin de, S nin bir alt monoidi olduğu açıktır. Bir S monoidinin bir A alt kümesi verilsin. O halde S nin A yı içeren tüm alt monoidlerinin arakesiti de bir monoiddir. Bu monoide A tarafından doğurulan monoid denir ve $\langle A \rangle$ ile gö sterilir. $\langle A \rangle$ monoidi S nin A yı içeren (kapsamaya göre) en küçük alt monoididir. Dahası eğer $0a = 0$ ve $0 \leq n$ için $(n+1)\,a = na + a$ ise

$$\langle A \rangle = \left\{ \sum_{i=1}^{r} n_i a_i \mid r \in \mathbb{N} \text{ ve her } 1 \leq i \leq r \text{ için } n_i \in \mathbb{N},\, a_i \in A \right\}$$

olduğu kolayca ispatlanabilir. Aşağıda aksi söylenmedikçe S sembolü bir monoidi gösterecektir.

Tanım 1.7 $S = \langle A \rangle$ ise S ye A tarafından doğurulmu ş ve A ya da S nin bir *doğuray kümesi* denir.

$S = \langle S \rangle$ olacağı açıktır.

Tanım 1.8 $S = \langle A \rangle$ ve A nın S yi doğuracak ş ekilde hiçbir özalt kümesi yok ise S ye A tarafından *minimal olarak doğurulmuş* ve A ya da S nin bir *minimal doğuray* kümesi denir.

Tanım 1.9 Eğer S nin, $S = \langle A \rangle$ olacak şekilde sonlu bir A alt kümesi varsa S ye *sonlu doğuraylı* denir.

A, S grubunun bir alt kümesi ise $\langle A \rangle$ monoidi aslında S nin bir alt grubudur. A tarafından doğurulan bu alt grubu $\mathbf{G}(A)$ ile göstereceğiz. Eğ er her $n \in \mathbb{N}$ ve $a \in A$ için $n(-a)$ elemanını $(-n)\,a$ şeklinde gösterirsek,

$$\mathbf{G}(A) = \left\{ \sum_{i=1}^{r} z_i a_i \mid r \in \mathbb{N} \text{ ve her } 1 \leq i \leq r \text{ için } z_i \in \mathbb{Z},\, a_i \in A \right\}$$

olduğu kolayca gösterilebilir.

Tanım 1.10 Eğer A sonlu ve $S = \mathbf{G}(A)$ ise S ye *sonlu doğuraylı grup* denir.

$S_1,\ldots,S_n$ bir monoidler (gruplar) dizisi olsun. $S_1,...,S_n$ kümelerinin kartezyen çarpımı olan $S_1\times\ldots\times S_n$ kümesi üzerinde

$$(a_1,\ldots,a_n)+(b_1,\ldots,b_n)=(a_1+b_1,\ldots,a_n+b_n)$$

şeklinde bir işlem tanımlarsak $(S_1\times\ldots\times S_n,+)$ bir monoid (grup) olur. Buna $S_1,\ldots,S_n$ nin *direk çarpımı* denir. Eğer S_i ler sadeleşmeli monoid ise $S_1\times\ldots\times S_n$ de sadeleşmeli monoiddir. S_i ler sonlu doğuraylı ise $S_1,\ldots,S_n$ nin direkç arpımı da sonlu doğuraylıdır.

Gösterim 1.11 $S\times\ldots\times S$ yi S^n ile göstereceğiz.

Tanım 1.12 σ, S üzerinde bir bağıntı olsun. Eğer her $a\sigma b$ ve her $c\in S$ için $(a+c)\,\sigma\,(b+c)$ oluyorsa σ ya S üzerinde *uyumlu bir bağ ıntı* denir.

σ, S üzerinde uyumlu bir bağıntı ise, kolayca görüleceği gibi her $a\sigma b$, $c\sigma d$ için $(a+c)\,\sigma\,(b+d)$ dir.

Tanım 1.13 S üzerindeki uyumlu bir σ denklik bağıntısı na S üzerinde bir *kongruans* denir.

Gösterim 1.14 Bir S monoidi üzerindeki verilen bir σ kongruansı için S/σ ile S nin tüm elemanlarının denklik sınıflarının kümesini göstereceğiz.

S/σ üzerinde $+$ işlemi $[a]+[b]=[a+b]$ olarak tanımlanırsa $(S/\sigma,+)$ nın bir değişmeli monoid olduğu kolayca gösterilebilir. S/σ monoidine S nin σ modülüne göre *bö lüm monoidi* denir. Benzer şekilde S bir grup ise S/σ nın bir grup ve S sonlu doğuraylı ise S/σ nin de sonlu doğ uraylı olduğu kolayca görülebilir.

Tanım 1.15 S,S' iki monoid olsun. $f:S\to S'$ bir fonksiyon ve $f(0)=0$, $\forall a,b\in S$ için $f(a+b)=f(a)+f(b)$ ise f ye monoid *morfizmi* denir. Eğer f bire-bir {örten [hem bire-bir hem ö rten]} ise f ye monomorfizm {epimorfizm [izomorfizm]} denir.

Eğer $f:S\to S'$ izomorfizm ise $f:S\simeq S'$ veya kısaca $S\simeq S'$ yazılır.

Eğer f izomorfizm ise f nin f^{-1} tersi de izomorfizmdir. Eğ er herhangi iki monoid arasında bir izomorfizm varsa bu iki monoid *izomorfiktir* denir. Sadeleşmeli olma, grup olma, sonlu doğuraylı olma özellikleri izomorfizmalar altında korunur.

$f : S \to S'$ bir monoid morfizmi olsun. S üzerinde $\ker(f)$ bağıntısını

$$\ker(f) = \{(a,b) \in S \times S \mid f(a) = f(b)\}$$

şeklinde tanımlayalım. f bir monoid morfizmi olduğundan $\ker(f)$ nin bir kongruans olduğu kolayca gösterilebilir. Bu kongruans f nin *çekirdek kongruansı* olarak bilinir.

f nin görüntü kümesi

$$\mathrm{Im}(f) = \{f(a) \mid a \in S\}$$

ile gösterilir. f bir monoid morfizmi olduğundan $\mathrm{Im}(f)$ de bir monoid olur.

Teorem 1.16 $f : S \to S'$ bir monoid morfizmi olsun. O zaman

$$\overline{f} : S/\ker(f) \to \mathrm{Im}(f)$$
$$\overline{f}([a]) = f(a)$$

bir izomorfizmdir. □

Teorem 1.17 S, $\{s_1, \ldots, s_n\}$ tarafından doğ urulan bir monoid olsun. O zaman $\mathbb{N}^n$ monoidi üzerinde, $S \simeq \mathbb{N}^n/\sigma$ olacak şekilde bir σ kongruansı vardır.

İspat: $f : \mathbb{N}^n \to S$, $f(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i s_i$ fonksiyonunu tanımlayalım. Bu fonksiyon bir epimorfizmdir. σ yı $\ker(f)$ olarak alırsak bir önceki teoremden $\overline{f}$ izomorfizm olur. ■

$\mathbb{N}^n$ üzerindeki kongruansların özelliklerini çalışmak, sonlu doğuraylı monoidlerin özelliklerini çalışmakla hemen hemen aynıdır.

önce sadeleşme özelliği ile başlayalım. Bu amaçla $\mathbb{Z}^n$ nin her alt grubuna bir kongruans, her kongruansa da $\mathbb{Z}^n$ nin bir alt grubunu karşılık getireceğiz.

Tanım 1.18 σ, $\mathbb{N}^n$ üzerinde bir kongruans olsun. $M_\sigma = \{a - b \mid (a,b) \in \sigma\} \subseteq \mathbb{Z}^n$ tanımlayalım. Burada $a - b \in \mathbb{Z}^n$, $a \in \mathbb{N}^n$ den $b \in \mathbb{N}^n$ nin bileşen bileşen çıkarılması sonucu elde edilen elemandır. σ bir kongruans olduğ undan M_σ, $\mathbb{Z}^n$ nin bir alt grubu olur.

Tanım 1.19 H, $\mathbb{Z}^n$ nin bir alt grubu olsun. $\sim_H = \{(a,b) \in \mathbb{N}^n \times \mathbb{N}^n \mid a - b \in H\}$ tanımlayalım. $\sim_H$, $\mathbb{N}^n$ üzerinde bir kongruansdır.

öncelikle $M_{\sim_H}$ ın yapısını inceleyelim.

$x \in M_{\sim_H}$ olsun. Bu durumda en az bir $(a,b) \in \sim_H$ için $x = a - b$ olur. O halde $(a,b) \in \mathbb{N}^n \times \mathbb{N}^n$ ve $x = a - b \in H$ dir. Karşıt olarak $x = a - b \in H$ ise $(a,b) \in \sim_H$ olup, $x \in M_{\sim_H}$ olur. O halde $M_{\sim_H} = H$ dir. Fakat σ ile $\sim_{M_\sigma}$ her zaman birbirine eşit değildir.

Lemma 1.20 σ, $\mathbb{N}^n$ üzerinde bir kongruans olsun. O halde;

1. $\sigma \subseteq \sim_{M_\sigma}$ dır,

2. Her $(a,b) \in \sim_{M_\sigma}$ için $(a+c,b+c) \in \sigma$ olacak şekilde $c \in \mathbb{N}^n$ vardır.

İspat: $(a,b) \in \sigma$ ise M_σ nın tanımı ndan $a-b \in M_\sigma$ olur. Dolayısıyla $(a,b) \in \sim_{M_\sigma}$ olur. şimdi $(a,b) \in \sim_{M_\sigma}$ alalım. $a-b \in M_\sigma$ olup M_σ nın tanımından bir $(x,y) \in \sigma$ için $x-y=a-b$ dir. σ bir kongruans olduğundan $(x+a,y+a) \in \sigma$ olur. $x+b=y+a$ olup $(x+a,x+b) \in \sigma$ elde edilir. $x=c$ alı rsak istenen elde edilmiş olur. ■

Bu sonuç sadeleşmeli olmayla σ nın $\sim_{M_\sigma}$ ya eşit olması arasındaki ilişkiyi ortaya koymaktadır.

Önerme 1.21 σ, $\mathbb{N}^n$ üzerinde bir kongruans olsun. O zaman

$$\mathbb{N}^n/\sigma \text{ sadeleşmelidir} \iff \sigma = \sim_{M_\sigma}$$

İspat: $\underline{\Longrightarrow:}$ $\mathbb{N}^n/\sigma$ sadeleşmeli olsun. $\sigma \subseteq \sim_{M_\sigma}$ olduğunu zaten biliyoruz. $(a,b) \in \sim_{M_\sigma}$ alalım. Bir önceki lemmadan bir $c \in \mathbb{N}^n$ için $(a+c,b+c) \in \sigma$ dir. Böylece $[a+c] = [a]+[c]$ ile $[b+c] = [b]+[c]$ nin $\mathbb{N}^n/\sigma$ da birbirine eşit olduğu g örülür. Sadeleşme özelliğinden $[a]=[b]$ yani $(a,b) \in \sigma$ elde edilir. Dolayısıyla $\sigma = \sim_{M_\sigma}$ dı r.

$\underline{\Longleftarrow:}$ $\sigma = \sim_{M_\sigma}$ olsun. Eğer $[a]+[c]=[b]+[c]$ ise $(a+c,b+c) \in \sigma$ dır. Buradan $(a+c)-(b+c) \in M_\sigma$ dolayısıyla $a-b \in M_\sigma$ elde edilir. Böylece $(a,b) \in \sim_{M_\sigma} = \sigma$ olur. O halde $[a]=[b]$ dir. Bö ylece $\mathbb{N}^n/\sigma$ nin sadeleşmeli olduğu görülür. ■

Tanım 1.22 $\mathbb{Z}^n$ nin bir H alt grubu verilsin. $\mathbb{Z}^n$ ü zerinde

$$a \equiv_H b \iff a-b \in H$$

olarak tanımlayalım. $\equiv_H$ bağıntısı daha açık olarak

$$\equiv_H = \{(a,b) \in \mathbb{Z}^n \times \mathbb{Z}^n \mid a-b \in H\}$$

dir.

$\sim_H$ da olduğu gibi $\equiv_H$ nin de bir kongruans olduğu kolayca gösterilebilir. Böylece $\mathbb{Z}^n/\equiv_H$ bir monoid olur. Ayrıca $[a-a]_{\equiv_H} = [a]_{\equiv_H} + [-a]_{\equiv_H} = [0]_{\equiv_H}$ olduğundan $\mathbb{Z}^n/\equiv_H$ bir grup olur. Genellikle bu grup $\mathbb{Z}^n/H$ ile gösterilir ve $\mathbb{Z}^n$ nin H üzerindeki *bölüm grubu*

olarak adlandırılır. Bu gruba bir örnek olarak $n = 1$ ve $1 < d \in \mathbb{N}$ için $H = (d) = \mathbf{G}(\{d\})$ olsun. $(a,b) \in \mathbb{Z} \times \mathbb{Z}$ için $a \equiv_H b$ olması için gerek ve yeter koşul $a - b \in H = (d)$ veya $d \mid a - b$ olmasıdır. O halde $\equiv_H$ bağıntısı nın e şdeğerlik sınıfları, d modülüne göre kalan sınıflarıdır. $\mathbb{Z}/\equiv_H$ de toplamanın tanımı gö z önüne alınırsa $\mathbb{Z}/\equiv_H = \mathbb{Z}_d$ olduğu görülür.

Aşağıdaki sonuç $\sim_M$ ile $\equiv_H$ arasındaki bağıntıyı vermektedir.

Önerme 1.23 H, $\mathbb{Z}^n$ nin bir alt grubu olsun. $i\left([a]_{\sim_H}\right) = [a]_{\equiv_H}$ olarak tanımlanan

$$i : \mathbb{N}^n/\sim_H \to \mathbb{Z}^n/\equiv_H$$

dönüşümü bir monoid monomorfizmidir.

İspat: $[a]_{\sim_H} = [b]_{\sim_H}$ olsun. O zaman $a - b \in H$ olup $[a]_{\equiv_H} = [b]_{\equiv_H}$dir. O halde i, iyi tanı mlıdır.

i nin monoid morfizmi olduğu açıktır. O halde i nin birebir olduğunu gösterelim. $a,b \in \mathbb{N}^n$ olmak üzere $[a]_{\equiv_H} = [b]_{\equiv_H}$ olsun. O halde $a - b \in H$ ve buradan $[a]_{\sim_H} = [b]_{\sim_H}$ elde edilir. Böylece i nin bir monomorfizm olduğu görülür. ■

Sonuç 1.24 S sonlu doğuraylı bir monoid olsun. O zaman, S nin sadeleşmeli olması için gerek ve yeter koşul S nin, bir grubun alt monoidine izomorfik olmasıdır.

İspat: Teorem 1.17 den $n \in \mathbb{N}$ ve σ, $\mathbb{N}^n$ ü zerinde bir kongruans olmak üzere, S, $\mathbb{N}^n/\sigma$ ya izomorfiktir. önerme 1.21 den de $\sigma = \sim_{M_\sigma}$ elde edilir. O zaman S, $\mathbb{N}^n/\sim_{M_\sigma}$ ya izomorfiktir. önerme 1.23 ise $\mathbb{N}^n/\sim_{M_\sigma}$ nın $\mathbb{Z}^n/\equiv_{M_\sigma} = \mathbb{Z}^n/M_\sigma$ nın bir alt monoidine izomorfik olduğunu söyler ki bu da ispatı bitirir. ■

Tanım 1.25 $x \in \mathbb{Z}^n$ nin her koordinatı pozitif ise x 'e *kuvvetli pozitif* denir. Benzer şekilde $x \in \mathbb{Z}^n$ nin her koordinatı sıfır yada sıfırdan büyükse x 'e *negatif olmayan* denir.

Önerme 1.26 S, sonlu doğuraylı bir monoid olsun. O halde aşağ ıdakiler birbirine denktir.

1. S bir gruptur.
2. $n \in \mathbb{N}$ ve $\mathbb{Z}^n$ nin kuvvetli pozitif eleman i çeren bir H alt grubu için $S \simeq \mathbb{N}^n/\sim_H$ dir.
3. $n \in \mathbb{N}$ ve $\mathbb{Z}^n$ nin bir H alt grubu için $S \simeq \mathbb{Z}^n/H = \mathbb{Z}^n/\equiv_H$ dir.

İspat: $\underline{1 \Longrightarrow 2:}$ S bir grup ise S sadeleşmeli bir monoiddir. Sonuç 1.24 nın ispatı ndan $n \in \mathbb{N}$ ve σ, $\mathbb{N}^n$ üzerinde bir kongruans olmak üzere $S \simeq \mathbb{N}^n / \sim_{M_\sigma}$ dır. S grup olduğ undan $\mathbb{N}^n / \sim_{M_\sigma}$ da gruptur. O zaman $[(1,\ldots,1)]_{\sim_{M_\sigma}}$ nın tersi vardır. $[(a_1,\ldots,a_n)]_{\sim_{M_\sigma}}$ tersi olsun. O halde

$$[(a_1+1,\ldots,a_n+1)]_{\sim_{M_\sigma}} = [(1,\ldots,1)]_{\sim_{M_\sigma}} + [(a_1,\ldots,a_n)]_{\sim_{M_\sigma}} = [(0,\ldots,0)]_{\sim_{M_\sigma}}$$

olur. Sonuç olarak $(a_1+1,\ldots,a_n+1) \sim_{M_\sigma} (0,\ldots,0)$ dır. Tanı mdan dolayı

$$(a_1+1,\ldots,a_n+1) - (0,\ldots,0) \in M_\sigma$$

olur. Dolayısıyla $(a_1+1,\ldots,a_n+1) \in M_\sigma$ olur ki zaten bu bir kuvvetli pozitif elemandır. O halde istenen elde edilmiş olur. Çünkü M_σ, $\mathbb{Z}^n$ nin bir alt grubudur.

$\underline{2 \Longrightarrow 3:}$ önerme 1.23 deki i dönüşümünün örten olduğunu göstermemiz yeterlidir. $a \in H$ bir kuvvetli pozitif eleman olsun. $[x]_{\equiv_H} \in \mathbb{Z}^n/H$ alalı m. a nın bütün koordinatları pozitif olduğu için $ka+x \in \mathbb{N}^n$ olacak şekilde $k \in \mathbb{N}$ vardır. $ka \in H$ olduğundan

$$[ka+x]_{\equiv_H} = [x]_{\equiv_H}$$

elde edilir. $i\left([ka+x]_{\sim_H}\right) = [ka+x]_{\equiv_H} = [x]_{\equiv_H}$ olduğundan i örtendir.

$\underline{3 \Longrightarrow 1:}$ $\mathbb{Z}^n/H$ bir grup olduğu için S de bir gruptur. ■

2. SONLU DOĞURAYLI DEĞİŞMELİ GRUPLAR

[Bu deneme]

Burada Grup teorinin klasik sonuçlarından biri olan sonlu doğ uraylı değişmeli grupların temel teoremi ile ilgileneceğiz. Önce bu teoremi ifade edelim.

Teorem 2.1 Sonlu doğuraylı her değişmeli grup, $d_1, \ldots, d_r, k$ her $1 \leq i \leq r-1$ için d_i, d_{i+1} i bölecek şekilde bir takım pozitif tamsayılar olmak üzere $\mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^k$ grubuna izomorftur.

Burada da $\mathbb{Z}_n$ ile n modülüne göre kalan sınıflarının toplama altındaki grubu gösterilmektedir.

Temel teoremin kanıtında kullanılan fikirler şunlardır:

1. H, $\mathbb{Z}^n$ nin bir alt grubu olmak üzere her sonlu do ğuraylı grup $\mathbb{Z}^n/H$ formundadır.

2. $\{f_1, \ldots, f_n\}$, $\mathbb{Z}^n$ nin bir bazı olsun. Her $i \in \{1, \ldots, r\}$ için d_i pozitif bir tamsayı olmak üzere $H = \mathbf{G}(\{d_1 f_1, \ldots, d_r f_r\})$ olsun. O zaman $\mathbb{Z}^n/H \simeq \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^{n-r}$ dir.

3. $\{f_1, \ldots, f_n\}, \mathbb{Z}^n$ nin bir bazı olsun. O zaman $\mathbb{Z}^n$ nin her alt grubunun $\{d_1 f_1, \ldots, d_r f_r\}$ formunda bir bazı vardır.

2.1 $\mathbb{Z}^n$ nin Alt Gruplarının Bazları ve Rankı

Tanım 2.2 M, $\mathbb{Z}^n$ nin bir alt grubu olsun. Eğer M deki her eleman, $z_1, \ldots, z_r \in \mathbb{Z}$ olmak üzere $m = \sum_{i=1}^{r} z_i m_i$ şeklinde tek türlü yazılabiliyorsa $\{m_1, \ldots, m_r\} \subset M$ kümesine M nin bir *bazı* denir. $(z_1, \ldots, z_r) \in \mathbb{Z}^r$ ye de m nin $\{m_1, \ldots, m_r\}$ sıralı bazına göre koordinatları denir.

Sonuç 2.3 $\{m_1, \ldots, m_r\}$ nin M nin bir bazı olması iç in gerek ve yeter koşul

1. Her $m \in M$ için $z_1, \ldots, z_r \in \mathbb{Z}$ olmak ü zere $m = \sum_{i=1}^{r} z_i m_i$ şeklinde yazılabilir,

2. Eğer $z_1, \ldots, z_r \in \mathbb{Z}$ olmak üzere $\sum_{i=1}^{r} z_i m_i = 0$ ise her $i \in \{1, \ldots, r\}$ için $z_i = 0$

olmasıdır.

Birinci koşul $\{m_1,\ldots,m_r\}$ kümesinin M i çin bir doğuray kümesi olduğunu yani $M=\mathbf{G}(\{m_1,\ldots,m_r\})$ olduğunu, ikinci ko şul ise bir bakımdan lineer bağımsızlığı ifade etmektedir. Bu son ifadeye kesinlik kazandırmak için

$$m_1,\ldots,m_r\in\mathbb{Z}^n \text{ lineer bağımsızdır} \iff m_1,\ldots,m_r \text{ , } \mathbb{Q}^n \text{ de lineer bağımsızdır}$$

olduğuna dikkat edelim.

Gösterim 2.4 e_i, $\mathbb{N}^n$ nin $i-$inci koordinatı 1, diğer büt ün koordinatları 0 olan elemanıdır.

$\{e_1,\ldots,e_n\}$ nin $\mathbb{Z}^n$ nin bir bazı olduğuna dikkat edelim.

Önerme 2.5 $M,\mathbb{Z}$ nin bir alt grubu ise $M=\mathbf{G}(\{z\})$ olacak şekilde bir $z\in M$ vardır.

İspat: $M=\{0\}$ ise $M=\mathbf{G}(\{0\})$ dır.

$M\neq\{0\}$ varsayalım.

M bir grup olduğundan M de pozitif elemanlar vardır. O halde $\{h\in M\mid h>0\}$ kümesi boştan farklıdır. Bu y üzden bu kümenin bir minimum elemanı vardır. Buna z diyelim. Şimdi $M=\mathbf{G}(\{z\})$ olduğunu gö sterelim. $\mathbf{G}(\{z\})\subseteq M$ olduğ u aşikar. O zaman $h\in M$ alalım. $\mathbb{Z}$ deki bölme algoritması gereği; $q,r\in\mathbb{Z}$ vardır öyleki $h=qz+r$, $0\leq r<z$. Dolayısıyla $0\leq r=h-qz<z$ dir. Fakat M deki sıfırdan büyük elemanların en küçüğü z idi. O halde $r=0$ dır. Dolayısıyla $h=qz\in\mathbf{G}(\{z\})$ dir. ■

$\{0\}\subseteq\mathbb{Z}^n$ ın bir bazının boş k üme olduğuna dikkat edelim. Şimdi $\mathbb{Z}^n$ nin bir alt grubunun bir bazının eleman sayısının en fazla ne olabileceğ ini söyleyeceğiz.

Önerme 2.6 $M,\mathbb{Z}^n$ nin bir alt grubu olsun. O zaman M nin bir bazı en fazla n eleman içerir.

İspat: Her $j\in\{1,\ldots,n\}$ için $M_j=M\cap\mathbf{G}\left(\left\{e_1,\ldots,e_j\right\}\right)$ tanımlayalım.

$$M_n=M\text{ , }M_1\subseteq M_2\subseteq M_3\subseteq\ldots$$

ve M_j lerin $\mathbb{Z}^n$ nin birer alt grubu olduğuna dikkat edelim. Tümevarım kullanarak M_i nin bir bazının en fazla i eleman içerebileceğini gösterelim.

$M_1=\{0\}$ ise M_1 in bazı boş kümedir.

$M_1\neq\{0\}$ ise $H=\{k\in\mathbb{Z}\mid ke_1\in M_1\}$ kümesini düşünelim. $H,\mathbb{Z}$ nin alt grubudur ve önerme 2.5 den $z\in\mathbb{Z}\setminus\{0\}$ için $H=\mathbf{G}(\{z\})$ dir. Buradan $M_1=\mathbf{G}(\{ze_1\})$ olacağ ı açıktır. Sonuç olarak $\{ze_1\}$, M_1 i çin bir baz olur. O halde Önerme $i=1$ için doğru olur.

Varsayalım ki $r \leq k$ olmak üzere $\{m_1,\ldots,m_r\}$, M_k için bir baz olsun. Eğer $M_{k+1} = \{0\}$ ise $M_k = \{0\}$ dır. Dolayısıyla $r = 0$ ve M_{k+1} in bazı boş küme olur.

$M_{k+1} \neq \{0\}$ varsayalım ve $H = \{z \in \mathbb{Z} \,|\, y + ze_{k+1} \in M_{k+1}, y \in \mathbf{G}(\{e_1,\ldots,e_k\})\}$ tanım- layalım. Yine $H, \mathbb{Z}$ nin alt grubu olur. O zaman bir $z \in \mathbb{Z}$ için $H = \mathbf{G}(\{z\})$ olur. Eğer $z = 0$ ise $M_{k+1} = M_k$ olur. Böylece M_{k+1}, r elemanlı bir baza sahiptir ve $r \leq k < k+1$ dir.

Eğer $z \neq 0$ ise $w = y + ze_{k+1} \in M_{k+1}$ olacak şekilde $y \in \mathbf{G}(\{e_1,\ldots,e_k\})$ vardır. Şimdi $\{m_1,\ldots,m_r,w\}$ nun M_{k+1} için bir baz olduğunu gösterelim. $m \in M_{k+1}$ alalım.

i) $m \in \mathbf{G}(\{e_1,\ldots,e_{k+1}\})$ olduğundan $z_1,\ldots,z_{k+1} \in \mathbb{Z}$ için $m = \sum_{i=1}^{k+1} z_i e_i$ dir. O halde $z_{k+1} \in H$ olur. Dolayısıyla $z_{k+1} = zt$ olacak şekilde bir $t \in \mathbb{Z}$ vardır. Buradan

$$m = \sum_{i=1}^{k} z_i e_i - ty + tw$$

ve sonuç olarak

$$m - tw \in M \cap \mathbf{G}(\{e_1,\ldots,e_k\}) = M_k$$

elde edilir. $\{m_1,\ldots,m_r\}$, M_k nın bir bazı olduğundan dolayı

$$m - tw = \sum_{i=1}^{r} t_i m_i$$

olacak şekilde $t_1,\ldots,t_r \in \mathbb{Z}$ vardır. O halde $m = \sum_{i=1}^{r} t_i m_i + tw$ dır.

ii) $s_1,\ldots,s_{r+1} \in \mathbb{Z}$ olmak üzere $\sum_{i=1}^{r} s_i m_i + s_{r+1} w = 0$ olsun. $\sum_{i=1}^{r} s_i m_i \in \mathbf{G}(\{e_1,\ldots,e_k\})$ ve w, $(k+1)$-inci koordinatı sıfırdan farklı olan tek eleman olduğundan $s_{r+1} = 0$ elde edilir. Dolayısıyla $\sum_{i=1}^{r} s_i m_i = 0$ dır ve böylece $s_1 = \cdots = s_r = 0$ elde edilir (çünkü $\{m_1,\ldots,m_r\}$, M_k nın bazı). Dolayısıyla $s_1 = \cdots = s_r = s_{r+1} = 0$ dır.

$\{m_1,\ldots,m_r,w\}$ nun M_{k+1} için bir baz olduğunu gösterdik ($r+1 \leq k+1$). ■

Tanım 2.7 $\mathbb{Q}^n$ nin A tarafından gerilen alt uzayı

$$\mathbf{L}_{\mathbb{Q}}(A) = \left\{ \sum_{i=1}^{n} q_i a_i \mid n \in \mathbb{N} \text{ ve her } 1 \leq i \leq n \text{ için } q_i \in \mathbb{Q},\ a_i \in A \right\}$$

şeklinde tanımlanır.

Önerme 2.8 M, $\mathbb{Z}^n$ nin alt grubu olsun. O zaman M nin büt ün bazlarında aynı sayıda eleman bulunur.

İspat: $B = \{m_i \mid i \in I\}$ ve $B' = \{n_i \mid i \in I'\}$, M nin iki farklı bazı olsun. B nin elemanları $\mathbb{Q}^n$ de lineer bağımsız olduklarından $|B|$ en fazla n olabilir. Benzer şey $|B'|$ içinde geçerlidir. $B, V = \mathbf{L}_{\mathbb{Q}}(B)$ nin bir bazı ve

$$B' \subset M = \mathbf{G}(\{m_1, \ldots, m_r\}) \subset \mathbf{L}_{\mathbb{Q}}(B)$$

V nin lineer bağımsız vektörlerinin kümesi olduğundan $|B'| \leq |B|$ olmak zorundadır. Tersini de benzer şekilde söyleyebiliriz. ■

Tanım 2.9 $\mathbb{Z}^n$ nin bir M alt grubunun *rankı*, M nin herhangi bir bazındaki eleman sayısı olarak tanımlanır ve $rank(M)$ ile gösterilir. $rank(M) \leq n$ olduğuna dikkat edelim.

Önerme 2.10 M, $\mathbb{Z}^n$ nin rankı k olan bir alt grubu ise $M \simeq \mathbb{Z}^k$ dır.

İspat: $B = \{m_1, \ldots, m_k\}$, M nin bir bazı olsun. M nin her elemanının B sıralı bazına göre koordinatı tek oldu ğundan

$$f : \mathbb{Z}^k \to M$$
$$f(z_1, \ldots, z_k) = \sum_{i=1}^{k} z_i m_i$$

dönüşümü bir izomorfizmdir. ■

Sonuç 2.11 M ve M', $\mathbb{Z}^n$ nin iki alt grubu olsun. O zaman

$$M \simeq M' \iff rank(M) = rank\left(M'\right)$$

dir.

2.2 Tamsayı Bileşenli Matrislerin Denkliği ve İnvaryant Faktörleri

Daha önce belirttiğimiz amaçlara ulaşmak için, bileş enleri tamsayı olan matrislerle ilgili bir kaç sonucu tekrar ele almamız gerekiyor. Temel fikir; $\mathbb{Z}^n$ nin verilen bir alt grubunun bazını, elementer işlemlerle, bu konunun başındaki şekilde olan bir baza dönüştürmektir. Satırları, verilen bir alt grubun bazının elemanları olan matrislerin transformasyonundan bahsedeceğiz. Bu prosese Gauss eliminasyonu diyoruz.

Verilen bir n tamsayısı için $1 \leq i, j \leq n$, $(i \neq j)$ olmak üzere şunları tanımlayalım.

1. $R_{i \leftrightarrow j}$, I_n nin i-inci satırı ile j -inci satırının yer değiştirmesiyle elde edilen matristir.

2. $R_{i\leftarrow -i}$, I_n nin i-inci satırının -1 ile çarpılmasıyla elde edilen matristir.

3. $R_{j\leftarrow j+zi}$, I_n nin i-inci satırının $z \in \mathbb{Z}$ ile çarpılıp j-inci satırına eklenmesiyle elde edilen matristir.

$C_{i\leftrightarrow j}$, $C_{i\leftarrow -i}$, $C_{j\leftarrow j+zi}$ de yukarıdakine benzer şekilde sadece satır yerine sütun yazmakla tanımlanmış olur.

Tanım 2.12 $R_{i\leftrightarrow j}$, $R_{i\leftarrow -i}$, $R_{j\leftarrow j+zi}$ matrislerine *elementer satır matrisleri*, $C_{i\leftrightarrow j}$, $C_{i\leftarrow -i}$, $C_{j\leftarrow j+zi}$ matrislerine de *elementer s ütun matrisleri* denir.

Önerme 2.13 i, j iki pozitif tamsayı, $z \in \mathbb{Z}$, A bileş enleri tamsayı olan $n \times n$ tipinde bir matris ve $\det(B)$ de B matrisinin determinantını göstersin. O zaman

1. $\det\left(R_{i\leftrightarrow j}\right) = \det\left(C_{i\leftrightarrow j}\right) = -1$,

2. $\left(R_{i\leftrightarrow j}\right)^{-1} = R_{j\leftrightarrow i}$ ve $\left(C_{i\leftrightarrow j}\right)^{-1} = C_{j\leftrightarrow i}$,

3. $\det\left(R_{i\leftarrow -i}\right) = \det\left(C_{i\leftarrow -i}\right) = -1$,

4. $\left(R_{i\leftarrow -i}\right)^{-1} = R_{i\leftarrow -i}$ ve $\left(C_{i\leftarrow -i}\right)^{-1} = C_{i\leftarrow -i}$,

5. $\det\left(R_{j\leftarrow j+zi}\right) = \det\left(C_{j\leftarrow j+zi}\right) = 1$,

6. $\left(R_{j\leftarrow j+zi}\right)^{-1} = R_{j\leftarrow j-zi}$ ve $\left(C_{j\leftarrow j+zi}\right)^{-1} = C_{j\leftarrow j-zi}$,

7. $R_{i\leftrightarrow j}A$ matrisi, A matrisinin i-inci satırı ile j-inci satırının yer değiştirilmesi sonucu elde edilen matristir. $AC_{i\leftrightarrow j}$ matrisi de, A matrisinin i-inci s ütunu ile j-inci sütununun yer değiştirilmesi sonucu elde edilen matristir.

8. $R_{i\leftarrow -i}A$ matrisi, A matrisinin i-inci satırının -1 ile çarpılması ile elde edilen matristir. $AC_{i\leftarrow -i}$ matrisi de, A matrisinin i-inci sütununun -1 ile çarpılması ile elde edilen matristir.

9. $R_{j\leftarrow j+zi}A$ matrisi, A matrisinin i-inci satırının z ile çarpımının j-inci satırına eklenmesiyle elde edilen matristir. $AC_{j\leftarrow j+zi}$ matrisi de A matrisinin i-inci sütununun z ile çarpımının j-inci sütununa eklenmesiyle elde edilen matristir.

Tanım 2.14 Bileşenleri tamsayı olan A ve B gibi iki matris verildiğinde

$$B = P_1 \dots P_r A Q_1 \dots Q_s$$

olacak şekilde $P_1, \dots, P_r$ elementer satır matrisleri ve $Q_1, \dots, Q_s$ elementer sütun matrisleri varsa A ve B ye *denktirler* denir.

Önerme 2.13 kullanılarak

$$A \sim B \iff A \text{ ve } B \text{ denktirler}$$

bağıntısının bir denklik bağıntısı olduğu kolayca g österilebilir.

Not 2.15 R_* tipindeki matrisler çarpılırken verilen matrisin satırları üzerinde elementer işlemler yapılır. C_* tipindeki matrisler çarpılırken de verilen matrisin sütunları üzerinde elementer işlemler yapılır.

Tamsayılar ve rasyonel sayılar veya herhangi bir cisim üzerinde ç alışırkenki temel fark şudur; q, cisimin sıfırdan farklı bir elemanı olmak üzere $R_{i \leftarrow qi}$ ve $C_{i \leftarrow qi}$ de elementer matrisler olur.

Gauss-Jordan eliminasyon metodunu bileşenleri cisimden gelen matrisler üzerinde şöyle yaparız.

İlk adım olarak matrisin sol üst köşesinin sıfırdan farklı olacak şekilde işlem yaparız. Bunu $R_{i \leftrightarrow j}$ ve $C_{i \leftrightarrow j}$ elementer matrislerinin kullanarak yaparız. Yani matrisimizin sol üst köşesi sıfırdan farklı olacak şekilde satır ve sütun değişimleri yaparız. Sonra birinci satırı veya birinci sütunu bu sol üst köşedeki sayıya bölerek sol üst köşeyi 1 yaparız. Bu kez sadece $R_{j \leftarrow j+z1}$ ve $C_{j \leftarrow j+z1}$ elementer matrislerini kullanarak orjinal matrisimize denk olan sol üst köşesi hariç birinci satırdaki ve sütundaki diğer bütün bileşenleri sıfır olan bir matris elde ederiz. Birinci satırı ve birinci sü tunu kapatıp bu prosedürü yeni matrise de uygularız. Bu prosese bu şekilde devam ederiz.

Fakat tamsayılarda bu prosedür bu kadar kolay işlemiyor. Çü nkü genellikle bir satırı veya sütunu 1 veya -1 den farklı bir tamsayıya bölemiyor ve dolayısıyla denk matrisi elde edemiyoruz.

Bu prosedürün tamsayılardaki halini aşağıdaki ş ekilde tanımlayabiliriz.

Önerme 2.16 A, bileşenleri tamsayı olan $s \times t$ tipinde bir matris olsun. $r \leq \min\{s,t\}$, $\{d_1,\ldots,d_r\} \subset \mathbb{N} \setminus \{0\}$ ve her $i \in \{1,\ldots,r-1\}$ için $d_i \mid d_{i+1}$ olmak üzere A matrisi

$$\begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

matrisine denktir.

İspat: A bir sıfır matrisi ise $r = 0$ olduğuna dikkat edelim. (i,j); i-inci satır, j-inci sütunda bulunan elemanı gö stersin. A matrisine elementer işlemler uygulanması sonucu $(1,1)$ deki elemanı mutlak değerce en küçük olacak şekilde ayarlayabiliriz. Elde ettiğimiz B matrisi A ya denk bir matristir. Şimdi birinci satırda bu eleman tarafından bölü nemeyen bir eleman var mı ona bakalım. Eğer varsa genelliği kaybetmeksizin bu eleman $(1,2)$ deki eleman olsun. Euclid algoritmasından dolayı $\gcd\{b_{11},b_{12}\} = d$ (b_{11} ile b_{12} nin en büyük ortak böleni) bulabiliriz. d hesaplanırkenki işlemleri elementer operasyonlara ç evirebilir ve matrisimizi önce $C_{1\leftarrow 1+z2}$ ve sonra $C_{1\leftrightarrow 2}$ ile çarparak yine orjinal matrisimize denk olan bir matris elde ederiz ve elde ettiğimiz bu matriste $(1,1)$ deki eleman $(1,2)$ deki elemanı böler. Bu prosedür ü devam ettirdiğimizde her i için $c_{11} \mid c_{1i}$ olacak şekilde A ya denk bir C matrisi elde ederiz. Birinci sütunu uygun sayılarla çarpıp diğer sütunlara eklediğimizde $(1,1)$ deki bileşen hariç birinci satırdaki tüm bileşenleri sıfırlamış oluruz. Benzer prosedürü birinci sütun için de uygulayabiliriz. Fakat bu sırada birinci satır bozulabilir (sıfırların bir kısmı kaybolabilir). Bu durumda prosedür birinci satır için tekrarlanmalıdır. Benzer şey birinci sütun içinde geçerlidir. Sonlu adım sonunda

$$\begin{pmatrix} f_{11} & 0 & \cdots & 0 \\ 0 & f_{22} & \cdots & f_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & f_{s2} & \cdots & f_{st} \end{pmatrix}$$

matrisini elde ederiz.

$f_{11} \nmid f_{ij}$ olacak şekilde i, j varsa i-inci satırı birinci satıra ekleyip biraz önce uyguladığımız prosedür ü tekrarlarsak her i, j için $f_{11} \mid f_{ij}$ elde ederiz. Sonlu adım sonunda $(1,1)$ deki eleman A nın bütün bile şenlerinin en büyük ortak böleni olur. Şimdi $d_1 = \gcd\left\{g_{ij} \mid \forall i, j \text{ için}\right\}$ olmak üzere orjinal matrisimize denk

$$\begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & g_{22} & \cdots & g_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & g_{s2} & \cdots & g_{st} \end{pmatrix}$$

matrisini elde etmiş olduk. Benzer yolla

$$\begin{pmatrix} g_{22} & \cdots & g_{2t} \\ \vdots & \ddots & \vdots \\ g_{s2} & \cdots & g_{st} \end{pmatrix}$$

alt matrisini de bu forma getirebiliriz.

Sonlu adım sonunda istediğimiz sonuca ulaşmış oluruz. ■

Tanım 2.17 $d_1, \ldots, d_r$ elemanlarına A nın *invaryant faktörleri* denir.

Bunların tek (unique) olduğunu kanıtlayacağız.

Tanım 2.18 Bir A matrisinin $k \times k$ tipindeki bir alt matrisinin determinantına A nın bir *k-minörü* denir. A nın bütü n k-minörlerinin en büyük ortak bölenini $D_k(A)$ ile göstereceğiz.

Önerme 2.19 A bileşenleri tamsayı olan bir matris olsun. O zaman

1. $D_k\left(R_{i\leftrightarrow j}A\right) = D_k(A) = D_k\left(AC_{i\leftrightarrow j}\right)$,
2. $D_k\left(R_{i\leftarrow -i}A\right) = D_k(A) = D_k\left(AC_{i\leftarrow -i}\right)$,
3. $D_k\left(R_{j\leftarrow j+zi}A\right) = D_k(A) = D_k\left(AC_{j\leftarrow j+zi}\right)$ dir.

Bunun bir sonucu olarak şunu elde ederiz: "A ve B matrisleri denk ise her k için $D_k(A) = D_k(B)$ dir."

Önerme 2.20 A ve B bileşenleri tamsayı olan iki matris olsun. O zaman A ile B nin denk olması için gerek ve yeter koşul A ile B nin aynı invaryant faktörlere sahip olmasıdır.

İspat: $\sim$ bağıntısı bir denklik bağıntısı olduğ undan önerme 2.16 kullanılarak

$$\delta_1 = \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}, \delta_2 = \begin{pmatrix} d'_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d'_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d'_s & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

ve $\{d_1,\ldots,d_r,d'_1,\ldots,d'_s\} \subset \mathbb{N}\setminus\{0\}$, her i için $d_i \mid d_{i+1}$, $d'_i \mid d'_{i+1}$ olmak üzere

$$\delta_1 \text{ ve } \delta_2 \text{ denktir} \iff r = s \text{ ve her } i \text{ için } d_i = d'_i$$

olduğunu göstermek yeterlidir.

Eğer $r = s$ ve her i için $d_i = d'_i$ ise δ_1 ve δ_2 denktir. Tersine eğer δ_1 ve δ_2 denk iseler

$$d_1 \cdots d_r = D_k(\delta_1) = D_k(\delta_2) = d'_1 \cdots d'_s$$

den dolayı sonuç açıktır. ■

Şimdi bütün bu sonuçları $\mathbb{Z}^n$ nin alt gruplarına uygulayabiliriz.

Önerme 2.21 $\{m_1,\ldots,m_i,\ldots,m_j,\ldots,m_r\}$, $\mathbb{Z}^n$ nin bir M alt grubunun bir bazı olsun. O zaman

1. $\{m_1,\ldots,m_j,\ldots,m_i,\ldots,m_r\}$ de M nin bir bazıdır.
2. $\{m_1,\ldots,-m_i,\ldots,m_j,\ldots,m_r\}$ de M nin bir bazıdır.
3. $\{m_1,\ldots,m_i,\ldots,m_j+zm_i,\ldots,m_r\}$ de M nin bir bazıdır.

Bu işlemler, bazlar için elementer işlemler olarak bilinir.

Önerme 2.22 Bileşenleri tamsayı olan

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$$

matrisi verilsin. $b_i = (b_{i1},\ldots,b_{in}) \in \mathbb{Z}^n$ alalım. O zaman $\{b_1,\ldots,b_n\}$ nin $\mathbb{Z}^n$ nin bir bazı olması için gerek ve yeter koşul $\det(B) \in \{-1,1\}$ olmasıdır.

İspat: $\underline{\Longrightarrow:}$ Her $i \in \{1,\dots,n\}$ için $\sum_{j=1}^{n} z_{ij} b_j = e_i$ olacak şekilde $z_{i1},\dots,z_{in} \in \mathbb{Z}$ vardır.

$$\begin{pmatrix} z_{11} & \cdots & z_{1n} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nn} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

dir. Böylece $\det\left(\left(z_{ij}\right)\right)\det(B) = 1$, dolayısıyla $\det(B) \in \{-1,1\}$ dir.

$\underline{\Longleftarrow:}$ $\det(B) \in \{-1,1\}$ olsun. O halde B tersinirdir ve buradan

$$(x_1,\dots,x_n)B = z$$

denklem sistemi tek bir çözüme sahiptir ve o da (her $z \in \mathbb{Z}^n$ için) zB^{-1} dir. Yani her eleman $\{b_1,\dots,b_n\}$ sırasında tek bir koordinata sahiptir. Dolayısıyla $\{b_1,\dots,b_n\}$, $\mathbb{Z}^n$ nin bazıdır. ■

Teorem 2.23 M, $rank(M) = r$ olacak şekilde $\mathbb{Z}^n$ nin bir alt grubu olsun. O zaman her i için $d_i \mid d_{i+1}$ ve $\{d_1 f_1,\dots,d_r f_r\}$, M için bir baz olacak şekilde $\mathbb{Z}^n$ nin bir $\{f_1,\dots,f_r,\dots,f_n\}$ bazı ve $\{d_1,\dots,d_r\} \subset \mathbb{N}\setminus\{0\}$ kümesi vardır.

İspat: Her i için $m_i = (m_{i1},\dots,m_{in})$ olmak ü zere $\{m_1,\dots,m_r\}$, M için bir baz olsun. Ö nerme 2.16 den $0 \le s \le \min\{r,n\}$ ve her i iç in $d_i \mid d_{i+1}$ ve $\{d_1,\dots,d_r\} \subset \mathbb{N}\setminus\{0\}$ olmak üzere

$$P\begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{r1} & \cdots & m_{rn} \end{pmatrix} Q = \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_s & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

olacak şekilde P ve Q matrisleri vardır. P ve Q tersinir olduklarından

$$\begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{r1} & \cdots & m_{rn} \end{pmatrix} \text{ ile } \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_s & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

nin rankı aynı olur (önerme 2.21 den).

Dolayısıyla $s = r$ dir. Ek olarak

$$\begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{r1} & \cdots & c_{rn} \end{pmatrix} = P \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{r1} & \cdots & m_{rn} \end{pmatrix}$$

alalım ve $c_i = (c_{i1}, \ldots, c_{in})$ koyalım. $\{c_1, \ldots, c_r\}$ kümesi, $\{m_1, \ldots, m_r\}$ bazından sonlu elementer işlemler uygulanarak elde edilmiştir. O halde $\{c_1, \ldots, c_r\}$ de M iç in bir bazdır.

Son olarak Q^{-1} i hesaplar ve

$$Q^{-1} = \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nn} \end{pmatrix}$$

şeklinde gösterirsek $f_i = (f_{i1}, \ldots, f_{in})$ olmak üzere $\{f_1, \ldots, f_n\}, \mathbb{Z}^n$ i çin bir baz olur (önerme 2.22 den), ve

$$\begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{r1} & \cdots & c_{rn} \end{pmatrix} = \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_r & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nn} \end{pmatrix}$$

bulunur. Dolayısıyla her $i \in \{1, \ldots, r\}$ için $c_i = d_i f_i$ dir. ■

$\{d_1, \ldots, d_r\}$ invaryant faktörlerin kümesi beklendiği gibi M nin bir alt grubudur.

$\{f_1, \ldots, f_r, \ldots, f_n\}, \mathbb{Z}^n$ nin bir bazı ve $\{d_1 f_1, \ldots, d_r f_r\}$ de M nin bir bazı olduğundan $x = (x_1, \ldots, x_n) \in \mathbb{Z}^n$ nin M nin elemanı olması için gerek ve yeter koşul x in $\{f_1, \ldots, f_n\}$

sıralı bazına göre koordinatları olan $(z_1,\ldots,z_n)$ nin

$$\begin{aligned} z_1 &\equiv 0 \pmod{d_1} \\ &\vdots \\ z_r &\equiv 0 \pmod{d_r} \\ z_{r+1} &= 0 \\ &\vdots \\ z_n &= 0 \end{aligned}$$

denklemlerini sağlamasıdır.

Teorem 2.23 nin ispatında olduğu gibi her i için $f_i = (f_{i1},\ldots,f_{in})$ alalım. $(z_1,\ldots,z_n)$, x in $\{f_1,\ldots,f_n\}$ sıralı bazına göre koordinatları olduğundan

$$x = (x_1,\ldots,x_n) = \sum_{i=1}^{n} z_i f_i$$

ve buradan

$$(x_1,\ldots,x_n) = (z_1,\ldots,z_n) \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nn} \end{pmatrix}$$

yazabiliriz. Dolayısıyla

$$(z_1,\ldots,z_n) = (x_1,\ldots,x_n) \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$$

dir (buradaki $\left(b_{ij}\right)$ matrisi Teorem 2.23 nin ispatındaki $\left(f_{ij}\right)$ matrisinin tersi olan matris yani Q matrisidir).

Buradan,

$$x = (x_1,\ldots,x_n) \in \mathbb{Z}^n, M \text{ dedir} \iff \begin{aligned} & b_{11}x_1 + \cdots + b_{n1}x_n \equiv 0 \pmod{d_1} \\ & \vdots \\ & b_{1r}x_1 + \cdots + b_{nr}x_n \equiv 0 \pmod{d_r} \\ & b_{1(r+1)}x_1 + \cdots + b_{n(r+1)}x_n = 0 \\ & \vdots \\ & b_{1n}x_1 + \cdots + b_{nn}x_n = 0 \end{aligned}$$

elde edilir. Bu denklemler genellikle $\{e_1,\ldots,e_n\}$ *sıralı bazına göre M nin denklemleri* yada daha basitçe *M nin denklemleri* olarak bilinir.

Eğer $d_i = 1$ ise bu d_i nin geçtiği denklem elimine edilebilir (çünkü her tamsayının 1 ile bölümü nden kalan 0 dır).

M nin tüm invaryant faktörleri 1 ise M ye *homojendir* denir.

Örnek 2.24 $M = \mathbf{G}(\{(2,-1,1),(3,2,-1)\}) \subseteq \mathbb{Z}^3$ alalım. Sü tunlarında yaptığımız elementer işlemleri sağdaki matrise, satırlarında yaptığımız elementer iş lemleri soldaki matrise uyguluyoruz.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 2 & -1 & 1 \\ 3 & 2 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 & 2 \\ -1 & 2 & 3 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 & 2 \\ 0 & 1 & 5 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 5 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 5 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & -2 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -5 \\ 1 & 1 & -7 \end{pmatrix}$$

elde edilir. Buradan

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 1 \\ 3 & 2 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -5 \\ 1 & 1 & -7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

elde edilir. Bu nedenle M nin denklemleri

$$\begin{aligned} x_3 &\equiv 0 \pmod 1 \\ x_2 + x_3 &\equiv 0 \pmod 1 \\ x_1 - 5x_2 - 7x_3 &= 0 \end{aligned}$$

dir. İlk iki denklemi elimine edebileceğimizden

$$(x_1, x_2, x_3) \in M \iff x_1 - 5x_2 - 7x_3 = 0$$

elde ederiz. $\Diamond$

Aşağıdaki teorem, sonlu doğuraylı değişmeli grupların temel teoremini göstermek için gerekli son kısmı verir.

Teorem 2.25 $M, \mathbb{Z}^n$ nin invaryant faktörleri $d_1, \ldots, d_r$ olan bir alt grubu ise

$$\mathbb{Z}^n / M \simeq \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^{n-r}$$

dir.

İspat: Teorem 2.23 den, $\{f_1, \ldots, f_r, \ldots, f_n\}$, $\mathbb{Z}^n$ nin bazı olacak şekilde, M nin $\{d_1 f_1, \ldots, d_r f_r\}$ formunda bir bazı olduğunu biliyoruz. Notasyonu basitleştirmek için $\mathbb{Z}_{d_i} = \mathbb{Z}/\mathbf{G}(\{d_i\})$ deki $[z_i]_{\equiv_{\mathbf{G}(\{d_i\})}}$ elemanını $[z_i]_{d_i}$ ile g österip

$$\varphi : \mathbb{Z}^n / M \to \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_r} \times \mathbb{Z}^{n-r}$$
$$\varphi\left(\left[\sum_{i=1}^{n} z_i f_i\right]_{\equiv_M}\right) = \left([z_1]_{d_1}, \ldots, [z_r]_{d_r}, z_{r+1}, \ldots, z_n\right)$$

dönüşümünü tanımlayalım.

φ iyi tanımlıdır ve φ bir izomorfizmdir. ■

2.3 Bazların Hesaplanışı İle İlgili Bazı Kullanışlı Sonuçlar

2.3.1 $\mathbb{Z}^n$ nin Bir Alt Grubunun Bir Doğuray Kü mesinden Bir Bazının Hesaplanışı

Eğer Q, bileşenleri tamsayı ve determinantı sıfırdan farklı olan bir matris ve f

$$\begin{aligned} f &: \mathbb{Z}^n \to \mathbb{Z}^n \\ f(x_1,\ldots,x_n) &= (x_1,\ldots,x_n)\,Q \end{aligned}$$

şeklinde tanımlanan bir fonksiyon ise $m_1,\ldots,m_r$ nin lineer bağımsız olması için gerek ve yeter koşul $f(m_1),\ldots,f(m_r)$ nin lineer bağımsız olmasıdır.

$$\begin{aligned} g &: \mathbb{Q}^n \to \mathbb{Q}^n \\ g(x) &= xQ \end{aligned}$$

dönüşümü bir izomorfizmdir. Dolayısıyla g , lineer bağımsızlığı korur.

Her $i \in \{1,\ldots,s\}$ için $m_i = (m_{i1},\ldots,m_{in})$ olmak üzere $\mathbb{Z}^n$ nin $\{m_1,\ldots,m_s\}$ tarafından doğurulan M alt grubunu alalım. A, satırları $m_1,\ldots,m_s$ olan bir matris olsun. $d_1,\ldots,d_r \in \mathbb{N}\setminus\{0\}$ olmak üzere A matrisinin

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

diagonal matrisine denk olduğunu biliyoruz. Dolayısıyla $PAQ = D$ olacak şekilde P ve Q matrisleri vardır. Üstelik P nin satır elementer matrislerin bir çarpımı ve Q nun da sütun elementer matrislerin bir çarpımı olduğunu biliyoruz. Bö ylece $PA = DQ^{-1}$ olduğuna ve buradan aşağıdakilerin sağ landığına dikkat edelim.

1. Q sadece D nin sütunları üzerinde elementer işlemler yaptığından ve $PA = DQ^{-1}$ olduğundan ve de D nin son $s-r$ satırındaki bütün bileşenler sıfır olduğundan PA nın son $s-r$ satırındaki bütün bileşenler sıfırdır.

2. P matrisi sadece A nın satırları üzerinde elementer i şlemler yaptığından PA nın satırları M nin bir doğ uray kümesi olur. Buradan PA nın ilk r satırının M nin bir doğuray kümesi olduğunu söylemiş oluruz.

3. D nin ilk r satırı lineer bağımsız ve $\det(Q) \neq 0$ olduğundan PA nın ilk r satırı lineer bağ ımsız olur.

Buradan, PA yı hesaplayıp onun ilk r satırını alarak, M nin bir bazını elde etmiş oluruz.

2.3.2 $\mathbb{Z}^n$ nin Bir Alt Grubunun Denklemlerinden Bir Bazının Hesaplanışı

Önce homojen olma durumunu (yani bütün invaryant faktörlerin 1 olduğu durumu) inceleyeceğiz. Sonra genel duruma bakacağız (aslında genel durumun homojen olma durumuna indirgenebileceğini gö receğiz).

$M, \mathbb{Z}^n$ nin homojen bir alt grubu ve A, bileş enleri tamsayı olan bir matris olsun. Varsayalımki

$$x \in M \iff Ax = 0$$

olsun. P, Q sırasıyla elementer satır matrislerinin ve elementer s ütun matrislerinin çarpımları olmak üzere $PAQ = D$ olacak şekilde P ve Q matrislerini bulabiliriz (D bir önceki sayfada verilen diagonal D matrisidir). Dolayısıyla $AQ = P^{-1}D$ dir ve bö ylece AQ matrisinin son $n-r$ sütunundaki tüm bileşenler sıfırdır. O halde Q nun son $n-r$ sütunu $Ax = 0$ denklemini sağ lar. Böylece bu son $n-r$ sütun M nin elemanıdır. Bu elemanlar determinantı sıfır olmayan bir matrisin bir parçası olduklarından lineer bağımsızdır. O halde, Q nun son $n-r$ sütunu M nin lineer bağımsız elemanlarıdır demiş olduk. Şimdi bunların M yi doğurduğunu gösterelim. $x \in M$ alalım. O zaman $Ax = 0$ dır. Dolayısıyla $P^{-1}DQ^{-1}x = 0$ dır. $\det(P) \neq 0$ olduğundan $D\left(Q^{-1}x\right) = 0$ dır. Bö ylece $Q^{-1}x$ in ilk r koordinatı sıfırdır. O halde $Q^{-1}x$

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ y_{r+1} \\ \vdots \\ y_n \end{pmatrix}$$

formundadır ve buradan

$$x = Q \begin{pmatrix} 0 \\ \vdots \\ 0 \\ y_{r+1} \\ \vdots \\ y_n \end{pmatrix}$$

olur. Dolayısıyla x, Q nun son $n-r$ sütununun bir lineer kombinasyonu şeklinde yazılabilir. Özetle, Q nun son $n-r$ sü tununun M için bir baz olduğunu göstermiş olduk.

Şimdi genel duruma bakalım. M kümesi, $\mathbb{Z}^n$ nin

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &\equiv 0 \pmod{b_1} \\ &\vdots \\ a_{k1}x_1 + \cdots + a_{kn}x_n &\equiv 0 \pmod{b_k} \\ a_{(k+1)1}x_1 + \cdots + a_{(k+1)n}x_n &= 0 \\ &\vdots \\ a_{s1}x_1 + \cdots + a_{sn}x_n &= 0 \end{aligned}$$

denklemlerini sağlayacak şekildeki elemanlarının kümesi olsun.

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n - b_1x_{n+1} &= 0 \\ &\vdots \\ a_{k1}x_1 + \cdots + a_{kn}x_n - b_kx_{n+k} &= 0 \\ a_{(k+1)1}x_1 + \cdots + a_{(k+1)n}x_n &= 0 \\ &\vdots \\ a_{s1}x_1 + \cdots + a_{sn}x_n &= 0 \end{aligned}$$

denklemlerini sağlayacak şekildeki $(x_1,\ldots,x_n,x_{n+1},\ldots,x_{n+k}) \in \mathbb{Z}^{n+k}$ elemanlarının oluşturduğu kümeye M' diyelim. M' homojen olduğu için $B' = \{m'_1,\ldots,m'_r\}$ gibi bir bazını bulabiliriz. $m'_i = \left(m_{i1},\ldots,m_{i(n+k)}\right)$ olarak varsayalım ve $m_i = (m_{i1},\ldots,m_{in})$ alalım. Şimdi $B = \{m_1,\ldots,m_r\}$ nin M için bir baz olduğ unu gösterelim.

- $x = (x_1,\ldots,x_n) \in M$ alalım. Her $i \in \{1,\ldots,k\}$ için

$$a_{i1}x_1 + \cdots + a_{in}x_n \equiv 0 \pmod{b_i}$$

olduğunu biliyoruz. Bu nedenle

$$a_{i1}x_1 + \cdots + a_{in}x_n = b_i x_{n+i}$$

olacak şekilde $x_{n+i} \in \mathbb{Z}$ vardır. Dolayısıyla $x' = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+k}) \in M'$ dir ve böylece $a_1, \ldots, a_r \in \mathbb{Z}$ için $x' = \sum_{i=1}^{r} a_i m'_i$ dir. Buradan $x = \sum_{i=1}^{r} a_i m_i$ olacağı açıktır. O halde B, M için bir doğuray kümesidir.

- $a_1, \ldots, a_r \in \mathbb{Z}$ nin $\sum_{i=1}^{r} a_i m_i = 0$ denklemini sağladığını varsayalım.

$$a_1 = \cdots = a_r = 0$$

olduğunu göstermek için, B' baz olduğundan, $\sum_{i=1}^{r} a_i m'_i = 0$ olduğunu göstermek yeterlidir. $\sum_{i=1}^{r} a_i m_i = 0$ olduğundan sadece, her $j \in \{1, \ldots, k\}$ için $\sum_{i=1}^{r} a_i m_{i(n+j)} = 0$ olduğu- nu göstermeliyiz. $m'_1, \ldots, m'_r \in M'$ olduğundan bunlar M' nin denklemlerini sağlar. Dolayısıyla

$$b_j m_{i(n+j)} = a_{j1}m_{i1} + \cdots + a_{jn}m_{in}$$

dir. O halde

$$m_{i(n+j)} = \frac{\left(a_{j1}m_{i1} + \cdots + a_{jn}m_{in}\right)}{b_j}$$

dir. Böylece

$$\begin{aligned} \sum_{i=1}^{r} a_i m_{i(n+j)} &= \sum_{i=1}^{r} a_i \left(\sum_{l=1}^{n} a_{jl} m_{il}\right)/b_j \\ &= \left(\sum_{l=1}^{n} a_{jl} \left(\sum_{i=1}^{r} a_i m_{il}\right)\right)/b_j \\ &= \left(\sum_{l=1}^{n} a_{jl} 0\right)/b_j \\ &= 0 \end{aligned}$$

dır. Dolayısıyla B deki vektörler lineer bağımsızdır.

3. GİRİŞ

Bu bölümün temel amacı FORTRAN komutlarını basit pratik uygulamalar kullanarak yeniden hatırlatmaktır. Özellikle SUBROUTINE ve FUNCTION alt programlarının nasıl kullanıldığını göstermek üzere basit örnek programlar sunulmuştur. Fortran komutlarının kısa bir özeti ve Fortran programa dili ile ilgili bazı önemli noktalar Ek-A'da verilmiştir. Ayrıca bu bölümde hata, hassasiyet ve kararlılık kavramları üzerinde kısaca durulmuştur. Uzun süredir programlamadan uzak kalmış veya daha önce programlama dersi almış fakat ayrıntılı uygulama fırsatı bulamamış olanların bu bölümü mutlaka izlemeleri önerilmektedir. Özellikle verilen örnek programların yazılıp derlenmesi başlangıç için çok önemlidir.

3.1 Giriş

Bilgisayarlar verilen sayıları yapıları gereği ancak sonlu hassasiyetle hafızalarında tutabilir ve bu nedenle ancak sonlu bir hassasiyetle işlem yapabilirler. Sayıların temsili *bit* (binary digit) ile veya bitlerin bir araya gelmesinden oluşan *byte* (8'li bitler) ile yapılır. Temsil edilen sayılar tam sayı (integer, fixed-point) olabileceği gibi gerçel sayı (real, floating point) da olabilirler. Sayıların temsilinde kullanılan sayı sistemi çoğunlukla *ikili* (binary) sayı sistemi olmakla beraber 16'lı veya 10'lu sistemlerde kullanılmaktadır.

Günlük hayatta kullandığımız onlu sayı sisteminde, örneğin 365 sayısı, aslında üç tane 100, altı tane 10 ve beş tane 1'in toplamından oluşur. Bu durumu

$$365 = 3 \cdot 100 + 6 \cdot 10 + 5 \cdot 1 = 3 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 = (365)_{10}$$

şeklinde gösterebiliriz. Onlu sayı sisteminde 0, 1, ..., 9 rakamları vardır. Bütün sayılar 10'nun uygun katlarının uygun katsayılar ile çarpılması ve bunların toplanması ile elde edilir. 10 bu sistemin *taban* sayısıdır. Aynı sayıyı *ikili* sayı sistemine göre temsil etmek istersek ne yapmamız gerekir? İkili sistemde de bütün sayılar, 2'nin uygun katlarının 0 veya 1 ile çarpımlarının toplanması ile elde edilecektir. O halde 7 sayısını ikili sistemde temsil etmek için 7 sayısının içinde 2'nin değişik katlarından kaç tane olduğunu bulmamız gerekir. Bunu verilen sayıyı sürekli 2'ye bölerek ve kalanları takip ederek yapabiliriz. Bu sayıyı

$$7 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (111)_2$$

şeklinde temsil edebileceğimizi hemen görebiliriz. Benze şekilde 9 sayısı

$$9 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (1001)_2$$

olarak temsil edilebilir. İkili sayı sisteminin görüldüğü gibi taban sayısı 2'dir. Birden küçük sayıların temsili ise kullanılan sayı sisteminin tabanının (onlu sistemde 10, ikili sistemde 2) negatif üsleri kullanılarak yapılır. Bunların ayrıntılarına burada girilmeyecektir.

Örnek 3.1 11 sayısını ikili ve *dörtlü* sayı sistemlerinde temsil ediniz.

Çözüm: İkili sayı ssitemine göre

$$11 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (1011)_2$$

olacaktır. Dörtlü sayı sisteminde temsil etmek için ise 0, 1, 2 ve 3 sayılarını kullanmamız gerekir. 4'ün uygun katları bu sayıların uygun olanları ile çarpılarak elde edilen sayılar toplanırsa, istenilen temsil bulunacaktır. Buna göre 11 sayısı içinde 2 tane 4^1 ve 3 tane 4^0 olduğundan aranan temsil

$$11 = 2 \cdot 4^1 + 3 \cdot 4^0 = (23)_4$$

olacaktır.

3.2 Hassasiyet, Hata ve Kararlılık

Sayıların bilgisayarlarda ancak sınırlı bir hassasiyetle temsil edilebilmesi nedeniyle ortaya çıkan hatalar kullanılan makineye bağlıdır. Bu durum 'makine hassasiyeti' kavramı ile açıklanmaya çalışılır ve değişik mertebedeki sayıları makine işlemcisinin bir birlerinde ayrılabilmesi yeteneğinin bir ölçüsü olarak düşünülebilir.

Makine hassasiyeti ε, 1.0 sayısına eklendiğinde 1.0'den farklı bir sayı veren en küçük sayının büyüklüğü olarak tanımlanır. Bu sayı kullanılan makineden makineye farklılıklar gösterebilir. 32 bitlik ikili sistemi kullanan kişisel bilgisayarların çoğunda tekli hassasiyet (single precision) kullanıldığında $\varepsilon = 1.1 \times 10^{-7}$ kadardır. Çifte hassasiyet (double precision) kullanıldığında $\varepsilon = 2.2 \times 10^{-16}$ mertebesindedir. Bilgisayarla yapılan hesaplamarda ε kadar bir hatanın yapılması muhtemeldir. Bu hata **yuvarlama hatası** (round-off error) olarak bilinir ve programcının bu hatayı azaltmak veya yoketmek için yapabileceği bir şey

yoktur. Fakat aşağıda da görüleceği gibi bu hatanın etkilerini en aza indirmek için çeşitli tedbirler alınabilir.

Bu noktada bir yanlış anlamaya meydan vermemek için hemen belirtelim: ε bir makinede kullanılabilecek en küçük sayı demek değildir. Örneğin makine hassasiyeti 1×10^{-12} olan bir bilgisayarda 1×10^{-44} gibi bir sayı rahatlıkla kullanılabilir ve bu sayı ile işlemlerde yapılabilir. Fakat bu sayı örneğin 1'e eklendiğinde sonuç yine 1 olacaktır. Bilgisayar bu kadar farklı mertebedeki iki sayıyı ayırt etme kapasitesine sahip dağildir.

Diğer yanda, bilgisayar ile yapılan hesaplamaların çoğunda sürekli bir değişkenin seçilen belirli kesikli noktalarda değeri bulunur. Örneğin sayısal olarak türev veya integral alırken her ikisinin analitik tanımlarında geçen ve çok küçük bir aralığı temsil eden Δx büyüklüğünü sıfır limit değerine götürmek pratik olarak mümkün değildir. Bu nedenle sayısal olarak yapılan hesaplamaların çoğunda belirli bir hata yapılır. Bu hataya **kesme hatası** (truncation error) denir. Kesme hatası genelde programcının kontrolünde olup, sayısal analizin hemen hemen tamamı bu hatayı en aza indirme veya kontrol altında tutma yöntemlerini ortaya çıkarmak üzerine kuruludur.

Hassasiyet sayısal olarak yapılan bir işlemin sonucunun gerçek değerine ne kadar yakın olduğu ile ilgili bir kavramdır. Kararlılık ise çalıştırılan programın ıraksamadan istikrarlı bir biçimde çalışması demektir. Burada önemli olan hassasiyetten çok hatalıda olsa işlemlerin sorunsuz yürütülebilmesidir. Bu nedenle kararlı çalışan bir programın çıktılarının da hassas olacağı sanılmamalıdır.

Örnek 3.2 Hesap makinenizin 'makine hassasiyetini' bulunuz. Bunu yapmak için 1 sayısına örneğin 10^{-5}'den başlayarak gittikçe küçülen sayılar ekleyiniz. Bu işleme toplamın sonucu yine 1 sayısını verene kadar devam ediniz.

Örnek 3.3 a) Kullandığınız bilgisayarın hassasiyetini bulmak için aşağıda verilen programı bilgisayarınızda bulunan bir editör kullanarak bir dosyaya yazınız. FORTRAN derleyicinizi kullanarak bu programı derleyiniz ve sonra çalıştırınız. Makinenizin hassasiyeti nedir?

b) Aynı programı, programda geçen 'REAL' komutunu, 'REAL*8' ile değiştirerek yeniden çalıştırınız. Makine hassasiyetinizde bir değişiklik meydana geldi mi?

```
PROGRAM epsilon
```

```
*****
* epsilon.for - KullandIgInIz makinenin hassasiyetini (epsilon)
* bulmak amacI ile yazIlmIstIr
*****

      REAL eps, bir, bireps

      eps = 1.0
      bir = 1.0
  100 bireps = bir + eps
      IF( bireps .LE. bir) GOTO 200
      eps = 0.5 * eps
      GOTO 100
  200 eps = eps * 2.0
      WRITE(*,*)
      WRITE(*,*) '-------------------- '
      WRITE(*,*) ' Makine Hassasiyeti: '
      WRITE(*,*) ' epsilon = ', eps
      WRITE(*,*) '-------------------- '
      END
```

FORTRAN programlama dilinde tekli hassasiyet kullanılarak işleme tabi tutulacak değişkenler REAL komutu ile tanıtılır. Bu kategorideki hesaplamalar hafızada 32 bit (4 bayt) yer ayrılarak yapılır. REAL*8 ise değişkenin çifte hassasiyetle hesaplanacağını ilan eder ve bu durumda işlemler 64 bit (8 bayt) yer ayrılarak yapılır. Günümüzde bilgisayarların bilgi işleme hızları ve kapasiteleri çok arttığından bütün hesaplamaları çifte hassasiyette yapmak yerinde olur. Bu konuya yeri geldikçe ileride değinilecektir.

3.3 Hassasiyet Kaybı ve İşlem Önceliği

Yukarıda sözü edilen makine hassasiyetinin sınırlı olmasından ve aşağıda bahsedilecek işlem önceliği nedeniyle bilgisayarlarla işlem yaparken çok farklı mertebelerdeki

sayıları bir araya getirmek doğru değildir. Bu nedenle işlemlerde en az hatayı yapmak için, örneğin bir çok sayı toplanıyorsa aynı mertebedeki sayıların gruplanarak toplanması daha uygundur. Veya sayıları küçükten büyüğe doğru sıralayarak, önce küçük sayılardan başlayarak toplama yapmak daha uygun olacaktır. Bu şekilde işlemlerde ortaya çıkabilecek hassasiyet kaybı önlenmiş olur. Örneğin makine hassasiyeti $\varepsilon = 2 \times 10^{-6}$ olan bir makinede 1 sayısına bir milyon defa 10^{-6} sayısı eklenirse 2 yerine yine 1 sayısı elde edilir. Burada yapılması gereken önce küçük sayıları ekledikten sonra ortaya çıkan sayının daha büyük sayıya eklenmesidir.

Başka bir hassasiyet kaybı ise birbirlerine çok yakın sayıların bir birinden çıkartılmasında ortaya çıkar. Örneğin $x = 0.123$, $y = 0.124$ olsun. Bu sayıların her ikisinde de 3 anlamlı rakam vardır ve son haneleri en az hassas olan rakamlardır. Bu iki sayı bir birinden çıkartıldığında, her iki sayınında en az hassas olan basamaklarının farkı alınmış olur. Sonuç sadece bir anlamlı rakam içerir ve o da en anlamsız iki rakamın farkı olur. Bir birlerine çok yakın sayıların çıkartılması payda da ise yapılan hatanın etkisi daha da abartılmış olur. Bu nedenle mümkün olan her yerde bir birlerine yakın sayıların farklarının alınmasından kaçınılmalı, aynı işlemi yapmanın değişik yolları aranmalıdır.

Örnek 3.4 Aşağıda verilen işlemleri hassasiyet kaybı en az olacak şekilde yapınız.

a) x sıfıra çok yakın bir sayı olmak üzere $y = 1 - \cos(x)$

b) x 1'e göre çok büyük bir sayı olmak üzere $y = \sqrt{x+1} - \sqrt{x}$

Çözüm:

a) x sıfıra çok yakın ise verilen sayılar bir birine çok yakın olacaktır. y'i eşleniği ile çarpıp bölersek

$$y = \frac{(1-\cos(x))(1+\cos(x))}{1+\cos(x)} = \frac{\sin^2(x)}{1+\cos(x)}$$

olur. Bu değer sağlıklı bir şekilde hesaplanbilir.

b) (a)'da kullanılan yöntemi kullanarak bir çözüm bulunuz.

İşlemler sırasında dikkat edilmesi gereken bir diğer konu *işlem önceliğidir*. FORTRAN da kullanılan artimetik işlemciler şunlardır: "toplama" (+), "çıkarma" (-), "çarpma" (*), "bölme" (/) ve "üs alma" (**). Bu işlemcilerin öncelik sırası ise şöyledir: 1. üs alma, 2. çarpma ve bölme, 3. toplama ve çıkarma. Aynı önceliğe sahip işlemlerde, parantez içindekilere öncelik tanınır. Sınırlı hassasiyet ve işlem önceliği nedeniyle

bilgisayarla yapılan işlemlerde örneğin toplamanın değişme ve birleşme özellikleri sağlanmayabilir.

Örnek 3.5 Aşağıda verilen aritmetik işlemlerin yapılması için FORTRAN programlama dilinde komut satırına bu değerler nasıl yazılmalıdır?

a) $a = \frac{b+c}{2}$, b) $a = \frac{b^3}{2\pi}$, c) $y = x - \frac{ax}{b^2 z^3}$

Çözüm: b)

```
PI=3.1415
A=B**3/2./PI
```

3.4 Örnek Programlar ve Öneriler

3.4.1 Verilerin Ekrandan Okutulması

Temel FORTRAN komutlarını hatırlamanız için EK A'da verilen kısa özeti kullanınız. Bu noktada temel FORTRAN komutlarının kullanıldığı basit algoritmalı programlar yazarak başlamak çok daha öğretici olacaktır. Daha sonra programların zorluk seviyesi ve karmaşıklığı arttırılabilir. Başlangıç olarak iki sayıyı bilgisayar ekranından okuyarak bunların toplamını bulup ekrana yazan bir program yazalım. Aşağıda verilen basit program bu işi yapmak için yeterlidir.

```
      PROGRAM GIRIS1
C************
C Ekrandan iki sayI okur ve bu sayIlarIn toplamInI ekrana yazar
C Son degistirme tarihi: 17 Ekim 2002
C***********
      REAL*8 A,B,TOPLAM
      WRITE(*,*)' A ve B nin degerini giriniz:'
      READ(*,*) A,B
      TOPLAM= A + B
      WRITE(*,*)' TOPLAM=',TOPLAM
```

```
      STOP
      END
```

Örnek 3.6 Bilgisayarınızda bulunan bir editör programını kullanarak yukarıda verilen programı bir dosyaya yazınız. Daha sonra bir FORTRAN derleyicisi kullanarak bu programı derleyip çalıştırınız. A ve B değişkenlerinin değerini ekrandan girerken bu değerleri boşluklarla veya virgülle ayırarak giriniz. Örneğin A ve B için 3.2 ve 6.7 değerleri girilecekse, ekrana

```
3.2 6.7
```

veya

```
3.2, 6.7
```

giriniz. A ve B için bu değerlerin girildiği durum için örnek çıktı aşağıda gösterilmiştir.

```
 A ve B nin degerini giriniz:
 3.2 6.7
 TOPLAM=        9.900000000000000
Stop - Program terminated.
```

3.4.2 Boyutlu Değişkenler ve Döngüler

Yukarıda verilen program biraz daha geliştirilebilir. Programın ikiden fazla sayıyı toplaması gerektiğini varsayalım. Bu durumda her sayı için A, B, C, $\cdots$ gibi farklı bir değişken kullanmak yerine boyutlu bir tek değişken kullanılabilir. Bu tip değişkenler vektör değişkenler olarak adlandırılır. Bu söylenen değişiklikleri yansıtacak şekilde GIRIS1 programı yeniden düzenlenerek aşağıdaki GIRIS2 programı elde edilmiştir.

3.4.3 'Subroutine' Alt Programı

GIRIS2 programını geliştirmeye devam edebiliriz. Programcılığın önemli basamaklarından biri yapılacak bir işi parçalara bölerek her parça için 'SUBROUTINE' veya 'FUNCTION' denen alt programları yazmaktır. Bu programın daha derli toplu olmasını,

kolay takip edilmesini ve tekrar tekrar yapılan işlemler varsa bunların daha etkin biçimde hesaplanmasını sağlar. Bizim geliştirmeye çalıstığımız programda toplama işlemini bir alt program yazarak yapabiliriz. Ayrıca bazı değişkenler için başlangıç değerlerini 'DATA' komutunu kullanarak atayabiliriz. 'DATA' komutu kullanılarak değeri atanan değiskenlerin değeri gerektiğinde program içinde daha sonra değiştirilebilir. Anılan değişikliklerin yapıldığı program GIRIS3 adı ile aşağıda verilmiştir.

3.4.4 'Function' Alt Programı

Başka bir alt program çeşidi ise 'FUNCTION' alt programıdır. Bir 'SUBROUTINE' alt programında birden fazla girdi ve birden fazla çıktı olabilir. Bir 'FUNCTION' alt programında ise birden fazla girdi olabilmesine karşın, alt programın kendi adı ile aynı olan sadece bir tek çıktısı olabilir. Bir 'SUBROUTINE' alt programı ile bir 'FUNCTION' alt programı arasındaki en önemli fark budur.

a ve b sabit sayılar olmak üzere, $F(x) = ax^2 - b$ fonksiyonu verilmiş olsun. Bu fonksiyonun belirli bir x_{min} değerinden başlayarak bir x_{max} değerine kadar eşit aralıklarla alacağı değerleri bulup tablo halinda yazmaya çalıştığımızı varsayalım. Bu işi yapabilecek bir program aşağıda verilmiştir.

3.4.5 Çıktıların Bir Dosyaya Yazdırılması

Yukarıda verilen bütün programlarda programın çıktısı ekrana yazılmaktadır. Eğer program çıktısı ekrandan takip edilemeyecek kadar çoksa veya sonuçlar ileride kullanılmak üzere kalıcı olarak saklanmak isteniyorsa bu durumda sonuçların bir dosyaya yazılması gerekir. Bunu yapmak için programlar içerisinde 'OPEN' komutunu kullanarak istediğimiz isimde bir dosyayı açıp sonuçlarımızı bu dosyaya yazabiliriz. Açılan dosya ile işimiz bitince bu dosyayı 'CLOSE' komutu ile kapatıyoruz. Aşağıda verilen GIRIS5 adlı program bu söylenenleri yapmak üzere GIRIS4 adlı programın yeniden düzenlenmiş halidir.

3.4.6 Verilerin Dosyadan Okutulması

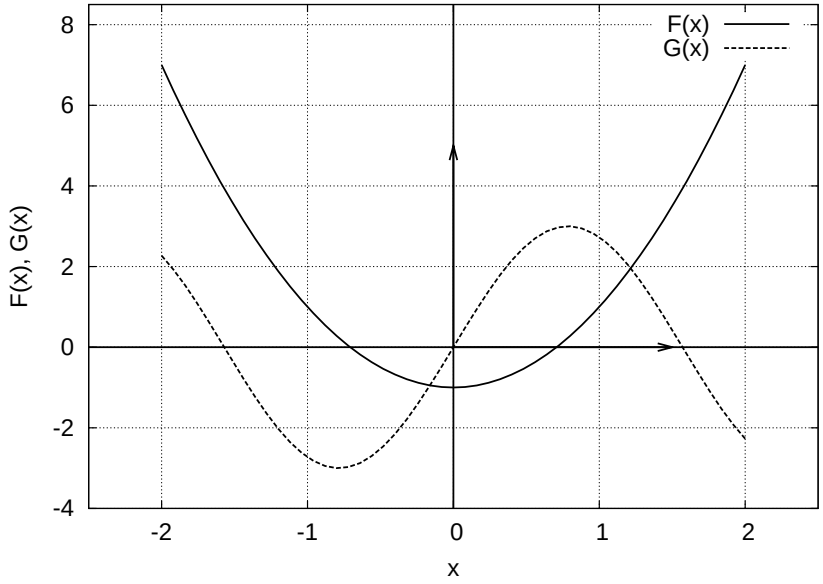
Bir programda kullanılacak veriler programda DATA komutu ile tanımlanabilir, daha önce gördüğümüz gibi ekrandan veya birazdan göreceğimiz gibi bir dosyadan da okutulabilir. Bu amaç için verilerin okunacağı dosyanın önceden OPEN komutu ile açılması gerekir. Bu açılan dosyada program için gereken veriler bulunmalıdır. Verilerin eksiksiz ve doğru okunabilmesi için, programda dosyada bulunan verilerin dizilişine uygun okuma komutları bulunmalıdır. Örneğin yukarıda kullandığımız GIRIS3 programında ekrandan okunan toplanacak sayı sayısı ve bu sayıların değeri bir dosyadan da okutulabilir. GIRIS3 programının bu amaca uygun şekilde değiştirilmiş hali GIRIS6 adı altında aşağıda verilmiştir. Buradaki en önemli farklılık READ(*,*) komutunun birinci argümanının verilerin okunacağı dosyanın birim numrası ile aynı olmasıdır.

3.4.7 Grafik Çizimi

Bilimsel çalışmalarda elde edilen verilerin uygun ve öz bir şekilde sunulması en az yapılan araştırma kadar önemlidir. Örneğin yukarıdaki GIRIS5 programının çalıştırılmasından sonra elde edilen dosyada bir çok rakamdan oluşan bir veri yığını vardır. Buradaki bilgi içeriğinin anlaşılır bir şekilde doğrudan başkalarına iletilmesinin kolay ve etkin yollarından biri elde edilen veriyi grafik ile temsil etmektir. GIRIS5 programının ürettiği verinin grafiği, yani $F(x)$ ve $G(x)$ fonksiyonlarının grafiği Şekil 1.1'de gösterilmiştir.

Örnek 3.7 Yukarıda verilen GIRIS5 programına benzeyen ve diyelimki adı GIRIS51 olan bir program yazınız. Bu program $f(x) = 2(x-1)^2$ ve onun birinci ve ikinci turevlerini $[x_{min}, x_{max}]$ aralığında eşit aralıklı N tane noktada hesaplayıp adı kullanıcı tarafından belirlenen bir dosyaya yazmalıdır. Çıktıların yazıldığı dosyada birinci kolona x değerleri yazılmalıdır. Benzer şekilde 2., 3. ve 4. kolonlara sırası ile $f(x)$, $f(x)$ fonksiyonunun birinci ve ikinci türevleri yazılmalıdır. Kullanıcı en küçük ve en büyük x değerlerini (x_{min}, x_{max}) ve bu aralıkta kaç tane değer basılacağını (N) girebilmelidir. Oluşturduğunuz bu çıktı dosyasını kullanarak verilen fonksiyonun ve onun türevlerinin grafiğini çiziniz. Önce her fonksiyonu tek tek çizerek görünüz. Daha sonra hepsini topluca aynı grafik

üzerinde gösteriniz.



Şekil 3.1 $F(x)$ ve $G(x)$ fonksiyonlarının grafikleri.

4. ADİ DİFERANSİYEL DENKLEMLERİNİN SAYISAL ÇÖZÜMÜ

Temel bilimlerde, mühendislik veya iktisatta bazı problemlerin çözümünde sık sık bir diferansiyel denkleminin çözülmesi gerekmektedir. Eğer diferansiyel denkleminin analitik bir çözümü bulunamıyorsa bu durumda sayısal yöntemlerin kullanılması kaçınılmaz olacaktır. Bu bölümde başlangıç şartlı adi diferansiyel denklemlerinin sayısal olarak nasıl çözüleceğini, kullanılan bir yöntemi geliştirerek hataları azaltmak için neler yapılabileceğini, programlamada dikkat edilmesi gereken noktaları adım adım göreceğiz. Konuların anlatımı ve programlama basitten karmaşığa doğru beraber yürütülecektir.

4.1 Giriş

Bir diferansiyel denkleminde bir veya birden fazla bağımlı değişkenin sadece bir bağımsız değişkene göre türevleri varsa bu denkleme adi (bayağı) diferansiyel denklem diyoruz. Örneğin basit harmonik bir salınıcının uyduğu diferansiyel denklem, t zamanı ve x salınıcının merkeze göre yer değiştirmesini temsil etmek üzere,

$$\frac{d^2x}{dt^2} + w^2x = 0 \tag{4.1}$$

ile verilir. Burada t bağımsız değişken, x ise bağımlı değişkendir. Bir diferansiyel denkleminin mertebesi denklemde görülen en yüksek dereceli türeve eşittir. Bu nedenle yukarıdaki denklem ikinci mertebe bir diferansiyel denklemdir. Adi diferansiyel denklemler sağladıkları şartlara göre başlangıç veya sınır şartlı olabilir. Başlangıç şartlı diferansiyel denklemlerinde denklemin sağlaması gereken şartlar sadece bir noktada tanımlıdır. Sınır şartlı diferansiyel denklemlerinde ise denklemin sağlaması gereken şartlar birden fazla noktada tanımlanmıştır. Örneğin

$$\frac{d^2x}{dt^2} + w^2x = 0, \quad x(t_0) = x_0, \quad \left.\frac{dx}{dt}\right|_{t_0} = x_0' \tag{4.2}$$

diferansiyel denklemi başlangıç şartlıdır, çünkü denklemin sağlaması gereken şartlar sadece $t = t_0$ noktasında tanımlıdır.

$$\frac{d^2x}{dt^2} + w^2x = 0, \quad x(t_1) = x_1, \quad \frac{dx}{dt}|_{t_2} = x_2' \tag{4.3}$$

denklemi ise denklemin sağlaması gereken şartlar x_1 ve x_2 gibi iki *farklı* noktada tanımlandığından sınır şartlıdır. Bu bölümde sadece başlangıç şartlı diferansiyel denklemleri gözönüne alınacaktır.

Birden yüksel mertebeli herhangi bir diferansiyel denklem, mertebe sayısı kadar birinci mertebe diferansiyel denkleme dönüştürülebilir. Bu yöntemle n. mertebe herhangi bir diferansiyel denklem, n tane birinci mertebe diferansiyel denklemi şeklinde yazılabilir.

Örnek 4.1 Denklem (3.1)'de verilen diferansiyel denklemini iki tane birinci mertebe diferansiyel denklem şeklinde yazınız.

çözüm: Bunu yapmak için

$$\begin{aligned} x_1(t) &= x(t) \\ x_2(t) &= \frac{dx}{dt} \end{aligned} \tag{4.4}$$

tanımlarını yapalım. Bunları asıl denklemde yerine koyarsak

$$\frac{dx_2}{dt} + w^2 x_1 = 0 \tag{4.5}$$

elde edilir. Yukarıda tanımladığımız ikinci denklem ise

$$x_2(t) = \frac{dx_1}{dt} \tag{4.6}$$

olur ve aradığımız diğer denklemin kendisidir. Denklem (3.5) ve (3.6) birleştirilirse aranan denklem sistemi

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -w^2 x_1 \end{aligned} \tag{4.7}$$

elde edilmiş olur. Bu son iki denklem birinci mertebe ve kuplajlıdır.

Örnek 4.2 üçüncü mertebe, lineer olmayan ve başlangıç şartları belli

$$\begin{aligned} y\frac{d^3y}{dx^3} + (\frac{dy}{dx})^2 + y^2 &= g(x) \\ y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad y''(x_0) &= y''_0, \end{aligned} \tag{4.8}$$

diferansiyel denklemini üç tane birinci mertebe diferansiyel denklemler halinde yazınız. Ayrıca yeni denklemlerin sağladığı başlangıç koşullarını da belirtiniz.

çözüm: Bunu yapmak için daha önce yaptığımız gibi

$$\begin{aligned} y_1(x) &= y(x) \\ y_2(x) &= \frac{dy}{dx} \\ y_3(x) &= \frac{d^2y}{dx^2} \end{aligned} \tag{4.9}$$

tanımlarını yapıp asıl diferansiyel denkleminde yerine yazalım. Dikkat edilirse yukarıdaki denklemlerden son ikisi aradığımız denklemlerden ikisini zaten tanımlamış olur. üçüncüsüde verilen denklemden bulunacaktır. (3.8)'deki diferansiyel denklemini y_1, y_2 ve y_3 cinsinden yazarsak

$$y_1 \frac{dy_3}{dx} + y_2^2 + y_1^2 = g(x) \tag{4.10}$$

elde ederiz. Yeniden düzenleme yapılırsa aradığımız denklem sistemi elde edilir.

$$\begin{aligned} \frac{dy_1}{dx} &= y_2, \quad y_1(x_0) = y_0, \\ \frac{dy_2}{dx} &= y_3, \quad y_2(x_0) = y_0', \\ \frac{dy_3}{dx} &= -y_2^2/y_1 - y_1 + g(x)/y_1, \quad y_3(x_0) = y_0'', \end{aligned} \tag{4.11}$$

Daha karmaşık bir örnek ise ters kare kuralına uyan merkezi bir kuvvet (örneğin kütle çekim) altında (x,y) düzleminde iki boyutta bir yörüngede dönen bir parçacığın hareketini temsil eden diferansiyel denklem olabilir. G problemin sabitlerini temsil etmek ve $\mathbf{r} = x\mathbf{i} + y\mathbf{j}$ olmak üzere, Newton'un ikinci yasasına göre hareket denklemi

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{G\hat{\mathbf{r}}}{r^2} = -\frac{G\mathbf{r}}{r^3} \tag{4.12}$$

şeklinde yazılabilir. Bu denklem görüldüğü gibi ikinci mertebe bir diferansiyel denklem olup iki boyutta yazılmıştır. Bu diferansiyel denklem her boyut için ayrı ayrı yazılabilir. Böylece

$$\begin{aligned} \frac{d^2x}{dt^2} &= -\frac{Gx}{r^3} \\ \frac{d^2y}{dt^2} &= -\frac{Gy}{r^3} \end{aligned} \tag{4.13}$$

elde edilir. Bunlar kuplajlı ikinci mertebe diferansiyel denklemlerdir.

Örnek 4.3 (3.13)'de verilen denklemleri birer mertebe daha indirgenerek birinci mertebe dört denklem elde ediniz.

çözüm: Bunu yapmak için hızın x ve y yönlerindeki bileşenlerini sırası ile $V_x = \frac{dx}{dt}$ ve

$V_y = \frac{dy}{dt}$ olarak yazabiliriz. Bu tanımları kullanarak

$$\begin{aligned} \frac{dx}{dt} &= V_x \\ \frac{dy}{dt} &= V_y \\ \frac{dV_x}{dt} &= -\frac{Gx}{r^3} \\ \frac{dV_y}{dt} &= -\frac{Gy}{r^3} \end{aligned} \tag{4.14}$$

şeklinde dört tane kuplajlı birinci mertebe diferansiyel denklem elde edilebilir. Bu denklemleri daha düzenli yazmak için $y_1 = x(t)$, $y_2 = y(t)$, $y_3 = \frac{dx}{dt}$ ve $y_4 = \frac{dy}{dt}$ tanımlarını kullanarak (3.14)'deki denklemler yeniden

$$\begin{aligned} \frac{dy_1}{dt} &= y_3 \\ \frac{dy_2}{dt} &= y_4 \\ \frac{dy_3}{dt} &= -\frac{Gy_1}{r^3}, \quad r = \sqrt{y_1^2 + y_2^2} \\ \frac{dy_4}{dt} &= -\frac{Gy_2}{r^3} \end{aligned} \tag{4.15}$$

şeklinde yazılabilir.

Yüksek mertebeli herhangi bir diferansiyel denklem yukarıda örnekleriyle gördüğümüz gibi birinci mertebe diferansiyel denklemler dizisi halinde yazılabildiğinden, başlangıç şartlı adi diferansiyel denklemlerini sayısal olarak çözme yöntemleri sadece birinci mertebe diferansiyel denklemler için geliştirilmiştir. Bu nedenle birinci mertebe bir diferansiyel denklem ele alalım. En genel haliyle böyle bir denklem, başlangıç şartı ile beraber,

$$\frac{dy}{dx} = f(x,y), \qquad y(x_0) = y_0 \tag{4.16}$$

şeklinde yazılabilir. Burada x bağımsız, y ise bağımlı değişkendir. $f(x,y)$ aslında y'nin eğimini bütün (x,y) düzlemi üzerinde x ve y'nin bir fonksiyonu olarak tanımlar. Fakat bu bir çözüm değil, bir çözümler kümesi oluşturur. Bu çözümlerden bir tanesini seçmek için (x,y) düzlemi üzerinde bir noktanın, örneğin (x_0, y_0) noktasının verilmesi gerekir. Bu durumda aranan çözüm bu noktadan geçen çözümün kendisi olur ve tektir. Bu noktanın seçilmesi aslında gözönüne alınan diferansiyel denklemine bir başlangıç şartı verilmesi anlamına gelir. Diferansiyel denklemlerini sayısal olarak çözeceğimizden, bir

diferansiyel denkleminde geçen türevleri sayısal olarak temsil edebileceğimiz yöntemler geliştirmeliyiz. Bu bir sonraki bölümün konusudur.

4.2 Sayısal Türev Alma

Denklem (3.16) veya benzeri denklemleri çözerken türev alınması gerekecektir. Bu nedenle sayısal olarak türev almak için bir yöntem bulmamız gerekir. Önce türevin tanımından başlayalım. Belirli bir aralıkta sürekli olan bir $f(x)$ fonksiyonunun herhangi bir x noktasındaki türevi hatırlanacağı gibi

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{y(x_0 + \Delta x) - y(x_0)}{\Delta x} \tag{4.17}$$

olarak tanımlanır. Bu tanımı kullanarak bilgisayarla bir foknsiyonun sayısal olarak türevini nasıl almamız gerekir? Δx kullandığımız makinenin hassasiyet limitine göre sonlu olmak üzere istediğimiz kadar küçük olabilir fakat limiti hiç bir zaman sıfıra götüremeyiz. Bu nedenle birinci ve gerektiğinde daha yüksek dereceli türevleri sayısal olarak alabilmek için uygun bir yöntem geliştirmemiz gerekir.

Böyle bir yöntemi fonksiyonların Taylor serisi açılımlarını kullanarak elde etmek mümkündür. $f(x)$ fonksiyonunun $x+\Delta x$ noktasındaki Taylor açılımı, $h = \Delta x$ olmak üzere,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \cdots \tag{4.18}$$

şeklindedir. Burada $f'(x)$, $f''(x)$, $\cdots$ $f(x)$ fonksiyonunun x noktasındaki birinci, ikinci, $\cdots$ türevleridir. Benzer şekilde $f(x)$'in $x-h$ noktasındaki Taylor açılımı

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \cdots \tag{4.19}$$

olur. Denklem (3.18)'i kullanarak $f(x)$'in türevi yaklaşık olarak

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h^2) \tag{4.20}$$

yazılabilir. Burada h'nin iki ve daha büyük kuvvetlerinin bulunduğu terimler ihmal edilmiştir. Yani türev almada yaptığımız hata h^2 ile orantılıdır. Herhangi yaklaşık bir yöntemde ihmal edilen terimler h^{k+1} ile orantılı ise, bu yöntemin k. mertebeden hassas olduğu söylenir. Bu tanıma göre denklem (3.20)'de verilen birinci türevin yaklaşık değeri birinci mertebeden hassastır. Birinci türev için elde ettiğimiz bu yaklaşık değer *ileriye*

doğru sonlu fark yaklaşımı olarak bilinir. Benzer şekilde denklem (3.19)'u kullanarak aynı türev *geriye doğru sonlu fark* olarak

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h^2) \tag{4.21}$$

şeklinde yazılabilir. Bu yaklaşık değerde birinci mertebeden hassastır. Bunların dışında denklem (3.19)'u denklem (3.18)'den taraf tarafa çıkartırsak türev için *merkezi sonlu fark*

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^3) \tag{4.22}$$

denklemini elde ederiz. Bu yöntemde dikkat edilirse hesaplanan türev değeri ikinci mertebeden hassastır. Bu nedenle merkezi sonlu fark yöntemi ileri ve geri sonlu fark yöntemlerinden bir mertebe daha hassastır. Yukarıda gösterildiği gibi bir fonksiyonun bir noktadaki türevini, fonksiyonun o noktaya komşu noktalarındaki değerleri cinsinden yazma yöntemine genel olarak *sonlu farklar yaklaşık metodu* denir.

Örnek 4.4 Denklem (3.18) ve (3.19)'un toplamından ikinci türev için yaklaşık

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^4) \tag{4.23}$$

değerinin elde edilebileceğini gösteriniz.

4.3 Euler Metodu

(3.16)'daki diferansiyel denklemini sayısal olarak çözmek için kullanılabilecek en basit yöntem Euler metodudur. Bu yöntemde çözümün eğiminin çok yavaş değiştiği kabul edilir. Bu nedenle sonlu fakat küçük bir bağımsız değişken aralığı Δx için y'nin çok yavaş değiştiği veya sabit olduğu kabul edilebilir. Sürekli herhangi bir fonksiyonun bir x noktasındaki türevinin (3.17) ile verildiğini görmüştük. Fakat bu denklemin sayısal hesaplamalarda pratik olarak pek işe yaramadığını ve daha farklı yöntemlerin geliştirilmesi gerektiğini bir önceki bölümde belirtmiştik. Bu nedenle (3.16)'daki diferansiyel denklemi, denklem (3.20)'de verilen birinci derece türevin ileriye doğru sonlu fark tanımından faydalanarak

$$\frac{y(x+\Delta x) - y(x)}{\Delta x} = f(x,y) \tag{4.24}$$

veya

$$y(x+\Delta x) = y(x) + \Delta x f(x,y) \tag{4.25}$$

olarak yazılabilir.

Yukarıdaki denklem, bir x_0 noktasındaki y_0 değerinin bilinmesi durumunda, daha sonraki bir $x_1 = x_0 + \Delta x$ noktasındaki $y_1 = y(x_0 + \Delta x)$ çözümünün bulunabileceğini gösterir. x_1 noktasındaki çözüm yaklaşık olarak

$$y_1 = y_0 + \Delta x f(x_0, y_0) \tag{4.26}$$

ile verilecektir. Böylece (x_1, y_1) noktası elde edilmiş olur. Bu nokta yeni bir başlangıç şartı olarak kullanılırsa bir sonraki $x_2 = x_0 + 2\Delta x$ noktasındaki çözüm yaklaşık olarak bulunabilir. x_2 noktasındaki yaklaşık çözüm

$$y_2 = y(x_0 + 2\Delta x) = y(x_1 + \Delta x) = y_1 + \Delta x f(x_1, y_1) \tag{4.27}$$

olacaktır. Bu şekilde adım adım devam ederek türevi $f(x,y)$ olan asıl $y(x)$ fonksiyonunu bulmuş oluyoruz. Yani integral alarak türevi alınmış ilkel fonksiyonu bulmaya çalışıyoruz. Δx sayısal çözümde kullanılan adım uzunluğu olarak bilinir.

$$\begin{aligned} h &= \Delta x \\ x_n &= x_0 + n\Delta x = x_0 + nh \\ y(x_n) &= y_n \\ y(x_{n+1}) &= y_{n+1} \end{aligned} \tag{4.28}$$

tanımlarını kullanarak ve denklem (3.25)'den hareketle Euler metodu

$$y_{n+1} = y_n + hf(x_n, y_n), \quad y(x_0) = y_0, \quad 0 \leq n \leq N \tag{4.29}$$

olarak yazılabilir. Bu bir tekrarlama bağıntısıdır. Başlangıçta (x_0, y_0) değerlerinin bilinmesi durumunda diğer bütün değerler bir önceki değerlerden faydalanarak adım adım bulunabilir.

Dikkat edilirse bu yöntemde aranan çözümün eğimi $f(x,y)$ her adımın başlangıcında hesaplanıp bu adım için kullanılmaktadır. Başlangıçtan itibaren bu süreç devam ettirilirse (x_0, y_0), (x_1, y_1), (x_2, y_2), $\cdots$, (x_N, y_N) dizisi elde edilir. Bu dizinin (x,y) düzleminde oluşturduğu noktalar kümesinin yaklaşık olarak aranan gerçek çözümü takip etmesi beklenir. x_0'dan başlayarak x_s'de son bulan belirli bir aralık için çözüm aranıyorsa, bu aralık K tane eşit aralığa bölünebilir. Δx'in küçük olmasını garantilemek için K'nın yeterince

büyük olması gerekir. Aralık uzunluğu

$$\Delta x = (x_s - x_0)/K \tag{4.30}$$

ile verilir.

4.3.1 Euler Metodu İçin Hata Tahmini

Euler metodunda yapılan hata ne kadardır? çözümün her adımında y yaklaşık olarak

$$y(x+\Delta x) = y(x) + \Delta x f(x,y) = y(x) + \Delta x \frac{dy}{dx} \tag{4.31}$$

olarak yazılabilir. Fakat doğru ve kesin açılım Taylor serisi ile verilir. Denklem (3.31)'in Taylor serisi açılımı

$$y(x+\Delta x) = y(x) + \Delta x \frac{dy}{dx} + \frac{(\Delta x)^2}{2}\frac{d^2y}{dx^2} + \cdots \tag{4.32}$$

olacaktır. Bu nedenle denklem (3.31)'in kullanılması ile yapılan her adımdaki hesap $(\Delta x)^2$ mertebesinde hatalıdır. Eğer Δx küçük ise, $(\Delta x)^2$ çok küçük olacaktır. Fakat Δx'in küçük olması için K'nın çok büyük seçilmesi gerekir, yani atılan adım sayısının arttırılması gerekir. x_0 noktasından x_s noktasına kadar yapılan çözümde toplam hata $K(\Delta x)^2 = (x_s - x_0)\Delta x$ olacaktır. Yani Euler metodunda yapılan yerel hata $(\Delta x)^2$, toplam hata ise Δx mertebesindedir. Prensipte K'yı yeterince büyük seçerek, hatayı istenildiği kadar küçültmek mümkündür fakat bu pratik değildir.

Örnek 4.5 örnek olarak aşağıda verilen diferansiyel denklemini Euler metodunu kullanarak $[0,1]$ aralığında çözelim.

$$\begin{aligned} \frac{dy}{dx} &= \frac{x}{2} \\ y(x_0) &= y_0 \end{aligned} \tag{4.33}$$

çözüm: Bu diferansiyel denkleminin analitik çözümünün

$$y(x) = y_0 + \frac{1}{4}(x^2 - x_0^2)$$

olduğunu göstermek ödev olarak size bırakılmıştır. Analitik çözüm sayısal çözümü test etmek için kullanılacaktır. Verilenlere göre $f(x,y) = \frac{1}{2}x$'dir. Tablo 3.1'de $x_0 = 0$, $y_0 =$

Tablo 4.1 Euler metodu ile $\frac{dy}{dx} = \frac{x}{2}, \quad y(0) = 0.25$ diferansiyel denkleminin çözümü. Analitik ve sayısal çözümler grafikte gösterilmiştir.

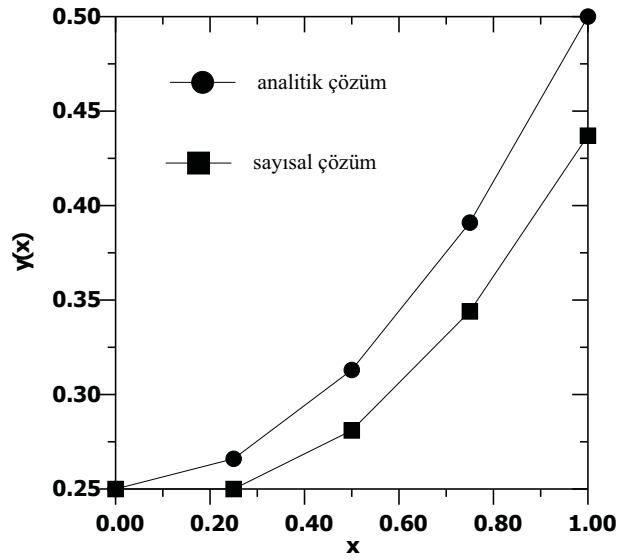
x	y_a	y_s	$\Delta y = y_a - y_s$
0	0	$y_0 = 0$	0
$\frac{1}{4}$	$\frac{1}{4}((\frac{1}{4})^2+1) = \frac{17}{64}$	$y_1 = y_0 + hf(x_0,y_0) = \frac{1}{4} + \frac{1}{4}\frac{0}{2} = \frac{16}{64}$	$\frac{17}{64} - \frac{16}{64} = \frac{1}{64}$
$\frac{2}{4}$	$\frac{1}{4}((\frac{2}{4})^2+1) = \frac{20}{64}$	$y_2 = y_1 + hf(x_1,y_1) = \frac{16}{64} + \frac{1}{4}\frac{1}{2}\frac{1}{4} = \frac{18}{64}$	$\frac{20}{64} - \frac{18}{64} = \frac{2}{64}$
$\frac{3}{4}$	$\frac{1}{4}((\frac{3}{4})^2+1) = \frac{25}{64}$	$y_3 = y_2 + hf(x_2,y_2) = \frac{18}{64} + \frac{1}{4}\frac{1}{2}\frac{2}{4} = \frac{22}{64}$	$\frac{25}{64} - \frac{22}{64} = \frac{3}{64}$
1	$\frac{1}{4}((1)^2+1) = \frac{32}{64}$	$y_4 = y_3 + hf(x_3,y_3) = \frac{22}{64} + \frac{1}{4}\frac{1}{2}\frac{3}{4} = \frac{28}{64}$	$\frac{32}{64} - \frac{28}{64} = \frac{4}{64}$

0.25 ve $h = \frac{1}{4} = 0.25$ için verilen denklemin analitik çözümü y_a, Euler yöntemine göre elde edilen sayısal çözümü y_s ve yapılan hata $\Delta y = y_a - y_s$ $x = 0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1$ noktalarında gösterilmiştir.

Tablodan da görüldüğü gibi sayısal çözümde yapılan hata h ile doğrusal olarak artmaktadır. şekil 1'de sayısal ve analitik çözümler aynı grafik üzerinde gösterilmiştir. ıki çözüm arasındaki fark yani sayısal çözümde yapılan hata şekilde görüldüğü gibi gittikçe artmaktadır. Bu tehlikeli bir duruma işaret eder. Bir süre sonra sayısal çözümde yapılan hata gerçek çözüm ile karşılaştırılabilir olacak ve daha sonra tamamen tanınmaz bir hal alarak gerçek çözüm ile bir ilgisi kalmayacaktır. Sayısal çözümde yapılan en ufak bir hata sınırsız şekilde artıyorsa bu durumda bir sonraki bölümde göreceğimiz gibi bir kararlılık sorunu var demektir.

4.3.2 Euler Metodunun Kararlılık Analizi

Sayısal çözümlerde göz önüne alınması gereken önemli kavramlardan bir tanesi kararlılıktır. Sayısal bir yöntemin kararlı olması çözümün kısa sürede çok büyüyerek veya sonsuza giderek tanınmaz hale gelmemesi olarak tanımlanabilir. Bazı yöntemler her şart altında kararlı, bazıları her zaman karasız olurken, bazı yöntemler ancak belirli parametre değerlerinde kararlı kalmaktadır. Kullanılan sayısal bir yöntemin detaylı bir kararlılık analizini yapmak her zaman mümkün olmayabilir. Euler metodunun kararlılık analizini yapmak için (3.29) denkleminde verilen sayısal çözümün, örneğin bilgisayarın sınırlı hassasiyetinden dolayı, gerçek sayısal çözümden ((3.16)'daki diferansiyel denkleminin anal-



Şekil 4.1 Örnek 3.5'de çözülen diferansiyel denkleminin sayısal ve analitik çözümleri.

itik çözümünden değil) δy kadar uzaklaştığını varsayalım. Bunu (3.29)'a eklersek

$$y_{n+1} + \delta y_{n+1} = y_n + \delta y_n + h * f(x_n, y_n + \delta y_n) \tag{4.34}$$

elde edilir. Denklemdeki son terimi Taylor serisine açıp sadece δy'ye göre birinci mertebe terimleri alırsak

$$\begin{aligned} y_{n+1} + \delta y_{n+1} &= y_n + \delta y_n + h * [f(x_n, y_n) + \left.\frac{\partial f(x,y)}{\partial y}\right|_n \delta y_n] \\ &= y_n + h * f(x_n, y_n) + \delta y_n + h * \left.\frac{\partial f(x,y)}{\partial y}\right|_n \delta y_n \\ &= y_{n+1} + \delta y_n + h * \left.\frac{\partial f(x,y)}{\partial y}\right|_n \delta y_n \end{aligned} \tag{4.35}$$

bulunur. Bu denklemden n.adımda yapılan bir hatanın bir sonraki adıma nasıl yansıyacağını veren

$$\delta y_{n+1} = [1 + h * \left.\frac{\partial f(x,y)}{\partial y}\right|_n] \delta y_n \tag{4.36}$$

denklemini elde ederiz. Buradan açık olarak görüldüğü gibi herhangi bir şekilde sayısal çözüme karışan bir hatanın nasıl ilerleyeceği $k = (1 + h * \frac{\partial f(x,y)}{\partial y})$ teriminin büyüklüğüne bağlıdır. $|k| < 1$ için hata gittikçe azalacak, $|k| > 1$ ise hata n arttıkça artarak çözümü kirletecek ve belkide tanınmaz hale getirecektir. Euler metodunun kararlı olması için bu nedenle

$$-1 \leq 1 + h * \frac{\partial f(x,y)}{\partial y} \leq +1 \tag{4.37}$$

şartının sağlanması gerekir. $h > 0$ olduğundan bu şart ancak $\frac{\partial f(x,y)}{\partial y} < 0$ ise mümkün olabilir. Adım uzunluğunun dikkatlice seçilmesi gereken bir parametre olduğu böylece açıklık kazanmış oldu.

Daha kesin bir tanım yapmak gerekirse sayısal bir çözüm eğer gerçek çözümden bir miktar uzaklaşıldığında büyüme eğilimi göstermiyorsa *kararlıdır*.

Örnek 4.6

$$\frac{dy}{dx} = y + x \tag{4.38}$$

diferansiyel denkleminin kararlılık analizini yapınız.

çözüm: Bu denklemde $f(x,y) = x + y$ olduğundan analiz için k değerini hesaplarsak

$$k = 1 + h * \frac{\partial f(x,y)}{\partial y} = 1 + h$$

elde ederiz. $-1 \leq k \leq 1$ şartı hiç bir zaman sağlanamayacağından Euler metodu verilen denklem için her şart altında kararsızdır.

Örnek 4.7 Aşağıda verilen diferansiyel denklemlerinin kararlılık analizini yapınız.
a) Büyüme problemi:

$$\frac{dy}{dx} = \alpha y, \quad \alpha > 0$$

b)Bozunma problemi:

$$\frac{dy}{dx} = -\alpha y \quad \alpha > 0$$

Örnek 4.8

$$\frac{dy}{dx} = y + x, \quad y(x_0) = y_0 \tag{4.39}$$

diferansiyel denklemini Euler yöntemini kullanarak çözen bir FORTRAN programı yazınız. Bu denklemin analitik çözümünün

$$y(x) = (y_0 + x_0 + 1)e^{(x-x_0)} - (x+1)$$

olduğunu gösteriniz ve sayısal çözümü test etmek için kullanınız.

çözüm: verilen diferansiyel denklemini Euler metodu ile çözecek çok basit bir program EK C'de ADD1 adı ile verilmiştir.

ADD1 programının bazı özellikleri not edilebilir. Program kendi kendini tekrar etmekte ve sadece K için sıfır veya daha küçük bir sayı girildiğinde durmaktadır. (x_0, y_0), (x_1, y_1), (x_2, y_2), $\cdots$, (x_s, y_s) değerlerinin hepsi X ve Y olarak adlandırılmakta, her adımda onların değeri güncelleştirilmektedir. Analitik çözüm YA hesaplanmış ve yapılan hata kaydedilmiştir.

Örnek 4.9 a) ADD1 programına bir önceki paragraftan da yararlanarak açıklayıcı notlar ekleyiniz.

b) ADD1 programını bir dosyaya editörünüzle kaydediniz, fortran derleyicinizi kullanarak derleyiniz ve çalıştırınız. Başlangıç şartları olarak X=0, Y=1 alınız.
i) XS=1 alarak programı K=5, 10, 20, 40 değerleri için çalıştırınız. Yapılan hata nasıl değişiyor? Yaklaşık olarak h ile orantılı mı? K iki kat arttırıldığında hata kaç kat azalıyor?
ii) (i)'yi XS=2 için tekrar ediniz.
iii) örnek 3.5'de gösterildiği gibi bu denklem Euler metoduna göre her zaman kararsızdır. XS'yi büyük seçerek h ne kadar küçük olursa olsun çözümün sürekli büyüdüğünü ve kararsızlaştığını gösteriniz.

Daha genel uygulamalar için ADD1 programı biraz daha geliştirilerek EK C'de verilen ADD2 programı elde edilmiştir. Yeni programda diferansiyel denkleminin türevini hesaplamak için bir alt program, kullanılan yönteme göre integrali almak için bir alt program eklenmiştir. Yeni programı oluşturan kısımlar şunlardır:

- ADD2: ana program
- EULER: herhangi bir diferansiyel denklem için sayısal integrasyonu yapan alt program
- DEQ: dışarıdan sağlanan ve diferansiyel denkleminin türevini (denklemin sağ tarafını) hesaplayan alt program

Ana program girdi-çıktı işlerini ve bazı hesaplamaları yapmaktadır. Başlangıç şartları tekrar tekrar atamayı önlemek üzere X0 ve Y0 değişkenlerinde saklanmaktadır. EULER alt programı verilen denklemden tamamen bağımsızdır. Başka bir diferansiyel denklem çözüleceği zaman sadece DEQ alt programının değiştirilmesi yeterlidir. DEQ'nün EULER alt programının bir argümanı olmasına dikkat ediniz. Bu durum programın en

başında 'EXTERNAL' komutu ile ilan edilmiştir. Daha iyi bir integrasyon yöntemi kullanılacaksa bu durumda EULER alt programının değiştirilmesi gerekir.

Örnek 4.10 a) ADD2 programını bir dosyaya yazarak derleyiniz ve çalıştırınız. Bu programla Örnek 3.6'da verilen diferansiyel denklemini Alıştırma 3.4'de kullanılan aynı başlangıc ve K değerlerini kullanarak çözünüz. Elde ettiğiniz sonuçları ADD1 programının sonuçları ile karşılaştırınız. çözümün hassasiyetinde bir değişme varmıdır?

b) ADD2 programını

$$\frac{dy}{dx} = x^2 + y, \qquad y(0) = 1$$

diferansiyel denklemini çözecek şekilde başka bir ad altında, örneğin ADD21, değiştiriniz. Bu denklemin analitik çözümünü bulup yeni programda bu çözümü kullanmanız gerekir. Analitik çözümün $y(x) = -(x^2 + 2x + 2) + 3e^x$ olması gerektiğini gösteriniz.

ADD2 yapı olarak uygun bir program olmasına karşın biraz daha geliştirilip daha kullanışlı özelliklerle donatılabilir. Bazı değişiklikler şunlar olabilir: (a) girdi sayısının arttırılması ve bu sayede programın daha kullanışlı hale getirilmesi, (b) bazı değişken veya sabitlere baştan değer atayarak aynı değerin defalarca girilmesinin önlenmesi ve (c) çıktıların formatlı yazdırılması.

Bu amaçlarla yazılan yeni programda başlangıç değerleri daha önce olduğu gibi X0 ve Y0'a kaydedilecektir fakat son x değeri olan XS yerine üç tane yeni değişken tanımlanmıstır. Bunlar L, P ve IC'dir. L kaç tane periyot üzerinden hesaplama yapılacağını, P her periyotun uzunluğunu (büyüklüğünü) ve IC hesaplamanın nasıl yapılacağını tanımlamaktadır. Eğer IC=1 ise diferansiyel denkleminin integrasyonunun kaldığı yerden devam edeceğini, IC=0 ise baştan başlayacağını göstermektedir. Programın ilk çalıştırılışında IC=0 olmalıdır. K değeri yeni organizasyonda bir P periyodu içinde atılacak adım sayısını tanımlamaktadır. Yeni tanımlamalara göre toplam integrasyon uzunluğunun L×P olduğuna dikkat ediniz. örnek olarak yukarıdaki diferansiyel denklemi [0,5] aralığında çözülecekse, başlangıçta L=5, P=1 ve IC=0 verilmesi yeterlidir. L ve P için olası başka bir seçenek ise L=10 ve P=0.5'tir. Integrasyona devam etmek için IC=1 verilmesi gerekir. Eğer L ve P için başka değerler girilmezse eski değerler kullanılarak integral almaya devam edilir. örneğin başlangıçta IC=0, L=10, P=0.5 iken, bir sonraki değerler girilirken sadece IC=1 girilip diğerleri değiştirilmeden bırakılırsa integral

kaldığı yerden 0.5 periyotlarla 10 defa alınarak X=10 değerine kadar alınır. Ortaya çıkan yeni program ADD3 adı altında EK C'de verilmiştir. Sadece ana programda değişiklikler yapılmış EULER ve DEQ alt programları değiştirilmeden bırakılmıştır.

Örnek 4.11 a) EK C'de verilen ADD3 progamını derleyerek çalıştırınız. Alıştırma 3.5(a)'da çözülen diferansiyel denklemini aynı başlangıç şartlarını kullanarak yeniden çözünüz. Sonuçları Alıştırma 3.5'in sonuçları ile karşılaştırınız. Ayrıca IC, L ve P parametrelerinin rolünü iyice ögrenmek için programı değişik L ve P değerleri için çalıştırınız.

b) ADD3 programını başka bir ad altında, örneğin ADD31, değiştirerek

$$\frac{dy}{dx} = x^2 + y$$

diferansiyel denklemini $y(0) = 1$ başlangıç şartını kullanarak çozünüz. Sadece DEQ alt programını değiştirmeniz yeterli olacaktır. Ayrıca bu denklemin analitik çözümünü bularak hata hesabında onu kullanmanız gerekir.

Euler metoduna dayanarak şimdiye kadar yaptığımız integral alma işlemleri ne yazık ki yeterli değildir. Pratik olarak bakacak olursak, hatayı azaltmak için K'yı arttırmak daha çok hesaplama dolayısı ile daha çok zaman kaybı demektir. Teknik olarak ise K'nın arttırılması ile hesaplamalarda yapılan yuvarlama hatalarının artması demektir ve bu hataların birikerek yaklaşık çözümde baskın hale gelmesi durumunda, K'nın arttırılması yarardan çok zarar vermeye başlar. Her iki sebepten dolayı daha iyi sonuçlar veren yaklaşık metotları aramakda yarar vadır. Euler metodunu geliştirmenin bir yolu aşağıda göreceğimiz gibi Euler Richardson metodunu kullanmaktır.

4.4 Euler-Richardson Metodu

Bir parametreye bağlı olan herhangi yaklaşık bir yöntemi geliştirmenin bir yolu Richardson ekstrapolasyonudur. Euler metodunda integral almada yapılan hata Δx ile orantılı idi. Adım uzunluğunu yarıya düşürdüğümüzde yapılan hata $\Delta x/2$ ile orantılı olacaktır. Euler metodunda herhangi bir x'den $x + \Delta x$ noktasına kadar yaptığımız integral alma işlemini şimdi iki türlü yapalım. Birincisinde sadece bir adım atarak $x + \Delta x$ noktasında bir çözüm bulalım ve bu yaklaşık çözüme y_1 diyelim. şimdi $x + \Delta x$ noktasına

$\Delta x/2$'lik iki adım atarak ulaşalım ve bu işlem sonucunda elde ettiğimiz yaklaşık çözüme y_2 diyelim. Böylece aynı $x+\Delta x$ noktasına karşılık gelen y_1 ve y_2 gibi iki yaklaşık çözüm elde etmiş olduk.

y_1 ve y_2'nin uygun bir lineer (doğrusal) bileşimini alarak her ikisinden de daha iyi başka yaklaşık bir çözüm elde edebiliriz. Bunu yapmak için denklem (3.16)'ı göz önünde tutarak $h=\Delta x$ olmak üzere $y(x+h)$ ve $y(x+h/2)$'i x civarında, $y((x+h/2))+h/2)$'i $(x+h/2)$ civarında Taylor serisine açalım.

$$y(x+h)=y(x)+hy'(x)+\frac{h^2}{2}y''(x)+\frac{h^3}{6}y'''(x)+\cdots \tag{4.40}$$

$$y(x+h/2)=y(x)+\frac{h}{2}y'(x)+\frac{h^2}{8}y''(x)+\frac{h^3}{48}y'''(x)+\cdots \tag{4.41}$$

$$\begin{aligned} y((x+h/2)+h/2) &= y(x+h/2)+\frac{h}{2}y'(x+h/2)+\frac{h^2}{8}y''(x+h/2) \\ &\quad +\frac{h^3}{48}y'''(x+h/2)+\cdots \end{aligned} \tag{4.42}$$

Denklem (3.41)'i denklem (3.42)'de yerine yazarsak

$$\begin{aligned} y((x+h/2)+h/2) &= y(x)+\frac{h}{2}y'(x)+\frac{h^2}{8}y''(x)+\frac{h}{2}y'(x+h/2)+ \\ &\quad \frac{h^2}{8}y''(x+h/2)+\frac{h^3}{48}y'''(x+h/2)+\cdots+ \\ &= y(x)+\frac{h}{2}[y'(x)+y'(x+h/2)] \\ &\quad +\frac{h^2}{8}[y''(x)+y'(x+h/2)] \\ &\quad +\frac{h^3}{48}[y'''(x)+y'''(x+h/2)]+\cdots \end{aligned} \tag{4.43}$$

elde edilir. Denklem (3.40) ve denklem (3.43) y'nin $(x+h)$ noktasındaki iki ayrı yaklaşık değeridir, bunlardan birincisine y_1 diğerine y_2 demiştik. Göz önüne aldığımız denklemden görüleceği gibi $y'(x)=f(x,y)$ ve $y'(x+h/2)=f(x+h/2,y(x+h/2))$ olacaktır. y'nin x ve $(x+h/2)$'deki bu türevlerine sırası ile S_1 ve S_2 diyelim. $x_h=x+h/2$ ve $y_h=y(x+h/2)$ olmak üzere

$$\begin{aligned} S1 &= f(x,y) \\ S2 &= f(x_h,y_h) \end{aligned} \tag{4.44}$$

yazabiliriz. Böylece denklem (3.40) ve (3.43) sırası ile

$$\begin{aligned} y_1 &= y(x)+hS_1+\frac{h^2}{2}y''(x)+\frac{h^3}{6}y'''(x)+\cdots \\ y_2 &= y(x)+\frac{h}{2}(S_1+S_2)++\frac{h^2}{8}[y''(x)+y''(x+h/2)] \\ &\quad +\frac{h^3}{48}[y'''(x)+y'''(x+h/2)]+\cdots \end{aligned} \tag{4.45}$$

olacaktır.

şimdi y_1 ve y_2'nin lineer birleşiminden, her ikisinden de daha iyi yaklaşık bir çözüm bulmak için $x+h$ noktasındaki yaklaşık çözümü $y=2y_2-y_1$ şeklinde yazalım. Denklem (3.45)'deki ilk köşeli parantez içindeki ikinci terim de Taylor serisine açılıp adı geçen lineer birleşim alınırsa

$$y=y(x)+hS_2+O(h^3) \tag{4.46}$$

elde edilir. Burada $O(h^3)$ terimi, h'nin üç ve daha yüksek kuvvetleri ile orantılı ihmal edilen artık terimleri temsil etmektedir. Burada da görüldüğü gibi Euler-Richardson metodu Euler metodundan bir mertebe daha hassastır. Dolayısı ile bu metotta toplam hatanın seçilen adım uzunluğunun yaklaşık karesi ile orantılı olmasını bekliyoruz.

Bu metot x_0'dan x_s'e kadar bütün bir bölge için değil, her Δx aralığı için ayrı ayrı uygulanmaktadır. Her aralıkta h adım uzunluğu kullanılarak yaklaşık bir çözüm ve iki tane $h/2$ adım uzunluğu kullanılarak başka yaklaşık bir çözüm bulunmaktadır. Buradaki gibi daha düşük mertebeli iki yaklaşık metottan daha yüksek mertebeli bir metot elde etme yöntemi genelde limite ekstrapolasyon yöntemi olarak bilinir. Euler-Richardson yönteminde herhangi bir x noktasından $x+h$ noktasına integral alırken yapılan işlemler aşağıda verilmiştir ve yöntemin algoritmasını oluşturur.

$$\begin{aligned} S1 &= f(x,y) \\ x_h &= x+\frac{h}{2} \\ y_h &= y+\frac{h}{2}*S1 \\ S2 &= f(x_h,y_h) \\ x &= x+h \\ y &= y+h*S2 \end{aligned} \tag{4.47}$$

Burada da görüldüğü gibi Euler metodunu bir mertebe daha hassas yapmak için

ödediğimiz bir bedel vardır. Diferansiyel denkleminin sağ tarafı (çözümün eğimi) şimdi iki ayrı noktada iki defa hesaplanmaktadır.

Bu yeni durumu programlarımıza uygulamak için integral alma yöntemimizi yani EULER alt programını değiştirmemiz gerekir. Bunların dışında sadece bazı ufak değişiklikler yapmak yeterlidir. Yeni durumların uygulandığı program ADD4 adı altında EK C'de verilmiştir.

Örnek 4.12 a) Alıştırma 3.5'da çözülen diferansiyel denklemini orada verilen aynı başlangıç ve parametre değerleri ile fakat program ADD4'ü kullanarak çözünüz. çözümdeki hata gerçekten h^2 ile orantılı mıdır?

b) ADD4 programını başka bir ad altında, örneğin ADD41, değiştirerek

$$\frac{dy}{dt} = x^2 + y$$

diferansiyel denklemini y(0)=1 başlangıç şartını kullanarak çözünüz. Bunun için sadece DEQ alt programını değiştirmeniz yeterlidir. Bulduğunuz sonuçları Alıştırma 3.5.b ile karşılaştırınız. Hangi yöntem daha hassastır?

4.5 Runge-Kutta Metodu

Runge-Kutta metodunun gerisinde yatan temel fikir aslında yukarıda gördüğümüz Euler-Richardson metodunda uygulanan fikir ile aynıdır. Runge-Kutta metotlarında diferansiyel denkleminin türevi ($f(x,y)$) her aralık içinde belirli noktalarda hesaplanarak çok daha yüksek hassasiyet sağlamaya çalışılır. şimdi çok basit ikinci mertebe hassasiyete sahip iki adımlı Runge-Kutta yöntemini inceleyelim.

4.5.1 İki-Adımlı Ruge-Kutta Metodu

$$\frac{dy}{dx} = f(x,y) \tag{4.48}$$

diferansiyel denklemi verilmiş olsun. Denklem (3.28)'de verilen tanımları kullanarak $x_{n+1} = x_n + h$ noktasındaki yaklaşık çözümü k_1 ve k_2 gibi iki fonksiyonun lineer bir

birleşimi olarak

$$y_{n+1} = y_n + ak_1 + bk_2 \tag{4.49}$$

şeklinde yazalım. Burada k_1 ve k_2

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \alpha h, y_n + \beta k_1) \end{aligned} \tag{4.50}$$

olarak aralığın başlangıçtaki ve aralık içinde en yüksek hassasiyeti sağlayacak bir noktadaki eğim olarak tanımlanmıştır. a, b, α ve β henüz bilinmeyen sabitlerdir. Bu sabitler $y(x+h)$'in Taylor açılımı ile (3.49)'daki terimler mümkün olan en yüksek mertebeden uyuşacak şekilde seçileceklerdir.

$y(x+h)$ Taylor serisine açılırsa

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \ldots \tag{4.51}$$

elde edilir. Dikkat edilirse y'nin türevleri $f(x,y)$ ve onun x ve y'e göre kısmi türevleri cinsinden yazılabilir. Bunu yukarıdaki denkleme uygularsak, f_x ve f_y sırası ile $f(x,y)$'in x ve y'ye göre kısmi türevleri olmak üzere

$$\begin{aligned} y' &= f(x,y) \\ y'' &= \frac{df(x,y)}{dx} = f_x + f_y y' = f_x + f_y y \\ y''' &= \frac{d^2 f(x,y)}{dx^2} = f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_x f_y + f_y^2 f \end{aligned} \tag{4.52}$$

elde edilir. Yukarıdaki denklemi çıkarma işlemi ödev olarak okuyucuya bırakılmıştır. Böylece Taylor açılımı $f(x,y)$ ve onun kısmi türevleri cinsinden

$$\begin{aligned} y(x_{n+1}) &= y(x_n) + hf(x_n, y_n) + \frac{h^2}{2}[f_x + ff_y]_n \\ &\quad + \frac{h^3}{6}[f_{xx} + 2ff_{xy} + f_{yy}f^2 + f_x f_y + f_y^2 f]_n \\ &\quad + O(h^4) \end{aligned} \tag{4.53}$$

olur. Burada alt indis n bütün değerlerin (x_n, y_n) noktasında hesaplanması gerektiğini gösterir.

ıki değişkenli fonksiyonlar için Taylor açılımını kullanarak k_2

$$\begin{aligned} \frac{k_2}{h} = f(x_n + \alpha h, y_n + \beta k_1) &= f(x_n, y_n) + \alpha h f_x + \beta k_1 f_y + \frac{\alpha^2 h^2}{2} f_{xx} + \\ &\quad \alpha h \beta k_1 f_{xy} + \frac{\beta^2 k_1^2}{2} f_{yy} + O(h^3) \end{aligned} \tag{4.54}$$

şeklinde yazılabilir. Burada da bütün türevlerin (x_n, y_n) noktasında hesaplanması gerekir.

$k_1 = hf(x_n, y_n)$ olduğu gerçeğini kullanarak k_2 için elde edilen değer denklem (3.49)'da yerine yazılır ve h'nin aynı kuvvetli katsayıları yeniden düzenlenirse

$$\begin{aligned} y_{n+1} &= y_n + (a+b)hf + bh^2[\alpha f_x + \beta f f_y] \\ &\quad + bh^3[\frac{\alpha^2}{2} f_{xx} + \alpha\beta f f_{xy} + \frac{\beta^2}{2} f^2 f_{yy}] + O(h^4) \end{aligned} \tag{4.55}$$

elde edilir. Bunun denklem (3.53) ile karşılaştırılarak h ve h^2'nin katsayılarının eşitlenmesi ile

$$\begin{aligned} a+b &= 1 \\ b\alpha = b\beta &= \frac{1}{2} \end{aligned} \tag{4.56}$$

elde edilir. Burada üç tane denklem fakat dört tane bilinmeyen vardır. Dolayısı ile parametrelerin seçiminde bir derece serbestlik olanağımız olacaktır. Bir çok olası çözüm arasından aşağıda verilenleri göz önüne alalım.

a) $a = 0$ seçilirse, bu durumda $b = 1$ ve $\alpha = \beta = \frac{1}{2}$ olur. Parametrelerin bu değerleri için Runge-Kutta denklemleri daha önce elde ettiğimiz Euler-Richardson denklemleri (3.47) ile tamamen aynı olur.

b) Başka olası bir çözüm kümesi $a = b = \frac{1}{2}$ ve $\alpha = \beta = 1$ olacaktır.

ıki adımlı Runge-Kutta yöntemini uygulamak için ADD4 programı değiştirilmiş ve yeni program ADD5 olarak adlandırılmıştır. Bu programın bir kopyesi EK C'de verilmiştir. Her adım için burada iki defa eğim hesaplanması gerekmektedir. ERICS alt programı RK2 ile değiştirilmiştir.

Örnek 4.13 a) ADD5 programını kullanarak Alıştırma 3.5'de verilen diferansiyel denklemini çözünüz ve sonuçları ADD4 programının sonuçları ile karşılaştırınız.

b) ADD5 programında a, b, α and β parametreleri için tutarlı farklı değerler kullanarak (a)'yı yeniden çözünüz. çözümde ve hassasiyet mertebesinde önemli bir değişiklik meydana geliyormu?

4.5.2 Dört Adımlı Runge-Kutta Metodu

ıki adımlı Runge-Kutta (RK) metodunun çıkarılışında izlenen yol kullanılarak dört adımlı daha hassas yöntemde elde edilebilir. Dört adımlı Runge-Kutta metodunda

(3.16)'da verilen diferansiyel denkleminin çözümünün $x_{n+1} = x_n + h$ noktasındaki yaklaşık değerini

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4.57}$$

şeklinde yazıyoruz. Burada k_i'ler

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_1) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_2) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned} \tag{4.58}$$

olarak tanımlanmıştır. Yukarıda verilen yöntemde her adımda yapılan yerel hata $O(h^5)$ mertebesindedir. Diğer yöntemlere göre hata çok daha küçük fakat bunun için ödenen bedel her adımda $f(x,y)$ fonksiyonunun dört defa hesaplanmasıdır. Dikkat edilirse her mertebeden Runge-Kutta metodu çıkartmak her zaman olasıdır fakat hesaplamalar çok karmaşık ve uzun olabilir.

Örnek 4.14 ADD5 programını başka bir ad altında, örneğin ADD51, değiştirerek yukarıda verilen dört adımlı Runge-Kutta algoritmasını programlayınız. Alıştırma 3.5'de verilen problemi yazdığınız bu programı kullanarak çözünüz. Sonuçlarınızı daha önce elde ettiğiniz bütün sonuçlar ile karşılaştırınız. En hassas olan yöntem hangisidir?

şimdiye kadar kullandığımız bütün programlar ve alt programlar sadece bir diferansiyel denkleminin çözümü için yazılmıştır. N tane denklemden oluşan bir diferansiyel denklem sistemini çözmek için değişkenlerin vektörler şeklinde yazılması gerekir. Yani gerekli değişkenler boyutlandırılmalıdır. Boyutlandırmaların nasıl yapılabileceğini göstermek üzere örnek (3.3)'de elde edilen (3.15)'deki denklem sisteminin çözülmesi için ADD4 programı ADD6 adı altında yeniden düzenlenmiş ve EK C'de verilmiştir.

Yeni programdaki bazı değişiklikleri not edelim. Toplam dört tane birinci mertebe diferansiyel denklem olduğu için 'PARAMETER' komutunda N=4 olarak verilmiş ve bu değer daha sonra boyutlu değişkenlerin boyutlarının tanımlanmasında kullanılmıştır. Başlangıç değerleri Y0 adlı vektörde tutulmuş, parçacığın x ve y yönündeki koordinatlarına sırası ile Y(1) ve Y(2), x ve y yönündeki hızlarına ise sırası ile Y(3) ve Y(4) adı verilmiştir. S1, S2 ve S3 vektörleri fonksiyon değerlerini hesaplayıp tutmak ve ara

Tablo 4.2 ADD6 programının örnek bir çıktısı

G= 1.0,	CT= 2000.				
T	X	Y	Vx	Vy	DX
0.000E+00	1.000E+00	0.000E+00	0.000E+00	5.000E-01	2.714E-03
2.714E-01	9.629E-01	1.340E-01	-2.757E-01	4.809E-01	2.714E-03
5.428E-01	8.478E-01	2.568E-01	-5.798E-01	4.141E-01	2.714E-03
8.142E-01	6.415E-01	3.507E-01	-9.595E-01	2.548E-01	2.714E-03
1.086E+00	3.102E-01	3.698E-01	-1.532E+00	-2.147E-01	2.714E-03
1.357E+00	-1.430E-01	1.279E-04	-4.287E-03	-3.498E+00	2.714E-03
1.628E+00	3.090E-01	-3.709E-01	1.531E+00	-2.199E-01	2.714E-03
1.900E+00	6.404E-01	-3.530E-01	9.604E-01	2.514E-01	2.714E-03
2.171E+00	8.471E-01	-2.598E-01	5.815E-01	4.119E-01	2.714E-03
2.443E+00	9.627E-01	-1.375E-01	2.777E-01	4.797E-01	2.714E-03
2.714E+00	1.000E+00	-3.679E-03	2.312E-03	4.998E-01	2.714E-03

işlemleri yapmak için kullanılmaktadır. Programda tanımlanan CT değişkeni DEQ alt programının kaç defa çağrıldığını yani kaç defa fonksiyon değer hesaplaması yapıldığını kaydetmektedir. Programı yakından inceleyerek diğer değişiklileri sizin bulmanız ve anlamaya çalışmanız yararlı olacaktır.

Örnek 4.15 ADD6 programını program içinde verilen $x_0 = 1$, $y_0 = 0$, $V_x = 0$ ve $V_y = 0.5$ başlangıç şartlarını kullanarak derleyip çalıştırınız. Bu başlangıç şartları Y0'lar cinsinden Y0(1)=1, Y0(2)=0, Y0(3)=0 ve Y0(4)=0.5 olacaktır.

ADD6 programında verilen toplam integrasyon süresi L×P = 2.714080941 bu parçacığın elips olan yörüngesini tamamlaması için gereken süreye, yörünge periyoduna eşittir. Bu nedenle bir periyot sonunda parçacık başladığı yere dönmeli ve buraya ulaştığı andaki hızları ile başlangıçtaki hızları eşit olmalıdır. Ayrıca parçacığın toplam enerjiside korunmalıdır (neden?). Bazı problemlerdeki bunlara benzer zamanla değişmeyen hareket sabitleri, kullanılan sayısal yöntemi test etmek için kullanılabilecek çok önemli parametrelerdir. Göz önüne aldığımız örnekte yörüngenin kapalı elips olması nedeni ile parçacık başladığı yere dönmelidir. Ayrıca başlangıç noktasına döndğü andaki hızları ilk hızlarına eşit olmalıdır. Bu programın bir çıktısı Tablo 3.2'de verilmiştir. Bu tabloyu inceleyerek yukarıda anlatılan özelliklerin sağlanıp sağlanmadığını kontrol ediniz.

Örnek 4.16 Alıştırma 3.14'de yazdığınız dört adımlı Runge-Kutta metodunu uygulayan ADD51 adlı programı ADD52 adı altında denklem (3.15)'de verilen kütle çekim prob-

lemini çözmek için yeniden düzenleyiniz. ADD6 ve ADD52 programlarının sonuçlarını karşılaştırınız. Hangisi daha hassas ve hangisi DEQ alt programını en az sayıda çağırmaktadır?

4.6 Adım Uzunluğu Kontrol Edilebilen Metotlar

Kütle çekim probleminin Tablo 3.2'de gösterilen çözümü dikkatlice incelendiğinde eliptik yörüngede dolaşan parçacığın T=1.357 olduğu zamanda (periyodun yarısında) merkeze en yakın ve hızının en yüksek olduğu görülecektir. Sayısal çözümde bu bölgede yapılan hatayı azaltmak ve yörüngeyi tam takip edebilmek için Δx adım uzunluğunun bu bölgede gerçekten küçük olması gerekir. Adım uzunluğunun yeniden ayarlanması için program durdurulup yeni bir adım uzunluğu verilerek yeniden başlatılabilir. Fakat yeni adımın ne zaman ve hangi büyüklükte seçilmesi gerektiği açık değildir. Bunun tam olarak bilinmesi için çözümün biliniyor olması gerekir ki buda mümkün değildir. Bu nedenle adım uzunluğunu çözümün akışı içerisinde dinamik olarak belirlemenin uygun bir yolunu bulmamız gerekir. Her aralıkta yapılan hesaplama hatası herhangi bir yolla tahmin edilebilirse, buna dayanarak adım uzunluğuda dinamik olarak seçilebilir. Bunu yapmanın bir yolu Euler-Richardson metodunda gördüğümüz gibi herhangi bir x_n noktasından $x_{n+1} = x_n + h$ noktasına integrasyonu iki ayrı şekilde gerçekleştirerek x_{n+1} noktasındaki $y(x)$ için iki ayrı yaklaşık değer bulmaktır. Bu yaklaşık değerlerden biri Δx adım uzunluğu, diğeri ise iki tane $\Delta x/2$ adım uzunluğu kullanılarak bulunabilir. Bu iki yaklaşık değerin karşılaştırılması adım başına yapılan yerel hatanın mertebesi hakkında bir fikir verir. Yerel olarak yapılan bu hata kullanıcı tarafından belirlenen bir hata tolerans parametresi ile karşılaştırılarak adım uzunluğu dinamik olarak çözümün akışı içerisinde belirlenebilir. Böylece sabit bir adım uzunluğu kullanmak yerine çözümün özelliklerine uygun ve onunla uyumlu olarak değişen adım uzunluğu kullanılmış olur. Bunun sağladığı bir çok fayda vardır.

Burada anlatılan yöntem daha önce kullandığımız Euler-Richardson yöntemine kolayca uygulanabilir. Yukarıda anlatıldığı gibi her aralık için iki yaklaşık çözüm bulunup bunların farkından yapılan hata tahmini HT elde edilir. Bu hata kullanıcı tarafından sağlanan hata toleransı HTOL ile karşılaştırılır ve adım uzunluğu hakkında bir karar verilir. üç olası durum mevcuttur:

i) Eğer HT, HTOL'den büyük ise gerekli tolerans sağlanana kadar adım uzunluğu yarılanır,

ii) Eğer HT yeterince küçük ise bu aralıktaki integrasyon etkin olan adım uzunluğu kullanılarak bitirilir,

iii) Eğer HT, HTOL'den çok çok küçük ise bu durumda adım uzunluğu arttırılır.

Böylece alınan integralin değişimine göre dinamik olarak uyarlanan bir adım uzunluğu olan bir program algoritması elde edilmiş oldu. Ayrıca birden fazla diferansiyel denkleminin çözüldüğü durumlarda her değişken için ağırlıklı bir hata payı AH tanımlayarak her değişken farklı bir hata toleransı ile hesaplanabilir.

EK C'de verilen ADD7 programı ADD6 programının adım kontrolü uygulanmış halidir. Yani adım uzunluğu kontrollü bir Euler-Richardson yöntemidir. Ayrıca her değişken için, örneğin *j*. değişken için ağırlıklı bir hata payı AH(J) tanımlanıp, o değişken için hata toleransı HTOL'ün AH(J) ile çarpılmasından elde edilmektedir. Böylece her değişken için ayrı bir göreli hata toleransı tanımlamak mümkündür.

Örnek 4.17 ADD7 programını çalıştırarak kütle çekim problemini yeniden çözünüz. Bu programın bir çıktısı Tablo 3.3'de gösterilmiştir.

Tablo 3.3'de verilen program çıktısı incelendiğinde DEQ alt programının sadece 466 kez çağrıldığı görülecektir. Adım kontrolünün olmadığı ADD6 programı için bu değer 2000 idi. Demek ki fonksiyon hesaplama sayısı yaklaşık dört kez daha az. Buna karşılık elde edilen sonuçlar daha hassas. Adım uzunluğunun sürekli değişmesine, Δx'in $t = 1.357$ civarında en küçük değerini almış olmasına özellikle dikkat ediniz.

4.6.1 Adım Uzunluğu Kontrollü RK-Verner Metodu

Birinci mertebe diferansiyel denklemleri sayısal olarak çözmek için adım kontrollü ileri düzeyde bir çok yöntem geliştirilmiştir. Runge-Kutta yöntemlerine de adım kontrolü sağlayan yeni metotlar eklenmiştir. Bunlardan çok uygun olan bir yöntem Runge-Kutta-Verner metodudur. Diğer benzer metotlar RK-Merson ve RK-Fehlberg metotlarıdır. RK-Verner algoritmasının detayları burada verilmeyecektir. Sadece bu algoritmaya dayalı olarak yazılan ve adım kontrolü sağlayan bir alt program verilmiştir. Euler-Richardson

Tablo 4.3 ADD7 programının örnek bir çıktısı

HTOL=	1.000E-03	MS=	1.000E+03		
AH=	1.000E+00	1.000E+00	1.000E+00	1.000E+00	
G=	1.0	CT=	466.		
T	X	Y	Vx	Vy	DX
0.000E+00	1.000E+00	0.000E+00	0.000E+00	5.000E-01	1.000E-02
2.714E-01	9.629E-01	1.340E-01	-2.755E-01	4.809E-01	5.647E-02
5.428E-01	8.480E-01	2.570E-01	-5.793E-01	4.142E-01	4.817E-02
8.142E-01	6.420E-01	3.511E-01	-9.583E-01	2.553E-01	3.328E-02
1.086E+00	3.112E-01	3.705E-01	-1.529E+00	-2.122E-01	1.100E-02
1.357E+00	-1.434E-01	4.918E-03	-6.990E-02	-3.491E+00	1.249E-03
1.628E+00	3.063E-01	-3.705E-01	1.536E+00	-2.242E-01	1.051E-02
1.900E+00	6.387E-01	-3.531E-01	9.634E-01	2.510E-01	2.299E-02
2.171E+00	8.460E-01	-2.598E-01	5.836E-01	4.121E-01	4.307E-02
2.443E+00	9.620E-01	-1.373E-01	2.795E-01	4.801E-01	5.307E-02
2.714E+00	1.000E+00	-3.321E-03	4.024E-03	5.001E-01	4.526E-02

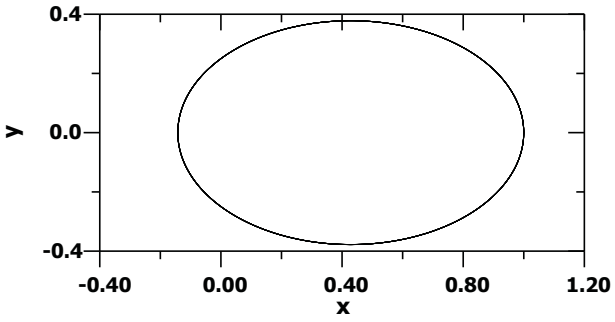
metodunda aralık başına iki türev değeri hesaplanmakta, birinci mertebe bir hata tahmini yapılmakta ve ikinci mertebeden hatalı bir çözüm üretilmektedir. Benzer şekilde RK-Verner metodunda aralık başına sekiz türev değeri hesaplanmakta, beşinci mertebe bir hata tahmini yapılmakta ve altıncı mertebeden hatalı bir çözüm üretilmektedir. Bu yöntemde $10^{-6} - 10^{-11}$ mertebesinde hata toleransları kullanmak hiç sorun yaratmayacaktır. Adım uzunluğu kontrollü Euler-Richardson metodunun uygulandığı ADD7 programı RKV metodu uygulanacak şekilde yeniden düzenlenmiş ve ortaya çıkan yeni program ADD8 adı ile EK C'de verilmiştir.

Örnek 4.18 ADD8 programını çalıştırarak kütle çekim problemini yeniden çözünüz. Bu programın bir çıktısı Tablo 3.4'de gösterilmiştir. Bu sonuçları Tablo 3.3'de gösterilen program ADD7'nin sonuçları ile karşılaştırınız. Hangisi daha hassas? Hangi programda fonksiyon hesaplaması en az?

Tablo 3.4'de görüldüğü gibi şimdi kullanılan hata toleransına uygun olarak yapılan hatalar 10^{-6} mertebesindedir. Adım uzunluğu parçacığın merkeze en yakın ve en hızlı olduğu anda en küçük değeri almıştır. Bu programla elde edilen yörüngenin grafiği şekil 3.2'de gösterilmiştir. Toplam integral süresi 10 periyota eşittir. çizimde (x, y) noktalarının arasındaki uzaklığın küçük olmasını sağlamak üzere P küçük seçilerek daha çok veri noktası elde edilmiştir. Hatanın en az olduğu bir hesaplamada yörüngenin sürekli elips olarak

Tablo 4.4 ADD8 programının örnek bir çıktısı

HTOL=	1.000E-06	MS=	1.000E+03		
AH=	1.000E+00	1.000E+00	1.000E+00	1.000E+00	
G= 1.0	CT=	424.			
T	X	Y	Vx	Vy	DX
0.000E+00	1.000E+00	0.000E+00	0.000E+00	5.000E-01	1.000E-02
2.714E-01	9.629E-01	1.340E-01	-2.757E-01	4.809E-01	1.420E-01
5.428E-01	8.478E-01	2.568E-01	-5.798E-01	4.141E-01	2.414E-01
8.142E-01	6.415E-01	3.507E-01	-9.595E-01	2.548E-01	2.172E-01
1.086E+00	3.101E-01	3.698E-01	-1.532E+00	-2.147E-01	1.271E-01
1.357E+00	-1.429E-01	-3.578E-07	-1.991E-06	-3.500E+00	1.040E-02
1.628E+00	3.101E-01	-3.698E-01	1.532E+00	-2.147E-01	6.875E-02
1.900E+00	6.415E-01	-3.508E-01	9.595E-01	2.548E-01	1.162E-01
2.171E+00	8.478E-01	-2.568E-01	5.798E-01	4.141E-01	1.964E-01
2.443E+00	9.629E-01	-1.340E-01	2.757E-01	4.809E-01	1.767E-01
2.714E+00	1.000E+00	-1.360E-05	9.157E-06	5.000E-01	2.297E-01



Şekil 4.2 (3.15)'de verilen kütle çekim probleminde parçacığın 10 periyot boyunca takip ettiği yörünge. Parçacığın yörüngesinin sabit kalması çözümün iyi olduğuna işaret eder.

kalması gerekir.

Bu bölümde gördüğümüz en hassas program RKV metoduna dayalı ADD8 programıdır. Bundan sonra çözdüğünüz başlangıç şartlı problemlerin hepsinde sadece RKV alt programını kullanmanız hem daha hızlı hem de daha hassas sonuçlar elde etme açısından daha uygun olacaktır.

Örnek 4.19 Denklem (3.7)'de verilen basit harmonik salınıcı denklemini $t_0 = 0$'da, $x(0) = 0$ ve $\dot{x}(0) = 1$ başlangıç şartlarını kullanarak çözmek için ADD8 programında gerekli değişiklikleri yapınız. Yeni programı ADD81 olarak adlandırınız. Analitik çözümüde bularak sayısal çözümü test etmek için kullanınız. HTOL=1×10^{-6} seçiniz. $m = w = 1$ olduğunu kabul ediniz.

a) ADD81 programında L=600, P=0.1 seçerek elde ettiğiniz x ve $\dot{x}$ değerlerini za-

manın (t) bir fonksiyonu olarak çiziniz. şekil 3.3'de gösterilen grafiği elde etmeniz gerekir.

b) Analitik çözüm ile sayısal çözüm arasındaki farkı (yapılan hatayı) zamanının (t) bir fonksiyonu olarak çiziniz. şekil 3.4'de gösterilen sonuçları bulmanız gerekir. Hatanın çok küçük de olsa salınım yapıyor ve gittikçe artıyor olmasına dikkat ediniz.

c) Basit harmonik salınıcının enerjisi

$$E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}mw^2x^2$$

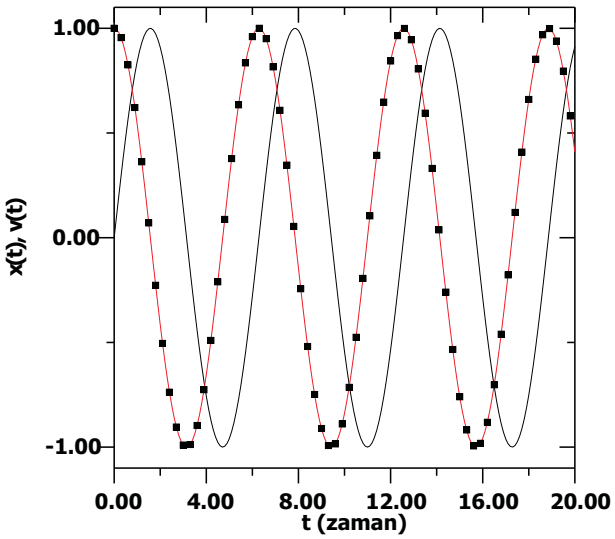
ile verilir. $m = w = 1$ olduğu varsayılırsa yukarıdaki enerji denklemi yarıçapı $\sqrt{2E}$ olan bir daire denklemi olacaktır. $\dot{x}$'i x'e karşılık çizerek daire elde edildiğini gösteriniz (Bu aynı zamanda faz uzayı olarak da bilinir). Bu grafiği en az 10 periyot için çizerek sonucun daireden ne kadar uzaklaştığını görmeye çalışınız. şekil 3.5'de gösterilen sonuca benzer bir grafik elde etmeniz gerekir.

4.7 Stif Problemler

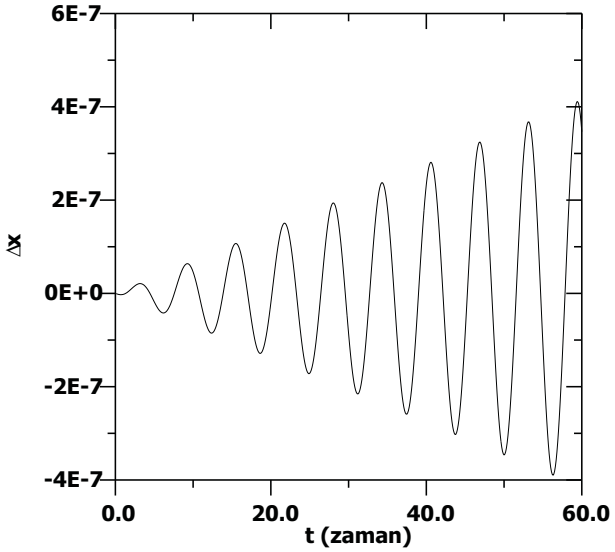
çözümünün bağımsız değişkene bağımlılığı çok farklı ölçeklerde eksponansiyel terimler içeren diferansiyel denklemlerine stif denklemler denir. Bir diferansiyel denkleminin çözümünün e^{-x} ile e^{-50x} terimlerinin lineer bir kombinasyonu ile verildiğini varsayalım. Bu iki terimin bağımsız değişkene bağımlılıkları çok farklı ölçeklerdedir. Bu çözümlerden biri çok hızlı şekilde azalırken diğeri uzun bir değer aralığı için (diğeri ile karşılaştırıldığında) sonlu kalacaktır. çok farklı boşalma zamanları olan kapasitörler bu yapıya uygun örneklerdir. Daha ayrıntılı bir inceleme için aşağıdaki diferansiyel denklemini göz önüne alalım.

$$\frac{d^2y}{dx^2} - 100y = 0 \tag{4.59}$$

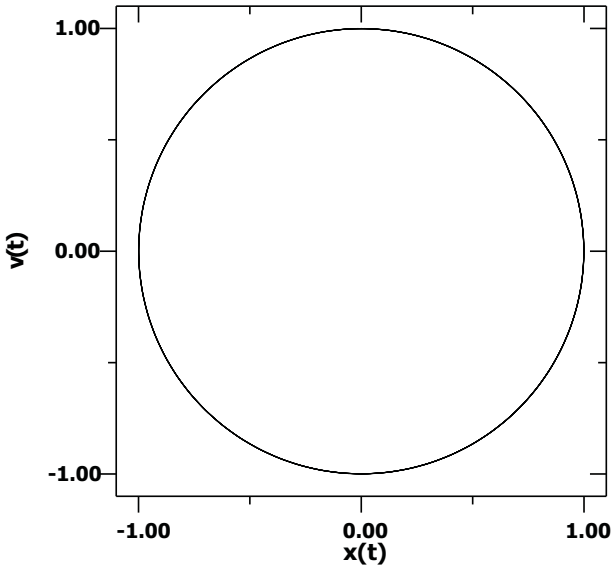
Bu denklemin genel çözümü, c_1 ve c_2 keyfi sabitlar olmak üzere, $y = c_1e^{-10x} + c_2e^{10x}$ ile verilir. $y(0) = 1$ ve $y'(0) = -10$ başlangıç şartları kullanılırsa $c_1 = 1, c_2 = 0$ olduğu gösterilebilir. Bu durumda analitik çözüm $y(x) = e^{-10x}$ olacaktır. Bu problemi sayısal olarak çözmeye çalıstığımızda, hızla artan e^{10x} terimine ait en ufak bir bileşenin sayısal çözüme bulaşması halinde bu bileşen hızla büyüyecek ve kısa sürede bütün çözümü etkisi altına alacaktır. Bu durumda problemin aslında çözümü olan e^{-x}'den hızla uzaklaşılacak



Şekil 4.3 (3.7)'de verilen basit harmonik salınıcı probleminin sayısal çözümünden bulunan parçacığın yerdeğişim ve hızının (kareler) zamanın bir fonksiyonu olarak değişimi.



Şekil 4.4 şekil 3.3'de gösterilen harmonik salınıcının x koordinatında yapılan hatanın zamanla değişimi.



Şekil 4.5 Şekil 3.3'de gösterilen harmonik salınıcının $v(t)$-$x(t)$ grafiği.

ve çözüm ıraksıyacaktır. Bu tip problemlerin çözümünda Runge-Kutta metotlarının hemen hepsinin çok kötü sonuçlar verdiği bilinmektedir. Bu nedenle stif problemler RK yöntemleri ile hiç bir zaman çözülmemelidir. Bunun yerine çok adımlı yöntemler temkinli şekilde kullanılabilir.

Örnek 4.20 ADD8 programını (3.59)'da verilen diferansiyel denklemini çözecek şekilde ADD82 adı altında yeniden düzenleyiniz. Sayısal ve analitik çözümün grafiklerini beraber çizerek sayısal çözümün hızla gerçek çözümden uzaklaştığını gözleyiniz. Farklı L ve P değerleri kullanarak sayısal çözümü gerçek çözümden fazla uzaklaşmadan ne kadar sürdürebildiğinize bakınız.

4.8 Sayısal Çözümler İçin Bazı Öneriler

Diferansiyel denklemlerini sayısal olarak çözmeye başlamadan önce ve çözüm elde edildikten sonra aşağıda belirtilen konuların mümkün olduğu yerlerde göz önüne alınması gerekir. Böylece hem denklem çözme işi kolaylaşacak hemde elde edilen çözümlerin gerçekten aranan çözüm olup olmadığı konusunda karar vermek daha kolay olacaktır.

a) Sadece belirli parametre değerlerinde veya bazı limitlerde geçerli bile olsa diferansiyel denklemlerin analitik çözümleri her zaman faydalıdır. Analitik çözümler çeşitli şartlar altında bulunabiliyorsa mutlaka elde edilmeli ve sayısal çözümün kontrol

edilmesinde kullanılmalıdır.

b) Diferansiyel denkleminde bulunan sabitler mümkün olduğunca sadeleştirilerek en az sayıya indirilmelidir. Bu zaman kazandıracak, bir çok parametre kullanmaktan doğan karmaşıklığı önleyecek ve sonuçların yorumlanmasını kolaylaştıracaktır.

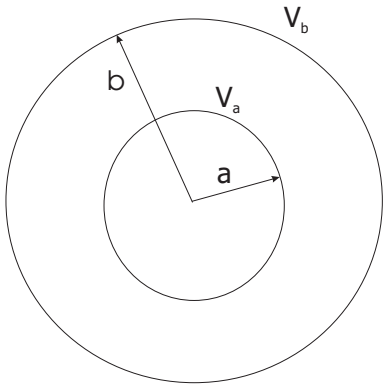
c) Mümkün olan durumlarda değişkenler yeniden ölçeklenerek boyutsuz hale getirilmelidir.

d) çözümde kullanılacak birimlere çözümden önce karar verilmelidir. Uygun olmayan birimlerin kullanılması durumunda yazılan program hiç çalışmayabilir. örneğin galaksilerin dinamiğini veya kuantum mekaniği problemlerini çalışmak üzere SI birimlerini kullanmak felaketle sonuçlanabilir. Galaksi problemlerinde R^3 gibi terimler kolayca ortaya çıkabilir ve bu sayı kullanılan makinede temsil edilebilecek en büyük sayıdan daha büyük olabilir. Benzer şekilde kuantum mekaniği problemlerinde $\hbar^4$ gibi terimler ortaya çıkabilir ve bu sayı makinede temsil edilebilecek en küçük sayıdan daha küçük olabilir. Bu durumda makine bu sayıyı sıfır olarak alacak ve elde edilen sonuçlar anlamsız olacaktır.

Göz önüne alınan problemin çoğunlukla kendi doğal birimleri vardır ve bu birimlerde çalışmak daha akıllıcadır. Doğal birimler kullanıldığında değişkenler çok büyük veya çok küçük olmaktansa çoğunlukla 1 mertebesindedir. Bir problemin doğal birimleri çoğunlukla ne SI nede cgs'dir.

e) Elde edilen sayısal çözümler mutlaka çeşitli yöntemlerle test edilmelidir. Kontrol için kullanılanlar çoğunlukla çözülen problemin matematiksel olmaktan ziyade fiziksel olarak sağlaması gereken şartlardır. örneğin problemde korunması gereken değerler varsa (enerji, momentum vb.) bunların korunup korunmadığı kontrol edilmelidir. Problemin analitik çözümü çeşitli limitlerde biliniyorsa, sayısal çözüm ile analitik çözüm bu limitlerde karşılaştırılmalıdır.

Yukarıdaki önerilerin bir kısmını uygulayabileceğimiz bir örnek göz önüne alalım. Elektrik ve manyetik alanlar altında hareket eden yüklü bir parçacığın yörüngesini ADD8 programını uygun şekilde değiştirerek bulmaya çalışalım.



Şekil 4.6 Sırası ile V_a ve V_b potansiyellerinde tutulan a ve b yarıçaplı aynı merkezli iki silindirik elektrotun üstten görünüşü.

4.8.1 Örnek: Radyal Elektrik Alanı Altında Hareket Eden Yüklü Bir Parçacık

şekil 3.6'da gösterildiği gibi yarıcapları a ve b ($b > a$) olan iki silindirik elektrot sırası ile V_a ve V_b potansiyellerinde tutulursa, bunların arasında

$$E = \frac{K}{r}, \qquad K = \frac{V_a - V_b}{log(b/a)} \tag{4.60}$$

ile verilen radyal bir elektrik alanı oluşacaktır. K sabiti V_a ve V_b'nin göreli değerlerine göre pozitif veya negatif olabilir. Q yüklü, m kütleli bir parçacığın bu alan altındaki hareket denklemi, **a** ivme olmak üzere, $Q\mathbf{E} = m\mathbf{a}$ olacaktır. Parçacığın yükünün negatif olduğu ($Q = -q$, $q > 0$) ve hareketin tamamen silindirik elektrotlara dik iki boyutlu bir düzlem üzerinde olduğu varsayılarak hareket denklemi

$$\frac{d^2\mathbf{r}}{dt^2} = -(\frac{qK}{m})\frac{\hat{\mathbf{r}}}{r} = -(\frac{qK}{m})\frac{\mathbf{r}}{r^2} \tag{4.61}$$

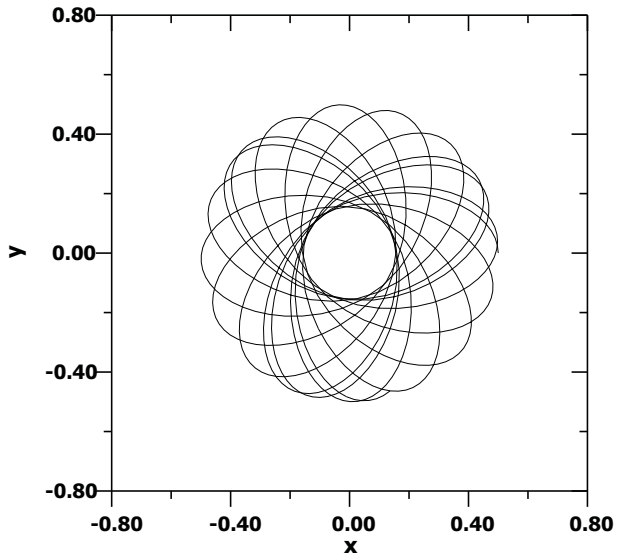
olarak yazılabilir.

a) Problemde geçen en büyük uzunluk dışta bulunan silindirin yarıçapı olacağından uzunlukları b'ye bölerek ölçeklendirelim. Yeni boyutsuz uzunlukları $r' = r/b$ ile gösterelim. Bu durumda (3.61) denklemi

$$\frac{d^2\mathbf{r}'}{dt^2} = -(\frac{qK}{mb^2})\frac{\mathbf{r}'}{r'^2} = -\alpha\frac{\mathbf{r}'}{r'^2} \tag{4.62}$$

olarak yeniden yazılabilir. Burada $\alpha = \frac{qK}{mb^2}$'dır.

b) α'nın biriminin zamanın karesinin tersi olduğunu gösteriniz.



Şekil 4.7 Sırası ile V_a ve V_b potansiyellerinde tutulan a ve b yarıçaplı aynı merkezli iki silindirik elektrotun arasında hareket eden yüklü bir parçacığın yörüngesi.

c) (3.62)'deki zamanı $t' = \sqrt{\alpha}t$ şeklinde yeniden ölçeklendiriniz ve ortaya çıkan denklemin

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mathbf{r}}{r^2} \tag{4.63}$$

olduğunu gösteriniz. Bu son denklemde karmaşıklığı önlemek için değişkenlerin üsleri yazılmamıştır ve görüldüğü gibi denklemde hiç bir parametre kalmamıştır. Dikkat edilirse bu denklem daha önce örnek 3.3'de gördüğümüz kütle çekim problemindeki denklemlere çok benzemektedir.

d) (3.63)'deki ikinci mertebe diferansiyel denklemini $\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$ kullanarak iki boyutlu (x, y) düzleminde dört tane birinci mertebe diferansiyel denkleme indirgeyiniz. (3.15)'de elde edilen denklemlere benzer denklemler elde etmeniz gerekir.

e) Ortaya çıkan denklemleri ADD8 programını uygun şekilde değiştirerek $x = 0.5$, $y = 0$, $V_x = 0$ and $V_y = 0.5$ başlangıç şartlarını kullanarak $t \in [0, 30]$ çözünüz. Sonuçlarınızı (t, x, y, V_x, V_y) bir dosyaya yazdırınız ve y-x grafiğini çizerek parçacığın izlediği yörüngeyi görünüz. Ne elde ettiniz? Elde etmeniz gereken yöründe şekil 3.7'de gösterilmiştir.

f) (e)'yi $t \in [0, 120]$ için tekrar ediniz. Parçacığın yörüngesi bir dış ve bir de iç bir daireye teğetler yapan elliptik yörüngeler olmalıdır. Bu şekilden faydalanarak bu parçacığın yörüngesinin kapalı olup olmadığına karar vermeye çalışınız.

g) Değişkenlerin zamanla nasıl değiştiğini görmek için x-t, y-t, V_x-t, V_y-t grafiklerini

çiziniz. Ayrıca V_x-x ve V_y-y grafiklerini çiziniz. Bu çizim aynı zamanda faz uzayı olarak bilinir. Faz uzayının her noktası ziyaret ediliyor mu?

(**Not:** (3.61)'deki diferansiyel denklemi silindirik koordinatlarda da yazılabilir. Yörünge problemlerinde kullanılan standart değişken değişimi $z = 1/r$ kullanılarak bu denklem, zamanın denklemlerde açıkça görülmediği ikinci mertebe bir denkleme dönüştürülebilir. Zaman yerine silindirik koordinatlardaki açı değişkeni probleme bir parametre olarak girer. Uygun değişken değişimleri ile ölçekleme yapılarak bu denklem de boyutsuz hale getirilebilir. Yalnız bu durumda çözülmesi gereken dört tane birinci mertebe diferansiyel denklem yerine sadece iki tane birinci mertebe diferansiyel denklem olacaktır).

4.8.2 Proje: Elektrik ve Manyetik Alanları Altında Hareket Eden Klasik Elektronlar

Bu bölümde ögrendiklerimizin pekiştirilmesi için aşağıdaki problemi çözmeye çalışınız.

B manyetik ve **E** elektrik alanı altında hareket eden kütlesi m, yükü q olan bir parçacığın hareket denklemi

$$m\frac{d^2\mathbf{r}}{dt^2} = q[\mathbf{V}\times\mathbf{B}+\mathbf{E}] - \gamma\mathbf{V} \tag{4.64}$$

ile verilir. Burada **V** parçacığin hızı olup $\gamma\mathbf{V}$ terimi parçacığın içinde hareket ettiği ortamdan kaynaklanan bir çeşit sürtünme kuvvetini temsil etmektedir.

a) Eletrik alanının sadece x-bileşeni (**E**=$(E,0,0)$) ve manyetik alanın sadece z-bileşeni (**B**=$(0,0,B)$) olduğunu, parçacığın başlangıçta z-yönünde hızı olmadığını varsayarak hareket denkleminin

$$\begin{aligned}\frac{d^2x}{dt^2} &= (\frac{qB}{m})V_y - (\frac{\gamma}{m})V_x + (\frac{qE}{m})\\ \frac{d^2y}{dt^2} &= -(\frac{qB}{m})V_x - (\frac{\gamma}{m})V_y\end{aligned} \tag{4.65}$$

olması gerektiğini gösteriniz. Veya hızın bileşenlerini türevli halleri ile yazarsak

$$\begin{aligned}\frac{d^2x}{dt^2} &= (\frac{qB}{m})\frac{dy}{dt} - (\frac{\gamma}{m})\frac{dx}{dt} + (\frac{qE}{m})\\ \frac{d^2y}{dt^2} &= -(\frac{qB}{m})\frac{dx}{dt} - (\frac{\gamma}{m})\frac{dy}{dt}\end{aligned} \tag{4.66}$$

elde ederiz.

b) Yukarıdaki denklemler dikkatlice incelendiğinde bu problemde üç tane bağımsız sabit olduğu görülecektir. Bunlar $(\frac{qB}{m})$, $(\frac{\gamma}{m})$ ve $(\frac{qE}{m})$'dir. $(\frac{qB}{m})$ ve $(\frac{\gamma}{m})$ sabitlerinin biriminin zamanın tersi olduğunu gösteriniz. Aslında $w_c = (\frac{qB}{m})$ *siklotron frekansı* ve $b = (\frac{\gamma}{m})$ *sönüm oranı* sabiti olarak bilinir. Bu nedenle bu problemde iki tane doğal zaman birimi vardır. Zamanı ya siklotron frekansının tersi veya sönüm oranı sabitinin tersi cinsinden ölçebiliriz. Bunlardan örneğin siklotron frekansını kullanmaya karar verirsek zaman $t' = w_c t$ şeklinde yeniden ölçeklendirilerek birimsiz hale getirilebilir. Bu durumda problemin zamanı artık siklotron frekansının tersi cinsinden ölçülmektedir.

c) (3.66)'da $t' = w_c t$ değişken değişimini yaparak

$$\begin{aligned} \frac{d^2x}{dt'^2} &= \frac{dy}{dt'} - (\frac{b}{w_c})\frac{dx}{dt'} + \alpha \\ \frac{d^2y}{dt'^2} &= -\frac{dx}{dt'} - (\frac{b}{w_c})\frac{dy}{dt'} \end{aligned} \tag{4.67}$$

denklemlerini elde ediniz. Burada $\alpha = (\frac{qE}{mw_c^2})$'dir.

d) α'nın biriminin uzunluk olduğunu gösteriniz. Böylece problemde geçen uzunlukları α'yı kullanarak birimsiz hale getiriniz. $x' = x/\alpha$ ve $y' = y/\alpha$ değişken değişimleri yapılırsa ortaya çıkan denklemler

$$\begin{aligned} \frac{d^2x'}{dt'^2} &= \frac{dy'}{dt'} - (\frac{b}{w_c})\frac{dx'}{dt'} + 1 \\ \frac{d^2y'}{dt'^2} &= -\frac{dx'}{dt'} - (\frac{b}{w_c})\frac{dy'}{dt'} \end{aligned} \tag{4.68}$$

olacaktır. Son denklem serisindeki üsleri düşürerek

$$\begin{aligned} \frac{d^2x}{dt^2} &= \frac{dy}{dt} - (\frac{b}{w_c})\frac{dx}{dt} + 1 \\ \frac{d^2y}{dt^2} &= -\frac{dx}{dt} - (\frac{b}{w_c})\frac{dy}{dt} \end{aligned} \tag{4.69}$$

denklemleri elde edilir. Son denklemlerde problemin doğal zamanlarının oranı olan sadece bir parametre olmasına dikkat ediniz. Bu denklemlerdeki değişkenler birimsiz olup, uzunluklar α, zaman ise siklotron frekansının tersi cinsinden ölçülmektedir.

e) (3.69)'daki denklemleri dört tane birinci mertebe diferansiyel denklemine dönüştürünüz ve (b/w_c)'nin ve başlangıç şartlarının çeşitli değerleri için bu denklemleri ADD8 programını uygun bir şekilde değiştirerek çözünüz. örneğin $x = 0$, $y = 0$, $V_x = 1$,

$V_y = 1$ başlangıç şartlarını ve $b/w_c = 0.01$ parametre değerini kullanarak $t \in [0,160]$ aralığı için çözüm yapınız. t, x, y, V_x ve V_y değerlerini bir dosyaya yazdırarak x-t, y-t, y-x, V_x-t ve V_y-t grafiklerini çiziniz. Elektrik alanı yönünde düzgün doğrusal hareket olmamasına dikkat ediniz. Hareket negatif y yönüne doğrudur. Aslında bu $\mathbf{E} \times \mathbf{B}$ vektörel çarpımının yönü ile aynıdır ve bu sebepten $\mathbf{E} \times \mathbf{B}$ sürüklenmesi olarak bilinir. Faz uzayının (V_x-x ve V_y-y) grafiklerini de çiziniz.

4.9 Verlet Algoritması ve Moleküler Dinamik

Bir parçacığa etkiyen net kuvvet $\mathbf{F}$ ve parçacığın konumu $\mathbf{r}(\mathbf{t})$ ise bilindiği gibi Newton'un ikinci hareket yasasına göre

$$\frac{d^2\mathbf{r}}{dt^2} = \mathbf{F} \tag{4.70}$$

yazılabilir. Burada kuvvet en genel haliyle zamanın, hızın, koordinatların veya bunlardan birkaçının veya hepsinin birden fonksiyonu olabilir. Eğer parçacık korunumlu bir kuvvet altında hareket ediyorsa bu durumda kuvvet koordinatların bir fonksiyonudur ve uygun bir skaler potansiyel fonksiyonunun gradiyenti şeklinde yazılabilir:

$$\mathbf{F} = -\nabla U(\mathbf{r}) \tag{4.71}$$

4.9.1 Hızdan Bağımsız Verlet Algoritması

Verlet tarafından geliştirilen algoritma, parçacığın konumunun zamana göre aşağıdaki gibi iki farklı şekilde Taylor serisine açılması ile elde edilir:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \Delta t\frac{d\mathbf{r}}{dt} + \frac{1}{2}(\Delta t)^2\frac{d^2\mathbf{r}}{dt^2} + \frac{1}{6}(\Delta t)^3\frac{d^3\mathbf{r}}{dt^3} + O(\Delta t)^4 \tag{4.72}$$

$$\mathbf{r}(t-\Delta t) = \mathbf{r}(t) - \Delta t\frac{d\mathbf{r}}{dt} + \frac{1}{2}(\Delta t)^2\frac{d^2\mathbf{r}}{dt^2} - \frac{1}{6}(\Delta t)^3\frac{d^3\mathbf{r}}{dt^3} + O(\Delta t)^4 \tag{4.73}$$

Yukarıdaki iki denklem taraf tarafa toplanırsa

$$\mathbf{r}(t+\Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\Delta t) + (\Delta t)^2\mathbf{a}(t) + O(\Delta t)^4 \tag{4.74}$$

elde edilir. Burada $\mathbf{a}$ ivme olup, m parçacığın kütlesi olmak üzere potansiyel fonksiyonundan

$$\mathbf{a}(t) = \frac{1}{m}\mathbf{F}(t) = -\frac{1}{m}\nabla U(\mathbf{r}(t)) \tag{4.75}$$

elde edilir. Görüldüğü gibi Verlet algoritması üçüncü mertebeden hassas yaklaşık bir yöntemdir. $t+\Delta t$ anındaki konumun bilinmesi için $t-\Delta t$ ve t anındaki konumların yanısıra t anındaki parçacığın ivmesininde bilinmesi gerekir. üç boyutlu bir problemde bunların her bir değeri için üç bileşen olduğundan bir tek parçacğın bir sonraki zaman adımındaki konumunun bilinmesi için dokuz değerin biliniyor olması gerekir. Burada dikkat edilirse parçacığın hızı işlemlere girmemektedir. Bu nedenle denklem (3.74)'de verilen algoritma hızdan bağımsız Verlet algoritması olarak bilinir.

4.9.2 Hıza Bağımlı Verlet Algoritması

Bazı durumlarda parçacığın sadece konumu değil, hızınında bilinmesi gerekebilir. örneğin kinetik enerji hesaplamaları için hızlar gerekecektir. Daha az hassas olmakla beraber $t+\Delta t$ anındaki konum ve hız, t anındaki değerlerinin Taylor serisi açılımlarından elde edilebilir. $\mathbf{v}(t) = d\mathbf{r}/dt$ olmak üzere

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \Delta t\mathbf{v}(t) + \frac{1}{2}(\Delta t)^2\mathbf{a}(t) \tag{4.76}$$

$$\mathbf{v}(t+\Delta t/2) = \mathbf{v}(t) + \frac{1}{2}(\Delta t)\mathbf{a}(t) \tag{4.77}$$

$$\mathbf{a}(t+\Delta t) = -\frac{1}{m}\nabla U(\mathbf{r}(t+\Delta t)) \tag{4.78}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\Delta t/2) + \frac{1}{2}(\Delta t)\mathbf{a}(t+\Delta t) \tag{4.79}$$

Yukarıdaki denklemler sırası ile kullanıldığında $t+\Delta t$ anındaki konum ve hız bulunabilir. t anındaki konum bilindiğinden prensipte bu konuma karşılık gelen kuvvet (ivme) bulunabilir. t anındaki hızda bilindiğinden, denklem (3.76) $t+\Delta t$ anındaki konum bulunabilir. (3.77) denklemi daha sonra kullanılmak üzere hızın bir ara değerini bulur. $t+\Delta t$ anındaki konum bilindiğinden, bu andaki konuma karşılik gelen kuvvet (ivme) de bulunabilir. Denklem (3.78) bunu ifade eder.

KAYNAKLAR

GRILLET, P. A., 1995. Semigroups. An introduction to the structure theory, Dekker, New York.

HANSEN, P. B., 1994. Multiple-length Division Revisited: a Tour of the Minefield, Software-Practice and Experience, 24(6), 579-601.

HERZOG, J., 1970. Generators and relations of abelian semigroups and semigroup rings, Manuscripta Math., 3, 175-193.

HOWIE, J.M., 1995. Fundamentals of Semigroup Theory, Oxford:Clarendon Press.

KNUTH, D.E., 1997a. The Art of Computer Programming (Vol 1, Fundamental algorithms), 3rd ed., Addison Wesley

_____, 1997b. The Art of Computer Programming (Vol 2, Seminumerical algorithms), 3rd ed., Addison Wesley

REDEI, L., 1965. The theory of finitely generated commutative semigroups, Pergamon.

ROSALES, J. C., 1995. On finitely generated submonoids of N^k, Semigroup Forum, 50, 251-262.

_____, 1996a. On numerical semigroups, Semigroup Forum, 52, 307-318.

_____, 1996b. On symmetric numerical semigroups, J. Algebra, 182, 422-434.

ROSALES, J. C., GARCIA-SANCHEZ, P. A., 1999. Finitely generated commutative monoids, Nova Science Publ., New York.

ROSALES, J. C., GARCIA-SANCHEZ, P. A., URBANO-BLANCO, J. M., 1999. On presentations of commutative monoids, Internat. J. Algebra Comput., 5, 539-553.

ROSALES, J. C., URBANO-BLANCO, J. M., 1996. A deterministic algorithm to decide if a finitely presented abelian monoid is cancellative, Comm. Algebra, 24, 4217-4224.

ÖZGEÇMİŞ

1981 yılında Adana'da doğdum. İlk, orta ve lise eğitimimi Adana'da tamamladıktan sonra 1998 yılında ÇÜ Matematik Bölümünde lisans eğitimime başladım vs.

EKLER

1. EK-A

Bu bölümün temel amacı Bölüm x.x'de verilen denklemin çıkarılışını göstermektir.

1.1 Giriş

$$\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{E} \tag{1.1}$$

$$S = \sum_{n=1}^{\infty} x^n = \frac{1}{1-x} \ , \quad |x| < 1 \tag{1.2}$$

$$I(a) = \int_0^{\infty} x^2 e^{-ax^2} dx \tag{1.3}$$

1.2 Gelişme

$$\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{E} \tag{1.4}$$

1.3 Gelişme

$$S = \sum_{n=1}^{\infty} x^n = \frac{1}{1-x} \ , \quad |x| < 1 \tag{1.5}$$

1.4 Sonuç

$$I(a) = \int_0^{\infty} x^2 e^{-ax^2} dx \tag{1.6}$$

1.5 Sonuç

$$\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{E} \tag{1.7}$$

1.6 Gelişme

$$S = \sum_{n=1}^{\infty} x^n = \frac{1}{1-x} \ , \quad |x| < 1 \tag{1.8}$$

1.7 Sonuç

$$I(a) = \int_0^{\infty} x^2 e^{-ax^2} dx \tag{1.9}$$

2. EK-B

Bu bölümün temel amacı FORTRAN komutlarını basit pratik uygulamalar kullanarak yeniden hatırlatmaktır. Özellikle SUBROUTINE ve FUNCTION alt programlarının nasıl kullanıldığını göstermek üzere basit örnek programlar sunulmuştur. Fortran komutlarının kısa bir özeti ve Fortran programa dili ile ilgili bazı önemli noktalar Ek-A'da verilmiştir. Ayrıca bu bölümde hata, hassasiyet ve kararlılık kavramları üzerinde kısaca durulmuştur. Uzun süredir programlamadan uzak kalmış veya daha önce programlama dersi almış fakat ayrıntılı uygulama fırsatı bulamamış olanların bu bölümü mutlaka izlemeleri önerilmektedir. Özellikle verilen örnek programların yazılıp derlenmesi başlangıç için çok önemlidir.

2.1 Giriş

Bilgisayarlar verilen sayıları yapıları gereği ancak sonlu hassasiyetle hafızalarında tutabilir ve bu nedenle ancak sonlu bir hassasiyetle işlem yapabilirler. Sayıların temsili *bit* (binary digit) ile veya bitlerin bir araya gelmesinden oluşan *byte* (8'li bitler) ile yapılır. Temsil edilen sayılar tam sayı (integer, fixed-point) olabileceği gibi gerçel sayı (real, floating point) da olabilirler. Sayıların temsilinde kullanılan sayı sistemi çoğunlukla *ikili* (binary) sayı sistemi olmakla beraber 16'lı veya 10'lu sistemlerde kullanılmaktadır.

Günlük hayatta kullandığımız onlu sayı sisteminde, örneğin 365 sayısı, aslında üç tane 100, altı tane 10 ve beş tane 1'in toplamından oluşur. Bu durumu

$$365 = 3 \cdot 100 + 6 \cdot 10 + 5 \cdot 1 = 3 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 = (365)_{10} \qquad (2.1)$$

şeklinde gösterebiliriz. Onlu sayı sisteminde 0, 1, ..., 9 rakamları vardır. Bütün sayılar 10'nun uygun katlarının uygun katsayılar ile çarpılması ve bunların toplanması ile elde edilir. 10 bu sistemin *taban* sayısıdır. Aynı sayıyı *ikili* sayı sistemine göre temsil etmek istersek ne yapmamız gerekir? İkili sistemde de bütün sayılar, 2'nin uygun katlarının 0 veya 1 ile çarpımlarının toplanması ile elde edilecektir. O halde 7 sayısını ikili sistemde temsil etmek için 7 sayısının içinde 2'nin değişik katlarından kaç tane olduğunu bulmamız gerekir. Bunu verilen sayıyı sürekli 2'ye bölerek ve kalanları takip ederek yapabiliriz. Bu sayıyı

$$7 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (111)_2$$

şeklinde temsil edebileceğimizi hemen görebiliriz. Benze şekilde 9 sayısı

$$9 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (1001)_2$$

olarak temsil edilebilir. İkili sayı sisteminin görüldüğü gibi taban sayısı 2'dir. Birden küçük sayıların temsili ise kullanılan sayı sisteminin tabanının (onlu sistemde 10, ikili sistemde 2) negatif üsleri kullanılarak yapılır. Bunların ayrıntılarına burada girilmeyecektir.