T.C.

AKDENİZ ÜNİVERSİTESİ

HAREKET DENKLEMİNİN NÜMERİK İNTEGRASYONUNDA ÇİFT HASSASİYET TEKNİĞİ UYGULAYARAK YUVARLAMA HATALARININ AZALTILMASI

Burçak YEŞİLIRMAK

FEN BİLİMLERİ ENSTİTÜSÜ

UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Haziran 2022

ANTALYA

T.C.

AKDENİZ ÜNİVERSİTESİ

HAREKET DENKLEMİNİN NÜMERİK İNTEGRASYONUNDA ÇİFT HASSASİYET TEKNİĞİ UYGULAYARAK YUVARLAMA HATALARININ AZALTILMASI

Burçak YEŞİLIRMAK

FEN BİLİMLERİ ENSTİTÜSÜ

UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Haziran 2022

ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAREKET DENKLEMİNİN NÜMERİK İNTEGRASYONUNDA ÇİFT HASSASİYET TEKNİĞİ UYGULAYARAK YUVARLAMA HATALARININ AZALTILMASI

Burçak YEŞİLIRMAK
FEN BİLİMLERİ ENSTİTÜSÜ

UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Bu tez ...../...../2022 tarihinde jüri tarafından Oybirliği/Oyçokluğu ile kabul edilmiştir.

Dr. Öğr. Üyesi Murat AK (Danışman)
Doç. Dr. Bekir Can LÜTFÜOĞLU
Doç Dr. Ali Özhan AKYÜZ

ÖZET

HAREKET DENKLEMİNİN NÜMERİK İNTEGRASYONUNDA ÇİFT HASSASİYET TEKNİĞİ UYGULAYARAK YUVARLAMA HATALARININ AZALTILMASI

Burçak YEŞİLIRMAK

Yüksek Lisans Tezi, Uzay Bilimleri ve Teknolojileri Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Murat AK

Haziran 2022; 48 sayfa

Nümerik integrasyon binlerce kayan noktalı sayı ile gerçekleştirilen bir hesaplama olup integrasyonun adım sayısı arttıkça yuvarlama hataları birikir. Bu çalışmanın amacı Newtonian hareket denkleminin nümerik integrasyonunda biriken yuvarlama hatalarını çift hassasiyet tekniği kullanarak azaltmaktır. Çift hassasiyet tekniği Güneş Sistemi'nin kararlılığı ile ilgili çalışmalarda daha hassas sonuçlarla dinamiksel süreçlerin araştırılması amacıyla sıkça kullanılmaktadır (Quinn ve Tremaine 1989; Quinlan 1994; Rein ve Spiegel 2015). Bu teknik ile 53-bit double hassasiyetli değişkenler kullanılarak 105-bit hassasiyetli kısa dönemli bir integrasyon gerçekleştirilmiştir. Elde edilen sonuçlar double hassasiyetli test integrasyonunun sonuçları ile iki cisim yörünge sabitleri kullanılarak karşılaştırılmıştır ve 8 basamak kazanç elde edilmiştir. Sonuç olarak tekniğin 3.01 kat CPU zamanı maliyetine karşın yerel integrasyon hataları 10^8 kat azaltılmıştır.

ANAHTAR KELİMELER: Çift hassasiyetli kayan nokta aritmetiği, Çift hassasiyet tekniği, Double-Double aritmetik, Hareket Denklemi, IEEE 724-2008 Standartı, İkiye ayırma algoritması, Kayan nokta aritmetiği, İki Cisim Problemi, Nümerik integrasyon, Runge Kutta metodları, Yuvarlama hatası, Yuvarlama hatalarının azaltılması

JÜRİ: Dr. Öğr. Üyesi Murat AK

Doç. Dr. Bekir Can LÜTFÜOĞLU

Doç Dr. Ali Özhan AKYÜZ

ABSTRACT

REDUCTION OF ROUND-OFF ERRORS IN NUMERICAL INTEGRATION OF EQUATION OF MOTION USING PAIR-PRECISION TECHNIQUE

Burçak YEŞİLIRMAK

MSc Thesis in Space Science and Technologies
Supervisor: Asst. Prof. Dr. Murat AK
June 2022; 48 pages

Calculation of numerical integration is realized with thousands of floating-point numbers and round-off errors accumulate as the number of steps increases in the integration. The aim of this study is to reduce the round-off errors accumulated in the numerical integration of Newtonian equation of motion using pair-precision technique. Pair-precision technique is frequently used in studies of the stability of the Solar System for the purpose to investigate dynamical processes with more precise results (Quinn and Tremaine 1989; Quinlan 1994; Rein and Spiegel 2015). Through this technique, a 105-bit precision integration was performed using 53-bit double-precision variables. The obtained results were compared with the results of double-precision test integration using two-body orbital constants and a gain of 8 digits has been achieved. Consequently, local integration errors were reduced 10^8 times despite by a factor of 3.01 CPU time cost of the technique.

KEYWORDS: Double-Double arithmetics, Equation of Motion, Floating-Point arithmetics, Floating-Point arithmetics with pair-precision, IEEE 724-2008 Standard, Numerical integration, Pair-Precision Technique, Reduction of round-off errors, round-off errors, Runge Kutta methods, Splitting algorithm, Two Body Problem

COMMITTEE: Asst. Prof. Dr. Murat AK
Assoc. Prof. Dr. Bekir Can LÜTFÜOĞLU
Assoc. Prof. Dr. Ali Özhan AKYÜZ

ÖNSÖZ

Doğal ve yapay gök cisimlerinin yörüngelerini hesaplamak disiplinlerarası ciddi bir çalışma gerektirir. Hesaplamaları daha doğru sonuçlar elde etmek için yapmak ise daha çok çalışmayı gerektirir. Yörünge integrasyonunda yuvarlama hatalarını azaltmak için uygulanan çift hassasiyet tekniğini öğrettiği için Dr. Anatoliy Ivantsov'a teşekkürü borç bilirim. Tez yazım sürecinde motive edici yaklaşımı ve yol göstericiliği için danışmanım sayın Dr. Murat AK'a teşekkürlerimi sunarım. Yüksek lisans eğitimim süresince manevi desteklerinden dolayı sayın Doç. Dr. Bekir Can LÜTFÜOĞLU'na teşekkürlerimi sunarım. Eğitim hayatım boyunca maddi ve manevi tüm katkılarından dolayı sevgili aileme teşekkürlerimi sunarım...

İÇİNDEKİLER

AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum "Hareket Denkleminin Nümerik İntegrasyonunda Çift Hassasiyetli Kayan Nokta Aritmetiği Uygulayarak Yuvarlama Hatalarının Azaltılması" adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak bulunduğunu belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

06/06/2022

Burçak YEŞİLIRMAK

SİMGELER VE KISALTMALAR

Simgeler:

d	: basamak sayısı
F	: kayan noktalı sayılar kümesi
h	: adım boyutu
k	: eğim katsayıları
p	: hassasiyet
$RN(x)$	: en yakına yuvarlama modu
β	: kayan noktalı sayı tabanı
μ	: gravitasyonel parametre

Kısaltmalar:

CPU	: Central Processing Unit
IEEE	: Institute of Electrical and Electronics Engineers
NaN	: Not a Number
RK4	: Dördüncü Dereceden Runge-Kutta Metodu
ulp	: Unit Last Place

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Yörünge integrasyonu hesaplamalarında çok uzun ve çok büyük hassasiyetli girdi ve çıktı sayılarına ihtiyaç duyulmaktadır. Bu büyük hassasiyetli sayılar bilgisayarda standart olarak desteklenmediği için kayan noktalı sayı olarak temsil edilemez ve yuvarlanır. Bunun yanısıra kayan noktalı sayılarla yapılan aritmetik işlem sonuçları çoğunlukla kayan noktalı sayı olarak temsil edilemediği için yuvarlanır (Overton 2001). Bilgisayarın sınırlı basamak sayısı problemi nedeniyle kayan noktalı aritmetik işlemlerin sonucunda yuvarlama hataları meydana gelir (Goldberg 1991). Yörünge integrasyonu binlerce kayan noktalı aritmetik işlemle gerçekleştirilen bir hesaplamadır ve integrasyonun adım sayısı arttıkça yuvarlama hataları birikir.

Biriken yuvarlama hataları Güneş Sistemi'nin evrimi ve kararlılığı gibi araştırmalarda en temel kısıtlamalardan birisidir (Milani ve Nobili 1988). Uzun dönemli yörünge integrasyonlarında yüksek dereceli simplektik ve çok adımlı yöntemler kullanılarak kesme hatası azaltılır fakat zamanla biriken yuvarlama hataları kesme hatasına baskın gelir ve integrasyonun doğruluğu azalır (Fukushima 2001). Brouwer yasası gereği integratörün türü ve derecesinden bağımsız olarak yuvarlama hataları zamanla $t^{3/2}$ kuvvetiyle büyür (Rein ve Spiegel 2015). Bu sebeple yuvarlama hatalarının yayılımını izlemek ve azaltmak oldukça önemlidir (Antoñana vd. 2017).

Yörünge integrasyonunda yuvarlama hatalarını azaltmak için ilk yaklaşım hassasiyeti iki katına çıkarmaktır (Goldberg 1991). 53-bit double hassasiyetli bir hesaplamada hassasiyeti iki katına çıkarmak için double veri tipindeki değişkenlerin quadruple veri tipine dönüştürülmesi gerekir. 128-bit quadruple hassasiyetli hesaplamalar genellikle yazılımsal olarak yapıldığı için yörünge integrasyonunda quadruple hassasiyet kullanımı oldukça maliyetlidir (Fukushima 2001).

Yuvarlama hatalarını azaltmak için diğer bir yaklaşım aritmetik işlemlerin yuvarlama hatalarını hesaplayıp sonuca ekleyen aritmetik işlem algoritmaları kullanmaktır. Algoritmalarda ana fikir ardışık aritmetik işlemler gerektiren bir hesaplamada kayan noktalı aritmetik işlemlerin yuvarlama hatasını hesaplamak ve hesaplanan yuvarlama hatalarını biriktirip sonuca ekleyerek global hatayı azaltmaktır. Kahan 1965'de literatürde compensated summation olarak bilinen kayan noktalı toplama işlemi algoritmasını tanıtmıştır.

Dekker 1971'de bu fikri ileri taşıyıp toplama, çıkarma, çarpma, bölme ve karekök işlemlerinden oluşan çift hassasiyetli kayan noktalı aritmetik işlem algoritmaları geliştirmiştir (Bebee 2017). Çift hassasiyetli kayan nokta aritmetiğinde 53-bit double hassasiyetli çiftli değişkenlerle aritmetik işlemler yapılır ve toplam 105-bit hassasiyetli çiftli sonuçlar üretilir. Dolayısıyla hassasiyet iki katına çıkar ve sınırlı basamak sayısı artırılmış olur. Çift hassasiyet tekniği ile yörünge integrasyonunda kaybedilen hassasiyetlerin yani yuvarlama hatalarının takip edilmesi ve azaltılması sağlanır (Rein ve Spiegel 2015).

Güneş Sistemi'nin uzun dönemli integrasyonlarında yuvarlama hatalarını azaltmak için Kahan compensated summation algoritması sıkça kullanılmaktadır. Bu çalışmalarda Kahan algoritması yalnızca integrasyon şeması içerisindeki toplama işlemlerine uygulanmaktadır (Quinn ve Tremaine 1989; Quinlan 1994; Rein ve Spiegel 2015). Uzun dönemli integrasyonlarda doğruluğu arttırmak amacıyla yüksek dereceli, algoritması karmaşık yapıda metotlar kullanılmaktadır. Bu metotlar trigonometrik, üssel ve logaritmik terimler içermektedir (Fukushima 2001). Bu sebeple yuvarlama hatalarını azaltmak için integrasyon şemasının tamamına çift hassasiyetli aritmetik işlemleri uygulamak oldukça zor ve maliyetlidir.

Bu çalışmanın amacı iki cisim probleminin nümerik çözümü için kullanılacak olan dördüncü dereceden Runge-Kutta (RK4) nümerik integrasyon şemasına çift hassasiyet tekniği uygulayarak yuvarlama hatalarını azaltmaktır. Bu teknik sayesinde 53-bit double hassasiyetli değişkenler kullanarak 105-bit hassasiyette sonuçlar üreten kısa dönemli bir integrasyon gerçekleştirilecektir ve daha hassas sonuçlar ile Jüpiter'in Güneş merkezli yörüngesi elde edilecektir. İki cisim yörünge integrasyonunda modifiye edilmiş RK4 integratörü ile elde edilen sonuçların double hassasiyetli integrasyonun sonuçlarına göre çok daha doğru olduğu analitik çözümle karşılaştırma testleriyle gösterilmiştir.

RK4 integratörü basit algoritmik yapısı sebebiyle çift hassasiyetli aritmetik işlemlerin integrasyon şemasındaki tüm aritmetik işlemlere uygulanabilmesi bakımından uygundur ve literatürdeki benzer kısa dönemli integrasyonların geçerliliğini kontrol etmek için seçilmiştir (Fukushima 2001). Güneş Sistemi'nin dinamiksel ve evrimsel süreçlerini daha iyi anlamak için yörünge integrasyonlarında mümkün olan en iyi çözümün elde edilmesi gerekir. Bu çalışmada uygulanan teknik ile Güneş Sistemi'nin evrimi ve kararlılığı ile ilgili çalışmalarda tahminlerin kullanışlılığını artırmak hedeflenmektedir.

2. KAYNAK TARAMASI

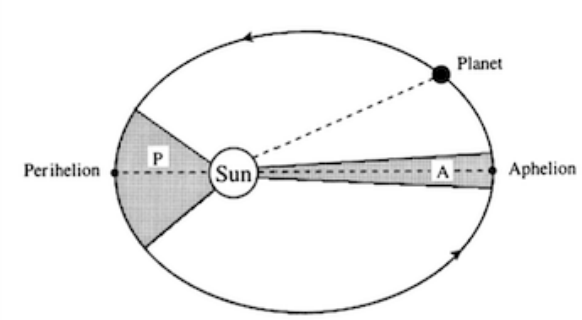
2.1. İki cisim problemi

Aristoteles döneminden Kepler dönemine kadar gök cisimlerinin hareketine ilahi bir atıf yapılıyor ve gezegenlerin çembersel yörüngelerde dolandığına inanılıyordu. Çembersel yörüngenin tek doğal ve mükemmel hareket olduğu ve hareketin küçük çemberlerin büyük çemberlerin kombinasyonlarında gerçekleştiği düşünülüyordu (Bate vd. 1971). Kepler, Tycho Brahe'nin gözlem verileri ile çembersel hareket teorisini uzlaştırmaya çalışırken çelişkiler tespit ederek Mars hariç o dönem bilinen bütün gezegenlerin yörüngelerini oldukça yüksek bir hassasiyette hesaplamıştır. Mars, Güneş Sistemi'nde Merkür'den sonra eksentrisitesi en büyük gezegendir ve bu sebeple kompleks bir yörüngeye sahiptir fakat bu o dönem henüz keşfedilememişti (Fiorelli ve Vilmart 2017).

Kepler, Brahe'nin Mars'ın konumuyla ilgili gözlem verilerine çeşitli geometrik eğriler uydurmaya çalışmış ve 1609'da Mars'ın odaklarından birinde Güneş'in yer aldığı eliptik bir yörüngede dolandığını keşfetmiştir. Kepler bu keşfi ile Güneş Sistemi'deki cisimlerin yörünge hareketine ilişkin modern çağdaki bilgilerimize yol açan üç gezegensel hareket yasasını türetmiştir (Prussing ve Conway 1993). Kepler'in gezegensel hareket yasaları aşağıdaki gibi tanımlanmaktadır:

1. Gezegenler odaklarından birinde Güneş'in bulunduğu eliptik yörüngelerde dolanır.
2. Güneş'ten gezegene uzanan çizgi eşit zaman aralıklarında eşit alanlar süpürür.
3. Gezegenin yörünge periyodunun karesi ile yarı büyük eksen uzunluğunun küpü birbirine eşittir.

Kepler'in hareket yasaları gözlem verileriyle elde edilmiş olup gezegensel hareketin neden bu üç yasayla açıklandığını belirten teorik bir temele dayanmamaktaydı. Dolayısıyla Kepler yasaları gezegensel hareketin kinematiğinden ibaret olup dinamiği hakkında bilgi vermiyordu (Bate vd. 1971). Sir Isaac Newton, Descartes'ın momentum çalışmaları ve Kepler'in hareket yasalarından faydalanarak gezegenlerin eliptik yörüngelerine neden olan kuvvetin ters kare yasasıyla ilişkili olduğunu keşfetmiş ve kendi hareket yasalarını ortaya koymuştur.



Şekil 2.1. Gezegenlerin Güneş etrafındaki eliptik yörüngesi

Newton, Kepler'in birinci yasasından yola çıkarak gezegenlerin Güneş etrafındaki eliptik yörüngesine iki cisim arasındaki uzaklığın ters karesiyle orantılı bir çekim kuvvetinin neden olduğunu keşfetmiştir. Newton ayrıca Kepler'in ikinci yasasının açısal momentumun korunumu ilkesinin bir sonucu olduğunu göstermiştir. Newton'un hareket yasaları aşağıdaki gibi tanımlanmaktadır:

1. Eylemsiz referans sisteminde bir cisim üzerine dengelenmemiş net bir kuvvet etki etmedikçe durgun halini veya sabit hızlı hareketini korur.

2. Çizgisel momentumun zamanla değişimi cisme etkiyen net kuvvetle aynı yöndedir ve orantılıdır.

3. Her etkiye karşılık eşit büyüklükte ve ters yönde bir tepki kuvveti vardır.

Eylemsiz bir koordinat sisteminde m kütleli bir cisme etkiyen tüm vektörel kuvvetlerin toplamı $\sum \vec{F}$ ile gösterilsin ve cismin ivmesi $\ddot{\vec{r}}$ ifadesiyle tanımlansın. Bu durumda Newton'un ikinci yasası matematiksel olarak,

$$\sum \vec{F} = \frac{d(m\vec{v})}{dt} = m\ddot{\vec{r}} \tag{2.1}$$

eşitliği ile ifade edilir.

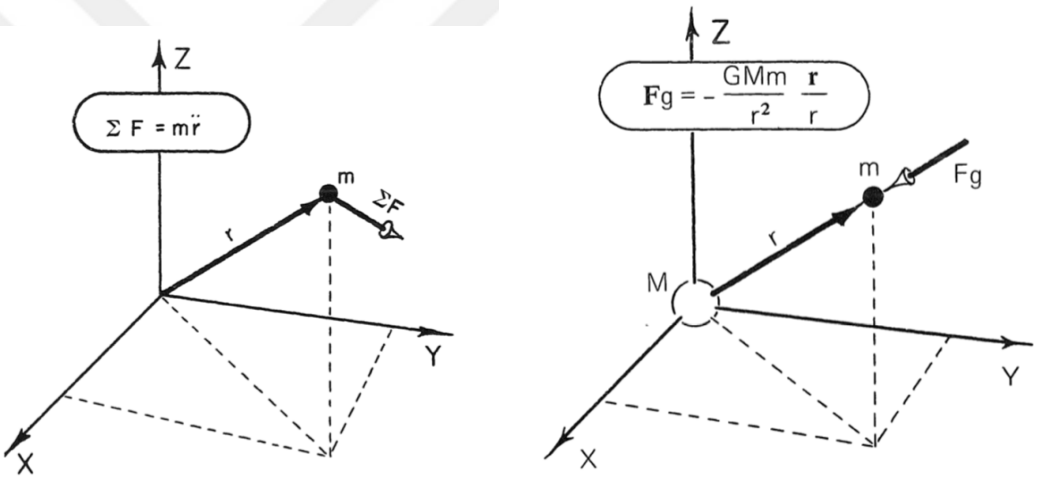
Newton ayrıca evrensel çekim yasasını formüle etmiştir. Evrensel çekim yasası iki cismin kütlelerinin çarpımıyla doğru ve aralarındaki uzaklığın karesiyle ters orantılı bir kuvvetle birbirlerini çektiğini belirtir. Bu kuvvet evrensel çekim kuvveti olup,

$$\vec{F}_g = -\frac{GmM}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.2}$$

eşitliği ile ifade edilir.

(2.2)'de verilen $\vec{F}_g$ vektörü M kütlesinin m kütlesi üzerine etkiyen çekim kuvvetini, $\vec{r}$ ifadesi iki cisim arasındaki vektörel uzaklığı belirtir. Evrensel kütle çekim sabiti G olarak tanımlanıp G$\simeq 6.67259\times 10^{-11} m^3 kg^{-1} s^{-2}$ değerindedir.

Güneş Sistemi'nde Güneş etrafında kapalı ve periyodik bir yörüngede dolanan herhangi bir cismin hareketi aynı mekanik yasalarla gerçekleşmektedir ve Kepler yasalarıyla uyumludur (Prussing ve Conway 1993). Dolayısıyla gezegenlerin, kuyruklu yıldızların, asteroidlerin veya gezegenler arası bir uzay aracının hareketi ayrıca doğal veya yapay bir uydunun bir gezegen etrafındaki hareketi aynı mekanik yasalara tabidir. Sonuç olarak, Kepler tarafından gezegenlerin Güneş etrafındaki hareketini tanımlayan iki cisim modeli günümüzde de geçerliliğini korumaktadır (Veris 2018).



Şekil 2.2. İkinci hareket yasası ve çekim yasasının geometrisi (Bate vd. 1971)

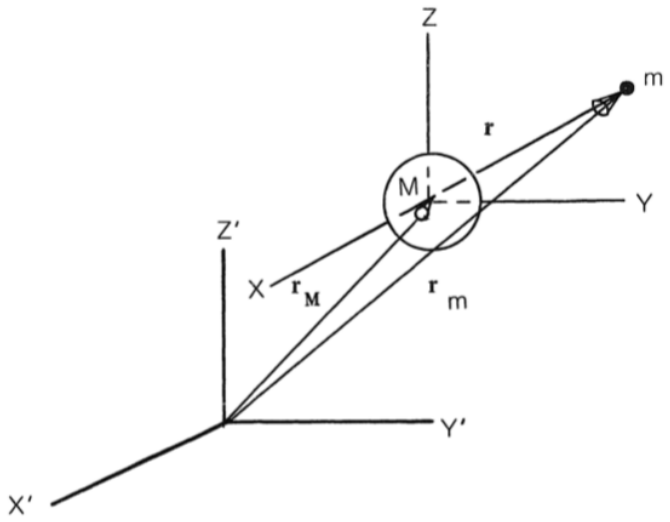
İki cisim problemi sırasıyla m ve M kütlelerine sahip iki izole cismin yörüngesini belirleme problemidir. Bu iki cisim birbirlerini yalnızca kütlelerinin çarpımıyla doğru ve uzaklıklarının karesiyle ters orantılı olan evrensel çekim kuvveti ile çeker. Newton'un ikinci hareket yasası $\vec{F} = m\vec{a}$ ve evrensel çekim yasası birleştirildiğinde ortaya bir cismin merkezdeki bir diğer cisme göre yörüngesini belirleyen hareket denklemi çıkmaktadır. İki cisim probleminin çözümünde amaç, verilen bir başlangıç zamanında izole edilmiş iki cismin konum ve hızlarını yalnızca karşılıklı çekim kuvveti etkisiyle hareket ettikleri varsayılırak istenilen herhangi bir zamanda tayin etmektir (Veris 2018).

2.1.1. Eylemsiz referans sisteminde iki cisim problemi için hareket denkleminin türetilmesi

Newton'un ikinci hareket yasası ve gravitasyonel çekim kuvveti yörünge hareketini belirlemek amacıyla birlikte kullanılmaktadır. Newton'un ikinci hareket yasası, çizgisel momentumun zamanla değişiminin cisimlere etkiyen kuvvetle orantılı olduğunu belirtir. Kepler'in gezegensel yörüngelere ilişkin yasalarının Newton'un hareket yasalarında geçerli olabilmesi ve hareket denkleminin türetilebilmesi için aşağıdaki varsayımlar yapılır:

1. Cisimler küresel simetrik materyal noktalar olarak tanımlanır yani kütlelerinin merkezlerinde toplandığı varsayılır. Böylelikle yalnızca küresel simetrik merkezi bir cisimden etkiyen kuvvet göz önüne alınır.

2. Eylemsiz koordinat sisteminde iki cismin merkezlerini birleştiren çizgi boyunca çekim kuvveti dışında etkiyen başka bir iç veya dış kuvvet olmadığı varsayılır ve böylece kuvvetin bileşenleri kolayca bulunabilir.



Şekil 2.3. İki cisim probleminin geometrisi (Bate vd. 1991)

İki cisim probleminde ya bir cismin diğerine göre hareketi ya da her iki cismin ortak kütle merkezi etrafındaki hareketi ile ilgilenilir (Battin 1999). Şekil 2.3'de eylemsiz referans sisteminde M ve m kütleli iki cismin hareket probleminin geometrisi gösterilmektedir. (X,Y,Z) koordinat sistemi dönmeyen ve ivmelenmeyen ideal bir eylemsiz referans sistemidir ve (X',Y',Z') eylemsiz kartezyen koordinat sistemine paraleldir.

Eylemsiz (X,Y,Z) kartezyen koordinat sisteminin orijini sistemin kütle merkezinde konumlanmıştır (Vallado 2013). Şekil 2.3'de gösterilen eylemsiz sistemde M ve m kütleli cisimlerin (X',Y',Z') koordinat sisteminin orijinine göre konum vektörleri $\vec{r}_M$ ve $\vec{r}_m$ olarak tanımlanmıştır. M kütleli cisimden m kütleli cisme doğru uzanan konum vektörü,

$$\vec{r} = \vec{r}_m - \vec{r}_M \tag{2.3}$$

eşitliği ile ifade edilir.

Eylemsiz koordinat sistemi (2.3)'de verilen eşitliğin (X,Y,Z) koordinat sisteminin her bir eksenini hesaba katmadan türevlenebilir olması bakımından önemlidir (Vallado 2013). m kütleli cismin M kütleli cismin kütle merkezine göre ivmesi (2.3)'de verilen vektörel eşitliğin ikinci türevi ile hesaplanır.

$$\ddot{\vec{r}} = \ddot{\vec{r}}_m - \ddot{\vec{r}}_M \tag{2.4}$$

Şekil 2.3'de verilen eylemsiz koordinat sisteminde (2.2)'de verilen evrensel çekim kuvveti yasasından faydalanılarak m ve M kütleli iki cismin birbiri üzerine etkiyen çekim kuvvetleri,

$$\vec{F}_1 = \frac{-GMm}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.5}$$

$$\vec{F}_2 = \frac{GMm}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.6}$$

eşitlikleriyle ifade edilir.

(X',Y',Z') eylemsiz koordinat sisteminin orijinine göre Newton'un ikinci hareket yasası ve evrensel çekim yasası birleştirilirse,

$$F_{g_m} = m\ddot{\vec{r}}_m = \frac{-GMm}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.7}$$

$$F_{g_M} = M\ddot{\vec{r}}_M = \frac{GMm}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.8}$$

eşitlikleri elde edilir.

(2.7) ve (2.8)'de verilen eşitlikler sadeleştirilip birbirinden çıkartılırsa ve (2.4)'de verilen göreli ivme için çözülürse m kütleli cismin M kütleli cisme göre hareketini tanımlayan,

$$\ddot{\vec{r}} = \frac{-G(M+m)}{r^2}\left(\frac{\vec{r}}{r}\right) \tag{2.9}$$

denklemi elde edilir.

Bir gezegenin Güneş etrafındaki hareketi, bir yapay uydunun veya uzay aracının bir gezegen etrafındaki hareketi iki cisim problemiyle tanımlanır (Bate vd. 1971). İki cisim probleminde yörüngede hareket eden cismin kütlesi m ve merkezdeki cismin kütlesi M olarak tanımlanırsa m kütlesinin M kütlesine oranla ihmal edilebilir ölçüde küçük olduğu varsayılır. Böylelikle $G(M+m) \simeq GM$ olarak tanımlanabilir. GM değeri gravitasyonel parametre olarak adlandırılır ve μ ile gösterilir. Böylelikle μ=GM olmak üzere (2.9)'da verilen denklem,

$$\boxed{\ddot{\vec{r}} = -\frac{\mu}{r^2}\left(\frac{\vec{r}}{r}\right)} \tag{2.10}$$

olarak ifade edilir.

(2.10)'da eylemsiz koordinat sisteminde m kütleli cismin M kütleli cisme göre yörüngesini belirlemek için kullanılan hareket denklemi verilmektedir. İki cisim hareketi merkezdeki M kütleli cisimden kaynaklanmaktadır. Hareket denklemi ikinci dereceden, nonlineer, vektörel, adi diferansiyel denklemdir (Vallado 2013). Bu çalışmada iki cisim probleminin analitik ve nümerik çözümü için küçük kütleli cisim Jüpiter ve merkezdeki büyük kütleli cisim Güneş olarak seçilmiştir. Jüpiter'in kütlesi Güneş'in kütlesinin yaklaşık binde biri kadar olup ihmal edilebilir ölçüde küçüktür.

2.1.2. İki cisim probleminin hareket sabitleri

Newtonian iki cisim hareket denkleminin nümerik çözümü ile elde edilen sonuçların Kepler tarafından ampirik olarak bulunan yasalarla doğrulanması gerekir (Boulet 1991). İki cisim yörünge hareketi Kepler yörüngesi olarak tanımlanır. Kepler yörüngesinde küçük kütleli cismin merkezdeki büyük kütleli cisme göre yörüngesini tanımlayan 6 adet yörünge elemanı bulunmaktadır. Yörünge elemanlarının küçük kütleli cismin yörüngesi boyunca değişmedikleri varsayılır. İki cisim hareketinde mekanik enerji sistemin kinetik enerjisi ile potansiyel enerjisinin toplamına eşittir ve bu iki enerji birbirlerine dönüşerek sistemde herhangi bir enerji kazancı veya kaybı gerçekleşmez. Dolayısıyla sistemin mekanik enerjisi korunur ve sabit bir değere eşit olduğu varsayılır. Ayrıca küçük kütleli cismin merkezi cisim etrafındaki açısal momentumu değişmez.

Bu çalışmada hareket denkleminin nümerik çözümü ile elde edilen sonuçlar yörünge elemanlarından yarı büyük eksen uzunluğu, eksentrisite ve hareket sabitlerinden mekanik enerji, açısal momentum değerleri ile karşılaştırılarak analiz edilmiştir.

2.1.3. Açısal momentum

İki cisim hareketinin açısal momentumu (2.10)'da verilen hareket denkleminin $\vec{r}$ ile vektörel olarak çarpılmasıyla türetilir. Vektörel çarpımda $\vec{r}\times\vec{r}=0$ olduğundan,

$$\vec{r}\times\ddot{\vec{r}}+\vec{r}\times\frac{\mu}{r^3}\vec{r}=0 \tag{2.11}$$

eşitliğinde ikinci terim ihmal edilir ve $\vec{r}\times\ddot{\vec{r}}=0$ elde edilir. Bu terim türev açılımıyla,

$$\frac{d}{dt}(\vec{r}\times\dot{\vec{r}})=\dot{\vec{r}}\times\dot{\vec{r}}+\vec{r}\times\ddot{\vec{r}} \tag{2.12}$$

$$\frac{d}{dt}(\vec{r}\times\dot{\vec{r}})=0 \tag{2.13}$$

olarak türetilir. (2.13)'de verilen eşitliğin integrali alınırsa açısal momentum ,

$$\vec{h}=\vec{r}\times\dot{\vec{r}} \tag{2.14}$$

eşitliği ile tanımlanır.

(2.14)'de verilen eşitlik konum ve hız vektörlerinin aynı düzlemde olduğunu belirtir. Açısal momentum vektörü $\vec{h}$ yörünge düzlemine diktir ve konum ve hız vektörlerinin vektörel çarpımına eşittir. Dolayısıyla iki cisim hareketi sabit bir yörünge düzlemi üzerinde gerçekleşir ve sistemin açısal momentumu korunur.

2.1.4. Mekanik enerji

İki cisim hareketinin mekanik enerji formülünü türetmek için (2.10)'da verilen hareket denklemi $\dot{\vec{r}}$ değeri ile skaler olarak çarpılır.

$$\dot{\vec{r}}\cdot\ddot{\vec{r}}+\dot{\vec{r}}\cdot\frac{\mu}{r^3}\vec{r}=0 \tag{2.15}$$

$\dot{\vec{r}}=\vec{v}$ ve $\ddot{\vec{r}}=\dot{\vec{v}}$ olarak tanımlanır ve (2.15)'de verilen eşitlik düzenlenirse,

$$v\cdot\dot{v}+\frac{\mu}{r^3}(r\cdot\dot{r})=0 \tag{2.16}$$

eşitliği elde edilir.

(2.16)'da eşitliğin içerisindeki skaler çarpım değerleri türev açılımı ile türetilerek,

$$\frac{d}{dt}\left(\frac{v^2}{2}\right)=v\cdot\dot{v} \qquad \frac{d}{dt}\left(\frac{-\mu}{r}\right)=\frac{\mu}{r^2}\dot{r} \tag{2.17}$$

eşitlikleri ile ifade edilir.

(2.17)'de verilen türev eşitlikleri (2.16)'da verilen eşitlikte yerine konulursa,

$$\frac{d}{dt}\left(\frac{v^2}{2}-\frac{\mu}{r}\right)=0 \tag{2.18}$$

denklemi elde edilir.

Yörünge mekaniğinde (2.18)'de verilen eşitliğin integral sabiti sıfır kabul edilir ve böylelikle özel mekanik enerji değeri elde edilir. Kapalı bir sistemin mekanik enerjisi sistemin kinetik enerjisi ile potansiyel enerjisinin birbirine dönüşmesi ile korunur yani sabit kalır. Mekanik enerji ξ ile tanımlanırsa,

$$\xi=\frac{v^2}{2}-\frac{\mu}{r} \tag{2.19}$$

denklemi ile hesaplanır.

2.1.5. Eksentrisite vektörü ve yarı büyük eksen uzunluğu

Eksentrisite, Kepler yörüngesinin şeklini belirleyen bir yörünge elemanıdır. Eksentrisite vektörü $\vec{e}$ ile tanımlanır ve (2.10)'da verilen hareket denklemi integre edilirse,

$$\vec{e}=\frac{\vec{v}\times\vec{h}}{\mu}-\frac{\vec{r}}{r} \tag{2.20}$$

eşitliği elde edilir.

(2.20)'de verilen eşitlikte açısal momentum vektörü $\vec{h}=\vec{r}\times\vec{v}$ yerine konulursa,

$$\vec{e}=\frac{\vec{v}\times(\vec{r}\times\vec{v})}{\mu}-\frac{\vec{r}}{r} \tag{2.21}$$

elde edilir.

(2.21)'de verilen eşitliğin sağ tarafındaki vektörel çarpım terimi açılır ve eşitlik düzenlenirse eksentrisite vektörü,

$$\vec{e}=\left(\frac{v^2}{\mu}-\frac{1}{r}\right)\vec{r}-\frac{\vec{r}\cdot\vec{v}}{\mu}\vec{v} \tag{2.22}$$

denklemi ile hesaplanır.

Kepler yörüngesini tanımlayan diğer bir yörünge elemanı yarı büyük eksen uzunluğudur. Küçük kütleli cismin yörünge periyodu yarı büyük eksen uzunluğu tarafından belirlenir. Yarı büyük eksen uzunluğu a ile tanımlanırsa,

$$\frac{1}{a}=\frac{2}{r}-\frac{v^2}{\mu} \tag{2.23}$$

denklemi ile hesaplanır.

2.2. Nümerik integrasyon

Yörünge mekaniğinde kullanılan temel hareket denklemleri, birinci ve ikinci dereceden nonlineer adi diferansiyel denklemlerdir (Vallado 2013). Tek bir bağımlı değişken içeren diferansiyel denklemlere adi diferansiyel denklemler denir. Adi diferansiyel denklemlerde bilinmeyen terim en az birinci dereceden türevli ifade içeren bir fonksiyondur. Bir adi diferansiyel denklemin en yüksek dereceli türevi birinci dereceden ise bu denklem birinci dereceden adi diferansiyel denklem olarak adlandırılır. Birinci dereceden adi diferansiyel denklemler,

$$\frac{dy}{dt} = f(t, y) \tag{2.24}$$

şeklinde ifade edilir.

Nonlineer adi diferansiyel denklemlerde kesin bir çözüm elde etmek için formül bulma işlemi oldukça zordur (Fiorelli ve Vilmart 2017). Karmaşık ve analitik çözümü bulunamayan problemler, problemi basitleştiren modellerle kolaylaştırılır. Bu basitleştirilmiş modeller problemin karakteristiğini yansıtan özelliklere sahiptir. İki cisim problemi, n cisim problemindeki yörünge dinamiklerinin analizi için basitleştirilmiş bir model örneğidir (Anzalone ve Chai 2015). İki cisim problemi literatürde iyi çalışılmış bir problem olup ikinci dereceden vektörel adi diferansiyel hareket denklemini içerir ve bu denklem 1744 yılında Euler tarafından analitik olarak tamamen çözülmüştür.

Adi diferansiyel denklemler modern bilgisayarlar yardımıyla nümerik integrasyon yöntemleri kullanılarak yaklaşık olarak çözülebilir. Nümerik çözüm için nonlineer bir diferansiyel denklem, birinci dereceden diferansiyel denklem sistemlerine indirgenir ve başlangıç değer problemiyle tanımlanır. Başlangıç değer problemi,

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0 \end{aligned} \tag{2.25}$$

şeklinde ifade edilir.

(2.25) eşitliğinde verilen $f(t, y(t))$ ifadesi integrand fonksiyonu, $y'(t)$, $y(t)$'nin bağımsız değişken t'ye göre birinci türevi, $y(t)$ diferansiyel denklemin bilinmeyeni ve y_0 değeri başlangıç koşulu olarak tanımlanır.

Fiziksel bir sistemin nümerik integrasyonu sistem modelinin zaman evriminin hesaplanması anlamına gelir. Nümerik integrasyon yöntemlerinde amaç, $y(t)$'nin zaman evrimini yaklaşık olarak hesaplamaktır (Scherer 2017). Bu tür yaklaşık çözümler elde etmek için kullanılan nümerik yöntemler çoğunlukla sonlu-farklar metodu veya Taylor seri açılımı kullanılarak geliştirilir (Veris 2018).

Sonlu-farklar metoduna dayanan nümerik integrasyon yöntemlerinde $(i = 0, 1, 2 \ldots, n)$ olmak üzere tanım kümesi ayrık ve sonlu t_i noktaları olarak tanımlanır. Ardışık iki ayrık nokta arasındaki uzaklık değeri adım boyutu olarak tanımlanır ve h ile gösterilir. Tek adımlı nümerik yöntemlerde adım boyutu $h = t_{i+1} - t_i$ olmak üzere t_i noktaları arasındaki uzaklığın eşit uzunlukta olduğu varsayılarak seçilir. Adım boyutu h sıfıra yaklaştıkça elde edilen yaklaşık çözüm problemin kesin çözümüne yakınsar (Fiorelli ve Vilmart 2017).

Sonlu-farklar metoduyla türetilen nümerik integrasyon yöntemlerinde integrasyonun başlangıç anı t_0 olmak üzere $y_0 = y(t_0)$ başlangıç koşulu kullanılarak $y_i \simeq y(t_i)$ yaklaşımı uygulanır ve $[t_1, t_2, t_3, \ldots, t_n]$ noktalarında $y(t_i)$ fonksiyonları yaklaşık olarak hesaplanır.

$$y'(t_i) \simeq \frac{y_{i+1} - y_i}{h} \tag{2.26}$$

Euler metodu (2.26)'da verilen sonlu-farklar metodu ile türetilen en basit nümerik integratör olarak kabul edilir (Fiorelli ve Vilmart 2017). Euler metodu, (2.26)'da verilen eşitliğe $y_i \simeq y(t_i)$ yaklaşımı uygulanarak,

$$y_{i+1} = y_i + (y_i)'h \tag{2.27}$$

denklemi ile tanımlanır.

Nümerik integrasyon yöntemleri ayrıca Taylor seri açılımı kullanılarak türetilebilir. Bir adi diferansiyel denklem $y'(t) = f(t, y(t))$ formunda tanımlanırsa $(i = 0, 1, 2 \ldots, n)$ için $h = t_{i+1} - t_i$ olmak üzere Taylor seri açılımı,

$$y_{i+1} = y_i + y_i'h + \frac{y_i''}{2!}h^2 + \cdots + \frac{y_i^{(n)}}{n!}h^n + O(h^{n+1}) \tag{2.28}$$

şeklinde ifade edilir.

(2.27) ve (2.28)'de verilen eşitlikler karşılaştırılırsa Taylor seri açılımının ilk iki teriminin toplamının Euler metoduna karşılık geldiği görülmektedir. (2.28)'de verilen $O(h^{n+1})$ ifadesi yerel kesme hatasını tanımlayan bir fonksiyondur ve yerel hatanın Taylor açılımının $n+1$ inci derecesinde h ile orantılı olduğunu belirtir (Chapra ve Canale 2010).

Herhangi bir nümerik integrasyon metodunun hatası yuvarlama ve kesme hatalarının birleşiminden oluşur. Yuvarlama hatası bilgisayarın sınırlı basamak sayısı problemi nedeniyle ortaya çıkar. Kesme hatası ise diferansiyel denklemin yaklaşık çözümünün bir sonucu olarak tanımlanabilir (Bate vd. 1971). Kesme hatası Taylor seri açılımındaki tüm terimlerin kullanılmamasından kaynaklanmaktadır. Euler integratöründe, Taylor seri açılımında ihmal edilen yüksek dereceden türevlerin neden olduğu hatayı azaltmak için adım boyutu h yeterince küçük bir değer seçilmelidir (Vallado 2013). Küçük adım boyutu kullanılması durumunda ise integrasyonda bir adım ilerlemek için çok fazla işlem yapılacağından yuvarlama hatası artar. Özetle, nümerik integrasyonda hata kaçınılmaz olmakla birlikte amaç, kullanılacak metodunun yuvarlama ve kesme hatalarının toplamını minimize edecek şekilde probleme uygun olarak seçilmesidir (Bate vd. 1971).

2.2.1. Hareket denkleminin nümerik integrasyonu için yöntemler

İki cisim probleminin hareket denklemi nümerik integrasyon yöntemleri ile çözülerek küçük kütleli cismin merkezi büyük kütleli cisme göre yörüngesi belirlenebilir. Yörünge integrasyonunda doğru sonuçlar üreten ve hızlı uygulanabilen nümerik bir yöntem gerekmektedir (Voesenek 2008). Hareket denkleminin çözümü için kullanılan nümerik integrasyon yöntemleri tek adımlı ve çok adımlı yöntemler olmak üzere ikiye ayrılır. Hareket denklemi Newton mekaniği içermektedir fakat Hamilton mekaniğine yönelik türetilen simplektik nümerik integrasyon yöntemleri de mevcuttur.

Yörünge mekaniğinde tek adımlı yöntemler olarak Runge-Kutta ve Bulirsch-Stoer, çok adımlı yöntemler olarak Adams, Störmer–Cowell ve Gauss–Jackson metodları sıkça kullanılmaktadır (Veris 2018). Çok adımlı yöntemler integrasyona başlamak için tek adımlı bir yöntem kullanmayı gerektirir yani özel bir başlangıç prosedürüne sahiptir. Çok adımlı yöntemler adım boyutu değişimine hassastır ve integrasyonun her adımında bir önceki adımdaki fonksiyon değerleri $(y_{n-1}, y_{n-2}, y_{n-3}, \dots)$ kullanılır. Dolayısıyla çok adımlı yöntemler karmaşık bir teorik mantık içermektedir (Battin 1999).

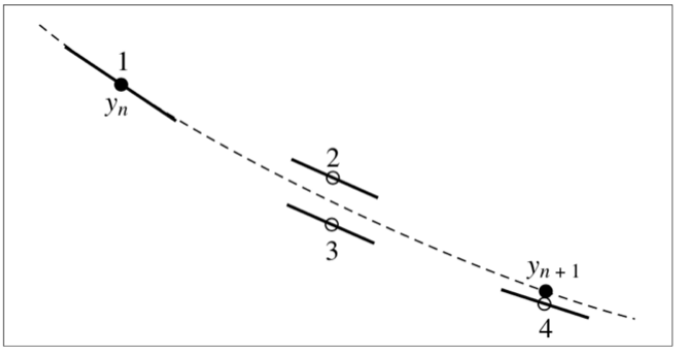
Yörünge integrasyonunda hangi yörünge tipi için hangi nümerik yöntemin kullanılacağı oldukça önemli bir problemdir. Seçilen yöntemin analitik çözüme en yakın olacak şekilde yeterli doğrulukta sonuç üretmesi beklenir (Bate vd. 1971). Bu sebeple birçok nümerik integrasyon metodunda hatayı azaltmak için adım aralığı teknikleri uygulanır.

Adım aralığı teknikleri sabit adım aralığı ve değişken adım aralığı teknikleri olmak üzere ikiye ayrılır. Sabit adım aralığı tekniği yörünge boyunca ardışık iki nokta arasındaki uzaklık yani adım boyutu sabit tutulabildiği için dairesel yörüngelerde tercih edilir. Değişken adım aralığı tekniği bir cismin hızlandığında yada yavaşladığında yörüngede kalabilmesi için ek hesaplamalar içerdiğinden eksentrisitesi yüksek yörüngelerde tercih edilir (Vallado 2013).

Bir nümerik integrasyon yönteminin seçimi için kriterler; kullanılan metodun teorik karmaşıklık içermemesi, düşük hesaplama maliyetine sahip olması, basit ve hızlı olması, mümkün olduğu kadar geniş adım aralığı içermesi, değişken adım aralığına dönüştürülebilmesi şeklinde sıralanabilir. Runge-Kutta tipi nümerik yöntemler uzun adım aralığında kararlı olması ve adım boyutunun kolayca değiştirilebilmesi, başlangıç prosedürü gerektirmemesi, doğruluk derecesinin arttırılabilmesi ve hızlı olması bakımından hareket denkleminin çözümü için uygun ve yeterlidir (Bate vd. 1971).

2.2.2. Dördüncü dereceden tek adımlı Runge-Kutta metodu (RK4)

Runge-Kutta metodu tek adımlı nümerik yöntemlerde en sık kullanılan ve en popüler olan yöntemdir (Veris 2018). Farklı derecelerde birçok Runge-Kutta metodu bulunmaktadır. Runge-Kutta metodlarının derecesi, yöntemin yerel kesme hatasının derecesine göre sınıflandılır. Dördüncü dereceden Runge-Kutta metodu (RK4) ile dördüncü dereceden Taylor seri açılımının yerel kesme hataları eşdeğerdir ve $O(h^5)$ fonksiyonu ile ölçülür (Vallado 2013). Tek adımlı Runge-Kutta metodlarında çok adımlı yöntemlerdeki gibi yüksek dereceli türev terimlerinin hesaplanmasıyla ilgili özel formüller yerine integrasyon yapılacak aralıktaki farklı noktaların eğim değerleri kullanılır.



Şekil 2.4. Dördüncü dereceden Runge-Kutta metodu (Press vd. 2002)

RK4, Euler tipi bir ekstrapolasyon yöntemidir ve adi diferansiyel denklemlerin nümerik çözümü için kullanılabilir (Press vd. 2002). RK4 metodunda Şekil 2.4'de gösterildiği gibi $(i = 0, 1, 2 \ldots, n)$ ve $h = t_{i+1} - t_i$ olmak üzere $[t_i, t_i + h]$ ekstrapolasyon aralığındaki 3 farklı noktadaki 4 eğim değeri kullanılarak problemin kesin çözümüne yakınsanır. Birinci dereceden adi diferansiyel bir denklem $y'(t_i) = f(t, y(t_i))$ ve $y(t_0) = y_0$ olacak şekilde başlangıç değer problemi ile tanımlansın. Eğim katsayıları (k_1, k_2, k_3, k_4) olmak üzere başlangıç değer problemi için RK4 metodu,

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.29}$$

denklemi ile ifade edilir.

(2.29) eşitliği içerisindeki k eğim katsayıları,

$$\begin{aligned} k_1 &= f(t_i,\ y_i) \\ k_2 &= f\left(t_i + \frac{h}{2},\ y_i + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t_i + \frac{h}{2},\ y_i + \frac{h}{2}k_2\right) \\ k_4 &= f(t_i + h,\ y_i + hk_3) \end{aligned} \tag{2.30}$$

fonksiyonları ile hesaplanır.

RK4 metodunda adım boyutu h ile $[t_i, t_i + h]$ zaman aralığındaki k eğim katsayılarının ağırlıklandırılmış ortalaması çarpılarak $(y_{i+1} - y_i)$ değişimi hesaplanır ve böylece zamanda tek bir adım ilerlenir. k eğim katsayıları $[t_i, t_i + h]$ zaman aralığının başlangıç, orta ve bitiş noktalarında hesaplanır. Başlangıç noktasında k_1 eğim katsayısı hesaplanır. Orta noktada önce k_1 eğim değeri kullanılarak k_2 değeri hesaplanır, sonra hesaplanan k_2 değeri kullanılarak k_3 değeri hesaplanır yani orta noktada iki kere hesaplama yapılır. Bitiş noktasında hesaplanan k_3 değeri kullanılarak k_4 değeri bulunur ve hesaplanan tüm eğim değerleri (2.29)'da verilen eşitlikte yerine konulur.

Yöntemin kesme hatası büyük kütleli merkezi bir cisim etrafında dolanan küçük kütleli cismin yörüngesini tahmin etme problemi için yeterli küçüklüktedir (Bate vd. 1971). RK4 metodu basit algoritmik yapısı sebebiyle herhangi bir programlama dili yardımıyla kolayca uygulanabilir. Bu çalışmada dördüncü dereceden tek adımlı Runge-Kutta metodu hareket denkleminin nümerik integrasyonu için C programlama dili yardımıyla kullanılmıştır.

2.3. Kayan Nokta Aritmetiği

Kayan nokta aritmetiği sayısal niceliklerin hesaplama işlemlerinin bilgisayarda yapılması olarak tanımlanabilir (Boldo 2014). Sayısal hesaplamalar yapmak için fiziksel araçların kullanılması fikri antik çağlara kadar uzanmaktadır. John Napier'in 17. yüzyılda logaritma cetvelini icat etmesiyle aritmetik işlemler hız kazanmıştır. Pascal ve Liebniz tarafından 17. yüzyılda icat edilen mekanik hesaplama makineleri 19. yüzyılın sonlarına kadar yaygın olarak kullanılmıştır. Charles Babbage 19. yüzyılda ilk programlanabilir makine fikrini ortaya atmış fakat mali anlaşmazlıklar nedeniyle projesini uygulayamamıştır (Overton 2001). Konrad Zuse, bilgisayar çağını başlatan ilk elektro-mekanik makineyi 1939-1942 yılları arasında icat etmiştir. Bu makine hesaplamalarını ikili kayan noktalı sayı sistemiyle yapan Z3 bilgisayarıdır.

2.3.1. Kayan noktalı sayı sistemi

Reel sayılar kümesi $\mathbb{R}$ ve kayan noktalı sayılar kümesi F olmak üzere $F \subseteq \mathbb{R}$ olacak şekilde tanımlanır. F sistemi dört tamsayı parametre ile karakterize edilir (Higham, 2002). $F(\beta, p, e_{\max}, e_{\min})$ olarak tanımlanırsa $\beta \geq 2$ ve $e_{\min} < 0 < e_{\max}$ olmak üzere F kümesinin parametreleri,

- sayı tabanı : β
- hassasiyet : p
- ekstremum üs aralığı : $(e_{\min}, e_{\max})$

olarak adlandırılır.

$x \in F$ ve d basamak sayısı olmak üzere F kümesindeki her bir x kayan noktalı sayısı $(d_1, \ldots, d_p) \in \mathbb{Z}$ olacak şekilde,

$$x = \pm \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_p}{\beta^p} \right) \cdot \beta^e \tag{2.31}$$

$$0 \leq d_i \leq \beta - 1 \quad (i = 1, \ldots, p) \tag{2.32}$$

(2.32)'de verilen koşulu sağlamak üzere (2.31)'de verilen eşitlikle ifade edilir.

Kayan nokta kavramı bir ondalık ifadedeki noktanın üs değiştikçe yer değiştirmesinden türetilmiştir. Dolayısıyla kayan noktalı sayıların temsili bilimsel gösterim ile benzerdir (Wittig 2008). F kümesinde sıfırdan farklı bir x kayan noktalı sayısı için $d_0 \neq 0$ ise x sayısı "normalize edilmiş" olarak tanımlanır (Forsythe vd. 1977). Normalize edilmiş kayan noktalı bir x sayısı (2.31) eşitliğinden faydalanılarak,

$$x = \pm \left(1.d_1 d_2 \ldots d_{p-1}\right) \cdot 2^e \tag{2.33}$$

olarak ifade edilir.

Kayan noktalı sayıların anlamlı basamak bitlerinin sayısı hassasiyet olarak tanımlanır. Kayan noktalı bir sayının hassasiyeti p ile gösterilirse,

$$p = \left(1.d_1 d_2 \ldots d_{p-1}\right)_2 \tag{2.34}$$

olarak ifade edilir.

Her kayan noktalı sayı dijital bilgisayarlarda 32-bit, 64-bit gibi sabit uzunluktaki bir bit dizisi ile temsil edilir ve elemanları 0 veya 1'den oluşur. Bu sebeple bir tamsayı ile reel sayının bilgisayarda temsil edilmesinde farklılıklar vardır ve reel sayıların temsili tamsayılara göre daha karmaşıktır (O'Regan 2016). Bir tamsayı bilgisayarda 32-bitlik sabit bir bit dizisi ile temsil edilebilir. Bir reel sayı kayan noktalı bir sayı olarak temsil edilirken sabit sayıda hassasiyet biti ve iki tabanında üs kullanılır.

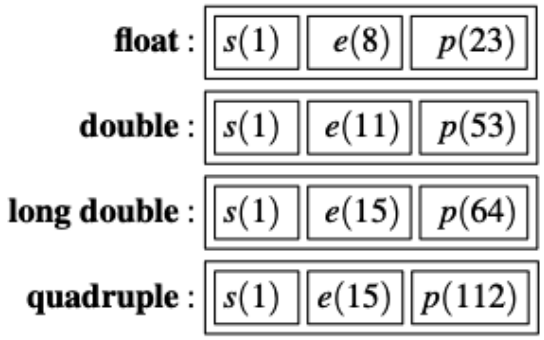
2.3.2. Kayan Nokta Aritmetiği için IEEE 754-2008 Standardı

Sonsuz ve sürekli reel sayı sistemini sonlu ve ayrık bilgisayar sayı sisteminde temsil edebilmek için sabit noktalı, logaritmik ve p-sel sayı sistemi gibi birçok farklı yaklaşım önerilmiştir. Günümüzde modern bilgisayarların çoğu kayan noktalı sayı sistemini kullanmaktadır (Muller vd. 2010). Kayan noktalı sayılar için IEEE 754 standardı 1985'de yayınlanmıştır ve 2008-2019 yıllarında revize edilmiştir (Boldo 2014). IEEE 754-2008 standartı kayan noktalı sayı formatlarını ve aritmetik işlem algoritmalarını içerir. IEEE standartını destekleyen bilgisayarlarda aynı aritmetik işlemler yapılırsa aynı sonuçlar üretilir. Bu özellik programların taşınabilmesini büyük ölçüde kolaylaştırır (Goldberg 1991). IEEE standartı ile ayrıca istisnai durumlar için (0, NaN, - ∞, +∞) tutarlı sonuçlar üretilir ve kayan noktalı aritmetik işlemlerin doğru yuvarlanması sağlanır (Overton 2001).

Çizelge 2.1. IEEE 754-2008 standardında kayan noktalı sayı formatları

Veri Tipi	Üs	Hassasiyet	Toplam Bit	Basamak Sayısı	Makine epsilonu (ε)
float	8	23	32	~ 7.2	$2^{-23} \approx 1.19 \times 10^{-7}$
double	11	53	64	~ 15.9	$2^{-52} \approx 2.22 \times 10^{-16}$
long double	15	64	80	~ 19.2	$2^{-63} \approx 1.08 \times 10^{-19}$
quadruple	15	112	128	~ 34.0	$2^{-112} \approx 1.93 \times 10^{-34}$

Çizelge 2.1'de gösterildiği gibi IEEE 754-2008 standartında kayan noktalı sayılar için dört temel temsiliyet formatı tanımlanmıştır ve bu formatlar veri tipi olarak adlandırılır (Panhaleux 2012). Kayan noktalı bir sayının hassasiyeti float, double, long double ve quadruple olmak üzere dört farklı kelime boyutunda tanımlanır. Bilgisayarın kelime boyutu normalize edilmiş kayan noktalı sayıları depolamak için üç alana ayrılır (Overton 2001). Bu üç alan sırasıyla işaret (s), üs (e) ve hassasiyet (p) biti olarak adlandırılır.



Şekil 2.5. Kayan noktalı sayılar için IEEE 754-2008 format şemaları

Kayan noktalı sayılar için format şeması Şekil 2.5'de her bir veri tipi için kutucuklarla temsil edilerek gösterilmiştir. Bu şemaya göre toplam 64-bit uzunluğuna sahip bir double kelime boyutu 1 işaret biti, 11 üs biti ve 53 hassasiyet biti olmak üzere üç alandan oluşur. Şekil 2.5'de verilen format şemasına göre double veri tipindeki bir kayan noktalı sayının virgülden sonra 52 basamağı $(d_0.d_1d_2\ldots d_{52})_2$ depolanabilir. Normalize edilmiş kayan noktalı bir sayı için $d_0 = 1$ olduğundan bu sayı depolanmaz böylelikle 1 bitlik kazanç elde edilir. Bir sayı Şekil 2.5'de verilen IEEE format şemasına uygun olarak depolanabiliyorsa kayan noktalı sayı olarak adlandırılır (Overton 2001).

2.3.3. Basitleştirilmiş kayan noktalı sayı sistemi

Kayan noktalı sayılar sayı doğrusu üzerinde reel sayılar gibi eşit aralıkta ve düzgün bir dağılıma sahip değildir (Higham 2002). Kayan noktalı sayı sistemi F'de 1'den büyük en küçük kayan noktalı sayı (2.35)'de verilen değeri alır.

$$(1.000\ldots1)_2 = 1+\beta^{1-p} \tag{2.35}$$

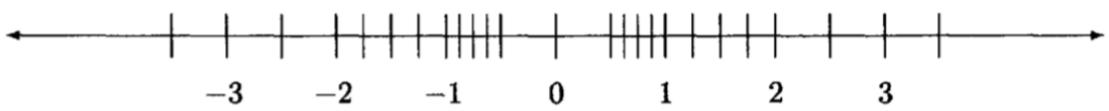
F sisteminde 1 ile 1'den büyük en küçük kayan noktalı sayı arasındaki uzaklık makine epsilonu olarak tanımlanır ve ε ile gösterilir. Makine epsilonu (2.36)'da verilen $\varepsilon = \beta^{1-p}$ değerine eşittir.

$$(0.000\ldots1)_2 = \beta^{1-p} \tag{2.36}$$

x kayan noktalı bir sayı ve $\beta = 2$ olarak tanımlansın. İki kayan noktalı sayı arasındaki uzaklık $ulp(x)$ fonksiyonu ile ölçülür ve $ulp(x) = \varepsilon * 2^e$ değerine eşittir.

$$\begin{aligned} ulp(x) &= (0.000\ldots1)_2 \cdot 2^e \\ &= 2^{1-p} * 2^e \\ &= \varepsilon * 2^e \end{aligned} \tag{2.37}$$

Kayan noktalı sayı sistemi basitleştirilmiş bir kayan noktalı sayı sistemi yardımıyla daha kolay anlaşılabilir. Basitleştirilmiş kayan nokta sisteminde tüm sayıların normalize edildiği ve $\pm(1.d_1d_2)_2 \cdot 2^e$ şeklinde depolandığı varsayılsın. F kümesinin parametreleri $\beta = 2$, $p = 3$, $e_{min} = -1$ ve $e_{max} = 1$ ve üs aralığı $e(-1,0,1)$ olarak tanımlansın.



Şekil 2.6. Basitleştirilmiş Kayan Noktalı Sayı Sistemi (Overton 2001)

Basitleştirilmiş sistemin parametreleri kullanılarak makine epsilonu (2.36)'da verilen eşitlikle $\varepsilon = 0.25$ olarak hesaplanır. (2.37)'de verilen eşitlik kullanılarak $e = 1$ için $ulp(x) = 2\varepsilon$ olarak hesaplanır ve bu üs aralığındaki sayılar arasındaki mesafe iki katına çıkar. $e = -1$ için $ulp(x) = \frac{1}{2}\varepsilon$ olarak hesaplanır ve bu üs aralığındaki sayılar arasındaki mesafe yarıya düşer (Overton 2001).

Şekil 2.6'da verilen sayı doğrusunda sayı değerleri arttıkça sayılar arasındaki mesafenin arttığı görülmektedir. Basitleştirilmiş kayan noktalı sayı sistemindeki tüm parametreler kullanılırsa $(0.5, 0.625, 0.750, 0.875, 1.0, 1.25, 1.50, 1.75, 2.0, 2.5, 3.0, 3.5)$ pozitif kayan noktalı sayıları elde edilir. Aynı üs değerine sahip kayan noktalı sayılar binade sayılar olarak adlandırılır (Boldo 2014). Ardışık binade sayılar arasındaki mesafe eşittir yani $ulp(x)$ değerleri eşittir. Üs değeri değiştiğinde farklı bir binade sayı grubuna geçilir ve sayılar arasındaki uzunluk taban değeriyle $(\beta = 2)$ çarpıldığı için sayılar arasındaki mesafe 2 çarpanı kadar değişir. Dolayısıyla kayan noktalı sayılar arasındaki mesafenin makine epsilonu ε ile karakterize edildiği ve sayı doğrusu üzerinde eşit dağılım göstermediği söylenebilir (Boldo 2014).

2.3.4. Yuvarlama hatası

F ayrık ve sonlu bir kümedir ve içerisinde $(2\varepsilon) \cdot (\beta - 1) \cdot (e_{\max} - e_{\min} - 1) + 1$ adet kayan noktalı sayı bulunur (Forsythe vd. 1977). Sonsuz ve sürekli reel sayılar kümesini sonlu bit sayılarına sıkıştırmak, yani kayan noktalı bir sayı olarak tanımlamak, reel sayıları yaklaşık olarak temsil etmeyi gerektirir (Goldberg 1991). F sonlu bir küme olduğu için sonsuz reel sayılar kümesi $\mathbb{R}$ bu kümede tam olarak temsil edilemez. F kümesinin maksimum elemanından daha büyük bir reel sayının kayan nokta temsili yoktur. Benzer şekilde F kümesindeki en küçük kayan noktalı sayıdan daha küçük bir reel sayı da temsil edilemez. Fakat F kümesindeki her bir eleman $F \subseteq \mathbb{R}$ olduğundan sonsuz reel sayılar kümesi $\mathbb{R}$'de temsil edilebilir (Forsythe vd. 1977).

Bir reel sayının kayan noktalı bir sayı şeklinde tam olarak temsil edilememesinin iki sebebi vardır (Goldberg 1991). Birincisi, bir x reel sayısının kayan noktalı sayı sisteminin dışında kalması durumudur. Eğer x reel sayısının mutlak değeri $\beta \times \beta^{e_{max}}$'dan büyük veya $1.0 \times \beta^{e_{min}}$'den küçükse F'nin dışında kalır ve temsil edilemez (Goldberg 1991). İkinci durum 0.1 reel sayısının iki tabanında gösterimi ile açıklanabilir. 0.1 sayısı ondalık tabanda sonlu bir sayı iken iki tabanında devirli bir sayıya dönüşür.

$$\frac{1}{10} = (0.000110011\ldots)_2 = \frac{1}{16} + \frac{1}{32} + \frac{0}{64} + \frac{0}{128} + \frac{1}{256} + \frac{1}{512} + \frac{0}{1024} + \cdots \quad (2.38)$$

$$(1.10011001100110011001100\ldots)_2 \times 2^{-4} \quad (2.39)$$

$$(0.1)_{10} = \boxed{\boxed{0}\ \boxed{01111011}\ \boxed{10011001100110011001100}} \tag{2.40}$$

0.1 sayısı hafızaya float veri tipinde kaydedilirse Şekil 2.5'de verilen IEEE 754-2008 float format şemasına uygun olarak depolanabilmesi için 2 tabanında sonsuza giden hassasiyeti 23. basamakta kesilir ve (2.40)'da gösterildiği gibi depolanır. Dolayısıyla 0.1 reel sayısı bilgisayarda temsil edilirken yaklaşık olarak temsil edilir ve bir hata oluşur. Bu hata yuvarlama hatası olarak adlandırılır (Overton 2001).

IEEE 754-2008 standartında reel sayıları kayan noktalı sayılara dönüştüren 5 farklı yuvarlama fonksiyonu tanımlanmaktadır. (Panhaleux 2012). Bir reel sayının bir kayan noktalı sayıya nasıl yuvarlandığı yuvarlama fonksiyonunu tanımlayan yuvarlama modlarından biri tarafından belirlenir (Muller vd. 2010). Bir x reel sayısı *RN(x)* en yakına yuvarlama modunda iki kayan noktalı sayı arasında kalır ve bu iki sayıdan kendisine en yakın olanı ile temsil edilir (Graillat vd. 2020). Örneğin, Şekil 2.5'de gösterilen basitleştirilmiş kayan nokta sisteminde 1.66 reel sayısı en yakına yuvarlama modunda $RN(1.66) = 1.75$ olarak yuvarlanır. Dolayısıyla *RN(x)* ifadesi x'e en yakın doğru yuvarlanmış kayan noktalı sayıyı temsil eder (Goldberg 1991).

$x \in \mathbb{R}$ ve ε makine epsilonu olmak üzere x sayısı bilgisayarda temsil edilirken oluşan rölatif yuvarlama hatası δ ile tanımlanırsa,

$$\delta = \left| \frac{RN(x) - x}{x} \right| \leq \frac{1}{2}\varepsilon \tag{2.41}$$

eşitsizliği ile ifade edilir.

Teorem 2.1. *$x \in \mathbb{R}$ olmak üzere x sayısı F kümesindeki bir aralıkta yer alsın. Rölatif yuvarlama hatası δ olmak üzere x reel sayısı en yakına yuvarlama modunda ,*

$$RN(x) = x(1+\delta) \tag{2.42}$$

şeklinde yuvarlanır.

(2.41) eşitsizliği bir reel sayının doğru temsil edilen basamak sayısının Çizelge 2.1'de verilen ε değeri kadar olduğunu ve Teorem 2.1. kayan noktalı sayı sisteminde bir x reel sayısının $(1+\delta)$ çarpanı kadar hatalı temsil edildiğini belirtir (Overton 2001).

IEEE standartında tanımlanan kayan noktalı aritmetik işlemler iki kayan noktalı sayı ile yapılır ve aritmetik işlemlerin sonucu çoğunlukla kayan noktalı bir sayı olarak temsil edilemez (Overton 2001). x ve y kayan noktalı sayılar olmak üzere $(x+y)$ işleminin sonucu kayan noktalı bir sayı olarak temsil edilemesin. Bu durumda $(x+y)$'nin sonucu en yakına yuvarlama modunda $RN(x+y)$ şeklinde doğru yuvarlanır ve kayan noktalı bir sayıya dönüşür. Dolayısıyla $RN(x+y)$ ifadesi $(x+y)$ işleminin kesin sonucunu yaklaşık olarak hesaplar (Overton 2001).

IEEE 754-2008 standartında varsayılan en yakına yuvarlama modunda x ve y sayıları ile yapılan doğru yuvarlanmış kayan noktalı aritmetik işlemler Teorem 2.1'de verilen (2.42) eşitliğiyle,

$$\begin{aligned} RN(x+y) &= (x+y)(1+\delta) \\ RN(x-y) &= (x-y)(1+\delta) \\ RN(x\times y) &= (x\times y)(1+\delta) \\ RN(x/y) &= (x/y)(1+\delta) \end{aligned} \tag{2.43}$$

olarak ifade edilir (Higham 2002).

Özetle, kayan noktalı sayılarla yapılan aritmetik işlemlerin sonucu Şekil 2.5'de verilen IEEE 754-2008 format şemalarının sınırlı bit sayılarına yerleşmesi için yuvarlanır. Bir kayan noktalı aritmetik işlemin sonucu kesin sonuca en yakın kayan noktalı sayı ise bu sayının Teorem 2.1'de verilen (2.42) eşitliği ile potansiyel olarak $0.5ulp$ hata içerdiği söylenebilir (Goldberg 1991).

2.3.5. Yörünge integrasyonunda yuvarlama hatalarının azaltılması

Kahan (1965) kayan noktalı toplama işlemi sonucu oluşan yuvarlama hatasını hesaplayan F2Sum algoritmasını geliştirmiştir (Riedy ve Demmel 2018). F2Sum algoritması literatürde Kahan compensated summation olarak bilinen toplama işlemi algoritmasında kullanılmaktadır. Compensated summation algoritması özyinelemeli bir toplamın her bir adımında yuvarlama hatasını F2Sum ile hesaplar ve biriken (global) hatayı sonuca ekleyerek yuvarlama hatalarını azaltır (Quinlan 1994). Güneş Sistemi'nin uzun dönemli integrasyonlarında biriken yuvarlama hatalarını azaltmak için compensated summation algoritması basit algoritmik yapısı ve etkisi sebebiyle sıkça başvurulan bir yöntemdir (Quinn ve Tremaine 1989; Quinlan 1994; Rein ve Spiegel 2015).

Kayan nokta aritmetiğinde yuvarlama hatalarının en temel kaynağı çok büyük ve çok küçük iki sayının toplanması ve çıkarılmasıdır (Wilkinson 1960). Bu sebeple uzun dönemli integrasyonlarda kullanılan metodun içerisindeki toplama işlemi sayısı azaltılarak yuvarlama hataları azaltılmaya çalışır (Milani ve Nobili 1988; Quinlan 1994). Hareket denkleminin nümerik integrasyonunda toplama, çıkarma, çarpma, bölme ve karekök işlemleri uygulanır. Her bir aritmetik işlemin sonucu potansiyel olarak $0.5ulp$ hata içerdiği için integrasyonda adım sayısı arttıkça yuvarlama hataları birikir. Biriken yuvarlama hatalarını azaltmanın bir başka yolu hassasiyeti iki katına çıkarmaktır (Goldberg 1991).

Hassasiyeti iki katına çıkarmak için veri tipleri birbirine dönüştürülebilir. Örneğin, float veri tipinde iki değişkenle aritmetik işlem yapılırsa ve sonuç double veri tipinde kaydedilirse Çizelde 2.1'de verildiği gibi double hassasiyet float hassasiyetin yaklaşık iki katı olduğu için daha doğru bir sonuç elde edilir. Double aritmetikte hassasiyeti iki katına çıkarmak için double veri tipindeki değişkenler quadruple veri tipine dönüştürülmelidir fakat bilgisayar sistemlerinin çoğunda 128-bit quadruple aritmetik standart olarak desteklenmemektedir. Bunun yanısıra veri tipi dönüşümü CPU zamanını 30-300 kat civarında artırmaktadır (Fukushima 2001).

Double değişkenlerle yapılan aritmetik işlemlerin hassasiyetini iki katına çıkarma problemi quadruple aritmetiğe göre daha basit ve daha az maliyetli çift hassasiyet tekniği ile çözülebilir. Çift hassasiyet tekniği ile double hassasiyeti genişletmek suretiyle yuvarlama hataları azaltılır (Dekker 1971).

3. MATERYAL VE METOT

3.1. Çift Hassasiyet (Pair-Precision) Tekniği

Dekker (1971) bilgisayardaki mevcut hassasiyetleri iki katına çıkarmak için Algol 60 dilinde toplama, çıkarma, çarpma, bölme ve karekök işlemlerinden oluşan çift hassasiyetli aritmetik işlem algoritmaları geliştirmiştir (Bebee 2017). Çift hassasiyetli kayan noktalı aritmetik işlemler iki adet 53-bit double hassasiyetli kayan noktalı sayı ile yapılan bir aritmetik işlemin kesin sonucunu üreten aritmetik işlem fonksiyonlardır (Dekker 1971).

53-bit double hassasiyetli a ve b kayan noktalı sayıları tanımlansın. a ve b sayıları ile yapılan herhangi bir kayan noktalı aritmetik işlemin sonucu $s = RN(a\ op\ b)$ olsun. Çift hassasiyetli aritmetik işlemlerde bilgisayarın sınırlı basamak sayısı problemi nedeniyle kayan noktalı aritmetik işlemler sonucu oluşan yuvarlama hatası hesaplanır ve sonuca dahil edilir (Higham 2002). Çift hassasiyet tekniğinde yuvarlama hatası c olmak üzere $s+c = (a\ op\ b)$ olacak şekilde a ve b sayıları ile yapılan kayan noktalı aritmetik işlemlerin sonucu kesin olarak üretilir.

Çift hassasiyetli aritmetik işlemlerin kodlanabilmesi için öncelikle a ve b sayıları ikiye ayırma (splitting) algoritması ile (a_h, a_l) ve (b_h, b_l) çiftlerine parçalanır ve çift hassasiyetli kayan noktalılara sayıya dönüştürülür (Bebee 2017). Çift hassasiyetli kayan noktalı sayılar iki adet 53-bit double hassasiyetli sayının toplamı olarak temsil edilir.

$$\begin{aligned} a &= a_h + a_l \\ b &= b_h + b_l \end{aligned} \tag{3.44}$$

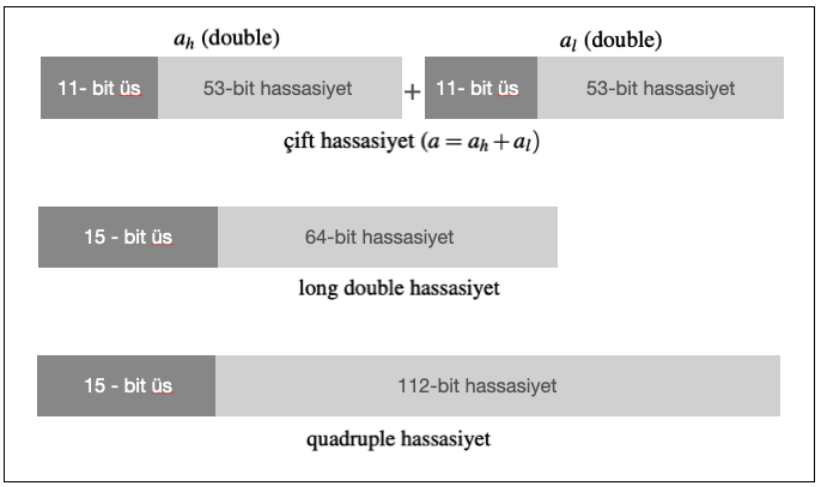
(3.44)'de verilen (a_h, a_l) ve (b_h, b_l) çiftleri sırasıyla a ve b sayılarının büyük dereceli ve küçük dereceli değişkenleri olarak tanımlanır ve hafızada ardışık iki kelime boyutu olarak depolanır (Sterbenz 1974). Çift hassasiyetli aritmetik işlemler sonucunda her bir değişkeni 53-bit double hassasiyetli (s, c) çift hassasiyetli sayısı üretilir.

Aritmetik işlem fonksiyonlarının geliştirilmesi için double hassasiyetli kayan nokta aritmetiği kullanılırsa uygulanan teknik ile toplam 105-bit hassasiyetli sonuçlar üretilir (Fukushima 2001). Çift hassasiyetli sayılar hafızada ardışık iki kelime boyutu olarak depolandığı ve aritmetik işlemler sonucunda her bir değişkeni 53-bit double hassasiyetli bir çift sayı üretildiği için bu teknik ile double hassasiyet iki katına çıkar (Graillat vd. 2020).

Sonuç olarak çift hassasiyetli aritmetik işlemler iki çift girdi (a_h, a_l) , (b_h, b_l) ve bir çift çıktı (s, c) olacak şekilde 6 argümanlı fonksiyonlar olarak tanımlanır. Çift hassasiyetli toplama-çıkarma, çarpma, bölme ve karekök işlemleri,

$$\begin{aligned} Summation(a_h,\ a_l,\ b_h,\ b_l,\ *s,\ *c) \\ Multiplication(a_h,\ a_l,\ b_h,\ b_l,\ *m,\ *c) \\ Division(a_h,\ a_l,\ b_h,\ b_l,\ *d,\ *c) \\ Sqrt(a_h,\ a_l,\ *sq_h,\ *sq_l) \end{aligned} \tag{3.45}$$

fonksiyonları ile tanımlanır.



Şekil 3.7. Çift hassasiyetli, long double ve quadruple sayıların hassasiyetleri

Çift hassasiyet tekniği literatürde double-double veya quasi-quadruple hassasiyetli aritmetik olarak da bilinmektedir (Møller 1965; Hida vd. 2001; Muller vd. 2010; Graillat vd. 2020). Bu çalışmada çift hassasiyetli kayan noktalı aritmetik işlem fonksiyonları IEEE 754-2008 standartında varsayılan $RN(x)$ en yakına yuvarlama modunda ve 53-bit double hassasiyetli kayan nokta aritmetiği kullanılarak dizayn edilmiştir. Algoritmalar C programlama dili yardımıyla geliştirilmiştir böylece hızlı ve verimli çalışan kodlar garanti edilmektedir.

3.1.1. İkiye ayırma (splitting) algoritması

Çift hassasiyetli kayan nokta aritmetiği çift hassasiyet uzunluklu kayan noktalı sayı sistemi ile uygulanır (Dekker 1971). Çift hassasiyetli kayan noktalı sayılar bir reel sayının ikiye ayırma (splitting) algoritması kullanılarak ikiye parçalanması ile elde edilir. İkiye ayırma algoritmasında (yuvarlanan sayı, yuvarlama hatası) çifti oluşturulur (Bebee 2017). Bu çiftler (3.45)'de gösterildiği gibi çift hassasiyetli aritmetik işlem fonksiyonlarının içerisine 53-bit double hassasiyetli büyük ve küçük dereceli değişkenler olarak tanımlanır. $a_h = RN(a)$ bir a reel sayısının bilgisayarda kayan noktalı sayı olarak yuvarlanan değeri olsun.

Bilgisayarın sınırlı digit problemi nedeniyle a sayısının yuvarlanması sonucu oluşan yuvarlama hatası a_l değişkeni ile tanımlanırsa yuvarlama hatası matematiksel olarak,

$$a_l = a - a_h \tag{3.46}$$

eşitliği ile tanımlanır.

a reel sayısı bilgisayarda kayan noktalı bir sayı olarak temsil edilemeyip a_h sayısına yuvarlandığından yuvarlama hatası a_l'yi hesaplamak için (3.46)'da verilen eşitlik kullanılamamaktadır (Higham 2001). Bu çalışmada yuvarlama hatası a_l'yi hesaplamak için geliştirilen algoritma dört basamakta tanımlanabilir:

1. a ve a_h sayılarının elemanları birebir okunacak şekilde karakter dizisine atanır.
2. İki dizinin her bir elemanı sırasıyla karşılaştırılır ve karakterlerin farklılaşmaya başladığı basamak tespit edilir ve m gibi bir değişkene atanır.
3. Dizinin m inci basamaktan sonraki karakterleri integer veri tipine dönüştürülüp I_1 ve I_2 gibi geçici değişkenlere atanır.
4. I_1 ve I_2 değerleri birbirinden çıkarılır ve fark normalize edilecek şekilde m inci basamağa (sağa) kaydırılır ve a_l değişkenine atanır.

$$\boxed{a_l = (I_1 - I_2) \times 10^{-m}} \tag{3.47}$$

Sonuç olarak geliştirilen algoritma ile a reel sayısı $a = a_h + a_l$ olacak şekilde iki adet double hassasiyetli sayının toplamı ile kesin olarak ifade edilir ve çift hassasiyetli kayan noktalı sayıya dönüştürülür (Bebee 2017).

3.1.2. Çift hassasiyetli toplama-çıkarma işlemi

a ve b sayıları F kümesinde tanımlı β tabanında ve 53-bit double hassasiyetli kayan noktalı sayılar olsun. a ve b sayılarının kayan noktalı toplama işleminin sonucu ve yuvarlama hatası sırasıyla 53-bit double hassasiyetli s ve c değişkenleri olarak tanımlansın. Çift hassasiyetli toplama işlemi a ve b sayılarının toplamını,

$$\begin{aligned} s &= RN(a+b) \\ s+c &= a+b \end{aligned} \tag{3.48}$$

olacak şekilde kesin olarak hesaplar (Dekker 1971).

Çift hassasiyetli toplama işlemini oluşturmak için öncelikle a ve b sayıları ikiye ayırma algoritmasıyla $a = a_h + a_l$ ve $b = b_h + b_l$ olacak şekilde (a_h, a_l) ve (b_h, b_l) çiftlerine parçalanır. Bu çiftler basamaklı toplama (cascaded summation) yöntemi ile derecelerine göre toplanarak çift hassasiyetli toplama işlemi algoritması oluşturulur (Ogita vd. 2005). Basamaklı toplama yönteminde TwoSum toplama algoritması kullanılmaktadır.

Teorem 3.2. *a ve b kayan noktalı sayılar olmak üzere, double hassasiyette kayan noktalı sayı sisteminde herhangi bir β tabanında yukarı ve aşağı taşma olmaması koşuluyla TwoSum Algoritması, s ve c sayılarını $s+c = a+b$ olacak şekilde kesin olarak hesaplar.*

Algorithm 1 TwoSum Algoritması

$s \leftarrow RN(a+b)$
$a' \leftarrow RN(s-b)$
$b' \leftarrow RN(s-a')$
$\delta_a \leftarrow RN(a-a')$
$\delta_b \leftarrow RN(b-b')$
$c \leftarrow RN(\delta_a + \delta_b)$
return (s, c)

TwoSum toplamı kayan noktalı toplama işlemi sonucu oluşan yuvarlama hatası c'yi hesaplayan bir algoritmadır (Wittig 2008). Basamaklı toplama yöntemi ile TwoSum algoritması iki kere uygulanarak büyük dereceli değişkenler (a_h, b_h) ve küçük dereceli değişkenler (a_l, b_l) ayrı ayrı toplanır (Muller vd. 2010).

(a_h, b_h) ve (a_l, b_l) çiftleri ile TwoSum toplamı sonucunda,

$$\begin{aligned} x_h + x_l &= a_h + b_h \\ y_h + y_l &= a_l + b_l \end{aligned} \tag{3.49}$$

olacak şekilde (x_h, x_l) ve (y_h, y_l) çiftleri elde edilir.

(3.49)'da verilen TwoSum algoritması ile elde edilen çiftlerde en büyük dereceli değişken x_h olduğundan çift hassasiyetli toplama işlemi sonucunun büyük dereceli değişkeni $s = x_h$ olarak tanımlanır.

Küçük dereceli tüm değişkenler $(x_l,\ y_h,\ y_l)$ kayan nokta aritmetiğinde toplanarak çift hassasiyetli toplama işleminin yuvarlama hatası c olarak tanımlanır. Böylelikle $(a+b)$ işleminin sonucu $s + c = a + b$ olacak şekilde kesin olarak hesaplanır (Muller vd. 2010). Çift hassasiyetli toplama işleminin sonucu,

$$\begin{aligned} s &= x_h \\ c &= y_l + x_l + y_h \end{aligned} \tag{3.50}$$

ile hesaplanır ve (s, c) çifti ile tanımlanır.

Algorithm 2 Çift hassasiyetli toplama-çıkarma işlemi algoritması

Require: $(a_h, a_l) \leftarrow Split(a)$
$(b_h, b_l) \leftarrow Split(b)$
$(x_h, x_l) \leftarrow TwoSum(a_h, b_h)$
$(y_h, y_l) \leftarrow TwoSum(a_l, b_l)$
$s \leftarrow x_h$
$c \leftarrow RN(y_l + x_l + y_h)$
return (s, c)

Algoritma-2 her biri toplam 105-bit hassasiyetli (a_h, a_l) ve (b_h, b_l) sayı çiftlerini toplayan ve 105-bit hassasiyetli (s, c) çiftli sonucunu üreten çift hassasiyetli toplama-çıkarma işlemi fonksiyonunu oluşturur. Çift hassasiyetli toplama-çıkarma işlemi fonksiyonu,

$$Sum(double\ a_h,\ double\ a_l,\ double\ b_h,\ double\ b_l,\ double\ *s,\ double\ *c) \tag{3.51}$$

olarak tanımlanır.

3.1.3. Çift hassasiyetli çarpma işlemi

a ve b sayıları F kümesinde tanımlı β tabanında ve 53-bit double hassasiyetli kayan noktalı sayılar olsun. a ve b sayılarının kayan noktalı çarpma işleminin sonucu ve yuvarlama hatası sırasıyla 53-bit double hassasiyetli m ve c değişkenleri olarak tanımlansın. Çift hassasiyetli çarpma işlemi a ve b sayılarının çarpımını,

$$\begin{aligned} m &= RN(a \times b) \\ m + c &= a \times b \end{aligned} \tag{3.52}$$

olacak şekilde kesin olarak hesaplar (Dekker 1971).

Dekker (1971) çift hassasiyetli çarpma işlemi için polinomal bir yaklaşım geliştirmiştir (Muller vd. 2010). Çift hassasiyetli toplama işlemini oluşturmak için öncelikle a ve b sayıları ikiye ayırma algoritmasıyla $a = a_h + a_l$ ve $b = b_h + b_l$ olacak şekilde (a_h, a_l) ve (b_h, b_l) çiftlerine parçalanır. Böylelikle $(a_h + a_l) \times (b_h + b_l)$ kayan noktalı çarpma işlemi polinomal olarak,

$$(a_h + a_l) \times (b_h + b_l) = a_h \cdot b_h + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l \tag{3.53}$$

eşitliği ile ifade edilir.

Çift hassasiyetli çarpma işleminde literatürde Veltkamp ayırması olarak bilinen bir algoritma kullanılmaktadır. Veltkamp ayırma algoritması 53-bit hassasiyete sahip bir x kayan noktalı sayısını $x = x_h + x_l$ olmak üzere her bir parçası 26-bit hassasiyete sahip $(x_h,\ x_l)$ çiftine ayırır (Hida vd. 2001). Ayrılan çiftler çift hassasiyetli çarpma algoritması içerisinde dinamik olarak kullanılır. Veltkamp algoritması ile ikiye ayrılan her bir sayının hassasiyeti sistemin hassasiyetinin yaklaşık yarısı kadar olur (Graillat vd. 2020).

Algorithm 3 Veltkamp ikiye ayırma algoritması

Require: $C = \beta^s + 1$

$\gamma \leftarrow RN(C \cdot x)$

$\delta \leftarrow RN(x - \gamma)$

$x_h \leftarrow RN(\gamma + \delta)$

$x_l \leftarrow RN(x - x_h)$

return (x_h, x_l)

Hassasiyet p ve taban β olmak üzere Veltkamp ayırma algoritmasında $s = \lceil p/2 \rceil$ için $C = \beta^s + 1$ sabit sayısı kullanılır. 53-bit double hassasiyet için $p = 53$ ve $\beta = 2$ için $s = 27$ olarak hesaplanır ve $C = 134217729$ sabit sayısı elde edilir (Muller vd. 2010). Veltkamp ayırması ile (3.53)'de verilen büyük dereceli değişkenler,

$$\begin{aligned} a_h &= a_{h_1} + a_{l_1} \\ b_h &= b_{h_1} + b_{l_1} \end{aligned} \tag{3.54}$$

olacak şekilde (a_{h_1}, a_{l_1}) ve (b_{h_1}, b_{l_1}) çiftlerine ayrılır.

Veltkamp algoritması ile (3.54)'de elde edilen çiftler kullanılarak (3.53)'de verilen $(a_h \cdot b_h)$ çarpımı için ikinci polinomal açılım yapılır. Böylelikle $(p,q,t,z) \in F$ olmak üzere $(a \times b)$ kayan noktalı çarpımı polinomal olarak,

$$\begin{aligned} m &= a_h \cdot b_h \\ y &= (a_h \cdot b_l) + (a_l \cdot b_h) + (a_l \cdot b_l) \\ p &= -m + (a_{h_1} \cdot b_{h_1}) \\ q &= p + (a_{h_1} \cdot b_{l_1}) \\ t &= q + (a_{l_1} \cdot b_{h_1}) \\ z &= t + (a_{l_1} \cdot b_{l_1}) \\ c &= z + y \end{aligned} \tag{3.55}$$

eşitlikleri ile ifade edilir.

(3.55)'de verilen eşitliklerde en büyük dereceli değişken m ve yuvarlama hatası c olduğundan çift hassasiyetli çarpma işleminin büyük dereceli değişkeni m ve küçük dereceli değişkeni c olarak tanımlanır.

Sonuç olarak çift hassasiyetli çarpma işlemi ile $(a \times b)$ işlemi $m + c = a \times b$ olacak şekilde kesin olarak hesaplanır. Algoritma-4 her biri toplam 105-bit hassasiyetli (a_h, a_l) ve (b_h, b_l) sayı çiftini çarpan ve toplam 105-bit hassasiyetli (m,c) çiftli sonucunu üreten çift hassasiyetli çarpma işlemi fonksiyonunu oluşturur. Çift hassasiyetli çarpma işlemi fonksiyonu,

$$Mult(double\ a_h,\ double\ a_l,\ double\ b_h,\ double\ b_l,\ double\ *m,\ double\ *c) \tag{3.56}$$

olarak tanımlanır.

Algorithm 4 Dekker polinomal çarpma işlemi algoritması

Require: $s = \lceil p/2 \rceil$

$(a_h, a_l) \leftarrow Split(a)$

$(b_h, b_l) \leftarrow Split(b)$

$(a_{h_1}, a_{l_1}) \leftarrow VeltkampSplit\,(a_h)$

$(b_{h_1}, b_{l_1}) \leftarrow VeltkampSplit\,(b_h)$

$m \leftarrow RN(a_h \cdot b_h)$

$y \leftarrow RN(a_h \cdot b_l) + RN(a_l \cdot b_h) + RN(a_l \cdot b_l)$

$p \leftarrow RN(-m + RN(a_{h_1} \cdot b_{h_1}))$

$q \leftarrow RN(p + RN(a_{h_1} \cdot b_{l_1}))$

$t \leftarrow RN(q + RN(a_{l_1} \cdot b_{h_1}))$

$z \leftarrow RN(t + RN(a_{l_1} \cdot b_{l_1}))$

$c \leftarrow RN(z + y)$

return (m, c)

3.1.4. Çift hassasiyetli bölme işlemi

a ve b sayıları F kümesinde tanımlı β tabanında ve double hassasiyetli kayan noktalı sayılar olsun. a ve b sayılarının kayan noktalı bölme işleminin sonucu ve yuvarlama hatası sırasıyla 53-bit double hassasiyetli d ve c değişkenleri olarak tanımlansın. Çift hassasiyetli bölme algoritması kesire uygulanan iteratif bir yaklaşım gerektirir ve en karmaşık çift hassasiyetli işlem olarak kabul edilir. Dekker, kesir ifadesini iteratif olarak hesaplamak yerine $1/(a+b)$ bölüm açılımını kullanarak çift hassasiyetli bölme işlemi algoritmasını kolaylaştırmıştır (Bebee 2017).

$$\frac{1}{a+b} = \frac{1}{a}\left(1 - \frac{b}{a} + \left(\frac{b}{a}\right)^2 - \cdots\right) \tag{3.57}$$

Çift hassasiyetli bölme işlemi algoritmasına $a = a_h + a_l$ ve $b = b_h + b_l$ olacak şekilde her bir sayıyı ikiye ayırma algoritması ile (a_h, a_l) ve (b_h, b_l) çiftlerine parçalayarak başlanır. Çift hassasiyetli bölme işlemi sonucunun büyük dereceli değişkeni için a_h/b_h sayısı başlangıç değeri olarak seçilir ve d değişkeni ile tanımlanır.

(3.57)'de verilen bölüm açılımı formülü kullanılarak,

$$\begin{aligned} d &= a_h/b_h \\ a/b &= (a_h/b_h) + (a/b - a_h/b_h) \\ &= d + (a/b - d) \\ &= d + (a - d \cdot b)/b \end{aligned} \tag{3.58}$$

elde edilir.

(3.58)'de verilen $(a/b - a_h/b_h)$ ifadesi (a/b) işleminin yuvarlama hatasına eşittir ve c değişkeni ile tanımlanır. Eşitlikler düzenlenip $a = a_h + a_l$ ve $b = b_h + b_l$ yerine konursa ve (3.57)'de verilen bölüm açılımı a_h/b_h kesrine uygulanıp düşük dereceli çarpan terimi ihmal edilirse yuvarlama hatası c yaklaşık olarak,

$$\begin{aligned} a/b &= d + (a_h + a_l - d \cdot b_h - d \cdot b_l)/b \\ &= d + ((a_h + a_l - d \cdot b_h - d \cdot b_l)/b_h) \cdot (1 - b_l/b_h + (b_l/b_h)^2 - \cdots) \\ &= d + c \end{aligned} \tag{3.59}$$

$$\begin{aligned} c &= ((a_h + a_l - d \cdot b_h - d \cdot b_l)/b_h) \cdot (1 - b_l/b_h + (b_l/b_h)^2 - \cdots) \\ c &\approx (a_h - d \cdot b_h + a_l - d \cdot b_l)/b_h \end{aligned} \tag{3.60}$$

formülü ile hesaplanır.

(3.60)'da verilen $d \cdot b_h$ terimi $u = d \cdot b_h$ olarak tanımlanırsa ve çift hassasiyetli çarpma işlemi uygulanıp $u = u_h + u_l$ olacak şekilde bir çift olarak temsil edilirse yuvarlama hatası c için daha hassas bir sonuç elde edilir (Bebee 2017).

$$\begin{aligned} u &= d \cdot b_h \\ &= u_h + u_l \\ c &= ((((a_h - u_h) - u_l) + a_l) - d \cdot b_l)/b_h \end{aligned} \tag{3.61}$$

Sonuç olarak çift hassasiyetli bölme işlemi (a/b) işlemini $d + c = a/b$ olacak şekilde kesin olarak hesaplar. Algoritma-5 her biri toplam 105-bit hassasiyetli (a_h, a_l) ve (b_h, b_l) sayı çiftini bölen ve toplam 105-bit hassasiyetli (d, c) çiftli sonucunu üreten çift hassasiyetli bölme işlemi fonksiyonunu oluşturur. Çift hassasiyetli bölme işlemi fonksiyonu,

$$Division(double\ a_h,\ double\ a_l,\ double\ b_h,\ double\ b_l,\ double\ *d,\ double\ *c) \tag{3.62}$$

olarak tanımlanır.

Algorithm 5 Dekker bölme işlemi algoritması

Require: $(a_h, a_l) \leftarrow Split(a)$

$(b_h, b_l) \leftarrow Split(b)$

$d \leftarrow RN(a_h/b_h)$

$(u_h, u_l) \leftarrow Multiplication(d,\ 0,\ b_h,\ 0)$

$p \leftarrow RN(a_h - u_h)$

$q \leftarrow RN(p - u_l)$

$t \leftarrow RN(q + a_l))$

$z \leftarrow RN(t - RN(d \cdot b_l))$

$k \leftarrow RN(z/b_h)$

$c \leftarrow k$

return (d, c)

3.1.5. Çift hassasiyetli karekök işlemi

a sayısı β tabanında ve 53-bit double hassasiyetli kayan noktalı bir sayı olsun. Dekker tarafından geliştirilen çift hassasiyetli karekök algoritması $b = \sqrt{a}$ işleminin sonucunu $c = c_h + c_l$ olmak üzere $c_h + c_l = \sqrt{a_h + a_l}$ olacak şekilde kesin olarak hesaplar.

Dekker çift hassasiyetli karekök algoritmasında fonksiyonun yakınsama hızı düşük olduğu için tek adımlı Newton-Raphson iterasyonu kullanılmaktadır (Bebee 2017). Newton-Raphson iterasyonu $b = \sqrt{a}$ işlemi için $F(b) = b^2 - a$ fonksiyonu ile tanımlanır.

Çift hassasiyetli karekök algoritmasını oluşturmak için öncelikle a sayısı ikiye ayırma algoritmasıyla $a = a_h + a_l$ olacak şekilde (a_h, a_l) çiftine parçalanır. İterasyon için bir b_h değişkeni tanımlanır ve $b_h = \sqrt{a_h}$ değeri başlangıç tahmini olarak seçilir. Tam kare açılımı ile b_h tahmininin düzeltmesi b_l=$\frac{1}{2}(b_h + a/b_h)$ olarak tanımlanırsa,

$$\begin{aligned} b_h &= \sqrt{a_h} \\ b_l &= \frac{1}{2}(b_h + a/b_h) \\ c_h &= b_h \\ c_l &= b_l - b_h \end{aligned} \tag{3.63}$$

elde edilir.

(3.63)'de başlangıç tahmini c_h ve yuvarlama hatası c_l değişkeni olarak tanımlanır.

Başlangıç tahmininin düzeltmesi c_l karekök işleminin yuvarlama hatasına karşılık gelmektedir. Yuvarlama hatası tek adımlı Newton-Raphson iterasyonu kullanılarak,

$$\begin{aligned} c_l &= \frac{1}{2}(b_h + a/b_h) - b_h \\ &= \frac{1}{2}(a/b_h - b_h) \\ &= (a - b_h^2)/(2b_h) \end{aligned} \tag{3.64}$$

eşitliği ile hesaplanır.

(3.64)'de verilen b_h^2 terimi $u = b_h^2$ olarak tanımlanırsa ve çift hassasiyetli çarpma işlemi uygulanıp u= u_h+u_l olacak şekilde bir çift olarak temsil edilirse yuvarlama hatası c_l,

$$\begin{aligned} c_l &= (a - u)/2b_h \\ &= (a_h - u_h - u_l + a_l)/(2b_h) \\ &= (a_h - u_h - u_l + a_l)/(2c_h) \end{aligned} \tag{3.65}$$

daha hassas hesaplanır (Bebee 2017).

Algorithm 6 Dekker çift hassasiyetli karekök işlemi algoritması

Require: $(a_h, a_l) \leftarrow Split(a)$

$c_h \leftarrow RN\left(\sqrt{a_h}\right)$

$(u_h, u_l) \leftarrow Multiplication(c_h,\ 0,\ c_h,\ 0)$

$p \leftarrow RN(a_h - u_h)$

$q \leftarrow RN(p - u_l)$

$t \leftarrow RN(q + a_l)$

$z \leftarrow RN(0.5 \cdot RN(t/c_h))$

$c_l \leftarrow z$

return (c_h, c_l)

Sonuç olarak çift hassasiyetli karekök işlemi $b = \sqrt{a}$ işlemini $c_h + c_l = \sqrt{a_h + a_l}$ olacak şekilde kesin olarak hesaplar. Algoritma-6 toplam 105-bit hassasiyetli (a_h, a_l) sayı çiftinin karekökünü alan ve toplam 105-bit hassasiyetli (c_h, c_l) çiftli sonucunu üreten çift hassasiyetli karekök işlemi fonksiyonunu oluşturur. Çift hassasiyetli karekök işlemi fonksiyonu,

$$Sqrt(double\ a_h,\ double\ a_l,\ double\ *c_h,\ double\ *c_l) \tag{3.66}$$

olarak tanımlanır.

3.1.6. Çift hassasiyetli skaler çarpım

$\vec{a}$ ve $\vec{b}$ n-boyutlu vektörler olmak üzere $\vec{a}$ ve $\vec{b}$ vektörlerinin skaler çarpımı,

$$\vec{a} \cdot \vec{b} = \sum_{k=0}^{n-1} a_k \cdot b_k \tag{3.67}$$

olarak tanımlanır.

(3.67)'de verildiği üzere skaler çarpım işlemi çarpma ve toplama işlemlerinden oluşan bir hesaplamadır. Dolayısıyla çift hassasiyetli skaler çarpım işlemi için öncelikle vektörlerin her bir bileşeni ikiye ayırma algoritmasıyla çiftlere parçalanır ve çift hassasiyetli çarpma ve toplama işlemleri kullanılır. (3.67)'de verilen a_k ve b_k vektör bileşenleri ikiye ayırma algoritmasıyla,

$$\begin{aligned} a_k &= a_{h_k} + a_{l_k} \\ b_k &= b_{h_k} + b_{l_k} \end{aligned} \tag{3.68}$$

(a_{h_k}, a_{l_k}) ve (b_{h_k}, b_{l_k}) çiftlerine parçalanır.

(3.67)'de verilen aritmetik işlemler gereği (a_{h_k}, a_{l_k}) ve (b_{h_k}, b_{l_k}) çiftleriyle n adet çift hassasiyetli çarpma işlemi ve $(n-1)$ adet çift hassasiyetli toplama işlemi uygulanarak çift hassasiyetli skaler çarpım işlemi oluşturulur. $\vec{a}$ ve $\vec{b}$ üç boyutlu iki vektör olsun. Bu iki vektörün çift hassasiyetli skaler çarpım işlemi için öncelikle $\vec{a}$ ve $\vec{b}$ vektörlerinin üç bileşeni ikiye ayırma algoritması ile çiftlere ayrılır. Elde edilen çiftlerle 3 adet çift hassasiyetli çarpma ve 2 adet çift hassasiyetli toplama işlemi uygulanır.

Algorithm 7 Çift hassasiyetli skaler çarpım işlemi algoritması

Require: $(a_{h_k}, a_{l_k}), (b_{h_k}, b_{l_k}) \leftarrow Split(a_k), Split(b_k)$

$(u_h, u_l) \leftarrow Mult\left(a_{h_0},\ a_{l_0},\ b_{h_0},\ b_{l_0}\right)$

$(v_h, v_l) \leftarrow Mult\left(a_{h_1},\ a_{l_1},\ b_{h_1},\ b_{l_1}\right)$

$(t_h, t_l) \leftarrow Mult\left(a_{h_2},\ a_{l_2},\ b_{h_2},\ b_{l_2}\right)$

$(k_h, k_l) \leftarrow Sum\left(u_h,\ u_l,\ v_h,\ v_l\right)$

$(p_h, p_l) \leftarrow Sum\left(t_h,\ t_l,\ k_h,\ k_l\right)$

$s \leftarrow p_h$

$c \leftarrow p_l$

return (s, c)

3.2. Hareket denkleminin RK4 metodu ile nümerik integrasyonu

İki cisim probleminin hareket denklemi eylemsiz koordinat sisteminde büyük kütleli merkezi bir cisim etrafında dolanan küçük kütleli cismin yörüngesini belirlemek için dördüncü dereceden Runge-Kutta metoduyla çözülebilir. Hareket denklemi vektörel, üç boyutlu, ikinci dereceden adi diferansiyel bir denklemdir. İki cisim probleminin hareket denklemi,

$$\ddot{\vec{r}} = \frac{-\mu \vec{r}}{r^3} \tag{3.69}$$

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3.70}$$

eşitliği ile ifade edilir.

$\vec{r}$ konum vektörü , $\vec{v}$ hız vektörü ve $\vec{a}$ ivme vektörü olmak üzere,

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \qquad \vec{a} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} \tag{3.71}$$

olarak tanımlanır.

(3.69)'da verilen hareket denklemi $(x,\ y,\ z)$ kartezyen koordinatları kullanılarak,

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{bmatrix} \frac{-\mu x}{r^3} \\ \frac{-\mu y}{r^3} \\ \frac{-\mu z}{r^3} \end{bmatrix} \tag{3.72}$$

olarak ifade edilebilir.

Yörünge integrasyonu için öncelikle iki cisim problemi t_0 başlangıç anında verilen (x_0, y_0, z_0) ve $(v_{x_0}, v_{y_0}, v_{z_0})$ başlangıç koşulları kullanılarak başlangıç değer problemine dönüştürülür. Hareket denklemi ikinci dereceden adi diferansiyel bir denklem olduğundan iki adet birinci dereceden denklem sistemine indirgenir. Hız vektörü $\dot{\vec{r}} = \vec{v}$ ve ivme vektörü $\dot{\vec{v}} = \vec{a}$ olmak üzere birinci denklem sistemi hız vektörü ve ikinci denklem sistemi ivme vektörü ile tanımlanır.

Birinci denklem sistemi $(v_{x_0}, v_{y_0}, v_{z_0})$ başlangıç koşulları kullanılarak hız vektörünün bileşenleri şeklinde,

$$\frac{dx}{dt} = v_{x_0} \qquad \frac{dy}{dt} = v_{y_0} \qquad \frac{dz}{dt} = v_{z_0} \tag{3.73}$$

olarak tanımlanır.

İkinci denklem sistemi (x_0, y_0, z_0) başlangıç koşulları kullanılarak ivme vektörünün bileşenleri şeklinde ,

$$\begin{aligned} a_x &= \frac{dv_x}{dt} = \frac{-\mu x_0}{r^3} \\ a_y &= \frac{dv_y}{dt} = \frac{-\mu y_0}{r^3} \\ a_z &= \frac{dv_z}{dt} = \frac{-\mu z_0}{r^3} \end{aligned} \tag{3.74}$$

olarak tanımlanır.

Dördüncü dereceden Runge-Kutta metodu hareket denkleminin vektörel formuna uygun olarak modifiye edilebilir (Voesenek 2008). Bunun için (2.30)'da verilen k eğim katsayıları (3.73) ve (3.74) eşitlikleri kullanılarak vektör fonksiyonu şeklinde tanımlanır. $(i = 0, 1, 2 \ldots, n)$ olmak üzere k eğim katsayıları (3.73) ve (3.74) eşitliklerindeki her bir koordinat bileşeni için ayrı ayrı hesaplanır ve vektörel olarak $\vec{k}_{v_{i+1}}$ ve $\vec{k}_{r_{i+1}}$ ile gösterilir.

$\vec{k}_{v_{i+1}}$ katsayıları (3.74)'de verilen ivme vektörünün bileşenleri kullanılarak,

$$\begin{aligned} \vec{k}_{1_{v_{i+1}}} &= \vec{a}\,(\vec{r}_i) \\ \vec{k}_{2_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + \frac{h}{2}\vec{k}_{1_{r_{i+1}}}\right) \\ \vec{k}_{3_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + \frac{h}{2}\vec{k}_{2_{r_{i+1}}}\right) \\ \vec{k}_{4_{v_{i+1}}} &= \vec{a}\left(\vec{r}_i + h\vec{k}_{3_{r_{i+1}}}\right) \end{aligned} \tag{3.75}$$

$\vec{k}_{r_{i+1}}$ katsayıları (3.73)'de verilen başlangıç hız vektörünün bileşenleri kullanılarak,

$$\begin{aligned} \vec{k}_{1_{r_{i+1}}} &= \vec{v}\,(\vec{v}_i) \\ \vec{k}_{2_{r_{i+1}}} &= \vec{v}\left(\vec{v}_i + \frac{h}{2}\vec{k}_{1_{v_{i+1}}}\right) \\ \vec{k}_{3_{r_{i+1}}} &= \vec{v}\left(\vec{v}_i + \frac{h}{2}\vec{k}_{2_{v_{i+1}}}\right) \\ \vec{k}_{4_{r_{i+1}}} &= \vec{v}\left(\vec{v}_i + h\vec{k}_{3_{v_{i+1}}}\right) \end{aligned} \tag{3.76}$$

fonksiyonları ile ifade edilir.

(3.75) ve (3.76)'da verilen $\vec{k}_{v_{i+1}}$ ve $\vec{k}_{r_{i+1}}$ eğim katsayıları sırasıyla birbiri içerisinde kullanılmaktadır. Dolayısıyla katsayılar algoritmik olarak verilmiştir ve döngü yardımıyla hesaplanmalıdır.

Sonuç olarak (2.29)'da verilen RK4 formülü vektörel eğim katsayıları kullanılarak yeniden tanımlanabilir. Dördüncü dereceden Runge-Kutta metodu vektörel olarak,

$$\begin{aligned}\vec{v}_{i+1} &= \vec{v}_i + \frac{h}{6}\left(\vec{k}_{1_{v_{i+1}}} + 2\vec{k}_{2_{v_{i+1}}} + +2\vec{k}_{3_{v_{i+1}}} + \vec{k}_{4_{v_{i+1}}}\right)\\ \vec{r}_{i+1} &= \vec{r}_i + \frac{h}{6}\left(\vec{k}_{1_{r_{i+1}}} + 2\vec{k}_{2_{r_{i+1}}} + +2\vec{k}_{3_{r_{i+1}}} + \vec{k}_{4_{r_{i+1}}}\right)\end{aligned} \tag{3.77}$$

formülleri ile tanımlanır.

3.3. Çift hassasiyetli aritmetik işlemler ile nümerik integrasyon

Çift hassasiyetli kayan nokta aritmetiğinin verimliliğini ve avantajını göstermek amacıyla iki cisim probleminde merkezi cisim Güneş ve yörüngedeki cisim Jüpiter seçilerek hareket denklemi RK4 metodu ile iki aşamada integre edilmiştir. Birinci adımda integrasyon hesabı IEEE 754-2008 standartında tanımlı 53-bit double hassasiyetli aritmetik kullanılarak yapılmıştır. İntegrasyonun başlangıç anı $t_0 = 2458274.5$ JD (5 Haziran 2018) ve adım aralığı $t = 6283$ gün olarak seçilmiştir. İntegrasyon için Jüpiter'in au cinsinden başlangıç konum ve hız vektörü değerleri ve gravitasyonel parametre NASA Horizon'dan alınmıştır ve 53-bit double veri tipinde değişkenler olarak tanımlanmıştır. Jüpiter'in Güneş merkezli yörüngesini elde etmek için hareket denklemi, $\vec{k}$ katsayıları i inci adımdaki konum ve hız vektörlerine karşılık gelecek şekilde (3.77)'de verilen RK4 formülleri kullanılarak 6283 gün ileri integre edilmiştir. RK4 şemasında tüm parametreler sabit tutularak $h = 0.1$, $h = 0.01$ ve $h = 0.001$ adım boyutlarında integrasyon tekrarlanmıştır. En optimal sonuçlar $h = 0.01$'de elde edildiği için integrasyonun sabit adım boyutu olarak seçilmiştir.

İkinci adımda birinci adımda verilen RK4 integrasyon şeması içerisindeki tüm aritmetik işlemler ile bu aritmetik işlemlere karşılık gelen çift hassasiyetli aritmetik işlem fonksiyonları yer değiştirilmiştir. İntegrasyon şemasının girdi parametreleri; başlangıç konum ve hız vektörü değerleri, gravitasyonel parametre, integrasyonun adım boyutu ve sayısıdır. Çift hassasiyetli aritmetik işlemlerin uygulanabilmesi için şemanın tüm girdileri ikiye ayırma algoritması ile parçalanarak 53-bit double hassasiyetli büyük ve küçük dereceli değişkenler olarak gruplandırılmıştır. Çift hassasiyetli aritmetik işlemlerle manipüle edilen RK4 şeması ile hareket denklemi 6283 gün ileri integre edilmiştir. İntegrasyon sonucunda Jüpiter'in Güneş merkezli 6283 günlük büyük ve küçük dereceli konum ve hız vektörü değerleri elde edilmiştir.

Çizelge 3.2. Büyük ve küçük dereceli konum ve hız vektörü değerleri

Çiftler	x	y	z
r	$-3.460167504309613E+00$	$-4.149454064629457E+00$	$9.465721330038770E-02$
v	$5.709741990408655E-03$	$-4.481465873394258E-03$	$-1.091471606521913E-04$
r_l	$-4.862578020000000E-17$	$-1.2690770530000000E-16$	$-4.757454799000000E-18$
v_l	$-2.100430652000000E-19$	$4.062370784000000E-19$	$1.520065599999999E-21$

Çizelge 3.2'de başlangıç konum ve hız vektörü değerlerinin toplam 105-bit hassasiyetli büyük ve küçük dereceli çiftleri verilmiştir. Bu çalışmada RK4 integrasyon şemasının girdisi olarak tanımlanan başlangıç koşulu değerlerinden birisi ikiye ayırma yöntemini algoritmik olarak incelemek için seçilebilir.

Jupiter'in 5 Haziran 2018 tarihinde Güneş merkezli kartezyen X koordinatı NASA Horizon'dan alınan $3.460167504309613E+00$ sayısıdır. Bu sayı double hassasiyetli x değişkeni olarak tanımlanırsa 15. basamaktan itibaren sıfır kullanılarak orijinal sayı olarak kabul edilebilir. En yakına yuvarlama modunda x sayısı (3.78) eşitliğinde gösterildiği gibi yuvarlanır ve yuvarlanan sayı double veri tipindeki x_h değişkeni olarak tanımlanır.

$$\begin{aligned} x &= 3.46016750430961\boxed{3}0000000000 \\ x_h &= 3.46016750430961\boxed{2}9513742198 \end{aligned} \tag{3.78}$$

(3.78)'de verilen x ve x_h değerlerinin 15. basamaktan itibaren farklılaşmaya başladığı görülmektedir. Dolayısıyla (3.45)'de verilen algoritmaya göre $m = 15$, $I_1 = 30000000000$ ve $I_2 = 29513742198$ sayıları elde edilir. Yuvarlama hatası (3.46)'da verilen denklem kullanılarak $x_l = 4.86257802 \times 10^{-17}$ olarak hesaplanır ve çift hassasiyetli sayı çiftinin küçük dereceli değişkeni olarak tanımlanır. Çizelge 3.2'de verilen 53-bit double hassasiyetli başlangıç konum ve hız vektörü değerleri (r, v) integrasyon şemasında virgülden sonra 15 basamak temsil edilmektedir. Bu sebeple çift hassasiyetli sayı çiftlerinin büyük dereceli değişkenleri olarak tanımlanmıştır.

4. BULGULAR VE TARTIŞMA

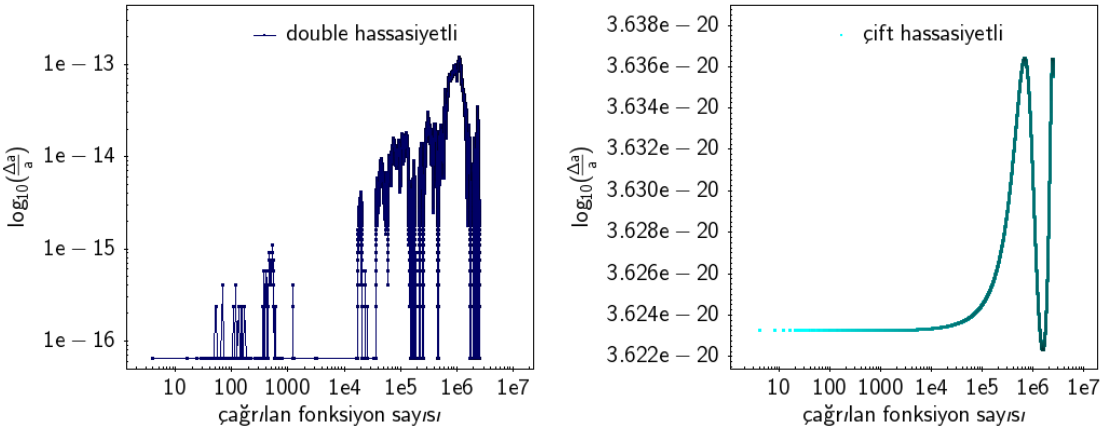
4.1. Analitik çözüm ile nümerik karşılaştırmalar

Çift hassasiyetli ve double hassasiyetli aritmetik ile gerçekleştirilen integrasyonlardan elde edilen sonuçları karşılaştırmak amacıyla iki cisim probleminin hareket sabitleri analitik çözüm olarak kullanılmıştır. Hareket sabitleri integrasyonun t_0 başlangıç anındaki konum ve hız vektörü değerleri kullanılarak (2.14), (2.19), (2.22) ve (2.23) denklemlerinin 80-bit long double hassasiyetli çözümü ile elde edilmiştir. Jüpiter'in Güneş merkezli yörüngesinin hareket sabitleri long double hassasiyette,

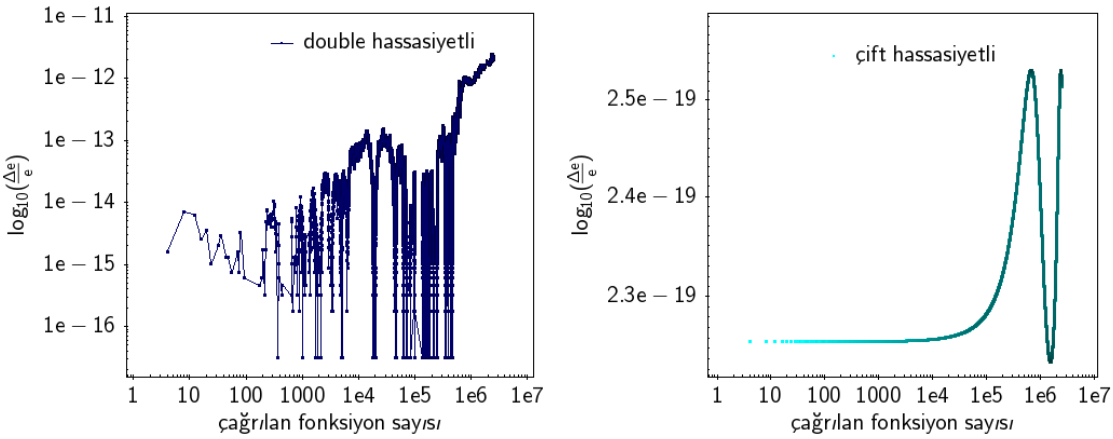
$$\begin{aligned} a &= 5.202735843552009 \\ e &= 4.880567975450338e-02 \\ h &= 3.920908437149964e-02 \\ E &= -2.846528747310223e-05 \end{aligned} \tag{4.79}$$

olarak hesaplanmıştır.

Her iki integrasyondan elde edilen sonuçlar ile çift hassasiyetli ve double hassasiyetli aritmetik kullanılarak (2.14), (2.19), (2.22) ve (2.23) denklemleri çözülmüştür ve integrasyonların her bir adımı için yarı büyük eksen uzunluğu, eksentrisite, açısal momentum ve mekanik enerji değerleri elde edilmiştir. Bu değerler (4.79)'da verilen long double hassasiyetli hareket sabitleriyle integrasyonun her bir adımı için rölatif olarak karşılaştırılarak yerel integrasyon hataları ölçülmüştür.



Şekil 4.8. Jüpiter'in yarı büyük eksen uzunluğu için yerel integrasyon hataları

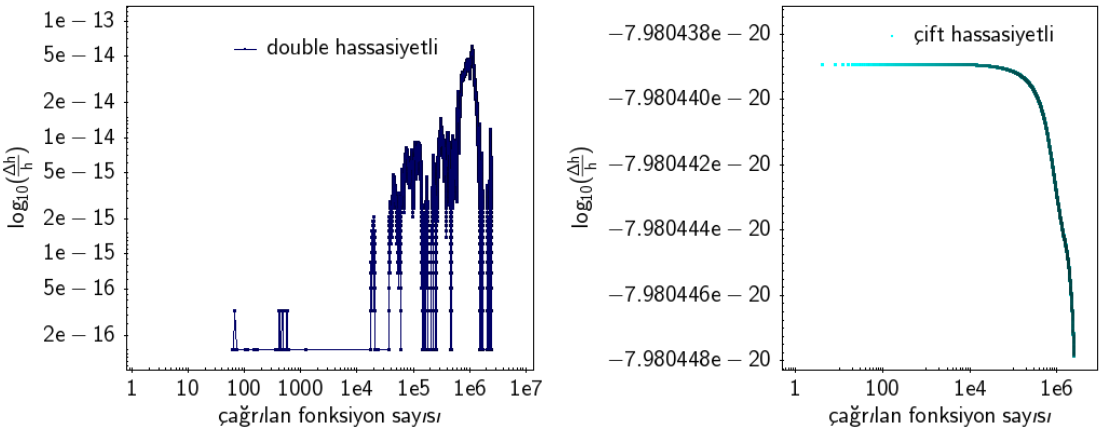


Şekil 4.9. Jüpiter'in eksentrisite değeri için yerel integrasyon hataları

Jüpiter'in çift hassasiyetli ve double hassasiyetli aritmetik ile hesaplanan yörünge sabitlerinin yerel integrasyon hataları, integrasyon için çağrılan ivme fonksiyonu sayısının fonksiyonu olarak tanımlanmıştır. Böylelikle integrasyonların doğruluğu Jüpiter'in Güneş etrafında 17 dolanımı sonrası hareket sabitlerinin rölatif hatası ölçülerek tahmin edilmiştir. Şekil 4.8 ve Şekil 4.9'da double hassasiyetli RK4 integrasyon şemasına çift hassasiyetli aritmetik işlemlerin uygulanması sonucunda Jüpiter'in yarı büyük eksen uzunluğu ve eksentrisite değerlerinin yerel integrasyon hatalarındaki azalma gösterilmektedir.

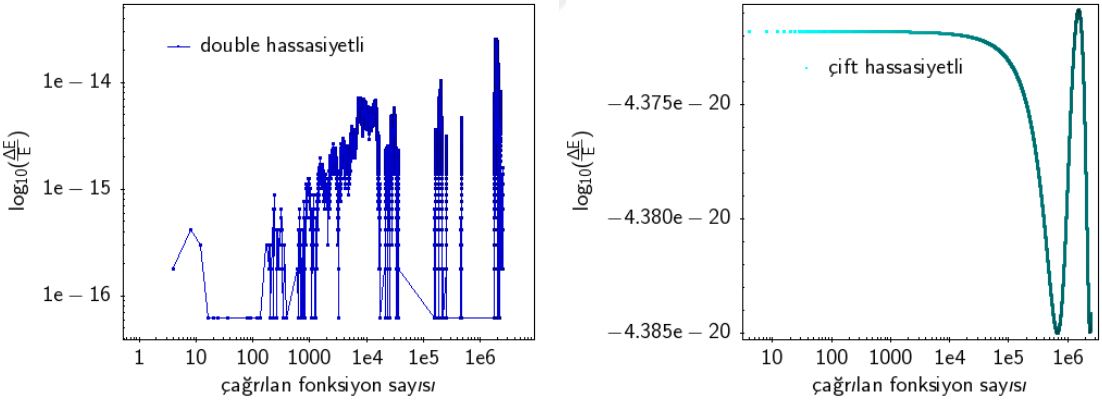
Çift hassasiyetli aritmetik ile yapılan RK4 integrasyon hesaplamasında double hassasiyetli hesaplamaya göre yarı büyük eksen uzunluğu ve eksentrisite değerlerinde ortalama 8 basamak kazanç elde edilmiştir. Dolayısıyla double hassasiyetli RK4 integrasyon şemasına çift hassasiyetli aritmetik işlemlerin uygulanması ile yerel integrasyon hatalarının yaklaşık 10^8 kat azaltıldığı sonucuna varılabilir.

Yüksek eksentrisiteye sahip cisimler yörünge boyunca hızlanır ve yavaşlar. Tek adımlı yöntemlerde adım boyutu yörünge boyunca sabit tutulduğu için yüksek eksentrisiteye sahip bir cismin yörüngesi tek adımlı bir yöntemle elde edilirse cisim hızlanırken ve yavaşlarken yörüngede kalması zorlaşır ve integrasyon hatası artar (Vallado 2013). Şekil 4.9'da çift hassasiyetli aritmetik ile hesaplanan eksentrisite değeri için kazanılan doğrulukta çift hassasiyetli aritmetikle hesaplanan Şekil 4.8'de verilen yarı büyük eksen uzunluğunun doğruluğuna göre 1 basamak kayıp olduğu görülmektedir. Bu durumun kullanılan tek adımlı nümerik integratörden ve Jüpiter'in görece yüksek eksentrisitesinden kaynaklandığı söylenebilir.



Şekil 4.10. Yörüngenin açısal momentum sabiti için yerel integrasyon hataları

İki cisim hareketi korunumlu bir sistemde gerçekleştiği için enerjinin integratör tarafından korunması gerekir (Rein vd. 2019). IEEE 754-2008 double hassasiyetli aritmetik kullanan bir integratörde enerji değeri için beklenen yuvarlama hatası Brouwer yasası ile $N = 628318,5$ adım sayısı için $E_{floor} = 2^{-52} \times \sqrt{N}$ formülü ile $E_{floor} = 1.76 \times 10^{-13}$ olarak hesaplanmıştır.



Şekil 4.11. Yörüngenin mekanik enerji sabiti için yerel integrasyon hataları

Şekil 4.10 ve Şekil 4.11'de gösterildiği gibi çift hassasiyetli aritmetik uygulanan integrasyonda double hassasiyetli aritmetiğe göre açısal momentum ve mekanik enerji değerlerinde yaklaşık 7 basamak kazanç elde edilmiştir. Çift hassasiyetli yörünge integrasyonunda açısal momentum ve mekanik enerji değerlerinin yerel integrasyon hataları yörünge boyunca 10^{-20} mertebesinde oldukça düşük ve kararlıdır. Dolayısıyla 6283 gün integrasyon süresince Jüpiter için oldukça kararlı bir yörünge elde edildiği sonucuna varılabilir.

RK4 nümerik integrasyon metodu ile yörünge hesaplaması için CPU zamanları double hassasiyet için 2.643365 s. ve çift hassasiyet için 7.943968 s. olarak ölçülmüştür. Dolayısıyla $N = 628318,5$ adım sayılı bir RK4 integrasyon hesaplamasında çift hassasiyetli aritmetiğin double hassasiyetli aritmetiğe göre 3.01 kat daha maliyetli olduğu söylenebilir. Her iki aritmetikle gerçekleştirilen integrasyonlar için CPU zamanları Intel Core i5 işlemcili kişisel bilgisayarda ölçülmüştür.

Güneş Sistemi'nin uzun dönemli integrasyonlarında farklı şemalar ile yuvarlama hatalarının azaltılmasına yönelik birçok çalışma yapılmıştır (Quinn ve Tremaine 1989; Quinlan 1994; Rein ve Spiegel 2015). Bu çalışmalarda kullanılan metot içerisinde yuvarlama hatalarının en temel kaynağı olan toplama işlemlerine Kahan compensated summation algoritması uygulanarak yuvarlama hataları azaltılmaya çalışılmıştır. Uzun dönemli hesaplamalarda integrasyon şemasını çift hassasiyetli aritmetik işlemler gibi ek hesaplamalar içerecek şekilde tamamen modifiye etmek oldukça maliyetlidir (Fukushima 2001).

Çift hassasiyetli aritmetik işlemleri kısa dönemli simplektik bir integrasyon şemasındaki aritmetik işlemlerin tamamına uygulayarak yuvarlama hatalarının azaltılmasına yönelik en kapsamlı çalışma Fukushima (2001) tarafından yapılmıştır. Fukushima'nın uyguladığı çift hassasiyetli aritmetik işlemler FORTRAN-77 dilinde yazılmış olmakla birlikte integrasyon karşılaştırma testleri double veri tipinde iki cisim hareket sabitleri ile yapılmıştır. İntegrasyon sonucunda çift hassasiyetli aritmetikle elde edilen hareket sabitlerinde yaklaşık 8 basamak kazanç elde edilmiştir ve yörünge boyunca 10^{-16} mertebesinde doğruluk sağlanmıştır.

Bu tez çalışmasında C programlama dili kullanılmıştır ve integrasyon hesaplamalarının sonuçları kişisel bilgisayarda 128-bit quadruple hassasiyet erişimi olmadığı için 80-bit long double hassasiyetli analitik çözümler ile karşılaştırılmıştır. Euler tipi basit bir ekstrapolasyon yöntemi olan RK4 nümerik integratörü kullanılarak yörünge boyunca yaklaşık 10^{-20} mertebesinde doğruluk elde edilerek hareket sabitlerinde ortalama 8 basamak kazanç elde edilmiştir. Dolayısıyla sadece kullanılan yazılım dili değiştirilerek ve güncel çift hassasiyetli aritmetik işlem algoritmaları uygulanarak kısa dönemli iki cisim yörünge integrasyonunda simplektik bir algoritmaya göre yerel integrasyon hataları 10^4 kat daha fazla azaltılmıştır ve CPU zamanında avantaj elde edilmiştir.

5. SONUÇLAR

Bu çalışmada öncelikle kayan noktalı aritmetik işlemlerin yuvarlama hataları incelenmiştir ve yörünge integrasyonu hesaplamalarında biriken yuvarlama hatalarının çift hassasiyet tekniği ile nasıl azaltılabileceği açıklanmıştır. Hareket denkleminin çözümü için kullanılan dördüncü dereceden Runge-Kutta nümerik integrasyon şemasının tamamına çift hassasiyetli aritmetik işlemler uygulanarak yörünge integrasyonunda aritmetik işlemler sebebiyle kaybedilen hassasiyetlerin geri kazanabileceği ve böylelikle yuvarlama hatalarının azaltılabileceği gösterilmiştir.

Jüpiter'in 5 Haziran 2018 tarihinden itibaren 6283 gün boyunca Güneş merkezli yörüngesini elde etmek için test integratörü olarak kullanılan double hassasiyetli RK4 integrasyon şemasının tamamına çift hassasiyetli aritmetik işlemler uygulanmıştır. Böylelikle hareket denklemi iki aşamada integre edilmiştir. İki cisim probleminin analitik çözümü (hareket sabitleri) çift hassasiyetli ve double hassasiyetli yörünge hesaplamalarının sonuçlarını karşılaştırmak için kullanılmıştır. Çift hassasiyetli aritmetik işlemler sonucunda 105-bit hassasiyetli sonuçlar üretilmektedir ($\sim$ 31 basamak). Kişisel bilgisayarda integrasyon sonuçlarının karşılaştırılabileceği en yüksek hassasiyet 80-bit long double olduğu için analitik çözüm ile karşılaştırma testleri long double hassasiyet ile yapılmıştır.

Çift hassasiyetli aritmetik ile gerçekleştirilen yörünge integrasyonunda double hassasiyetli test integrasyonuna göre Jüpiter'in yörüngesi boyunca yarı büyük eksen uzunluğu, eksentrisite, açısal momentum ve mekanik enerji değerlerinde 3.01 kat CPU zamanı maliyetine karşın her bir hareket sabiti için ortalama 8 basamak kazanç elde edilmiştir. Çift hassasiyetli integrasyon sonucunda Jüpiter'in hareket sabitleri yörünge boyunca yaklaşık 20 basamak korunarak oldukça kararlı bir yörünge elde edilmiştir.

Sonuç olarak, çift hassasiyetli aritmetik işlemler ile manipüle edilen RK4 integrasyon şeması ile 53-bit double hassasiyetli değişkenler kullanılarak 105-bit hassasiyetli nümerik integrasyon gerçekleştirilmiş. Uygulanan teknik ile iki cisim hareket sabitlerinin yerel integrasyon hataları yaklaşık 10^8 kat azaltılmıştır. Bu çalışmada algoritmik olarak oldukça basit ve düşük dereceli bir nümerik yöntem kullanılmasına rağmen uygulanan teknik ve kullanılan yazılım dili farkı ile literatürde benzer kısa dönemli iki cisim yörünge integrasyonunda simplektik bir algoritmaya göre avantaj sağlanmıştır (Fukushima 2001).

6. KAYNAKLAR

Anzalone E. and Chai P. 2015. Numerical Integration Techniques in Orbital Mechanics Applications Retrieved from https://www.researchgate.net/publication/281108779 [Son erişim tarihi: 25.05.2021].

Antoñana, M. and Makazaga, J. and Murua, A. 2017. Reducing and monitoring round-off error propagation for symplectic implicit Runge-Kutta schemes from https://arxiv.org/abs/1702.03354 [Son erişim tarihi: 10.02.2017].

Bate, R.R. and Mueller, D.D. and White, J.E. 1971. Fundamentals of Astrodynamics. Dover Publications, New York, 449 p.

Battin, R.H. 1999. An Introduction to the Mathematics and Methods of Astrodynamics. American Institute of Aeronautics and Astronautics, Virginia, 785 p.

Beebe, N.H.F. 2017. The Mathematical-Function Computation Handbook. Springer Nature, Salt Lake, 990 p.

Boldo, S. 2014. Deductive Formal Verification: How To Make Your Floating-Point Programs Behave. Phd Thesis, Paris University, Paris, 80 p.

Boulet D.L. 1991. Methods of Orbit Determination for the Micro Computer. Willmann Bell, Virginia, 559 p.

Brouwer, D. 1937. On the Accumulation of Errors in Numerical Integration. *The Astronomical Journal*, 46(16): 149-153.

Chapra, S. C. and Canale, R. P. 2010. Numerical Methods for Engineers. McGraw-Hill, New York, 955 p.

Dekker, T.J. 1971. A Floating-Point Technique for Extending the Available Precision. *Numerische Mathematik*, 18: 224-242.

Fiorelli, S. and Vilmart G. 2017. Computing the Long-Term Evolution of the Solar System with Geometric Numerical Integrators. *Snapshots of modern mathematics from Oberwolfach*, 9(9):1-16.

Forsythe, G.E. and Malcolm, M.A. and Moler, C.B. 1977. Computer Methods for Mathematical Computations. Prentice-Hall, New Jersey, 255 p.

Fukushima, T. 1996. Reduction of Round-off Errors in the Extrapolation Methods and its Application to the Integration of Orbital Motion. *The Astronomical Journal*, 121(3): 1298-1301.

Fukushima, T. 2001. Reduction of Round-off Errors in Symplectic Integrators. *The Astronomical Journal*, 121(3): 1768-1775.

Graillat, S. and Lefèvre, V. and Muller, J. M. 2020. Alternative Split Functions and Dekker's Product. Retrieved from https://hal.archives-ouvertes.fr/hal-02470782 [Son erişim tarihi: 14.05.2020]

Goldberg, D. 1991. What Every Computer Scientist Should Know about Floating-Point Arithmetic. *ACM Comput. Surv.*, 23(1): 5-48.

Hida, Y. and Li, X.S. and Bailey, D.H. 2001. 15th IEEE Symposium on Computer Arithmetic , pp. 155-162, 11-13 June, IEEE, Vail.

Higham, N.J. 2002. Accuracy and Stability of Numerical Algorithms. SIAM, Manchester, 667 p.

Jeannerod, C.P. and Muller, J.M. and Zimmermann, P. 2018. ARITH 2018 - 25th IEEE Symposium on Computer Arithmetic, pp. 53–60, 25 June, IEEE, Lyon.

Kornerup, P. and Lefèvre, V. and Louvet, N. and Muller, J.M. 2009. On the Computation of Correctly-Rounded Sums. Retrieved from https://hal.inria.fr/inria-00367584v2 [Son erişim tarihi: 4 Şubat 2022]

Milani, A. and Nobili, A.M. 1988. Integration Error Over Very Long Time Spans. *Celestial Mechanics*, 43(1-4): 1-34.

Møller, O. 1965. Quasi double-precision in floating point addition. *BIT Numerical Mathematics*, 5: 37-50.

Muller, J. M. and Brisebarre N. and Dinechin D.F. and Jeannerod C.P. and Lefèvre, V. and Melquiond, G. and Revol, N. and Stehle, D. and Torres, S. 2010. Handbook of Floating-Point Arithmetic. Birkäuser, New York, 514 p.

Ogita, T. and Rump, S. M. and Oishi, S. 2005. Accurate Sum and Dot Product. *SIAM Journal on Scientific Computing*, 26(6): 1955-1988.

O'Regan, G. 2016. Guide to Discrete Mathematics. Springer Nature, Mallow, 365 p.

Overton, M.L. 2001. Numerical Computing with IEEE Floating point Arithmetic. SIAM, New York, 101 p.

Panhaleux, A. 2012. Contributions to floating-point arithmetic: Coding and correct rounding of algebraic functions, Phd thesis, Ecole Normale Supérieure de Lyon, Lyon, 105 p.

Press, W.H. and Teukolsky, S.A. and Vetterling, W.T. and Flannery, B.P. 2002. Numerical Recipes in C - The Art of Scientific Computing. Cambridge University Press, New York, 965 p.

Prussing, J.E. and Conway, B.E. - 1993. Orbital Mechanics. Oxford University Press, New York, 191 p.

Quinlan, G.D. 1994. Roundoff Error in Long-Term Orbital Integrations Using Multistep Methods. *Celestial Mechanics and Dynamical Astronomy*, 58(4): 339-351.

Quinn, T. and Tremaine, S. 1989. Roundoff Error in Long-Term Planetery Orbit Integrations. *The Astronomical Journal*, 99(3): 1016-1023.

Rein, H. and Spiegel, S. 2015. IAS15: A fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. *Monthly Notices of the Royal Astronomical Society*, 446(2): 1424–1437.

Rein, H. and Brown, G. and Tamayo, D. 2019. On the accuracy of symplectic integrators for secularly evolving planetary systems. *Monthly Notices of the Royal Astronomical Society*, 490(4): 5122–5133.

Riedy, J. and Demmel, J. 2018. Augmented Arithmetic Operations Proposed for IEEE-754 2018 Retrieved from https://ieeexplore.ieee.org/document/8464813 [Son erişim tarihi: 10.02.2017].

Scherer, P.O.J. 2017. Computational Physics Simulation of Classical and Quantum Systems. Springer, Garching, 627 p.

Sterbenz, P.H. 1974. Floating point Computation. Prentice-Hall, New York, 311 p.

Vallado, D.A. 2013. Fundamentals of Astrodynamics and Applications. Microcosm Press, Hawthorne, 1056 p.

Varadi, F. and Runnegar, B. and Ghil, M. 2003. Successive Refinements in Long-Term Integrations of Planetary Orbits. *Celestial Mechanics, methods: N-body simulations, Solar System: General*, 592(1): 620-630.

Veris, A.I. 2018. Practical Astrodynamics. Springer Nature, Rome, 1308 p.

Voesenek, C. J. 2008. Implementing a Fourth Order Runge-Kutta Method for Orbit Simulation Retrieved from http://spiff.rit.edu/richmond/nbody/OrbitRungeKutta4.pdf [Son erişim tarihi: 19.03.2022].

Wilkinson, J.H. 1960. Error Analysis of Floating-Point Computation. *Numerische Mathematik*, 2: 319-340.

ÖZGEÇMİŞ

BURÇAK YEŞİLIRMAK

ÖĞRENİM BİLGİLERİ

Yüksek Lisans: 2019-2022	Akdeniz Üniversitesi, Antalya Fen Bilimleri Enstitüsü, Uzay Bilimleri ve Teknolojileri Anabilim Dalı
Lisans: 2016 –	Akdeniz Üniversitesi, Antalya Fen Fakültesi, Uzay Bilimleri ve Teknolojileri Bölümü
Lisans: 2010-2016	Çukurova Üniversitesi, Adana Fen Fakültesi, Matematik Bölümü