

ABSTRACT

Title of Document: AN INVESTIGATION INTO THE MULTI-
POLE MOMENTS OF A GRAVITATING
BODY AND ITS EFFECTS ON ORBITS

Angeleene S. Ang
BS Physics 2015

Directed By: Dr. Jerrold J. Garcia
Department of Physics

Compiled: February 24, 2015

In the Kepler Problem, it is assumed that the attractor and the orbiting body to be both point particles. In this paper, we removed this assumption for the attractor and assume that it has a shape. To do this, we derived the general form of the additional terms in the multipole expansion. Specifically, we look at the quadrupole moment of a gravitating body and used this as the approximation of our extended body. Using analytical methods, we were able to show that only the the z-component of the angular momentum vector remains constant, and that orbits starting at the equatorial or azimuthal plane stay on a single plane. We also used numerical methods to show that the orbit varies as the initial conditions approach the attractor.

AN INVESTIGATION INTO THE MULTIPOLE MOMENTS OF
A GRAVITATING BODY AND ITS EFFECTS ON ORBITS

By

Angeleene Sayaboc Ang

Thesis submitted to the Faculty of the Physics Department of the
Ateneo de Manila University, in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Physics
2015

Advisory Committee:
Dr. Jerrold J. Garcia
Dr. Quirino M. Sugon, Jr.
Dr. Eduardo C. Cuansing, Jr.
Dr. Joel T. Maquiling

This is to certify that undergraduate thesis entitled, AN INVESTIGATION INTO THE MULTIPOLE MOMENTS OF A GRAVITATING BODY AND ITS EFFECTS ON ORBITS and submitted by Angeleene S. Ang to fulfill part of requirements for the degree of BS Physics was successfully defended and approved on 2015 Feb 23

Joel T. Maquiling, Ph.D.
Panel Member

Eduardo C. Cuansing, Jr., Ph.D.
Panel Member

Quirino M. Sugon, Jr., Ph.D.
Thesis Co-Adviser

Jerrold J. Garcia, Ph.D.
Thesis Adviser

James Bernard B. Simpas, Ph.D.
Chair
Physics Department

TABLE OF CONTENTS

Table of Contents	v
List of Figures	v
List of Source Codes	vi
List of Symbols	vii
Chapter 1: Introduction	1
Chapter 2: Kepler Problem and Multipole Moment Expansion	2
2.1 Central Forces and Kepler's First Law	2
2.2 Multipole Moments	4
2.2.1 Multipole Expansion in Cartesian Coordinates	6
2.2.2 Multipole Expansion in Spherical Coordinates	11
2.2.3 Laplace's Equation	12
Chapter 3: The Ring Attractor	14
3.1 Monopole Moment	16
3.2 Dipole Moment	17
3.2.1 In Cartesian Coordinates	17
3.2.2 In Spherical Coordinates	18
3.3 Quadrupole Moment	19
3.3.1 In Cartesian Coordinates	19
3.3.2 In Spherical Coordinates	21
3.4 Using Laplace's Equation	22
3.5 On the plane	26
Chapter 4: Orbit for the Quadrupole Potential	29
4.1 Dynamical Variables	29
4.1.1 Lagrangian and Hamiltonian	29
4.1.2 Angular Momentum on the Equatorial Plane	30
4.2 Orbit Simulation	31
4.3 Classification of Orbits	33
4.3.1 Bound Planar Orbits	33
4.3.2 Invalid and Unbound Orbits	35
4.3.3 General Case of the Bound Orbit	36
4.4 Orbit Return Map	37
Chapter 5: Conclusion	39
Appendices	40
Chapter A: List of Spherical Harmonics	40
Chapter B: FORTRAN Code	41
B.1 Exact Ring Potential	41
B.2 Ring Orbit Simulation	46
B.3 Return Map	53
References	57

LIST OF FIGURES

2.1	Orbits of the Kepler Problem	3
2.2	Potential due to Arbitrary Source	4
3.1	Geometry for the Ring of Mass	14
3.2	Approximate Ring Multipole Potential on the Plane	25
3.3	Potential on the Plane for a Ring of Mass	26
3.4	Exact Ring Multipole Potential on the Plane	27
3.5	Ring Multipole Potential vs the exact potential	27
4.1	Comparison of Monopole and Quadrupole effects on Near-Attractor Equatorial Orbit	33
4.2	Comparison of Monopole and Quadrupole effects on Far-Attractor Equatorial Orbit	33
4.3	Polar Orbit	34
4.4	Orbit where the particle falls into the attractor	35
4.5	Unbound orbit at Near-Attractor	35
4.6	Orbit confined in a topological torus	36
4.7	Angular Momentum of Fig 4.6	36
4.8	Return Map for $\mathbf{r}_0 = (3, 0, 0)$	38
4.9	Return Map for $\mathbf{r}_0 = (2, 0, 1)$	38

LIST OF SOURCE CODES

B.1	Romberg Integration	41
B.2	Trapezoid Rule	42
B.3	Polynomial Interpolation	42
B.4	Exact Potential Integrand	43
B.5	Multipole Potential Function	43
B.6	Numerical Integration of Exact Potential Function	44
B.7	RK2 Subroutine for Orbit Solution	46
B.8	Derivation of Actual Orbit for Ring Potential	47
B.9	Print XYZ plot to file	50
B.10	Plot magnitude of Angular Momentum vs time to file	51
B.11	Reverse Input Array	52
B.12	Return Map	53

LIST OF SYMBOLS

$\delta(r' - a)$	Dirac Delta Function
δ_{ij}	Kronecker Delta
dM	Arbitrarily small unit of mass
\mathcal{L}	Lagrangian
\mathbf{F}	Force
\mathbf{N}	Torque
\mathbf{p}	Dipole Moment Vector
\mathbf{r}'	Position vector of the source mass
$\hat{\mathbf{r}}$	Unit radial vector
$\overset{\leftrightarrow}{\mathbf{q}}$	Traceless Quadrupole Moment Tensor
∇^2	Laplacian Operator
Φ	Gravitational Potential
ϕ	Azimuth Angle
$\Phi^{(0)}$	Monopole Potential
$\Phi^{(1)}$	Dipole Moment Potential
$\Phi^{(2)}$	Quadrupole Moment Potential
$\rho(\mathbf{r}')$	Mass Density
θ	Polar Angle
$\text{Tr } Q$	Trace of Matrix Q
e	Eccentricity of the conic equation
$f(r)$	Magnitude of a central force
G	Gravitational Constant
H	Hamiltonian
m	Mass of the orbiting body
p	Parameter of the conic equation; length of the latus rectum
p_i	i^{th} component of the Dipole Moment
q_{ij}	Component of the Traceless Quadrupole Moment Tensor
U	Potential Energy
$Y_{lm}(\theta, \phi)$	Spherical Harmonics Function

CHAPTER 1

INTRODUCTION

While the gravitational two-body problem - the Kepler Problem - is normally a part of any Classical Mechanics course, be it in the undergraduate or graduate level, it is only solved in under the assumption that both the attractor and the orbiting body are both point particles. This assumption limits its validity to orbiting bodies far enough such that the attractor can be assumed to be a point. [1, 2, 3] The two-body problem, assuming that only the force of interaction between the two particles is considered, can be reduced to a central force problem in the limit as the reduced mass of the system is approximately the mass of the attractor, such that the center of mass of the system lies very close to the position of the attractor.

To reduce the approximations in the Kepler Problem, we use the multipole expansion - a method that uses the Taylor Series expansion on the formula for finding the potential. [4, 5, 6] We use this method to find the orbit of a body about a simple non-point extended body - a ring of uniform mass. We chose this configuration since it is the simplest physical object with a quadrupole moment.

Chapter 2 is dedicated to the discussion of both the Kepler Problem and the Multipole Expansion. In the following chapter, we use the equations derived in order to derive the approximate gravitational potential of a ring of uniform mass. We will then use the potential in 4 to examine the orbit of the potential, both analytically and numerically.

CHAPTER 2

KEPLER PROBLEM AND MULTIPOLE MOMENT EXPANSION

2.1 CENTRAL FORCES AND KEPLER'S FIRST LAW

When the magnitude of the force depends only on the radial distance away from a previously-defined origin, the force field is called *central*. The force can be written as

$$\mathbf{F} = f(r)\hat{\mathbf{r}}. \quad (2.1)$$

which is a standard topic in undergraduate Classical Mechanics texts, such as Marion and Thornton [2]. From the form of the central force, we note that

$$\mathbf{N} = \mathbf{r} \times \mathbf{F} = \mathbf{0}. \quad (2.2)$$

In this result, we find the conservation of angular momentum - which implies that the orbit is restricted on a single plane for all time. From the same text, given the inverse-square potential, the orbit of the planet can be described as a polar equation

$$r = \frac{p}{1 + e \cos \varphi}. \quad (*)$$

where p is the *parameter* of the orbit, e is the *eccentricity*, and

$$p := \frac{L^2}{m^2 GM} \quad (2.3a)$$

$$e := \sqrt{\frac{2EL^2}{m^3 G^2 M^2} + 1} \quad (2.3b)$$

and given the energy, the shape of the orbit changes

* Marion and Thornton[2], page 300, Eq. (8.41)

Arnold[3], page 39

Goldstein[1], page 94, Eq. (3.56)

unbound orbits		
$ e > 1$	$E > 0$	hyperbolic
$ e = 1$	$E = 0$	parabolic
bound orbits		
$ e < 1$	$E < 0$	elliptic
$e = 0$	$E = -\frac{m^3 G^2 M^2}{2L^2}$	circular

Table 2.1: Relationship between the eccentricity e and the energy E , as well as the resulting shape of the orbits.

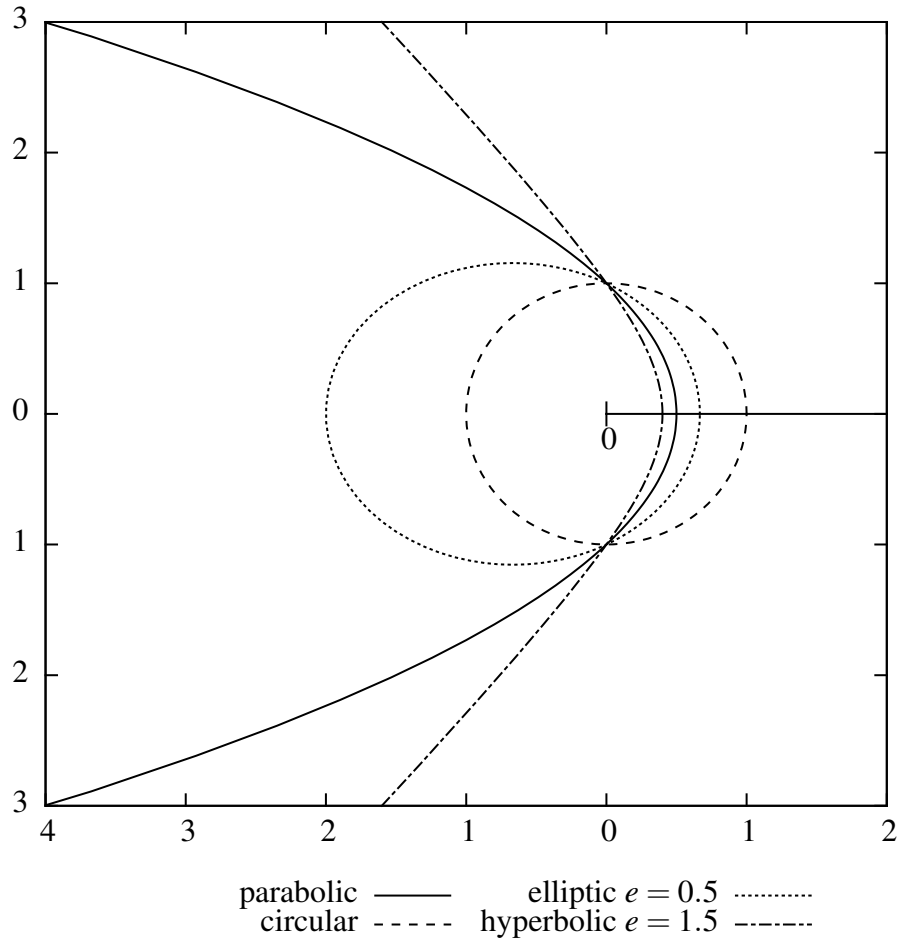


Figure 2.1: For a given energy E , the Kepler Problem produces orbits that are either bound or unbound.

These results are under the assumption that the attractor is a point mass. However, point masses have zero volume, and these do not exist in nature - all celestial bodies have a shape of some kind, and this shape has little to no symmetry. In order to investigate the orbits about extended bodies, we are going to use a method of analytically obtaining an approximate potential based on the Taylor series expansion called the Multipole Expansion.

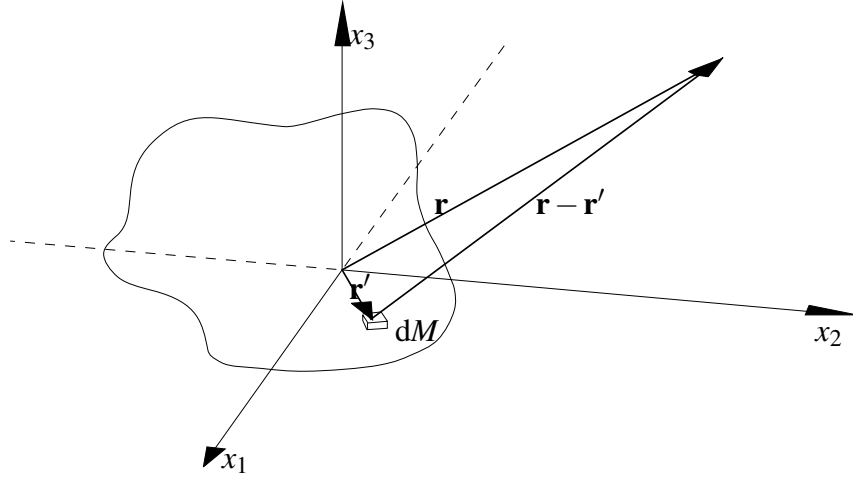


Figure 2.2: Given a source with an arbitrary structure, we can find the potential of the entire figure by finding the the potential due to each mass element dM .

2.2 MULTIPOLE MOMENTS

We start with a general distribution of mass M . This mass is fixed in space, generating a gravitational potential Φ at the field point \mathbf{r} . If the mass were not located at the origin of the coordinate system, we use the source position vector \mathbf{r}' . The potential would then depend on the distance between the source and the field, or

$$d\Phi = -\frac{G dM}{|\mathbf{r} - \mathbf{r}'|}. \quad (2.4)$$

We note that the gravitational potential Φ is related to the potential energy of the system using

$$U = m\Phi. \quad (2.5)$$

We can only define a single position vector for the source if the mass were all concentrated to a point. We change the mass distribution into a density $\rho(\mathbf{r}')$. With this, we can find the potential potential using

$$dM = \rho(\mathbf{r}') d\tau', \quad (2.6)$$

so that we can integrate Eq. (2.4)

$$\Phi = -G \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\tau'. \quad (2.7)$$

The challenge with Eq. (2.7) lies in integrating the expression for $1/|\mathbf{r} - \mathbf{r}'|$. Even for simple cases such as a ring of mass with constant density, obtaining an exact solution would be difficult without imposing certain conditions. For this, we turn to the multipole expansion. We will look at the expansion for both the cartesian and spherical coordinate systems in the following sections.

2.2.1 MULTIPOLE EXPANSION IN CARTESIAN COORDINATES

Using the Cartesian coordinate system, we know that the magnitude of $\mathbf{r} - \mathbf{r}'$ is given by

$$|\mathbf{r} - \mathbf{r}'| = \left[\sum_{i=1}^3 (x_i - x'_i)^2 \right]^{\frac{1}{2}}. \quad (2.8)$$

In order to obtain a simpler expression, we recall the Taylor series

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left[f^{(k)}(x) \right]_{x=x_0} (x - x_0)^k \quad (2.9)$$

where $f^{(k)}(x)$ is the k -th derivative of $f(x)$. For a multivariable function, the Taylor series can be generalized as

$$\begin{aligned} f(\mathbf{x}) = f(\mathbf{x}_0) &+ \sum_j \frac{\partial}{\partial x_j} f(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_0} (x_j - x_{0,j}) \\ &+ \frac{1}{2} \sum_{j,k} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_k} f(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_0} (x_j - x_{0,j})(x_k - x_{0,k}) + \dots \end{aligned} \quad (2.10)$$

If we expand $1/|\mathbf{r} - \mathbf{r}'|$ using Eq. (2.10) about some arbitrary vector \mathbf{x}_0 , we obtain

$$\begin{aligned} \frac{1}{|\mathbf{r} - \mathbf{r}'|} &= \frac{1}{0!} \left[\frac{1}{|\mathbf{r} - \mathbf{r}'|} \right]_{\mathbf{r}'=\mathbf{x}_0} \\ &+ \frac{1}{1!} \sum_{i=1}^3 \left[\frac{\partial}{\partial x'_i} \frac{1}{|\mathbf{r} - \mathbf{r}'|} \right]_{\mathbf{r}'=\mathbf{x}_0} (x'_i - x_{0,i}) \\ &+ \frac{1}{2!} \sum_{i,j=1}^3 \left[\frac{\partial^2}{\partial x'_i \partial x'_j} \frac{1}{|\mathbf{r} - \mathbf{r}'|} \right]_{\mathbf{r}'=\mathbf{x}_0} (x'_i - x_{0,i})(x'_j - x_{0,j}) \\ &+ \dots \end{aligned} \quad (2.11)$$

where $x_{0,i}$ indicates the i -th component to \mathbf{x}_0 .

To obtain the second term in the right-hand side of Eq. (2.11), we perform the first

derivative of $1/|\mathbf{r}-\mathbf{r}'|$, as follows:

$$\sum_{j=1}^3 \left[\frac{\partial}{\partial x'_j} \frac{1}{|\mathbf{r}-\mathbf{r}'|} \right] = -\frac{1}{2} \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \sum_{j=1}^3 \sum_{i=1}^3 \left[\frac{\partial}{\partial x'_j} (x_i - x'_i)^2 \right] \quad (2.12)$$

$$= \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \sum_{j=1}^3 \sum_{i=1}^3 (x_i - x'_i) \left[\frac{\partial x'_i}{\partial x'_j} \right] \quad (2.13)$$

$$= \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \sum_{j=1}^3 \sum_{i=1}^3 (x_i - x'_i) \delta_{ij}, \quad (2.14)$$

where δ_{ij} is the Kronecker Delta. Taking the sum over j , we obtain

$$\sum_{j=1}^3 \frac{\partial}{\partial x'_j} \frac{1}{|\mathbf{r}-\mathbf{r}'|} = \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \sum_{i=1}^3 (x_i - x'_i) \quad (2.15)$$

Note that we will also need the second derivative of $1/|\mathbf{r}-\mathbf{r}'|$ for the Taylor expansion. To simplify matters, we can simply take the derivative of Eq. (2.15).

$$\sum_{i,j} \left[\frac{\partial^2}{\partial x'_j \partial x'_i} \frac{1}{|\mathbf{r}-\mathbf{r}'|} \right] = \sum_{i,j} \frac{\partial}{\partial x'_j} \left(\frac{\partial}{\partial x'_i} \frac{1}{|\mathbf{r}-\mathbf{r}'|} \right) \quad (2.16)$$

$$= \sum_{i,j} \frac{\partial}{\partial x'_j} \left(\frac{\partial}{\partial x'_i} \left[\sum_{k=1}^3 (x_k - x'_k)^2 \right]^{-\frac{1}{2}} \right) \quad (2.17)$$

$$= \sum_{i,j} \frac{\partial}{\partial x'_j} \left(\frac{1}{|\mathbf{r}-\mathbf{r}'|^3} (x_i - x'_i) \right) \quad (2.18)$$

$$= \sum_{i,j} \left(\frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \frac{\partial}{\partial x'_j} (x_i - x'_i) + (x_i - x'_i) \frac{\partial}{\partial x'_j} \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \right) \quad (2.19)$$

$$= \sum_{i,j} \left(-\frac{\delta_{ij}}{|\mathbf{r}-\mathbf{r}'|^3} + (x_i - x'_i) \frac{\partial}{\partial x'_j} \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \right) \quad (2.20)$$

$$= \sum_{i,j} \left(-\frac{\delta_{ij}}{|\mathbf{r}-\mathbf{r}'|^3} + 3 \frac{1}{|\mathbf{r}-\mathbf{r}'|^5} (x_j - x'_j) (x_i - x'_i) \right) \quad (2.21)$$

We go back to Eq. (2.11). Plugging in the expressions for the derivatives of

$1/|\mathbf{r}-\mathbf{r}'|$ from Eqs. (2.15) and (2.21), we get

$$\begin{aligned}
\frac{1}{|\mathbf{r}-\mathbf{r}'|} &= \frac{1}{|\mathbf{r}-\mathbf{r}'|} \Big|_{\mathbf{r}'=\mathbf{x}_0} \\
&+ \frac{1}{|\mathbf{r}-\mathbf{r}'|^3} \sum_{i=1}^3 (x_i - x'_i) \Big|_{\mathbf{r}'=\mathbf{x}_0} (x_i - x_{0,i}) \\
&+ \frac{1}{2} \sum_{i,j=1}^3 \left[-\frac{\delta_{ij}}{|\mathbf{r}-\mathbf{r}'|^3} + 3 \frac{(x_j - x'_j)(x_i - x'_i)}{|\mathbf{r}-\mathbf{r}'|^5} \right]_{\mathbf{r}'=\mathbf{x}_0} (x'_i - x_{0,i})(x'_j - x_{0,j}) \\
&+ \dots
\end{aligned} \tag{2.22}$$

Now we set the arbitrary point \mathbf{x}_0 to be the origin.

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \frac{1}{|\mathbf{r}|} + \frac{1}{|\mathbf{r}|^3} \sum_{i=1}^3 x_i x'_i + \frac{1}{2} \sum_{i,j} \left[-\frac{\delta_{ij}}{|\mathbf{r}|^3} + 3 \frac{x_j x_i}{|\mathbf{r}|^5} \right] x'_i x'_j \tag{2.23}$$

Substituting Eq. (2.23) in the expression for the potential Eq. (2.7), we get

$$\Phi = -G \int_{\mathbb{R}^3} \left(\frac{\rho(\mathbf{r}')}{|\mathbf{r}|} + \frac{\rho(\mathbf{r}')}{|\mathbf{r}|^3} \sum_{i=1}^3 x_i x'_i + \frac{1}{2} \sum_{i,j} \rho(\mathbf{r}') \left[-\frac{\delta_{ij}}{|\mathbf{r}|^3} + 3 \frac{x_j x_i}{|\mathbf{r}|^5} \right] x'_i x'_j \right) d\tau', \tag{2.24}$$

where we have truncated the higher-order terms.

We can now integrate the preceding expression over all space. Starting from the first term alone, we obtain

$$\Phi = -\frac{G}{|\mathbf{r}|} \int_{\mathbb{R}^3} \rho d\tau' = -\frac{GM}{|\mathbf{r}|}, \tag{2.25}$$

where we used Eq. (2.6). This is the form of the potential for a point mass, and this is the form of a potential for any mass distribution as $|\mathbf{r}| \rightarrow \infty$. We will call this the *monopole potential* $\Phi^{(0)}$.

The second term is the *dipole potential* $\Phi^{(1)}$, with the corresponding *dipole moment* \mathbf{p} , given by p_i

$$p_i := \int_{\mathbb{R}^3} \rho(\mathbf{r}') x'_i d\tau'. \tag{2.26}$$

Integrating the second term of Eq. (2.24),

$$\Phi^{(1)} = -G \frac{1}{|\mathbf{r}|^3} \sum_{i=1}^3 \int_{\mathbb{R}^3} \rho(\mathbf{r}') x_i x'_i d\tau' \quad (2.27)$$

$$= -\frac{G}{|\mathbf{r}|^3} \sum_{i=1}^3 x_i p_i \quad (2.28)$$

$$= -\frac{G}{|\mathbf{r}|^3} \mathbf{r} \cdot \mathbf{p}. \quad (2.29)$$

The last term is called the *quadrupole potential* $\Phi^{(2)}$, obtained from the third term of Eq. (2.24):

$$\Phi^{(2)} = -\frac{G}{2} \sum_{i,j=1}^3 \left[\frac{3x_i x_j}{|\mathbf{r}|^5} - \frac{\delta_{ij}}{|\mathbf{r}|^3} \right] \int_{\mathbb{R}^3} \rho(\mathbf{r}') x'_i x'_j d\tau'. \quad (2.30)$$

Defining the tensor Q_{ij}

$$Q_{ij} = \int_{\mathbb{R}^3} \rho(\mathbf{r}') x'_i x'_j d\tau', \quad (2.31)$$

we can write $\Phi^{(2)}$ as

$$\Phi^{(2)} = -\frac{G}{2} \left[\sum_{i,j=1}^3 \frac{3Q_{ij} x_i x_j}{|\mathbf{r}|^5} - \sum_{i,j=1}^3 \frac{Q_{ij} \delta_{ij}}{|\mathbf{r}|^3} \right] \quad (2.32)$$

We define the trace $\text{Tr } Q$

$$\text{Tr } Q := \sum_j Q_{jj} := \sum_{ij} Q_{ij} \delta_{ij} \quad (2.33)$$

With this, Eq. (2.32) becomes

$$\Phi^{(2)} = -\frac{G}{2} \frac{1}{|\mathbf{r}|^5} \sum_{i,j=1}^3 [3Q_{ij} - \delta_{ij} \text{Tr } Q] x_i x_j \quad (2.34)$$

We define the Quadrupole Moment Tensor q_{ij}

$$q_{ij} = 3Q_{ij} - \delta_{ij} \text{Tr } Q \quad (2.35)$$

which simplifies Eq. (2.34) as

$$\Phi^{(2)} = -\frac{G}{2} \frac{1}{|\mathbf{r}|^5} \sum_{i,j=1}^3 q_{ij} x_i x_j. \quad (2.36)$$

This is now the form of the product q_{ij} and the position vector,

$$\Phi^{(2)} = -\frac{G}{2} \frac{\mathbf{r} \cdot \overset{\leftrightarrow}{\mathbf{q}} \cdot \mathbf{r}}{|\mathbf{r}|^5} \quad (2.37)$$

where $\overset{\leftrightarrow}{\mathbf{q}}$ is given by Eq. (2.35).

The quadrupole moment tensor is symmetric, meaning that it has six independent components. However, we further reduce that number to five by showing that it is traceless:

$$\text{Tr } q = \sum_{i,j=1}^3 \delta_{ij} (3Q_{ij} - \delta_{ij} \text{Tr } Q). \quad (2.38)$$

The first term reduces to simply the trace of Q , whereas the sum of a Kronecker Delta over *both* its indices produces a 3

$$\text{Tr } q = 3 \text{Tr } Q - 3 \text{Tr } Q = 0. \quad (2.39)$$

2.2.2 MULTIPOLE EXPANSION IN SPHERICAL COORDINATES

Henceforth, we use the symbol φ for the azimuthal angle in the Spherical Coordinate system, and θ for the polar angle. Using the Addition Theorem, we can expand $1/|\mathbf{r}-\mathbf{r}'|$ in terms of the Spherical Harmonics

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{2l+1} \frac{r_{<}^l}{r_{>}^{l+1}} Y_{lm}^*(\theta', \varphi') Y_{lm}(\theta, \varphi). \quad (*)$$

where

$$r_{<} = \min(r, r') \quad (2.40a)$$

$$r_{>} = \max(r, r') \quad (2.40b)$$

and the position vectors are now functions of r, θ, φ . Eq. (2.7) then becomes

$$\Phi = -4\pi G \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{1}{2l+1} \left(\int_{\mathbb{R}^3} \rho(\mathbf{r}') \frac{r_{<}^l}{r_{>}^{l+1}} Y_{lm}^*(\theta', \varphi') d\tau' \right) Y_{lm}(\theta, \varphi), \quad (2.41)$$

We define the quantity q_{lm} as

$$q_{lm} = \begin{cases} \int_{\mathbb{R}^3} \rho(\mathbf{r}') (r')^l Y_{lm}^*(\theta', \varphi') d\tau' & r' < r \\ \int_{\mathbb{R}^3} \rho(\mathbf{r}') \frac{1}{(r')^{l+1}} Y_{lm}^*(\theta', \varphi') d\tau' & r < r' \end{cases} \quad (2.42)$$

where the spherical monopole moment is q_{00} , the spherical dipole moment is q_{1m} , the spherical quadrupole moment is q_{2m} , and so on, where m goes from $-l$ to $+l$.

A table of the first few Spherical Harmonics can be found in Appendix A.

* Jackson[6], page 102, Eq. (3.70)
Zangwill[5], page 109, Eqs. (4.83, 4.84)

2.2.3 LAPLACE'S EQUATION

We also recall that the potential function Φ must also satisfy Laplace's Equation

$$\nabla^2 \Phi = 0, \quad (2.43)$$

it is known in literature that the solution to this is

$$\Phi(r, \theta, \varphi) = \sum_{l,m} \left[A_l r^l + B_l \frac{1}{r^{l+1}} \right] Y_{lm}(\theta, \varphi) \quad (*)$$

where A and B are constants which are determined by the boundary conditions. These conditions usually come in the form of a defined constant potential at some location, but in our case, we only have the mass distribution itself. What we can do is to find the potential of the distribution at a given location using direct integration or other methods, then use this to find the unique solution everywhere.

We also note that given a body isolated in space all A_l 's are zero for $l > 0$ as the potential should not be increasing as $r \rightarrow \infty$, and that for the case with azimuthal symmetry, $m = 0$. For A_0 , the physical interpretation of the potential, the force, remains the same as the gradient of a constant is zero. In Subsection 2.2.1, we have shown that the quadrupole moment potential $\Phi^{(2)}$ goes as r^{-3} . We can take the same term in the solution of the Laplace Equation

$$\Phi^{(2)} = B_2 \frac{1}{r^3} Y_{20}(\theta, \varphi) = B_2 \sqrt{\frac{5}{16\pi}} \frac{1}{r^3} (3 \cos^2 \theta - 1) \quad (2.44)$$

where we now need the boundary condition to find B_2 .

* Jackson[6], page 95-96, Eqs. (3.2), (3.5), (3.6), and (3.8),
Panofsky[4], page 82, Eqs. (5.6, 5.7),
Zangwill[5], page 213, Eqs. (7.76, 7.77)

Now that we have a formulation for solving for the approximate potential of a body with arbitrary shape, what we will do next is now to look at a specific body to study. In this paper, we are going to look at the ring of uniform mass.

As we will show in the following chapter, the ring is one of the simplest physical objects with a non-vanishing quadrupole moment. At the same time, the shape is simple enough to integrate in the Spherical Coordinate system using the methods derived in Section 2.2. Lastly, the ring is the basic element of other shapes with azimuthal symmetry - a disc can be formed by adding enough rings on a plane, and discs of varying radii and at various heights can be used to describe spheroids. Once we have derived the form of its potential, the potential for the other shapes would only differ by multiplying a constant in the original solution.

CHAPTER 3

THE RING ATTRACTOR

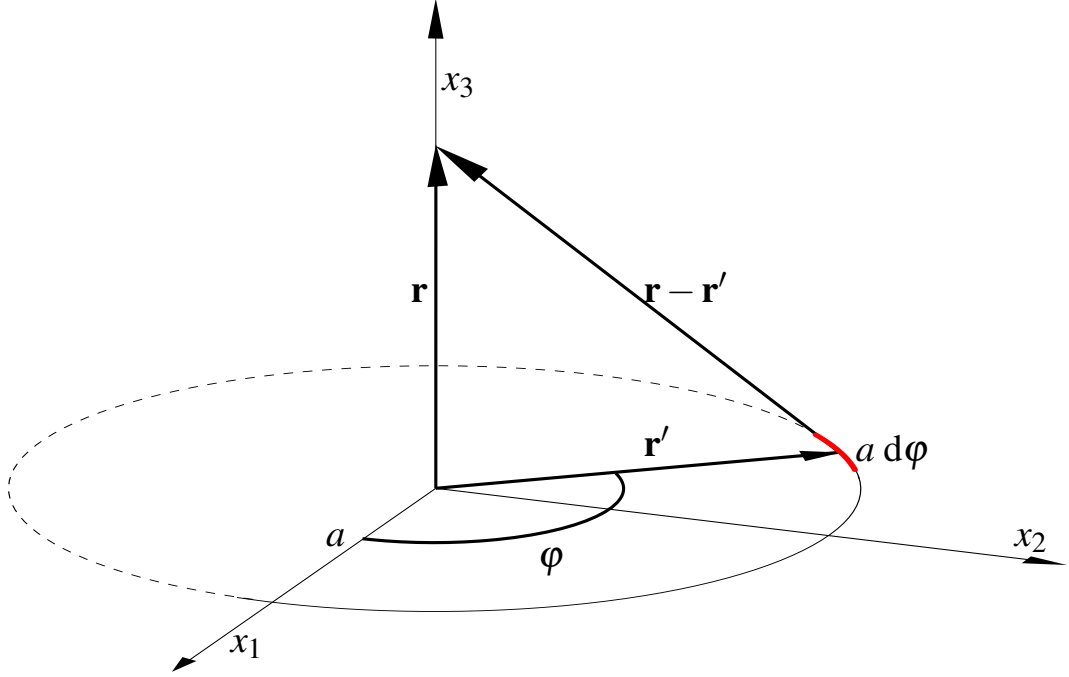


Figure 3.1: The field point is indicated by the vector \mathbf{r} , and a small mass dM is located using the vector \mathbf{r}' .

For this chapter, we note that our "volume element" is defined only on the equatorial plane, and at $r = a$. We combine these two in order to write the density ρ

$$\rho(\mathbf{r}') = \lambda \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) \quad (3.1)$$

where λ is the linear density, which we will find an expression for, and δ is the Dirac Delta function. We note that if we integrate the density over all space, we should obtain the total mass, or

$$M_{tot} = \int \rho(\mathbf{r}') d\tau = \int_{r'=0}^{\infty} \int_{\varphi'=0}^{2\pi} \int_{\theta'=0}^{\pi} \lambda \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) (r')^2 \sin \theta' dr' d\theta' d\varphi' \quad (3.2)$$

and we note that $\delta\left(\theta' - \frac{\pi}{2}\right)$ has the same properties as $\delta(\cos\theta')$. Using the following property of the delta function,

$$\int_{a-\varepsilon}^{a+\varepsilon} f(x)\delta(x-a)dx = f(a) \quad \forall \varepsilon > 0, \quad (3.3)$$

we get

$$\lambda = \frac{M}{2\pi a^2} \quad (3.4)$$

and our expression for the density is now

$$\rho(\mathbf{r}') = \frac{M}{2\pi a^2} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a). \quad (3.5)$$

We can now use Eq. (3.5) in Eq. (2.31) and Eq. (2.42) to find p_i and Q_{ij} , and q_{lm} .

3.1 MONOPOLE MOMENT

The Cartesian Expansion of the Monopole moment is obvious. We use Eq. (2.25), to obtain

$$\Phi_{\text{cart}}^{(0)} = -\frac{GM}{r}. \quad (3.6)$$

For the monopole moment using the Spherical Expansion, we use Eq. (2.42). We will divide the region at $r' = a$, as this is where the ring lies. However, we still need to integrate over r' , so these terms inside the integrals are left as is. If we plug in Eq. (3.5) into Eq. (2.42),

$$q_{lm} = \begin{cases} \frac{M}{2\pi a^2} \int_{\mathbb{R}^3} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) (r')^l Y_{lm}^*(\theta', \varphi') d\tau' & r > a \\ \frac{M}{2\pi a^2} \int_{\mathbb{R}^3} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) \frac{1}{(r')^{l+1}} Y_{lm}^*(\theta', \varphi') d\tau' & r < a \end{cases} \quad (3.7)$$

The monopole term is given by q_{00} , which we can evaluate

$$q_{00} = \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{1}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) (r')^2 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{1}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) (r') dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' & r < a \end{cases} \quad (3.8a)$$

$$= \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{1}{4\pi}} (a)^2 (1) (2\pi) = M \sqrt{\frac{1}{4\pi}} & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{1}{4\pi}} (a) (1) (2\pi) = \frac{M}{a} \sqrt{\frac{1}{4\pi}} & r < a \end{cases} \quad (3.8b)$$

using Eq. (3.8), we can find the monopole potential

$$\Phi_{\text{sphr}}^{(0)} = \begin{cases} -\frac{GM}{r} & r > a \\ -\frac{GM}{a} & r < a \end{cases} \quad (3.9)$$

3.2 DIPOLE MOMENT

3.2.1 IN CARTESIAN COORDINATES

Here, we will have to use the following transformations in order to evaluate p and later Q :

$$x_1 = r \sin \theta \cos \varphi \quad (3.10a)$$

$$x_2 = r \sin \theta \sin \varphi \quad (3.10b)$$

$$x_3 = r \cos \theta \quad (3.10c)$$

$$d\tau = r^2 \sin \theta dr d\theta d\varphi \quad (3.10d)$$

We can show that the dipole moment, defined in Eq. (2.26), is also zero for all i :

$$\begin{aligned} p_1 &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} (r')^3 \delta(r' - a) dr' \int_{\theta'=0}^{\pi} \sin^2 \theta' \delta\left(\theta' - \frac{\pi}{2}\right) d\theta' \int_{\varphi'=0}^{2\pi} \cos \varphi' d\varphi' \\ &= \frac{M}{2\pi a^2} (a)^3 (1) (0) = 0 \end{aligned} \quad (3.11a)$$

$$\begin{aligned} p_2 &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r' - a) (r')^3 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^2 \theta' d\theta' \int_{\varphi'=0}^{2\pi} \sin \varphi' d\varphi' \\ &= \frac{M}{2\pi a^2} (a)^3 (1) (0) = 0 \end{aligned} \quad (3.11b)$$

$$\begin{aligned} p_3 &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} (r')^3 \delta(r' - a) dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \cos \theta' \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' \\ &= \frac{M}{2\pi a^2} (a)^3 (0) (2\pi) = 0 \end{aligned} \quad (3.11c)$$

From Eq. (3.11), we can see that

$$\Phi_{\text{cart}}^{(1)} = 0. \quad (3.12)$$

3.2.2 IN SPHERICAL COORDINATES

For the dipole moment in terms of q_{lm} , we find q_{10}

$$q_{10} = \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{3}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) (r')^3 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \cos \theta' \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{3}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \cos \theta' \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' & r < a \end{cases} \quad (3.13a)$$

$$= \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{3}{4\pi}} (a)^3 (0) (2\pi) = 0 & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{3}{4\pi}} (1) (0) (2\pi) = 0 & r < a \end{cases} \quad (3.13b)$$

Similar to the Dipole Moment vector \mathbf{p} in Cartesian Coordinates, we have also shown here that the Dipole Moment in Spherical Coordinates is also zero. Therefore,

$$\Phi_{\text{sphr}}^{(1)} = 0 \quad \forall r \neq a \quad (3.14)$$

3.3 QUADRUPOLE MOMENT

3.3.1 IN CARTESIAN COORDINATES

Recall our expression for the Quadrupole Moment Tensor Q_{ij} for Cartesian coordinates, Eq. (2.31),

$$Q_{ij} = \int_{\mathbb{R}^3} \rho(\mathbf{r}') x'_i x'_j d\tau' \quad (2.31)$$

which, for this problem, becomes

$$Q_{ij} = \frac{M}{2\pi a^2} \int_{\mathbb{R}^3} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) x'_i x'_j d\tau' \quad (3.15)$$

Since we already know that Q_{ij} is symmetric, we only need to find six terms.

Using Eq. (3.15) and Eq. (3.10), we get

$$\begin{aligned} Q_{11} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r' - a) (r')^4 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^3 \theta' d\theta' \int_{\varphi'=0}^{2\pi} \cos^2 \varphi' d\varphi' \\ &= \frac{M}{2\pi a^2} (a^4) (1) (\pi) = \frac{M a^2}{2} \end{aligned} \quad (3.16a)$$

$$\begin{aligned} Q_{22} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r' - a) (r')^4 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^3 \theta' d\theta' \int_{\varphi'=0}^{2\pi} \sin^2 \varphi' d\varphi' \\ &= \frac{M}{2\pi a^2} (a^4) (1) (\pi) = \frac{M a^2}{2} \end{aligned} \quad (3.16b)$$

$$\begin{aligned} Q_{33} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r' - a) (r')^4 dr' \int_{\theta'=0}^{\pi} \cos^2 \theta' \sin \theta' \delta\left(\theta' - \frac{\pi}{2}\right) d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' \\ &= \frac{M}{2\pi a^2} (a) (0) (2\pi) = 0 \end{aligned} \quad (3.16c)$$

$$\begin{aligned} Q_{13} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} (r')^4 \delta(r' - a) dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^2 \theta' \cos \theta' d\theta' \int_{\varphi'=0}^{2\pi} \cos \varphi' d\varphi' \\ &= \frac{M}{2\pi a^2} (a')^4 (0) (0) = 0 \end{aligned} \quad (3.16d)$$

It can be easily shown that all other off-diagonal terms of Q_{ij} are zeros.

If we use q_{ij} defined in Eq. (2.35), we obtain the following

$$q_{11} = q_{22} = \frac{Ma^2}{2} \quad (3.17a)$$

$$q_{33} = -Ma^2 \quad (3.17b)$$

$$q_{12} = q_{13} = q_{23} = 0. \quad (3.17c)$$

However, for $\Phi^{(2)}$, we need to use Eq. (2.37),

$$\Phi_{\text{cart}}^{(2)} = -\frac{G}{2} \frac{\mathbf{r} \cdot \overset{\leftrightarrow}{\mathbf{q}} \cdot \mathbf{r}}{|\mathbf{r}|^5} \quad (3.18)$$

$$= -\frac{MGa^2}{2r^5} \begin{bmatrix} r \sin \theta \cos \varphi & r \sin \theta \sin \varphi & r \cos \theta \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} r \sin \theta \cos \varphi \\ r \sin \theta \sin \varphi \\ r \cos \theta \end{bmatrix} \quad (3.19)$$

$$= -\frac{MGa^2}{2r^5} \left(\frac{1}{2} r^2 \sin^2 \theta \cos^2 \varphi + \frac{1}{2} r^2 \sin^2 \theta \sin^2 \varphi + -r^2 \cos^2 \theta \right) \quad (3.20)$$

$$= -\frac{MGa^2}{2r^3} \left(\frac{1}{2} \sin^2 \theta - \cos^2 \theta \right) \quad (3.21)$$

$$= -\frac{MGa^2}{4r^3} (1 - 3 \cos^2 \theta) \quad (3.22)$$

Finally, we have an expression for the quadrupole potential using the Cartesian expansion.

$$\Phi_{\text{cart}}^{(2)} = -\frac{GMa^2}{4r^3} (1 - 3 \cos^2 \theta) \quad (3.23)$$

3.3.2 IN SPHERICAL COORDINATES

For the quadrupole moment, we find q_{20}

$$q_{20} = \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{5}{16\pi}} \int_{r'=0}^{\infty} \delta(r' - a) (r')^4 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) (3\cos^2\theta' - 1) \sin\theta' d\theta' \int_{\phi'=0}^{2\pi} d\phi' & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{5}{16\pi}} \int_{r'=0}^{\infty} \delta(r' - a) \frac{1}{r'} dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) (3\cos^2\theta' - 1) \sin\theta' d\theta' \int_{\phi'=0}^{2\pi} d\phi' & r < a \end{cases} \quad (3.24a)$$

$$= \begin{cases} \frac{M}{2\pi a^2} \sqrt{\frac{5}{16\pi}} (a)^4 (-1) (2\pi) = -Ma^2 \sqrt{\frac{5}{16\pi}} & r > a \\ \frac{M}{2\pi a^2} \sqrt{\frac{5}{16\pi}} \frac{1}{a} (-1) (2\pi) = -\frac{M}{a^3} \sqrt{\frac{5}{16\pi}} & r < a \end{cases} \quad (3.24b)$$

As an aside, we also show that $q_{21} = 0$,

$$q_{21} = \begin{cases} \frac{-M}{2\pi a^2} \sqrt{\frac{15}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) (r')^4 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^2\theta' \cos\theta' d\theta' \int_{\phi'=0}^{2\pi} e^{i\phi} d\phi' & r > a \\ \frac{-M}{2\pi a^2} \sqrt{\frac{15}{4\pi}} \int_{r'=0}^{\infty} \delta(r' - a) \frac{1}{r'} dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^2\theta' \cos\theta' d\theta' \int_{\phi'=0}^{2\pi} e^{i\phi} d\phi' & r < a \end{cases} \quad (3.25)$$

$$= \begin{cases} \frac{-M}{2\pi a^2} \sqrt{\frac{15}{4\pi}} (a^4) (0) (0) = 0 & r > a \\ \frac{-M}{2\pi a^2} \sqrt{\frac{15}{4\pi}} \frac{1}{a} (0) (0) = 0 & r < a \end{cases} \quad (3.26)$$

With this, the potential due to the quadrupole moment $\Phi^{(2)}$ becomes

$$\Phi_{\text{sphr}}^{(2)} = \begin{cases} \frac{1}{4} G M a^2 \frac{1}{r^3} (3\cos^2\theta - 1) & r > a \\ \frac{1}{4} \frac{G M}{a^3} r^2 (3\cos^2\theta - 1) & r < a \end{cases} \quad (3.27)$$

3.4 USING LAPLACE'S EQUATION

Using the method of direct integration, we can find the potential of a ring of mass M for field points at the z -axis. That is, if we directly use Eq. (2.7), where we let

$$\mathbf{r} = a \cos \varphi \hat{\mathbf{i}} + a \sin \varphi \hat{\mathbf{j}} \quad (3.28a)$$

$$\mathbf{r}' = z \hat{\mathbf{k}} \quad (3.28b)$$

along with Eq. (3.5) and Eq. (3.10d), we obtain

$$d\Phi = -\frac{GM}{2\pi a^2} \frac{1}{\sqrt{z^2 + a^2}} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a)(r')^2 \sin \theta' dr' d\theta' d\varphi'. \quad (3.29)$$

Integrating,

$$\Phi = -\frac{GM}{2\pi a^2} \frac{1}{\sqrt{z^2 + a^2}} \int_{r'=0}^{\infty} \delta(r' - a)(r')^2 dr' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin \theta' d\theta' \int_{\varphi'=0}^{2\pi} d\varphi' \quad (3.30)$$

$$= -\frac{GM}{2\pi a^2} \frac{1}{\sqrt{z^2 + a^2}} (a^2)(1)(2\pi) = -\frac{GM}{\sqrt{z^2 + a^2}} \quad (3.31)$$

We note that this is the potential only at the z -axis. We indicate this by writing $\Phi(r, \theta) = \Phi(z, \theta = 0, \pi)$.

In order to find the potential everywhere, we use the solution of the Laplace Equation

$$\Phi(r, \theta) = \sum_{l=0}^{\infty} \left[A_l r^l + B_l r^{-(l+1)} \right] P_l(\cos \theta). \quad (3.32)$$

If we use the case for a field point at $r > a$, we need to set $A_l = 0, \forall l \in \mathbb{Z}$. The potential then reduces to

$$\Phi(r, \theta) = \sum_{l=0}^{\infty} \left[B_l r^{-(l+1)} \right] P_l(\cos \theta). \quad (3.33)$$

In order to find the coefficients B_l , we first apply the boundary condition Eq. (3.31),

$$\Phi(z, \theta = 0) = \sum_{l=0}^{\infty} \left[B_l z^{-(l+1)} \right], \quad (3.34)$$

where we used $P_l(1) = 1$. We can now equate this with the boundary condition

$$\sum_{l=0}^{\infty} [B_l z^{-(l+1)}] = -\frac{GM}{\sqrt{z^2 + a^2}}, \quad (3.35)$$

where we now need to solve for the constants B_l .

The way to do this is to expand the right-hand side using the binomial series expansion:

$$(1+x)^n = \sum_{k=0}^{\infty} \binom{n}{k} x^k, \quad (3.36)$$

where we require $|x| < 1$ for the series to converge, and $\binom{n}{k}$ is the binomial coefficient, defined by

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \quad (3.37)$$

However, the right-hand side of Eq. (3.35) does not appear to satisfy this condition.

We rewrite the equation as follows:

$$-\frac{GM}{\sqrt{z^2 + a^2}} = -\frac{GM}{z} \frac{1}{\sqrt{1 + (a/z)^2}}, \quad (3.38)$$

and now this satisfies the convergence condition, as $z > a$. We can now apply Eq. (3.36),

$$(1 + (a/z)^2)^{-1/2} = \sum_{k=0}^{\infty} \binom{-1/2}{k} (a/z)^{2k}, \quad (3.39)$$

and expand this in the first two terms:

$$(1 + (a/z)^2)^{-1/2} = \binom{-1/2}{0} (a/z)^0 + \binom{-1/2}{1} (a/z)^2 + \sum_{k=2}^{\infty} \binom{-1/2}{k} (a/z)^{2k}. \quad (3.40)$$

Evaluating the binomial coefficients, and dropping the remainder terms, we obtain

$$(1 + (a/z)^2)^{-1/2} = 1 - \frac{1}{2} a^2 \frac{1}{z^2}. \quad (3.41)$$

We can now plug this back in Eq. (3.38)

$$-\frac{GM}{\sqrt{z^2 + a^2}} = -\frac{GM}{z} + \frac{GMa^2}{2} \frac{1}{z^3}. \quad (3.42)$$

and again, we plug this in Eq. (3.35), and we also expand the left-hand side and

drop the remainder terms:

$$B_0 \frac{1}{z} + B_1 \frac{1}{z^2} + B_2 \frac{1}{z^3} = -\frac{GM}{z} + \frac{GMa^2}{2} \frac{1}{z^3}. \quad (3.43)$$

In comparing the coefficients, we obtain the following:

$$B_0 = -GM \quad (3.44a)$$

$$B_1 = 0 \quad (3.44b)$$

$$B_2 = \frac{GMa^2}{2} \quad (3.44c)$$

From these coefficients, we can now obtain the potential everywhere using Eq. (3.33)

$$\Phi(r, \theta) = -GM \frac{1}{r} P_0(\cos \theta) + \frac{GMa^2}{2} \frac{1}{r^3} P_2(\cos \theta) \quad (3.45)$$

We write out the Legendre Polynomials:

$$\Phi^{(0)} = -GM \frac{1}{r} \quad (3.46a)$$

$$\Phi^{(2)} = \frac{GMa^2}{4} \frac{1}{r^3} (3 \cos^2 \theta - 1) \quad (3.46b)$$

We note that Eq. (3.46a) is the same form as Eq. (2.25).

We also have to consider $r < a$. Using the form of the solution again, but noting that the $B_l r^{-(l+1)}$ term goes to infinity as r goes to zero, the potential now becomes

$$\Phi(r, \theta) = \sum_{l=0}^{\infty} [A_l r^l] P_l(\cos \theta). \quad (3.47)$$

Evaluating this at the z -axis, and applying the boundary condition,

$$\sum_{l=0}^{\infty} [A_l r^l] = -\frac{GM}{\sqrt{z^2 + a^2}}. \quad (3.48)$$

We consider the first three terms again, and expand the right-hand side using the Taylor Series.

$$A_0 + A_1 z + A_2 z^2 = -\frac{GM}{a} + \frac{1}{2} \frac{GM}{a^3} z^2. \quad (3.49)$$

Comparing the coefficients, we have

$$A_0 = -\frac{GM}{a} \quad (3.50a)$$

$$A_1 = 0 \quad (3.50b)$$

$$A_2 = \frac{GM}{2a^3} \quad (3.50c)$$

Now we have a full expression for the potential

$$\Phi_{\text{lapl}}(r, \theta) = \begin{cases} -\frac{GM}{r} + \frac{GMa^2}{4} \frac{1}{r^3} (3 \cos^2 \theta - 1) & r > a \\ -\frac{GM}{a} + \frac{GM}{4a^3} r^2 (3 \cos^2 \theta - 1) & r < a \end{cases} \quad (3.51)$$

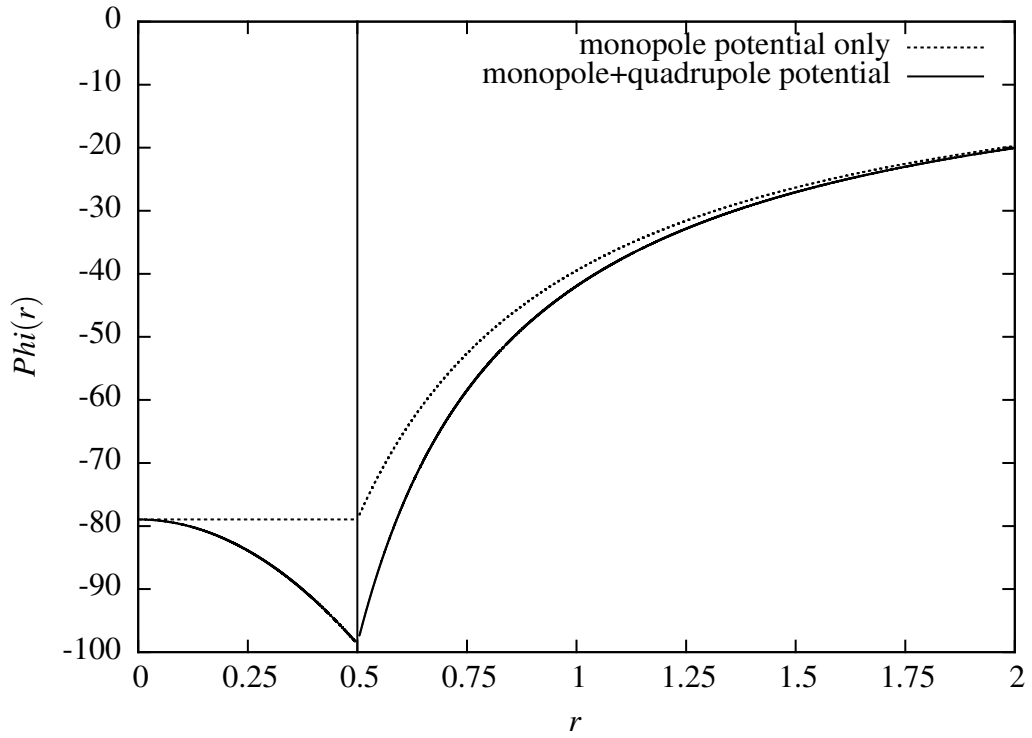


Figure 3.2: Comparison of the Monopole-only and Monopole with Quadrupole Potential of the ring on the equatorial plane

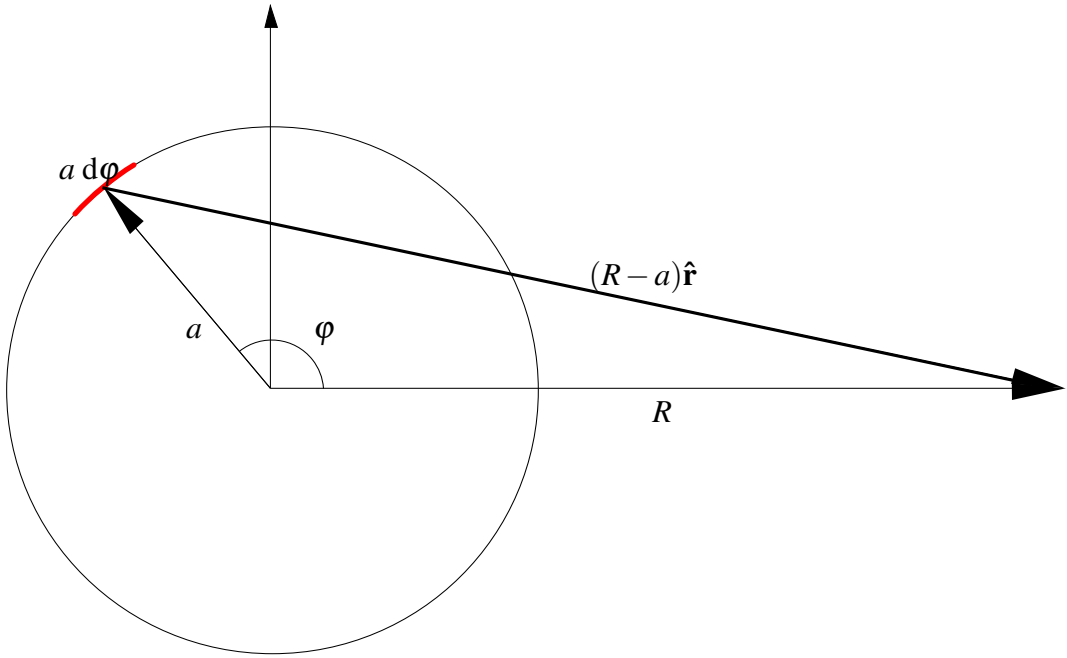


Figure 3.3: Origin to field point is R , radius of the ring is a .

3.5 ON THE PLANE

In order to confirm the potential obtained using the multipole expansion, we can solve for an exact expression for the potential by directly integrating the ring of mass on the equatorial plane. From the geometry of in Figure 3.3, and using the cosine law, we know that

$$|\mathbf{r} - \mathbf{r}'|^2 = R^2 + a^2 - 2aR \cos \varphi \quad (3.52)$$

and

$$dM = \frac{M}{2\pi a^2} \delta(r' - a) (r')^2 dr' d\varphi' \quad (3.53)$$

so we substitute these in Eq. (2.4), and integrating

$$\Phi = -\frac{GM}{2\pi} \int_{\varphi'=0}^{2\pi} \frac{d\varphi'}{\sqrt{R^2 + a^2 - 2aR \cos \varphi}} \quad (3.54)$$

While it is possible to find an analytical expression for Eq. (3.54), we can also integrate it numerically. Here, we will use the Romberg integration method. The result of the integration is in Figure 3.4.

In Figure 3.5, we see that the error in the multipole potential approximation

only increases as $r \rightarrow a$. Given this information, in the next chapter, we will only consider orbits where the error is sufficiently small.

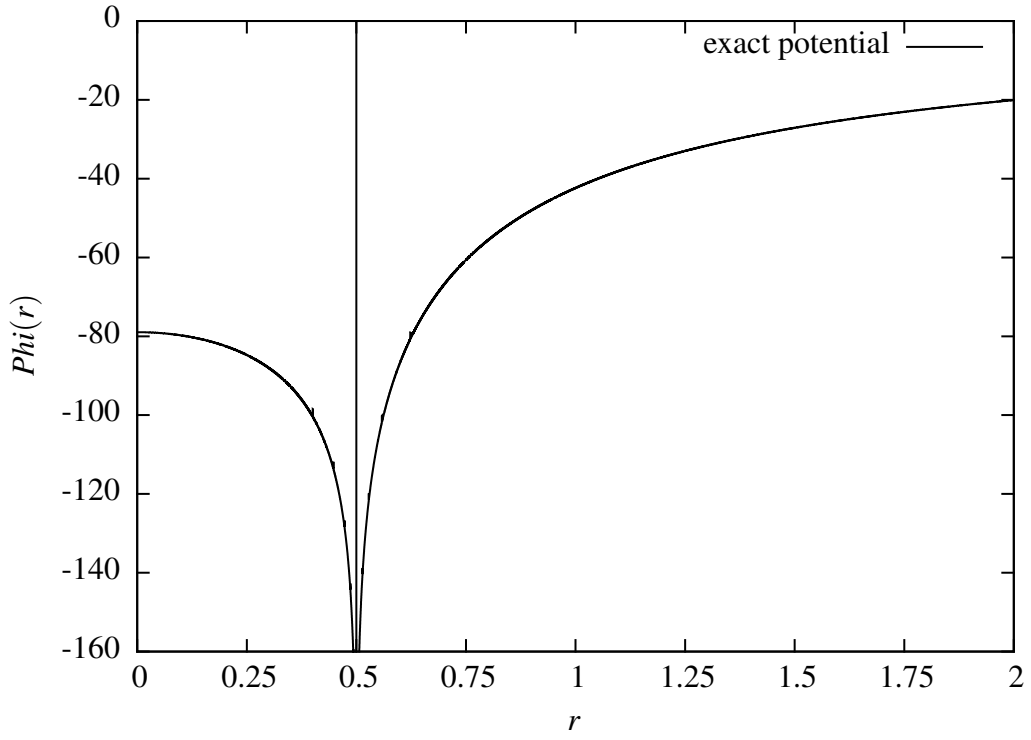


Figure 3.4: Exact Ring multipole potential on the plane

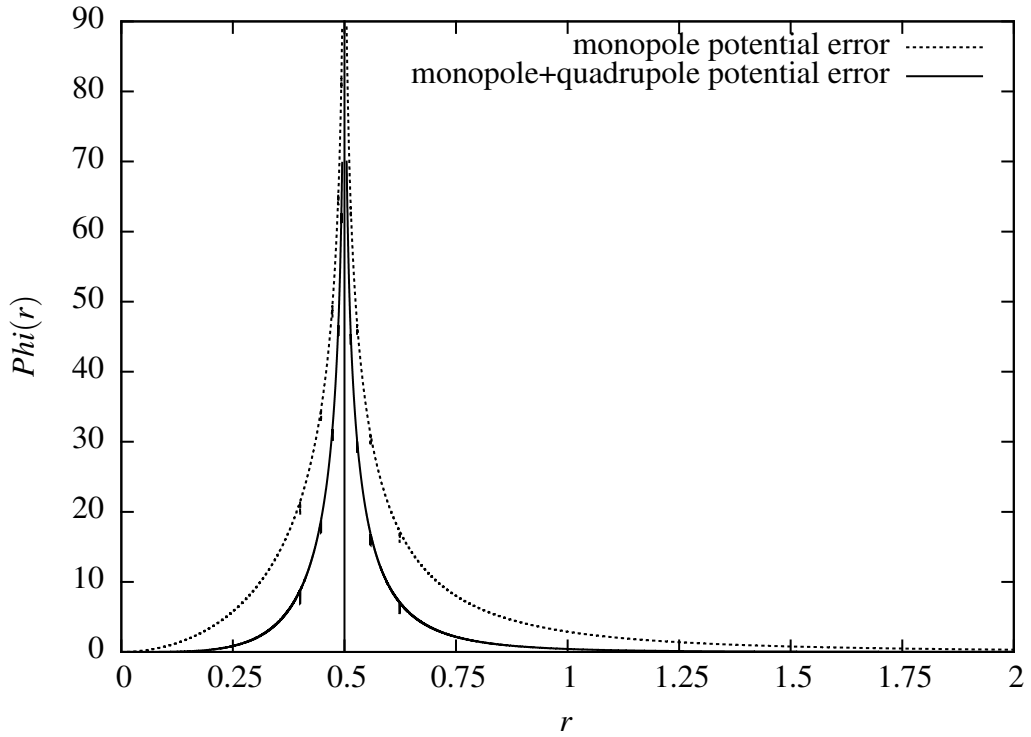


Figure 3.5: The difference of the ring multipole potential at each r and the exact potential obtain from numerical integration.

We now have the potential up to the quadrupole moment for a simple physical object given by

$$\Phi(r, \theta) = \begin{cases} -GM\frac{1}{r} + \frac{GMa^2}{4} \frac{1}{r^3} (3\cos^2 \theta - 1) & r > a \\ -\frac{GM}{a} + \frac{GM}{4a^3} r^2 (3\cos^2 \theta - 1) & r < a \end{cases}$$

However, we only consider orbits outside the sphere bounding the object. Why? This is an approximation of an extended physical body, and in most cases, this body will be a solid. The orbiting body cannot penetrate the surface of a solid. Even in the case something like a star, an orbiting planet cannot enter the surface a star and exit on another side without being obliterated.

Given that in mind, in the following section, we look at certain characteristics of the orbit given the potential, as well as finding the actual orbit using numerical methods.

CHAPTER 4

ORBIT FOR THE QUADRUPOLE POTENTIAL

4.1 DYNAMICAL VARIABLES

4.1.1 LAGRANGIAN AND HAMILTONIAN

We construct the Lagrangian* \mathcal{L} for this system

$$\mathcal{L} = \frac{1}{2}m(\dot{r}^2 + r^2\dot{\theta}^2 + r^2\sin^2\theta\dot{\phi}^2) + \frac{GMm}{r} - \frac{GMma^2}{4}\frac{1}{r^3}(3\cos^2\theta - 1) \quad (4.1)$$

where we see that ϕ is a cyclic coordinate. As a result, the conjugate angular momentum $L_\phi = (=L_z)$ is conserved.

Similarly, we can construct the Hamiltonian. We can simply use $H = T + U$ since the derived potential energy does not depend on the velocity.

$$H = \frac{p_r^2}{2m} + \frac{L_\theta^2}{2m} + \frac{L_\phi^2}{2m\sin^2\theta} - \frac{GMm}{r} + \frac{GMma^2}{4r^3}(3\cos^2\theta - 1) \quad (4.2)$$

Using the canonical equations, we get

$$\begin{aligned} \dot{r} &= \frac{p_r}{m} & \dot{p}_r &= -\frac{GMm}{r^2} + \frac{3}{4}\frac{GMma^2}{r^4}(3\cos^2\theta - 1) \\ \dot{\theta} &= \frac{L_\theta}{m} & \dot{L}_\theta &= \frac{p_\phi^2}{mr^2}\frac{\cos\theta}{\sin^3\theta} - \frac{3}{4}\frac{GMma^2}{r^3}\sin 2\theta \\ \dot{\phi} &= \frac{L_\phi}{m\sin^2\theta} & \dot{L}_\phi &= 0 \end{aligned} \quad (4.3)$$

* In order to prevent confusion between the magnitude of the angular momentum L and the Lagrangian, the latter is written as a curly \mathcal{L} .

4.1.2 ANGULAR MOMENTUM ON THE EQUATORIAL PLANE

We can also show that the Angular Momentum of the orbit if it is in the equatorial plane is constant. If we take the gradient of the potential and restrict it such that $\theta = \pi/2$, the force becomes

$$\mathbf{F} = \left(-GMm\frac{1}{r^2} - \frac{3}{4}GMma^2\frac{1}{r^4} \right) \hat{\mathbf{r}}. \quad (4.4)$$

Note that this is a central force, and from the symmetry, we can find a conserved quantity.

The potential of the equatorial case can be written as

$$U(r) = -GMm\frac{1}{r} - \frac{GMma^2}{4}\frac{1}{r^3} \quad (4.5)$$

Consider the time derivative of angular momentum

$$\frac{d\mathbf{L}}{dt} = \dot{\mathbf{r}} \times \mathbf{p} + \mathbf{r} \times \dot{\mathbf{p}} \quad (4.6)$$

The first term is zero, and we use Newton's second law on the second.

$$\frac{d\mathbf{L}}{dt} = \mathbf{r} \times \mathbf{F} = \mathbf{r} \times f(r)\hat{\mathbf{r}} = \mathbf{0} \implies \mathbf{L} = \text{constant} \quad (4.7)$$

However, the angular momentum in the equatorial plane is entirely in the z-direction. This indicates that an orbit in the equatorial plane will stay on the plane. The difference between the equatorial orbit and the more general case will be shown when we use a numerical methods in order to obtain the actual orbit in the following section.

4.2 ORBIT SIMULATION

In order to simplify the code used, we will have to rewrite Φ in terms of Cartesian Coordinates

$$\Phi(x, y, z) = -\frac{GM}{r} + \frac{GMa^2}{4r^5}(3z^2 - r^2), \quad (4.8)$$

where

$$r = \sqrt{x^2 + y^2 + z^2} \quad (4.9)$$

Taking the gradient in order to find the acceleration, we have

$$\ddot{x} = -\frac{GM}{r^3}x + \frac{5GMa^2}{4r^7}(3z^2)x - \frac{3GMa^2}{4r^5}x \quad (4.10a)$$

$$\ddot{y} = -\frac{GM}{r^3}y + \frac{5GMa^2}{4r^7}(3z^2)y - \frac{3GMa^2}{4r^5}y \quad (4.10b)$$

$$\ddot{z} = -\frac{GM}{r^3}z + \frac{5GMa^2}{4r^7}(3z^2)z - \frac{3GMa^2}{4r^5}z - \frac{6GMa^2}{4r^5}z \quad (4.10c)$$

We then separate each component into a pair of first order differential equations

$$\begin{aligned} \dot{v}_x &= -\frac{GM}{r^3}x + \frac{5GMa^2}{4r^7}(3z^2)x - \frac{3GMa^2}{4r^5}x & \dot{x} &= v_x \\ \dot{v}_y &= -\frac{GM}{r^3}y + \frac{5GMa^2}{4r^7}(3z^2)y - \frac{3GMa^2}{4r^5}y & \dot{y} &= v_y \\ \dot{v}_z &= -\frac{GM}{r^3}z + \frac{5GMa^2}{4r^7}(3z^2)z - \frac{3GMa^2}{4r^5}z - \frac{6GMa^2}{4r^5}z & \dot{z} &= v_z \end{aligned} \quad (4.11)$$

For the purposes of the simulation, we will use a system of units where the unit of length is the Astronomical Unit, the unit of time is the year, and the unit of mass is defined as setting the mass of the attractor to unity. Using Kepler's Third Law, we have

$$GM = 4\pi^2. \quad (4.12)$$

For the numerical solution, we will use the second-order Runge-Kutta method, which is commonly discussed in Computational Physics texts as an improvement of the Euler-Cromer Method. The actual code used can be found the Appendix.

The results of the simulations will be presented in the following section.

We have to note that none of the orbits are meant to be an exhaustive representation of all possible orbits. The initial conditions are all selected at random.

To avoid redundancy of legends in the plots, all red orbits represent the Keplerian case where only the monopole moment is considered, the black orbits represent the orbit due to the monopole and quadrupole moments. For the two-dimensional plots, the hatched areas represent the bounding surface of the attractor, and in the three-dimensional plots, this surface will be **represented** by a sphere centered at the origin.

4.3 CLASSIFICATION OF ORBITS

4.3.1 BOUND PLANAR ORBITS

Orbits confined in the equatorial plane stay on the equatorial plane for all time, due to the conservation of angular momentum along the z direction. Naturally, neither presented orbits have any velocity component in the z -direction.

Figure 4.1 has initial conditions of $\mathbf{r}_0 = (1, 0, 0)$ and $\mathbf{v}_0 = (0, 7, 0)$. From the earlier discussion, we showed how the error for the monopole-only potential is large compared to the monopole-quadrupole potential. This difference in error is then seen in the variation of the orbit generated by both potentials.

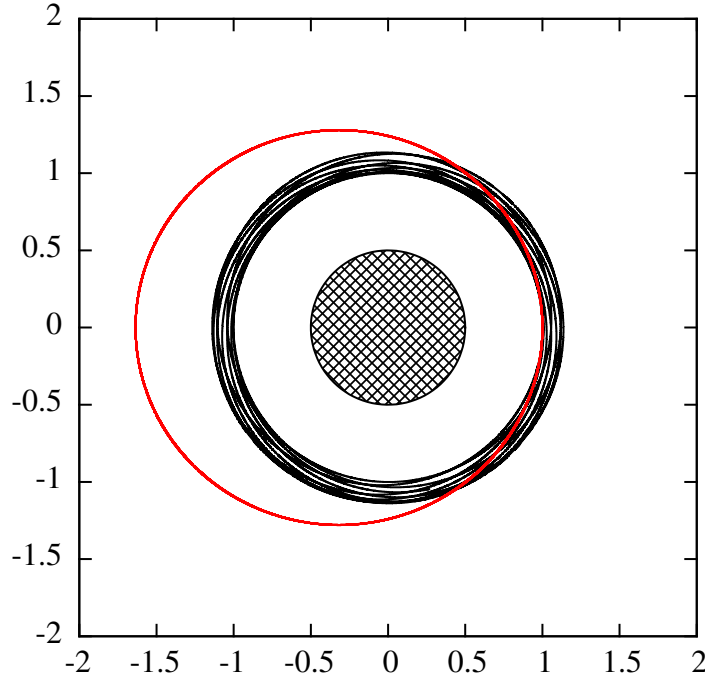


Figure 4.1: Near-attractor orbit comparison. Notice the difference between the monopole-only orbit and the orbit with the additional term.

Figure 4.2 has initial conditions of $\mathbf{r}_0 = (3, 0, 0)$ and $\mathbf{v}_0 = (0, 4, 0)$. Here, there is less difference in the error between the monopole-only and monopole-quadrupole potential. However, from Bertrand's Theorem, only the Keplerian orbit is closed, while the additional potential term causes the orbit to remain open.

We also look at the case where the orbit is on a polar plane. From the conservation of angular momentum, this will generate an orbit on a single plane per-

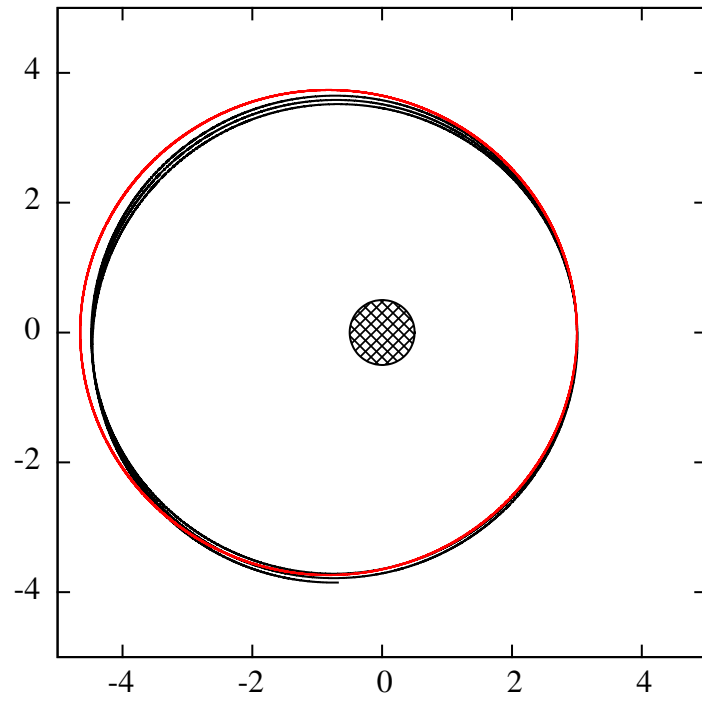


Figure 4.2: Far-attractor orbit comparison. Upon closer inspection, it is seen that the red orbit closes, as expected, and the black orbit does not.

pendicular to the xy plane. Figure 4.3 has initial conditions of $\mathbf{r}_0 = (0,0,1)$ and $\mathbf{v}_0 = (0,7,0)$.

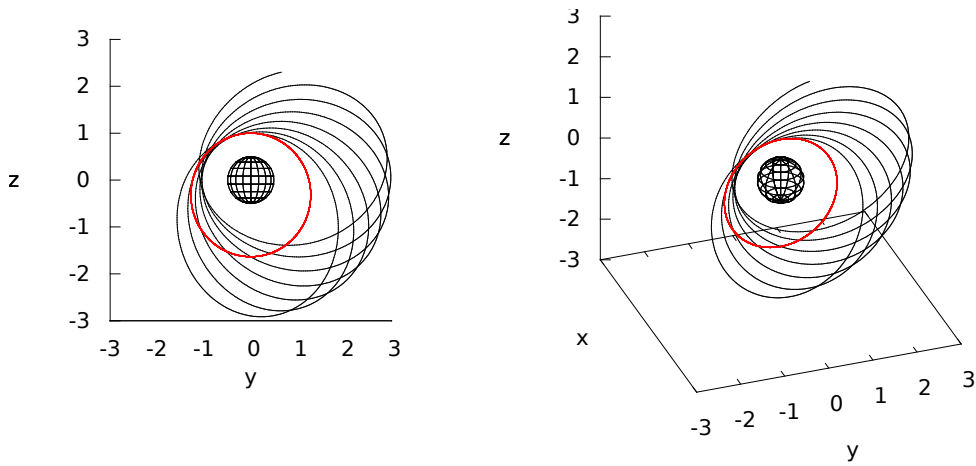


Figure 4.3: Given an initial condition where the initial position is at the z -axis, and the initial velocity lies only on the azimuthal direction, we get an orbit restricted in a polar plane.

4.3.2 INVALID AND UNBOUND ORBITS

Since the attractor is no longer a point mass, we can also look at orbits where the particle enters, or falls into, the sphere separating the attractor from the rest of space.

The simulation will end as soon as this happens.

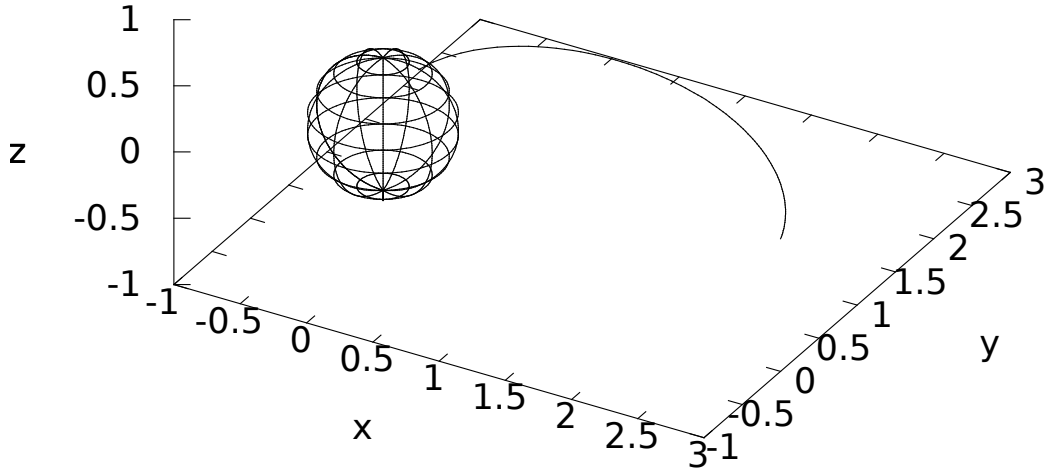


Figure 4.4: Given a low enough velocity, the orbiting particle will fall into the attractor.

Similar to how parabolic and hyperbolic orbits can result from the Keplerian orbit, the added potential term in the equatorial plane, can also generate unbound orbits. In Figure 4.5, the initial conditions are $\mathbf{r}_0 = (1, 0, 0)$ and $\mathbf{v}_0 = (0, 12, 0)$.

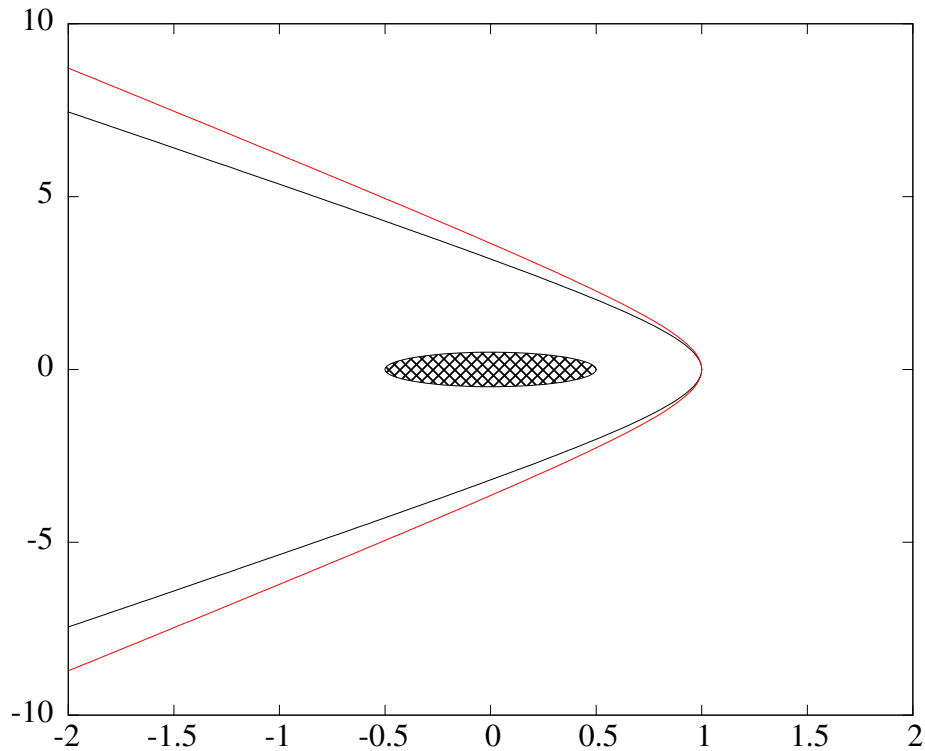


Figure 4.5: An unbound orbit that approaches the attractor at $r = 1$

4.3.3 GENERAL CASE OF THE BOUND ORBIT

Lastly, we consider the most general case of the bound orbit - these are bound within a topological torus. However, due to the conservation of angular momentum, only specific regions of the shell are covered by the orbit. In this section, we look at an orbit with initial conditions $\mathbf{r}_0 = (3, 0, 0)$ and $\mathbf{v}_0 = (0, 2, 2)$ on Figure 4.6. We also look at the surface traced out by the angular momentum vector in Figure 4.7.

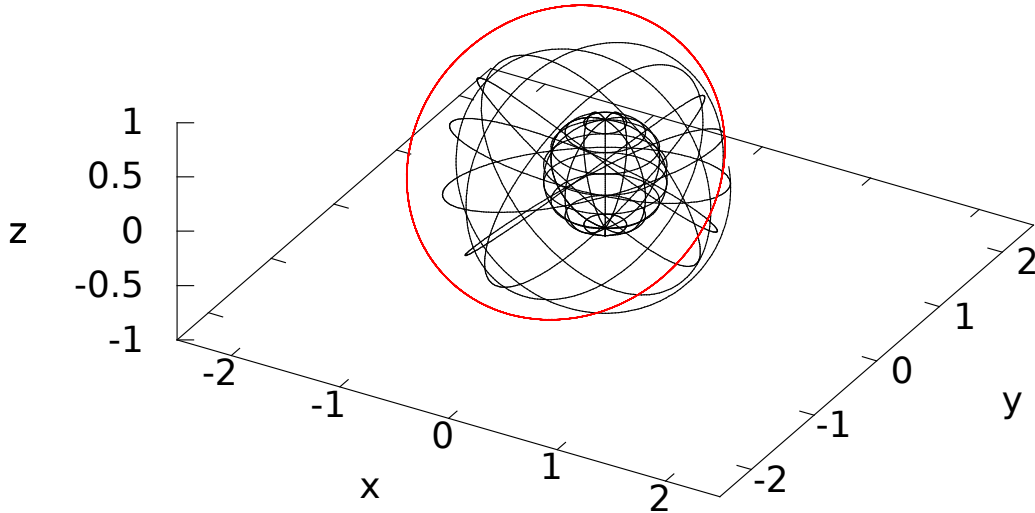


Figure 4.6: Given the proper initial conditions, we can get an orbit that is confined in a topological torus outside the attractor.

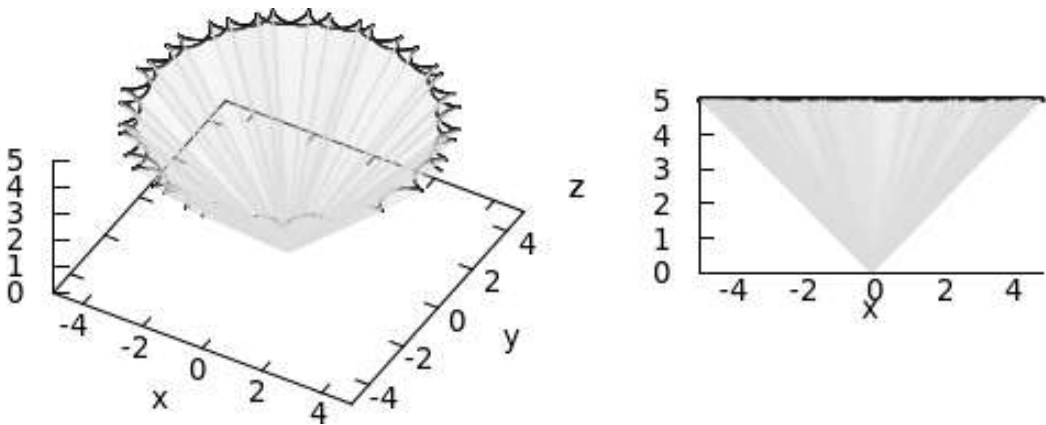
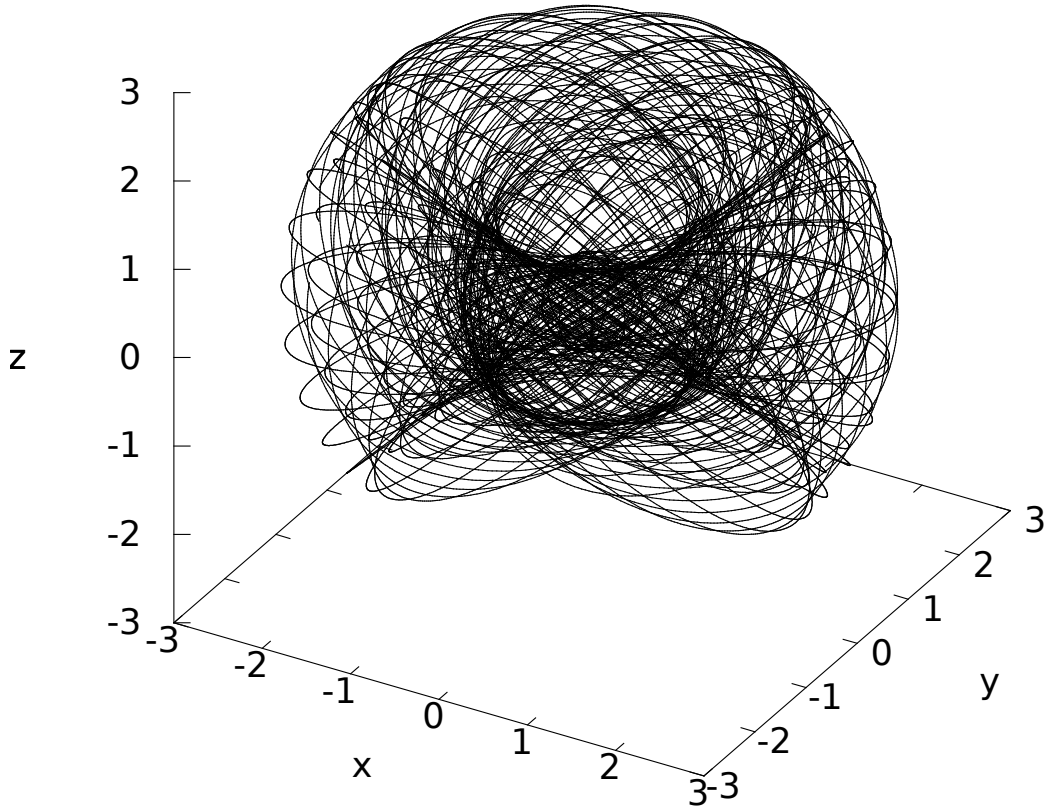


Figure 4.7: This is the cone generated by the angular momentum vectors. We see here that the cone of constant height indicates that the angular momentum along the z-direction is constant.

4.4 ORBIT RETURN MAP

From the last orbit presented in the previous section, we cannot truly say that the orbit remains confined in a topological torus. If we let the code run for a longer time interval, we get a rather messy figure that appears as follows.



In order to have a better idea of how the orbit is confined in a topological torus, we plot the intersections with a plane that cuts the torus. The generated image is called the return map - called as such since the orbit is mapped from one point to another as it returns in the cross-sectional plane.

In order to generate the return map, points crossing the plane of interest were filtered out. Then, a parametric equation of the line containing the points crossing was intersected with the yz plane, which then generates the points indicated in the map. The code used to generate these plots are listed in the Appendix. The return map in Figure 4.8 has initial conditions $\mathbf{r}_0 = (3, 0, 0)$ and $\mathbf{v}_0 = (0, 2, 2)$, and the in Figure 4.9 has initial conditions $\mathbf{r}_0 = (2, 0, 1)$ and $\mathbf{v}_0 = (4, 3, 0)$.

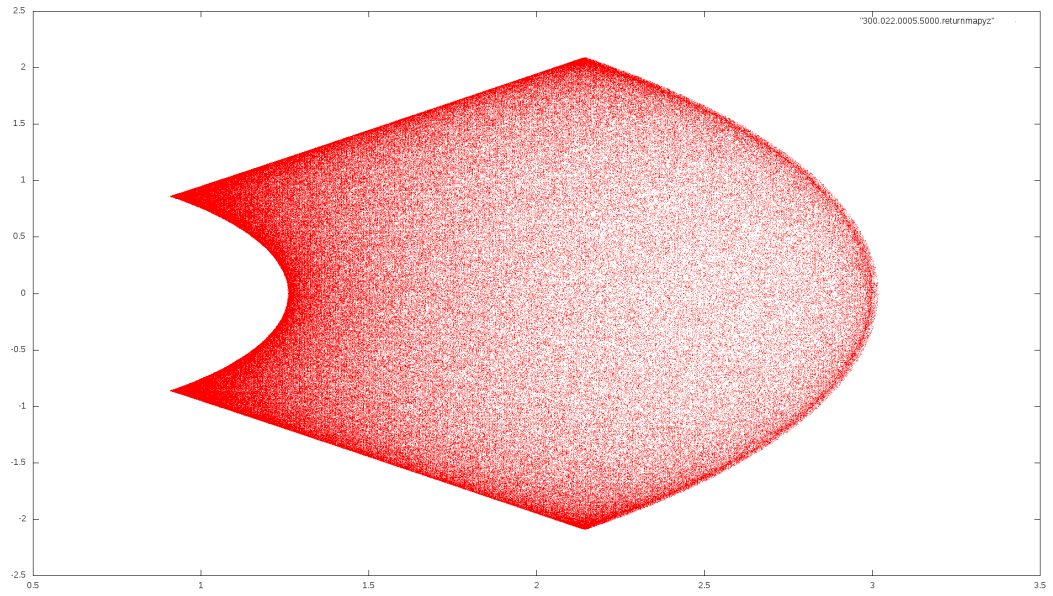


Figure 4.8: Return Map for $\mathbf{r}_0 = (3, 0, 0)$

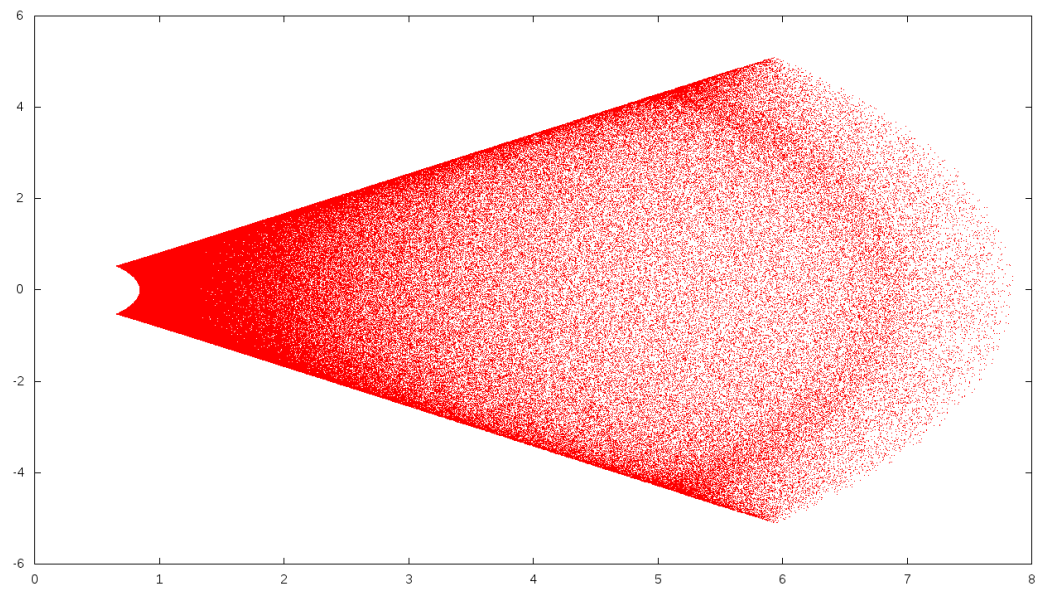


Figure 4.9: Return Map for $\mathbf{r}_0 = (2, 0, 1)$

CHAPTER 5

CONCLUSION

In this paper, we looked at the orbits about an extended body where we used the Multipole Moment Expansion of the Gravitational Potential in order to derive an approximate potential. Specifically, we derived the potential up to the quadrupole moment of a ring of uniform mass, which can be easily extended into other shapes exhibiting azimuthal symmetry. We also compared the numerical value of the potential

with the answer derived from the multipole moments, and we were able to show that the error is less with the added term.

From this potential we then showed that the z-component of the angular momentum is constant, as the azimuthal component is a cyclic coordinate. With this result, we were able to say that the orbit is confined into a torus-like region inside a spherical shell. We were also able to visualize these conclusions using numerical methods.

CHAPTER A

LIST OF SPHERICAL HARMONICS

$$Y_{l,-m}(\theta, \varphi) = (-1)^m Y_{l,m}^*(\theta, \varphi), \quad (\text{A.1})$$

$$Y_{0,0}(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{1}{\pi}} \quad (\text{A.2})$$

$$Y_{1,0}(\theta, \varphi) = \frac{1}{2} \sqrt{\frac{3}{\pi}} \cdot \cos \theta \quad (\text{A.3})$$

$$Y_{1,1}(\theta, \varphi) = \frac{-1}{2} \sqrt{\frac{3}{2\pi}} \cdot e^{i\varphi} \cdot \sin \theta \quad (\text{A.4})$$

$$Y_{2,0}(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{5}{\pi}} \cdot (3 \cos^2 \theta - 1) \quad (\text{A.5})$$

$$Y_{2,1}(\theta, \varphi) = \frac{-1}{2} \sqrt{\frac{15}{2\pi}} \cdot e^{i\varphi} \cdot \sin \theta \cdot \cos \theta \quad (\text{A.6})$$

$$Y_{2,2}(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{15}{2\pi}} \cdot e^{2i\varphi} \cdot \sin^2 \theta \quad (\text{A.7})$$

$$Y_{3,0}(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{7}{\pi}} \cdot (5 \cos^3 \theta - 3 \cos \theta) \quad (\text{A.8})$$

$$Y_{3,1}(\theta, \varphi) = \frac{-1}{8} \sqrt{\frac{21}{\pi}} \cdot e^{i\varphi} \cdot \sin \theta \cdot (5 \cos^2 \theta - 1) \quad (\text{A.9})$$

$$Y_{3,2}(\theta, \varphi) = \frac{1}{4} \sqrt{\frac{105}{2\pi}} \cdot e^{2i\varphi} \cdot \sin^2 \theta \cdot \cos \theta \quad (\text{A.10})$$

$$Y_{3,3}(\theta, \varphi) = \frac{-1}{8} \sqrt{\frac{35}{\pi}} \cdot e^{3i\varphi} \cdot \sin^3 \theta \quad (\text{A.11})$$

CHAPTER B

FORTRAN CODE

B.1 EXACT RING POTENTIAL

Subroutines B.1, B.2 and B.3 were taken from a text for Numerical Methods [7], pages 134-135. The code was modified slightly to accomodate the radius parameter and is preproduced here without permission.

Source Code B.1: Romberg Integration

```
1 SUBROUTINE qromb(func,a,b,ss,r,radius)
2 INTEGER :: jmax, jmaxp, k, km
3 REAL :: a, b, func, ss, eps
4 REAL :: r, radius
5 EXTERNAL :: func
6 PARAMETER (eps=1.e-6, jmax=20, jmaxp=jmax+1, k=5, km=k-1)
7 INTEGER :: j
8 REAL :: dss, h(jmaxp), s(jmaxp)
9 h(1) = 1
10 DO j=1, jmax
11   CALL trapzd(func,a,b,s(j),j,r,radius)
12   IF (j .GE. k) THEN
13     CALL polint(h(j-km),s(j-km),k,0.,ss,dss)
14     IF (ABS(dss).LE.eps*ABS(ss)) RETURN
15   END IF
16   s(j+1) = s(j)
17   h(j+1) = 0.25*h(j)
18 END DO
19 STOP !'too many steps in qromb'
20 END SUBROUTINE qromb
```

Source Code B.2: Trapezoid Rule

```
1 SUBROUTINE trapzd(func,a,b,s,n,r,radius)
2 INTEGER :: n
3 REAL :: a,b,s,func
4 REAL :: r, radius
5 EXTERNAL func
6 INTEGER :: it,j
7 REAL :: del,sum,tnm,x
8 IF (n.EQ.1) THEN
9   s=0.5*(b-a)*(func(r,a,radius)+func(r,b,radius))
10 ELSE
11   it = 2*(n-2)
12   tnm=it
13   del=(b-a)/tnm
14   x=a+0.5*del
15   sum = 0.0
16   DO j=1, it
17     sum=sum+func(r,x,radius)
18     x=x+del
19   END DO
20   s=0.5*(s+(b-a)*sum/tnm)
21 ENDIF
22 RETURN
23 END SUBROUTINE trapzd
```

Source Code B.3: Polynomial Interpolation

```
1 SUBROUTINE polint(xa,ya,n,x,y,dy)
2 INTEGER n,nmax
3 REAL dy,x,y,xa(n),ya(n)
4 PARAMETER (nmax=10)
5 INTEGER i,m,ns
6 REAL den,dif,dift,ho,hp,w,c(nmax),d(nmax)
7 ns=1
8 dif=ABS(x-xa(1))
9 DO i=1,n
10   dift=ABS(x-xa(i))
11   IF (dift.LT.dif) THEN
12     ns=i
13     dif=dift
14   ENDIF
15   c(i)=ya(i)
16   d(i)=ya(i)
17 END DO
18 y=ya(ns)
19 ns=ns-1
20 DO m=1,n-1
21   DO i=1,n-m
22     ho=xa(i)-x
23     hp=xa(i+m)-x
24     w=c(i+1)-d(i)
25     den=ho-hp
26     IF (den.EQ.0) STOP !'failure in polint'
27     den=w/den
```

```

28      d(i)=hp*den
29      c(i)=ho*den
30  END DO
31  IF (2*ns.LT.n-m) THEN
32      dy=c(ns+1)
33  ELSE
34      dy=d(ns)
35      ns=ns-1
36  END IF
37  y=y+dy
38 END DO
39 RETURN
40 END SUBROUTINE polint

```

Source Code B.4: Exact Potential Integrand

```

1  FUNCTION func(r,th,a)
2  REAL r,th,a,func
3      func = 1/SQRT(r**2+a**2-(2*a*r*COS(th)))
4  RETURN
5  END FUNCTION func

```

Source Code B.5: Multipole Potential Function

```

1  FUNCTION multipolev(r,th,a,mo)
2  IMPLICIT NONE
3  REAL :: r,th,a,multipolev
4  integer :: mo
5  REAL, PARAMETER :: PI = 3.14159265
6  REAL, PARAMETER :: GM = 39.4784176
7  if (mo == 0) then
8      IF (r>a) THEN
9          multipolev = -GM/r
10     ELSE IF (r<a) THEN
11         multipolev = -GM/a
12     ELSE
13         WRITE(*,*) "r!=a"
14         STOP
15     END IF
16 else if (mo == 1) then
17     IF (r>a) THEN
18         multipolev = -GM/r
19     ELSE IF (r<a) THEN
20         multipolev = -GM/a
21     ELSE
22         WRITE(*,*) "r!=a"
23         STOP
24     END IF
25 else if (mo == 2) then
26     IF (r>a) THEN
27         multipolev = -GM/r + ((GM*a**2)/(4.0*r**3))*((3*(COS(th)**2))-1)
28     ELSE IF (r<a) THEN
29         multipolev = -GM/a + ((GM*r**2)/(4.0*a**3))*((3*(COS(th)**2))-1)
30     ELSE

```

```

31     WRITE(*,*) "r!=a"
32     STOP
33 END IF
34 else
35     write(*,*) "moment unavailable"
36 end if
37
38 RETURN
39 END FUNCTION multipolev

```

Source Code B.6: Numerical Integration of Exact Potential Function

```

1  PROGRAM main
2  IMPLICIT NONE
3  REAL, PARAMETER :: PI = 3.14159265
4  REAL, PARAMETER :: GM = 39.4784176
5  REAL, PARAMETER :: m = 3e-3
6  REAL, DIMENSION(:), ALLOCATABLE :: r
7  REAL :: rstep
8  REAL :: phi, res
9  INTEGER :: n,i
10 REAL :: r0,rf,radius
11 REAL, EXTERNAL :: func, multipolev
12 character(len=20) :: filename
13
14 ! Ask the user for input
15 WRITE(*,*) "This program calculates the exact potential"
16 WRITE(*,*) "of the ring on the equatorial plane."
17 WRITE(*,*) "radius"
18 READ(*,*) radius
19 WRITE(*,*) "Set the range of the R"
20 READ(*,*) r0, rf
21 WRITE(*,*) "Step size of R"
22 READ(*,*) rstep
23 WRITE(*,*) "Filename of output"
24 READ(*,*) filename
25
26 n = INT( (rf-r0)/rstep )
27 ALLOCATE(r(n))
28
29 r(1) = r0
30 DO i = 1,n-1
31     r(i+1) = r(i) + rstep
32 END DO
33
34 ! This program generates five files:
35 ! filename.exact where the program will create two-columns
36 !   of data, the first for r and the second for the exact potential
37 OPEN ( unit = 101, file = trim(filename)//".exact", status = 'replace', &
38       & action = 'write')
39 ! filename.mon, used to generate a r vs monopole potential at r
40 OPEN ( unit = 103, file = trim(filename)//".mon", status = 'replace', &
41       & action = 'write')
42 ! filename.monquad, used to generate a r vs monopole and quadrupole potential
43 OPEN ( unit = 104, file = trim(filename)//".monquad", status = 'replace', &

```

```

44      & action = 'write')
45  ! filename.monerr, used to generate a r vs the error assuming that
46  ! we only use the monopole potential only
47  OPEN ( unit = 105, file = trim(filename)//".monerr", status = 'replace', &
48        & action = 'write')
49  ! filename.mqerr, used to generate a r vs the error assuming that
50  ! we only use the monopole and quadrupole potential
51  OPEN ( unit = 106, file = trim(filename)//".mqerr", status = 'replace', &
52        & action = 'write')
53
54  DO i=1,n
55
56  ! given a certain r(i), the subroutine qromb integrates the potential
57  ! and stores it in the variable res. this is then written to file
58    CALL qromb(func,0.0,2*PI,res,r(i),radius)
59    WRITE(101,200) r(i), -(res*GM)/(2*PI)
60
61    WRITE(103,200) r(i), multipolev(r(i),PI/2,radius, 0)
62    WRITE(104,200) r(i), multipolev(r(i),PI/2,radius, 2)
63    WRITE(105,200) r(i), &
64      & ABS(-(res*GM)/(2*PI)) - ABS(multipolev(r(i),PI/2,radius, 0))
65    WRITE(106,200) r(i), &
66      & ABS(-(res*GM)/(2*PI)) - ABS(multipolev(r(i),PI/2,radius, 2))
67  END DO
68
69  200 FORMAT (f12.9, 3x, f12.3)
70  END PROGRAM

```

B.2 RING ORBIT SIMULATION

Source Code B.7: RK2 Subroutine for Orbit Solution

```
1  SUBROUTINE planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, error)
2  !-----
3  !  SUBROUTINE planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, error)
4  !    input: integer n (number of elements in the following arrays)
5  !          number GM (product of attractor mass and grav. const.)
6  !          array x (of n elements)
7  !          array y (of n elements)
8  !          array z (of n elements)
9  !          array vx (of n elements)
10 !          array vy (of n elements)
11 !          array vz (of n elements)
12 !          array t (of n elements)
13 !    output: arrays x y z vx vy vz t
14 !          integer error (returns either zero or one)
15 !          integer num (if n!=num, this indicates that the planet stopped)
16 !    note: this program allows numbers up to 10 decimal places
17 !          : all arrays must already have a set initial value at n=1
18 !          : as outlined in Numerical Methods, page 704,
19 !          expanded to accomodate three independent variables.
20 !-----
21
22 IMPLICIT NONE
23 INTEGER, PARAMETER :: dbl = selected_real_kind(p=10)
24 INTEGER, INTENT(in) :: n
25 INTEGER, INTENT(out) :: error, num
26 REAL(dbl), INTENT(in) :: radius, GM, dt
27 REAL(dbl), INTENT(out), DIMENSION(n) :: t
28 REAL(dbl), INTENT(out), DIMENSION(n) :: x, y, z
29 REAL(dbl), INTENT(out), DIMENSION(n) :: vx, vy, vz
30 INTEGER :: i, j
31 REAL(dbl) :: r, rpri, p, ppri, o, opri, q, qpri, m, mpri
32
33 REAL(dbl) :: xpri, ypri, zpri
34 REAL(dbl) :: vxpri, vypri, vzpri
35 REAL(dbl) :: vmag
36
37 num = 0
38
39 DO i = 1, n-1
40   t(i+1) = t(i) + dt
41
42   r = SQRT(x(i)**2 + y(i)**2 + z(i)**2)
43   xpri = x(i) + ( ( dt*vx(i) ) / 2.0 )
44   ypri = y(i) + ( ( dt*vy(i) ) / 2.0 )
45   zpri = z(i) + ( ( dt*vz(i) ) / 2.0 )
46
47   m = GM/(r**3)
48   o = -3.75*( (GM*radius**2)/(r**7) )*(z(i)**2)
49   p = 0.75*( (GM*radius**2)/(r**5) )
```

```

50  q = 1.50*( (GM*radius**2)/(r**5) )
51  vxpri = vx(i) - (dt*x(i)/2.0)*( m + o + p )
52  vypri = vy(i) - (dt*y(i)/2.0)*( m + o + p )
53  vzpri = vz(i) - (dt*z(i)/2.0)*( m + o + p + q )
54
55  rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
56  x(i+1) = x(i) + (dt*vxpri)
57  y(i+1) = y(i) + (dt*vypri)
58  z(i+1) = z(i) + (dt*vzpri)
59
60  mpri = GM/(rpri**3)
61  opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
62  ppri = 0.75*( (GM*radius**2)/(rpri**5) )
63  qpri = 1.50*( (GM*radius**2)/(rpri**5) )
64  vx(i+1) = vx(i) - (dt*xpri)*( mpri + opri + ppri )
65  vy(i+1) = vy(i) - (dt*ypri)*( mpri + opri + ppri )
66  vz(i+1) = vz(i) - (dt*zpri)*( mpri + opri + ppri + qpri )
67
68  IF ( r < radius ) THEN
69      num = i
70      error = 1                                !particle stopped
71
72      DO j = num, n-1
73          x(j+1) = x(i)
74          y(j+1) = y(i)
75          z(j+1) = z(i)
76          vx(j+1) = 0.0
77          vy(j+1) = 0.0
78          vz(j+1) = 0.0
79      END DO
80
81      EXIT
82  ELSE
83      num = i
84      IF (num == n-1) THEN
85          error = 0
86      END IF
87  END IF
88
89  END DO
90
91  RETURN
92  END SUBROUTINE planetrk2

```

Source Code B.8: Derivation of Actual Orbit for Ring Potential

```

1  PROGRAM planeorbit
2  !-----
3  !   PROGRAM planeorbit
4  !   generates the numerical solution to the orbit
5  !   given the monopole and quadrupole potential
6  !-----
7  IMPLICIT NONE
8
9  INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)

```

```

10 INTEGER :: n, i, num
11 INTEGER :: outputerr, allocerr, rk2error
12 CHARACTER(len=20) :: filename
13
14 REAL(dbl), PARAMETER :: PI = 3.14159265359
15 REAL(dbl) :: GM
16
17 REAL(dbl), DIMENSION(:), ALLOCATABLE :: t
18 REAL(dbl), DIMENSION(:), ALLOCATABLE :: x, y, z
19 REAL(dbl), DIMENSION(:), ALLOCATABLE :: vx, vy, vz
20
21 REAL(dbl) :: dt, tf, r, vmag
22 REAL(dbl) :: x0, y0, z0, vx0, vy0, vz0
23 REAL(dbl) :: radius
24
25 REAL(dbl) :: E, KE, PH
26 !     energies
27
28 REAL(dbl), PARAMETER :: mass = 1.65956463e-7
29 !     uses the ratio of mercury and the sun as the mass of the attractor
30 !     this is only relevant in the computation of the
31 !     energies of the problem
32
33 WRITE(*,*) "This program generates the actual trajectory of a planet "
34 WRITE(*,*) "orbiting about a ring of mass on the equatorial plane"
35 WRITE(*,*) "using the multipole approximation and RK2."
36 WRITE(*,*) "This uses a unit system scaled such that G*M_sun = 4pi^2
    (default)"
37 WRITE(*,*)
38 WRITE(*,*) "Input the following information:"
39 WRITE(*,*) "initial position (cartesian coordinates)"
40 READ(*,*) x0, y0, z0
41 WRITE(*,*) "Initial velocity"
42 READ(*,*) vx0, vy0, vz0
43 WRITE(*,*) "Attractor radius"
44 READ(*,*) radius
45 WRITE(*,*) "Final time, timestep"
46 READ(*,*) tf, dt
47 WRITE(*,*) "(debug) value for GM. enter 0 for default"
48 READ(*,*) GM
49 WRITE(*,*) "Filename for output"
50 READ(*,*) filename
51
52 IF ( GM == 0 ) THEN
53     GM = 4.0*(PI**2)
54 END IF
55
56 n = ABS(INT(tf/dt))
57 WRITE(*,*) "Plotting ", n, "points..."
58
59 ALLOCATE( t(n) , x(n) , y(n) , z(n) , vx(n) , vy(n) , vz(n) , stat=allocerr)
60 IF ( allocerr /= 0 ) THEN
61     WRITE(*,*) "Error in allocation. Exiting..."
62     STOP
63 END IF
64

```

```

65 !      set initial conditions
66 t(1) = 0.0
67 x(1) = x0
68 y(1) = y0
69 z(1) = z0
70 vx(1) = vx0
71 vy(1) = vy0
72 vz(1) = vz0
73
74 OPEN ( unit = 100, file = TRIM(filename)//".gp", status = 'replace', &
75       & action = 'write', iostat=outputerr)
76 IF ( outputerr /= 0 ) THEN
77     WRITE(*,*) "Error in", TRIM(filename)//".gp", ". Exiting..."
78     STOP
79 ELSE
80     WRITE(100,*) "# echo input data"
81     WRITE(100,*) "# Initial Position: ", x0, y0, z0
82     WRITE(100,*) "# Initial velocity: ", vx0, vy0, vz0
83
84     r = SQRT(x0**2 + y0**2 + z0**2)
85     vmag = SQRT(vx0**2 + vy0**2 + vz0**2)
86     WRITE(100,*) "# Initial velocity magnitude: ", vmag
87     KE = 0.5*mass*vmag**2
88     PH = -GM/r + ((0.75*GM*(radius**2)*(z0**2))/r**5) -
90         ((0.25*GM*radius**2)/r**3)
89     E = KE + mass*PH
90     WRITE(100,*) "# Initial kinetic energy: ", KE
91     WRITE(100,*) "# Initial potential energy: ", mass*PH
92     WRITE(100,*) "# Initial total energy: ", E
93     WRITE(100,*) "# Attractor Radius"
94     WRITE(100,*) "r = ", radius
95     WRITE(100,*) "# GM =", GM
96
97     IF ( dt > 0 ) THEN
98         WRITE(100,*) "# Initial time: ", 0.0
99         WRITE(100,*) "# Final time: ", tf
100     ELSE IF ( dt < 0 ) THEN
101         WRITE(100,*) "# Initial time: ", -tf
102         WRITE(100,*) "# Final time: ", 0.0
103     END IF
104
105     WRITE(100,*) "# Timestep: ", ABS(dt)
106 END IF
107
108 IF ( dt > 0 ) THEN
109     CALL planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, rk2error)
110 ELSE IF ( dt < 0 ) THEN
111     CALL planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, rk2error)
112
113     CALL reversearray(x, n)
114     CALL reversearray(y, n)
115     CALL reversearray(z, n)
116     CALL reversearray(vx, n)
117     CALL reversearray(vy, n)
118     CALL reversearray(vz, n)
119 END IF

```



```

120
121
122 WRITE(100,*) "# Points plotted: "
123 WRITE(100,*) "n = ", num
124
125 WRITE(*,*) "rk2error", rk2error
126 WRITE(*,*)
127
128 WRITE(100,*) "plotfilename = ", ""//TRIM(filename)//".plot'"
129 WRITE(100,*) "lvecfilename = ", ""//TRIM(filename)//".lvec'"
130 WRITE(100,*) "outtmpl = ", "plot/"//TRIM(filename)//"%07d.png'"
131
132 WRITE(100,*) "# > bounding box"
133 WRITE(100,*) "# x", MINVAL(x), MAXVAL(x)
134 WRITE(100,*) "# y", MINVAL(y), MAXVAL(y)
135 WRITE(100,*) "# z", MINVAL(z), MAXVAL(z)
136 WRITE(100,*) "set xrange [", MINVAL(x)-0.5, ":", MAXVAL(x)+0.5, "]"
137 WRITE(100,*) "set yrange [", MINVAL(y)-0.5, ":", MAXVAL(y)+0.5, "]"
138 WRITE(100,*) "set zrange [", MINVAL(z)-0.5, ":", MAXVAL(z)+0.5, "]"
139
140 WRITE(100,*) 'set label "attractor radius:', radius, &
141   & '" font "sans,8" right at screen 1, character 6'
142 WRITE(100,203) 'set label "initial position:', x0, y0, z0, &
143   & '" font "sans,8" right at screen 1, character 5'
144 WRITE(100,203) 'set label "initial velocity:', vx0, vy0, vz0, &
145   & '" font "sans,8" right at screen 1, character 4'
146
147 CALL printxyzarray(x, y, z, n, filename)
148 CALL printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)
149
150 203 FORMAT (a, f6.2, f6.2, f6.2, a)
151 DEALLOCATE(t, x, y, z, vx, vy, vz)
152 CLOSE(100)
153 END PROGRAM

```

Source Code B.9: Print XYZ plot to file

```

1 SUBROUTINE printxyzarray(x, y, z, n, filename)
2 !-----
3 ! SUBROUTINE printxyzarray(x, y, z, n, filename)
4 !   input: integer n (number of elements in the following arrays)
5 !           array x (of n elements)
6 !           array y (of n elements)
7 !           array z (of n elements)
8 !           string (20 chars max) filename
9 !   output: file "filename.plot"
10 !           three columns: x y z
11 !   note: this program allows numbers up to 10 decimal places
12 !-----
13
14 IMPLICIT NONE
15 INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
16 REAL(dbl), DIMENSION(n), INTENT(in) :: x, y, z
17 INTEGER, INTENT(in) :: n
18 CHARACTER(len=20), INTENT(in) :: filename

```

```

19
20 INTEGER :: outputerr, i
21
22 OPEN ( unit = 111, file = TRIM(filename)//".plot", status = 'replace', &
23       & action = 'write', iostat=outputerr )
24
25 IF ( outputerr == 0 ) THEN
26   DO i = 1, n
27     WRITE(111,201) x(i), y(i), z(i)
28   END DO
29 END IF
30
31 CLOSE(111)
32 201 FORMAT (f30.10, 3x, f30.10, 3x, f30.10)
33 RETURN
34 END SUBROUTINE printxyzarray

```

Source Code B.10: Plot magnitude of Angular Momentum vs time to file

```

1 SUBROUTINE printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)
2 !-----
3 ! SUBROUTINE printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)
4 !   input: integer n (number of elements in the following arrays)
5 !       array x (of n elements)
6 !       array y (of n elements)
7 !       array z (of n elements)
8 !       array vx (of n elements)
9 !       array vy (of n elements)
10 !       array vz (of n elements)
11 !       array t (of n elements)
12 !       real mass (for debugging purposes)
13 !       string (20 chars max) filename
14 !   output : file "filename.lmag"
15 !           two columns: t lmag
16 !   output : file "filename.lvec"
17 !           three columns: lx ly lz
18 !   note: this program allows numbers up to 10 decimal places
19 !           to generate the surface of angular momentum vectors,
20 !           run "splot 'filename.plot' with vectors nohead" in gnuplot
21 !           this program prints out the angular momentum per unit mass
22 !-----
23
24 IMPLICIT NONE
25 INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
26 REAL(dbl), DIMENSION(n), INTENT(in) :: t
27 REAL(dbl), DIMENSION(n), INTENT(in) :: x,y,z
28 REAL(dbl), DIMENSION(n), INTENT(in) :: vx,vy,vz
29 REAL(dbl), INTENT(in) :: mass
30 INTEGER, INTENT(in) :: n
31 CHARACTER(len=20), INTENT(in) :: filename
32 INTEGER :: outputerr, i
33 REAL(dbl) :: lx, ly, lz, lmag
34
35 OPEN ( unit = 107, file = TRIM(filename)//".lmag", status = 'replace', &
36       & action = 'write', iostat=outputerr)

```

```

37
38 OPEN ( unit = 108, file = TRIM(filename)//".lvec", status = 'replace', &
39       & action = 'write', iostat=outputerr)
40
41 IF ( outputerr == 0 ) THEN
42   DO i = 1, n
43     lx = ((y(i)*vz(i)) - (vy(i)*z(i)))
44     ly = ((z(i)*vx(i)) - (x(i)*vz(i)))
45     lz = ((x(i)*vy(i)) - (y(i)*vx(i)))
46     lmag = SQRT(lx**2 + ly**2 + lz**2)
47     WRITE(107,200) t(i), lmag
48     WRITE(108,202) 0.0, 0.0, 0.0, lx, ly, lz
49   END DO
50 END IF
51
52 CLOSE(107)
53 CLOSE(108)
54 200 FORMAT (f30.10, 3x, f30.10)
55 202 FORMAT (f30.10, 3x, f30.10, 3x, f30.10, 3x, f30.10, 3x, f30.10, 3x,
56           f30.10)
57 END SUBROUTINE printlmag

```

Source Code B.11: Reverse Input Array

```

1 SUBROUTINE reversearray(array, n)
2 !-----
3 ! SUBROUTINE reversearray(array, n)
4 !   description: this subroutine takes the input array and
5 !               outputs the same array in reverse order
6 !   input: integer n (number of elements in the following array)
7 !           array array (of n elements)
8 !   output: array array
9 !   note: this program allows numbers up to 10 decimal places
10 !-----
11
12 IMPLICIT NONE
13
14 INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
15 INTEGER, INTENT(in) :: n
16 REAL(dbl), INTENT(out), DIMENSION(n) :: array
17 REAL(dbl), DIMENSION(n) :: temp
18 INTEGER :: i
19 REAL(dbl) :: othertemp
20
21 DO i = 1, n
22   temp(i) = array(n-i+1)
23 END DO
24
25 DO i = 1, n
26   array(i) = temp(i)
27 END DO
28
29 RETURN
30 END SUBROUTINE reversearray

```

B.3 RETURN MAP

Source Code B.12: Return Map

```
1 PROGRAM returnmap
2 !-----
3 ! PROGRAM returnmap
4 !   description: used to generate the return map on the yz plane
5 !               given a set of initial conditions
6 !   note: this program does not allow modifiable attractor mass
7 !-----
8
9 IMPLICIT NONE
10 REAL, PARAMETER :: PI = 3.14159265359
11 REAL, PARAMETER :: GM = 4.0*PI**2
12 REAL :: radius, dt
13 REAL :: x1, x2, y1, y2, z1, z2
14 REAL :: vx1, vx2, vy1, vy2, vz1, vz2
15 real, dimension(100) :: ycept, zcept
16
17 INTEGER :: i=0, j, n=1, repcount=0, num, rep, outputerr, forbac=1
18 REAL :: r, rpri, p, ppri, vmag, o, opri, q, qpri, m, mpri
19 real :: temp, xc, yc, zc
20
21 REAL :: xpri, ypri, zpri, vxpri, vypri, vzpri
22
23 write(*,*) "generate yz return map of orbit"
24 write(*,*) "warning: not for unbound and invalid orbits"
25 write(*,*) "Initial Position"
26 read(*,*) x1, y1, z1
27 write(*,*) "Initial Velocity"
28 read(*,*) vx1, vy1, vz1
29 write(*,*) "timestep, radius"
30 read(*,*) dt, radius
31 write(*,*) "repeat"
32 read(*,*) rep
33
34 ! this is simply a marker indicating that the program is running
35 write(*,*) "1"
36
37 OPEN ( unit = 100, file = "returnmapyz", status = 'replace', &
38       & action = 'write', iostat=outputerr)
39
40 ! this is a marker indicating that the file is sucessfully opened
41 write(*,*) "2", outputerr
42
43 write(100,*) "# return map using initial position"
44 write(100,*) "#", x1, y1, z1
45 write(100,*) "#initial velocity"
46 write(100,*) "#", vx1, vy1, vz1
47 write(100,*) "# timestep", dt
48 write(100,*) "# radius", radius
49 write(100,*)
```

```

50
51 ! this is a marker indicating that the main program is ready
52 write(*,*) "3"
53
54 DO
55 ! this loop only generates one point in the orbit after another
56 ! no unused data is stored
57   if (forbac == 1) then
58     forbac = 2
59
60   ! generate the next point using the initial conditions
61   r = SQRT(x1**2 + y1**2 + z1**2)
62   xpri = x1 + ( ( dt*vx1 ) / 2.0 )
63   ypri = y1 + ( ( dt*vy1 ) / 2.0 )
64   zpri = z1 + ( ( dt*vz1 ) / 2.0 )
65
66   m = GM/(r**3)
67   o = -3.75*( (GM*radius**2)/(r**7) )*(z1**2)
68   p = 0.75*( (GM*radius**2)/(r**5) )
69   q = 1.50*( (GM*radius**2)/(r**5) )
70   vxpri = vx1 - (dt*x1/2.0)*( m + o + p )
71   vypri = vy1 - (dt*y1/2.0)*( m + o + p )
72   vzpri = vz1 - (dt*z1/2.0)*( m + o + p + q )
73
74   rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
75   x2 = x1 + (dt*vxpri)
76   y2 = y1 + (dt*vypri)
77   z2 = z1 + (dt*vzpri)
78
79   mpri = GM/(rpri**3)
80   opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
81   ppri = 0.75*( (GM*radius**2)/(rpri**5) )
82   qpri = 1.50*( (GM*radius**2)/(rpri**5) )
83   vx2 = vx1 - (dt*xpri)*( mpri + opri + ppri )
84   vy2 = vy1 - (dt*ypri)*( mpri + opri + ppri )
85   vz2 = vz1 - (dt*zpri)*( mpri + opri + ppri + qpri )
86
87   ! if the generated point is a match on the conditions
88   ! the intercept is found and stored in the array
89   ! n counts how many times a match has been found
90   IF ( (x1 > 0.0) .AND. (x2 < 0.0) ) THEN
91     temp = -x1/(x2 - x1)
92
93     yc = y1 + temp*(y2-y1)
94     zc = z1 + temp*(z2-z1)
95
96     if ( (yc > 0.0) ) then
97       ycept(n) = yc
98       zcept(n) = zc
99       n = n + 1
100    end if
101  ELSE IF ( (x1 < 0.0) .AND. (x2 > 0.0) ) THEN
102    temp = -x1/(x2 - x1)
103
104    yc = y1 + temp*(y2-y1)
105    zc = z1 + temp*(z2-z1)

```

```

106
107     if ( (yc > 0.0) ) then
108         ycept(n) = yc
109         zcept(n) = zc
110         n = n + 1
111     end if
112     ELSE IF ( x1 == 0.0 ) THEN
113         if ( (y1 > 0.0) ) then
114             ycept(n) = y1
115             zcept(n) = z1
116             n = n + 1
117         end if
118     END IF
119     else if ( forbac == 2 ) then
120 ! this process repeats the
121         forbac = 1
122
123         r = SQRT(x2**2 + y2**2 + z2**2)
124         xpri = x2 + ( ( dt*vx2 ) / 2.0 )
125         ypri = y2 + ( ( dt*vy2 ) / 2.0 )
126         zpri = z2 + ( ( dt*vz2 ) / 2.0 )
127
128         m = GM/(r**3)
129         o = -3.75*( (GM*radius**2)/(r**7) )*(z2**2)
130         p = 0.75*( (GM*radius**2)/(r**5) )
131         q = 1.50*( (GM*radius**2)/(r**5) )
132         vxpri = vx2 - (dt*x2/2.0)*( m + o + p )
133         vypri = vy2 - (dt*y2/2.0)*( m + o + p )
134         vzpri = vz2 - (dt*z2/2.0)*( m + o + p + q )
135
136         rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
137         x1 = x2 + (dt*vxpri)
138         y1 = y2 + (dt*vypri)
139         z1 = z2 + (dt*vzpri)
140
141         mpri = GM/(rpri**3)
142         opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
143         ppri = 0.75*( (GM*radius**2)/(rpri**5) )
144         qpri = 1.50*( (GM*radius**2)/(rpri**5) )
145         vx1 = vx2 - (dt*xpri)*( mpri + opri + ppri )
146         vy1 = vy2 - (dt*ypri)*( mpri + opri + ppri )
147         vz1 = vz2 - (dt*zpri)*( mpri + opri + ppri + qpri )
148
149
150     IF ( (x2 > 0.0) .AND. (x1 < 0.0) ) THEN
151         temp = -x2/(x1 - x2)
152
153         yc = y2 + temp*(y1-y2)
154         zc = z2 + temp*(z1-z2)
155
156         if ( (yc > 0.0) ) then
157             ycept(n) = yc
158             zcept(n) = zc
159             n = n + 1
160         end if
161     ELSE IF ( (x2 < 0.0) .AND. (x1 > 0.0) ) THEN

```

```

162         temp = -x2/(x1 - x2)
163
164         yc = y2 + temp*(y1-y2)
165         zc = z2 + temp*(z1-z2)
166
167         if ( (yc > 0.0) ) then
168             ycept(n) = yc
169             zcept(n) = zc
170             n = n + 1 !n counts how many times a match has been found
171         end if
172     END IF
173 end if
174
175 ! i = i + 1 ! i counts how many times that the entire code has been run
176 if ( n == 100 ) then ! once we have filled up the 100 element array...
177     repcount = repcount + 1 ! so now the array is filled and we write to file
178     write(*,*) repcount
179     do j = 1, 100
180
181         if ( (repcount /= rep) .and. (j /= 100) ) write(100,200) ycept(j),
182             zcept(j)
183     end do
184     n = 1 !start over
185 end if
186
187 if (repcount == rep) exit
188
189 END DO
190
191 close(100)
192 200 FORMAT (f12.3, 3x, f12.3)
193 END PROGRAM

```

BIBLIOGRAPHY

- [1] Herbert Goldstein. *Classical Mechanics*. Addison Wesley, 3rd edition, 2002.
- [2] Stephen T. Marion and Jerry B. Thornton. *Classical Dynamics of Particles and Systems*. Thomson Brooks-Cole, 5th edition, 2004.
- [3] V. I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer-Verlag, 2nd edition, 1989.
- [4] Wolfgang K. H. Panofsky. *Classical Electricity and Magnetism*. Dover Publications.
- [5] Andrew Zangwill. *Modern Electrodynamics*. Cambridge University Press, 2013.
- [6] John David Jackson. *Classical Electrodynamics*. John Wiley and Sons, 2nd edition, 1975.
- [7] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in FORTRAN*. Cambridge University Press, 2nd edition, 1992.