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

List of Figures

LIST OF SOURCE CODES

LIST OF SYMBOLS

$\delta(r'-a)$	Dirac Delta Function
δ_{ij}	Kronecker Delta
$\mathrm{d}M$	Arbitrarily small unit of mass
$\mathscr{L}$	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
$\overleftrightarrow{\mathbf{q}}$	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
$\mathrm{Tr}\,Q$	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

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

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

$$\mathbf{N} = \mathbf{r} \times \mathbf{F} = \mathbf{0}. \tag{2.2}$$

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

$$r = \frac{p}{1 + e\cos\varphi}. \tag{*}$$

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

$$p := \frac{L^2}{m^2 GM} \tag{2.3a}$$

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

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
$e = 0$	$E = -\frac{m^3G^2M^2}{2L^2}$	circular

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

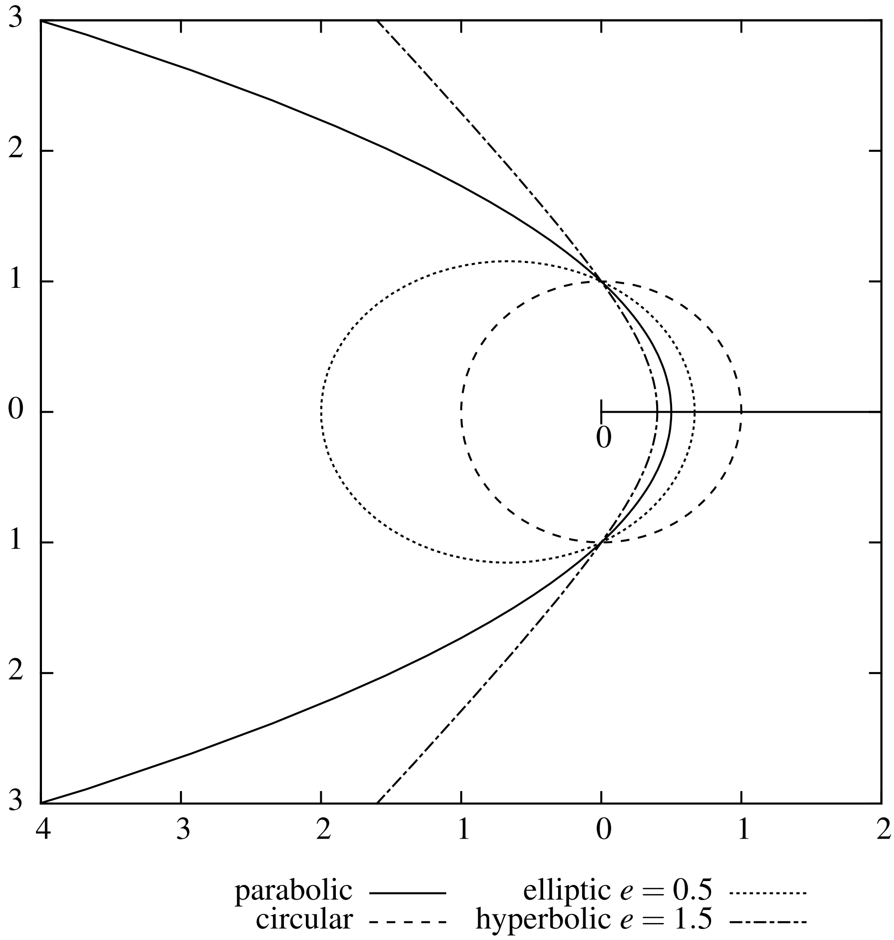


Figure 2.1: For a given energy E, the Kepler Problem produces orbits that are either bound or unbound.

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.

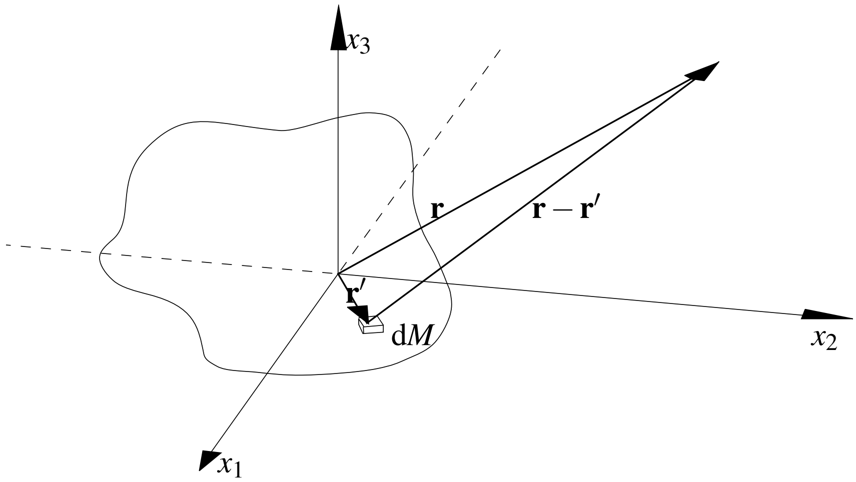


Figure 2.2: Given a source with an arbitrary structure, we can find the potential of the entire figure by finding the the potential due to each mass element dM.

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

$$\mathrm{d}\Phi = -\frac{G\ \mathrm{d}M}{|\mathbf{r}-\mathbf{r}'|}. \tag{2.4}$$

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

$$U = m\Phi. \tag{2.5}$$

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

$$\mathrm{d}M = \rho(\mathbf{r}')\ \mathrm{d}\tau', \tag{2.6}$$

so that we can integrate Eq. (2.4)

$$\Phi = -G \int\limits_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \, \mathrm{d}\tau'. \tag{2.7}$$

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

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

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

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

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

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

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

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

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

To obtain the second terin the right-hand side of Eq. (2.11), we perform the first derivative of ${}^{1}/_{|\mathbf{r}-\mathbf{r}'|}$, as follows:

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

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

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

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

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

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).

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $|\mathbf{r}| \to \infty$. 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

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

Integrating the second term of Eq. (2.24),

$$\Phi^{(1)} = -G\frac{1}{|\mathbf{r}|^3}\sum_{i=1}^{3}\int\limits_{\mathbb{R}^3}\rho(\mathbf{r}')\,x_i x_i'\,\mathrm{d}\tau' \tag{2.27}$$

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

$$= -\frac{G}{|\mathbf{r}|^3}\mathbf{r}\boldsymbol{\cdot}\mathbf{p}. \tag{2.29}$$

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

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

Defining the tensor Q_{ij}

$$Q_{ij} = \int\limits_{\mathbb{R}^3}\rho(\mathbf{r}')\,x_i'\,x_j'\,\mathrm{d}\tau', \tag{2.31}$$

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

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

We define the trace $\mathrm{Tr}\,Q$

$$\mathrm{Tr}\,Q := \sum_j Q_{jj} := \sum_{ij} Q_{ij}\delta_{ij} \tag{2.33}$$

With this, Eq. (2.32) becomes

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

We define the Quadrupole Moment Tensor q_{ij}

$$q_{ij} = 3\,Q_{ij}-\delta_{ij}\,\mathrm{Tr}\,Q \tag{2.35}$$

which simplifies Eq. (2.34) as

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

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

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

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 showng that it is traceless:

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

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

$$\mathrm{Tr}\, q = 3\,\mathrm{Tr}\, Q - 3\,\mathrm{Tr}\, Q = 0. \tag{2.39}$$

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

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

where

$$r_< = \min(r, r') \qquad (2.40a)$$

$$r_> = \max(r, r') \qquad (2.40b)$$

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

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

We define the quantity q_{lm} as

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

where the spherical monopole moment as 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

$$\nabla^2 \Phi = 0, \tag{2.43}$$

it is known in literature that the solution to this is

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

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 $r \to \infty$, 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

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

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

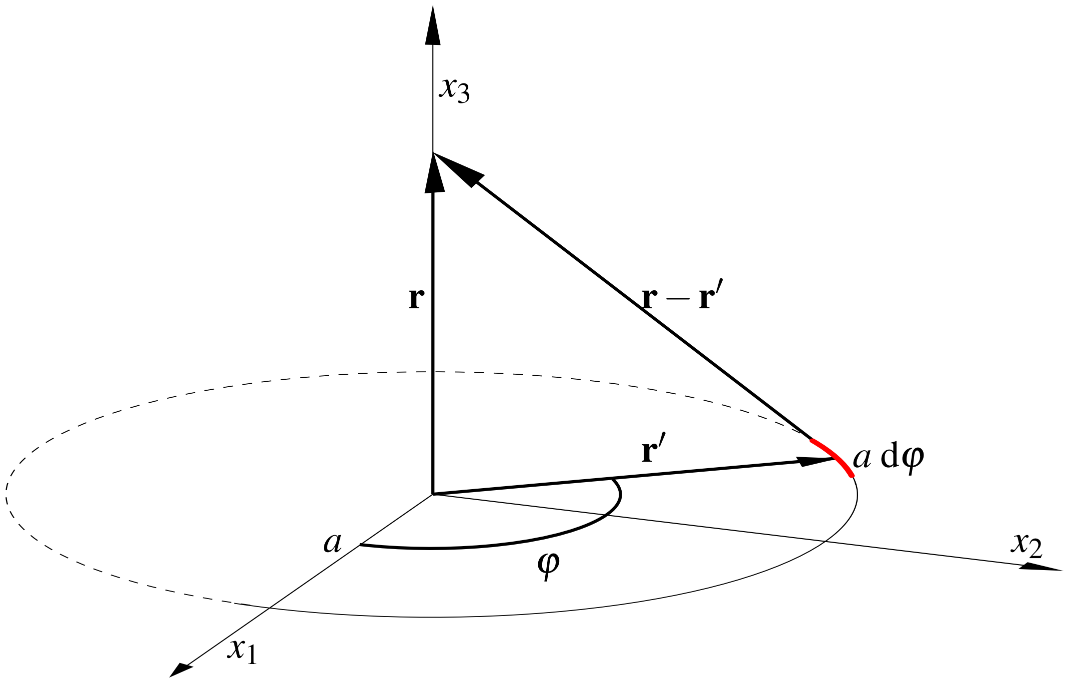


Figure 3.1: The field point is indicated by the vector $\mathbf{r}$, and a small mass dM is located using the vector $\mathbf{r}'$.

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

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

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

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

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,

$$\int_{a-\varepsilon}^{a+\varepsilon} f(x)\delta(x-a)\,\mathrm{d}x = f(a) \qquad \forall \varepsilon > 0, \tag{3.3}$$

we get

$$\lambda = \frac{M}{2\pi a^2} \tag{3.4}$$

and our expression for the density is now

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

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

$$\Phi^{(0)}_{\text{cart}} = -\frac{GM}{r}. \tag{3.6}$$

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) ,

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

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

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

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

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

$$\Phi^{(0)}_{\text{sphr}} = \begin{cases} -\dfrac{GM}{r} & r > a \\ -\dfrac{GM}{a} & r < a \end{cases} \tag{3.9}$$

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:

$$x_1 = r\sin\theta\cos\varphi \tag{3.10a}$$

$$x_2 = r\sin\theta\sin\varphi \tag{3.10b}$$

$$x_3 = r\cos\theta \tag{3.10c}$$

$$\mathrm{d}\tau = r^2\sin\theta\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}\varphi \tag{3.10d}$$

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

$$p_1 = \frac{M}{2\pi a^2}\int\limits_{r'=0}^{\infty}(r')^3\delta(r'-a)\,\mathrm{d}r'\int\limits_{\theta'=0}^{\pi}\sin^2\theta'\delta\left(\theta'-\frac{\pi}{2}\right)\mathrm{d}\theta'\int\limits_{\varphi'=0}^{2\pi}\cos\varphi'\,\mathrm{d}\varphi'$$

$$= \frac{M}{2\pi a^2}(a)^3\,(1)\,(0) = 0 \tag{3.11a}$$

$$p_2 = \frac{M}{2\pi a^2}\int\limits_{r'=0}^{\infty}\delta(r'-a)(r')^3\,\mathrm{d}r'\int\limits_{\theta'=0}^{\pi}\delta\left(\theta'-\frac{\pi}{2}\right)\sin^2\theta'\,\mathrm{d}\theta'\int\limits_{\varphi'=0}^{2\pi}\sin\varphi'\,\mathrm{d}\varphi'$$

$$= \frac{M}{2\pi a^2}(a)^3\,(1)\,(0) = 0 \tag{3.11b}$$

$$p_3 = \frac{M}{2\pi a^2}\int\limits_{r'=0}^{\infty}(r')^3\delta(r'-a)\,\mathrm{d}r'\int\limits_{\theta'=0}^{\pi}\delta\left(\theta'-\frac{\pi}{2}\right)\cos\theta'\sin\theta'\,\mathrm{d}\theta'\int\limits_{\varphi'=0}^{2\pi}\mathrm{d}\varphi'$$

$$= \frac{M}{2\pi a^2}(a)^3\,(0)\,(2\pi) = 0 \tag{3.11c}$$

From Eq. (3.11), we can see that

$$\Phi^{(1)}_{\text{cart}} = 0. \tag{3.12}$$

3.2.2 IN SPHERICAL COORDINATES

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

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

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

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,

$$\Phi^{(1)}_{\text{sphr}} = 0 \qquad \forall r \neq a \tag{3.14}$$

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),

$$Q_{ij} = \int_{\mathbb{R}^3} \rho(\mathbf{r}') x_i' x_j' \, \mathrm{d}\tau' \tag{2.31}$$

which, for this problem, becomes

$$Q_{ij} = \frac{M}{2\pi a^2} \int_{\mathbb{R}^3} \delta\left(\theta' - \frac{\pi}{2}\right) \delta(r' - a) x_i' x_j' \, \mathrm{d}\tau' \tag{3.15}$$

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

$$\begin{aligned}
Q_{11} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r'-a)(r')^4 \mathrm{d}r' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^3\theta' \, \mathrm{d}\theta' \int_{\varphi'=0}^{2\pi} \cos^2\varphi' \, \mathrm{d}\varphi' \\
&= \frac{M}{2\pi a^2} (a^4)\,(1)\,(\pi) = \frac{M a^2}{2} \qquad (3.16a) \\
Q_{22} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r'-a)(r')^4 \mathrm{d}r' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^3\theta' \, \mathrm{d}\theta' \int_{\varphi'=0}^{2\pi} \sin^2\varphi' \, \mathrm{d}\varphi' \\
&= \frac{M}{2\pi a^2} (a^4)\,(1)\,(\pi) = \frac{M a^2}{2} \qquad (3.16b) \\
Q_{33} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} \delta(r'-a)(r')^4 \mathrm{d}r' \int_{\theta'=0}^{\pi} \cos^2\theta' \sin\theta' \delta\left(\theta' - \frac{\pi}{2}\right) \mathrm{d}\theta' \int_{\varphi'=0}^{2\pi} \mathrm{d}\varphi' \\
&= \frac{M}{2\pi a^2} (a)\,(0)\,(2\pi) = 0 \qquad (3.16c) \\
Q_{13} &= \frac{M}{2\pi a^2} \int_{r'=0}^{\infty} (r')^4 \delta(r'-a) \, \mathrm{d}r' \int_{\theta'=0}^{\pi} \delta\left(\theta' - \frac{\pi}{2}\right) \sin^2\theta' \cos\theta' \, \mathrm{d}\theta' \int_{\varphi'=0}^{2\pi} \cos\varphi' \, \mathrm{d}\varphi' \\
&= \frac{M}{2\pi a^2} (a')^4\,(0)\,(0) = 0 \qquad (3.16d)
\end{aligned}$$

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

$$q_{11} = q_{22} = \frac{Ma^2}{2} \tag{3.17a}$$

$$q_{33} = -Ma^2 \tag{3.17b}$$

$$q_{12} = q_{13} = q_{23} = 0. \tag{3.17c}$$

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

$$\Phi^{(2)}_{\text{cart}} = -\frac{G}{2}\,\frac{\mathbf{r}\cdot\overset{\leftrightarrow}{\mathbf{q}}\cdot\mathbf{r}}{|\mathbf{r}|^5} \tag{3.18}$$

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

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

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

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

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

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

3.3.2 In Spherical Coordinates

For the quadrupole moment, we find q_{20}

$$q_{20} = \begin{cases} \dfrac{M}{2\pi a^2}\sqrt{\dfrac{5}{16\pi}} \int\limits_{r'=0}^{\infty} \delta(r'-a)(r')^4 \,\mathrm{d}r' \int\limits_{\theta'=0}^{\pi} \delta\left(\theta' - \dfrac{\pi}{2}\right)(3\cos^2\theta' - 1)\sin\theta' \,\mathrm{d}\theta' \int\limits_{\varphi'=0}^{2\pi} \mathrm{d}\varphi' & r > a \\ \dfrac{M}{2\pi a^2}\sqrt{\dfrac{5}{16\pi}} \int\limits_{r'=0}^{\infty} \delta(r'-a)\dfrac{1}{r'} \,\mathrm{d}r' \int\limits_{\theta'=0}^{\pi} \delta\left(\theta' - \dfrac{\pi}{2}\right)(3\cos^2\theta' - 1)\sin\theta' \,\mathrm{d}\theta' \int\limits_{\varphi'=0}^{2\pi} \mathrm{d}\varphi' & r < a \end{cases} \tag{3.24a}$$

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

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

$$q_{21} = \begin{cases} \dfrac{-M}{2\pi a^2}\sqrt{\dfrac{15}{4\pi}} \int\limits_{r'=0}^{\infty} \delta(r'-a)(r')^4 \,\mathrm{d}r' \int\limits_{\theta'=0}^{\pi} \delta\left(\theta' - \dfrac{\pi}{2}\right)\sin^2\theta'\cos\theta' \,\mathrm{d}\theta' \int\limits_{\varphi'=0}^{2\pi} e^{i\varphi} \,\mathrm{d}\varphi' & r > a \\ \dfrac{-M}{2\pi a^2}\sqrt{\dfrac{15}{4\pi}} \int\limits_{r'=0}^{\infty} \delta(r'-a)\dfrac{1}{r'} \,\mathrm{d}r' \int\limits_{\theta'=0}^{\pi} \delta\left(\theta' - \dfrac{\pi}{2}\right)\sin^2\theta'\cos\theta' \,\mathrm{d}\theta' \int\limits_{\varphi'=0}^{2\pi} e^{i\varphi} \,\mathrm{d}\varphi' & r < a \end{cases} \tag{3.25}$$

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

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

$$\Phi^{(2)}_{\mathrm{sphr}} = \begin{cases} \dfrac{1}{4}GMa^2\dfrac{1}{r^3}(3\cos^2\theta - 1) & r > a \\ \dfrac{1}{4}\dfrac{GM}{a^3}r^2(3\cos^2\theta - 1) & r < a \end{cases} \tag{3.27}$$

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

$$\mathbf{r} = a\cos\varphi\,\hat{\mathbf{i}} + a\sin\varphi\,\hat{\mathbf{j}} \tag{3.28a}$$

$$\mathbf{r}' = z\,\hat{\mathbf{k}} \tag{3.28b}$$

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

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

Integrating,

$$\Phi = -\frac{GM}{2\pi a^2}\frac{1}{\sqrt{z^2+a^2}}\int\limits_{r'=0}^{\infty}\delta(r'-a)(r')^2\,\mathrm{d}r'\int\limits_{\theta'=0}^{\pi}\delta\left(\theta'-\frac{\pi}{2}\right)\sin\theta'\,\mathrm{d}\theta'\int\limits_{\varphi'=0}^{2\pi}\mathrm{d}\varphi' \tag{3.30}$$

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

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

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

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

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

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

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

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

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

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:

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

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

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

However, the right-hand side of Eq. (3.35) does not appear to satisfy this condition. We rewrite the equation as follows:

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

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

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

and expand this in the first two terms:

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

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

$$\left(1+({}^{a}/_{z})^2\right)^{-1/2} = 1 - \frac{1}{2}a^2\frac{1}{z^2}. \tag{3.41}$$

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

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

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

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

In comparing the coefficients, we obtain the following:

$$B_0=-GM \tag{3.44a}$$

$$B_1=0 \tag{3.44b}$$

$$B_2=\frac{GMa^2}{2} \tag{3.44c}$$

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

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

We write out the Legendre Polynomials:

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

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

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

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

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

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

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

$$A_0+A_1z+A_2z^2=-\frac{GM}{a}+\frac{1}{2}\frac{GM}{a^3}z^2. \tag{3.49}$$

Comparing the coefficients, we have

$$A_0 = -\frac{GM}{a} \tag{3.50a}$$

$$A_1 = 0 \tag{3.50b}$$

$$A_2 = \frac{GM}{2a^3} \tag{3.50c}$$

Now we have a full expression for the potential

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

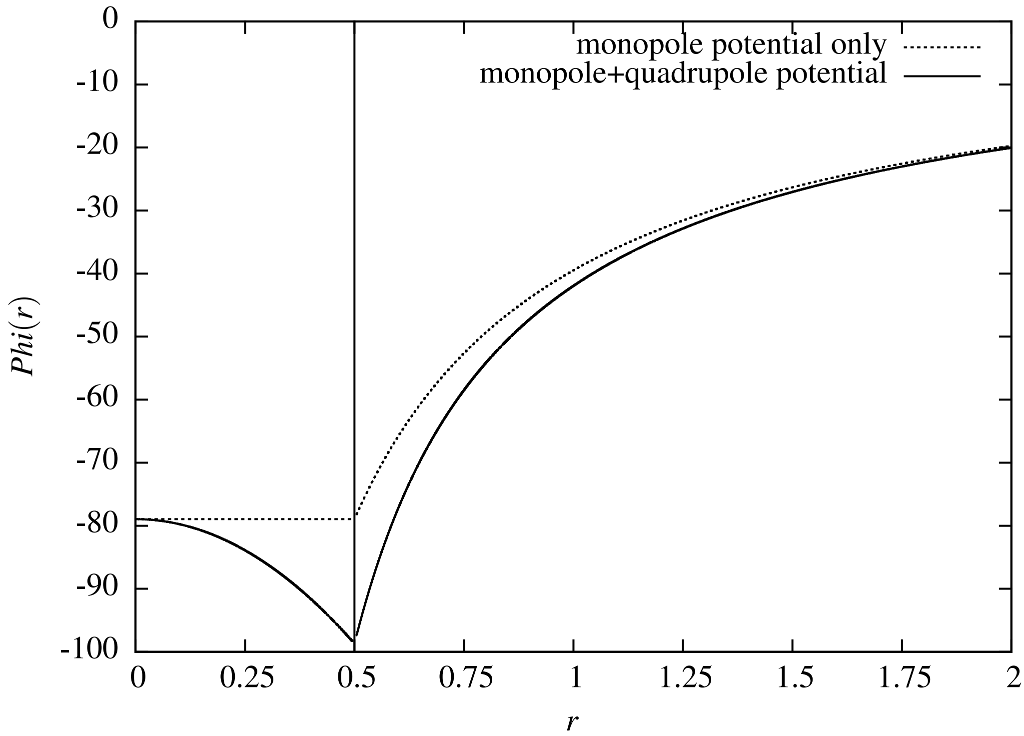


Figure 3.2: Comparison of the Monopole-only and Monopole with Quadrupole Potential of the ring on the equatorial plane

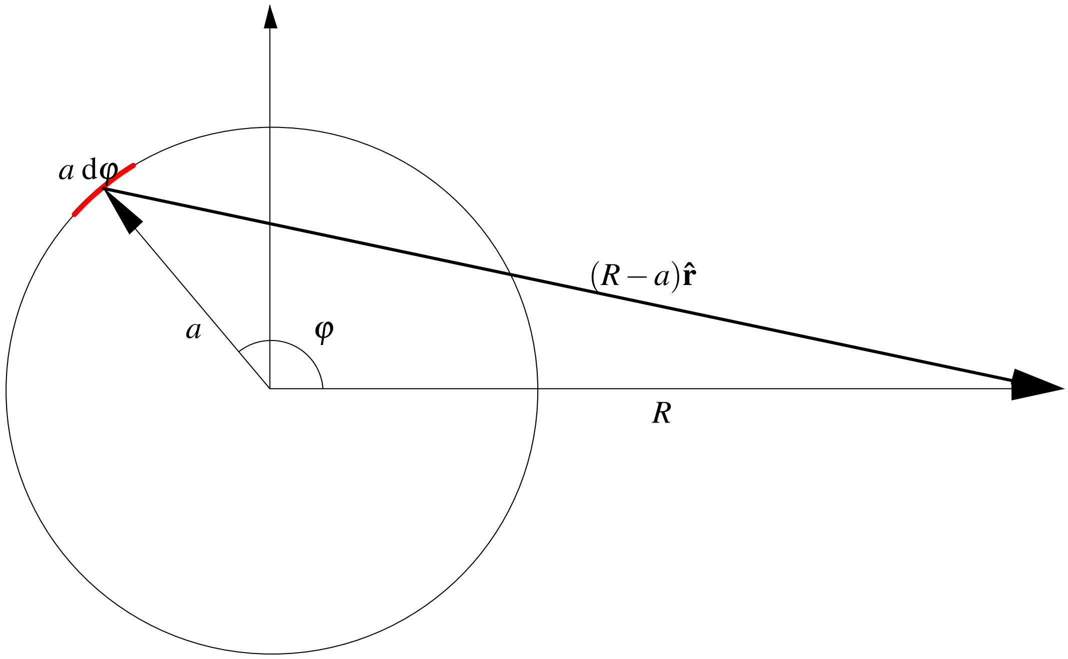


Figure 3.3: Origin to field point is R, radius of the ring is a.

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

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

and

$$\mathrm{d}M = \frac{M}{2\pi a^2}\delta(r'-a)(r')^2\,\mathrm{d}r'\,\mathrm{d}\varphi' \tag{3.53}$$

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

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

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

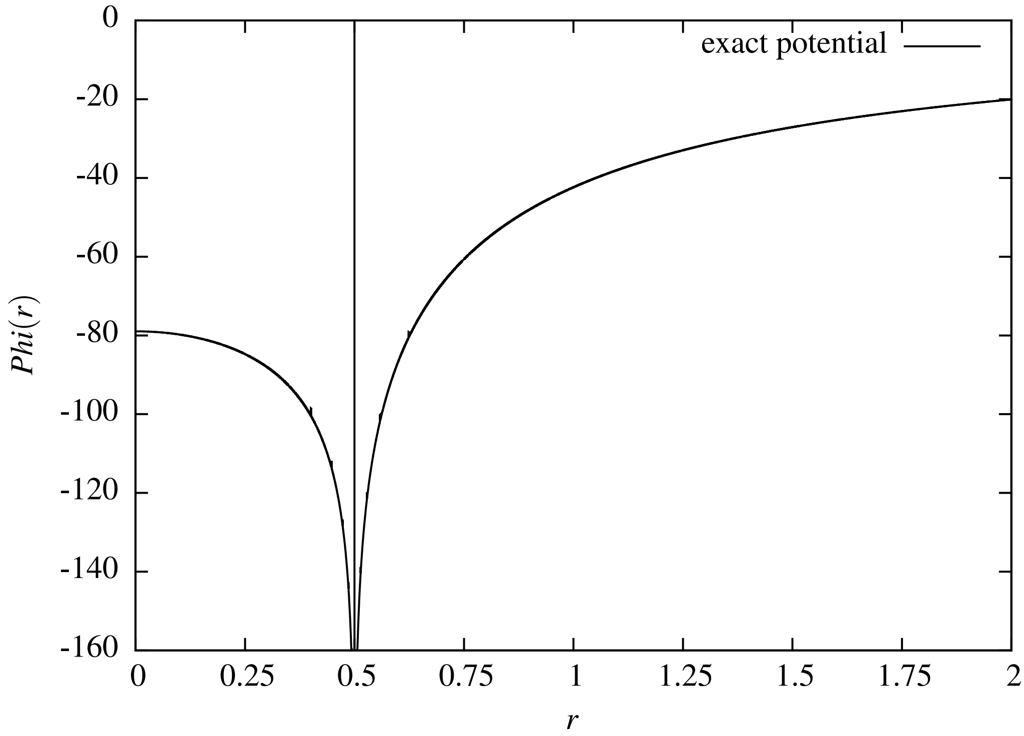


Figure 3.4: Exact Ring multipole potential on the plane

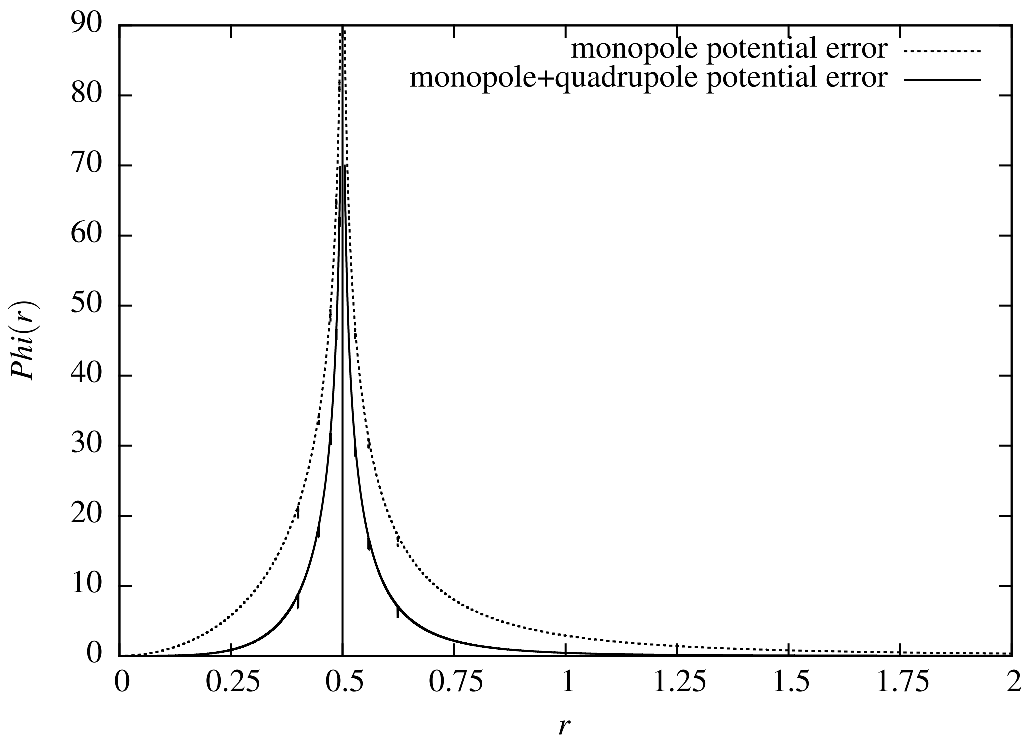


Figure 3.5: The difference of the ring multipole potential at each r and the exact potential obtain from numerical integration.

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

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

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* $\mathscr{L}$ for this system

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

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

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

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

Using the canonical equations, we get

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

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

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

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

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

Consider the time derivative of angular momentum

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t} = \dot{\mathbf{r}} \times \mathbf{p} + \mathbf{r} \times \dot{\mathbf{p}} \tag{4.6}$$

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

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

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

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

where

$$r = \sqrt{x^2+y^2+z^2} \tag{4.9}$$

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

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

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

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

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

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

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

$$GM = 4\pi^2. \tag{4.12}$$

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.

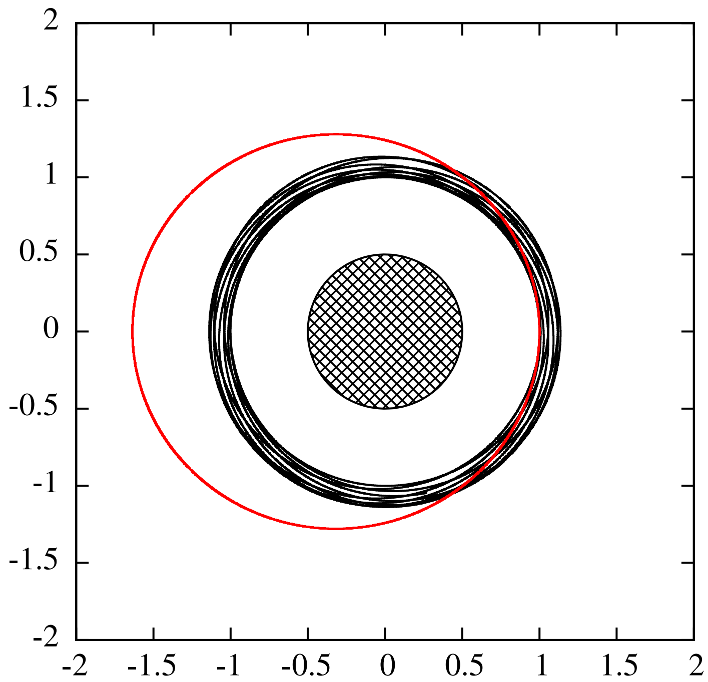


Figure 4.1: Near-attractor orbit comparison. Notice the difference between the monopole-only orbit and the orbit with the additional term.

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-

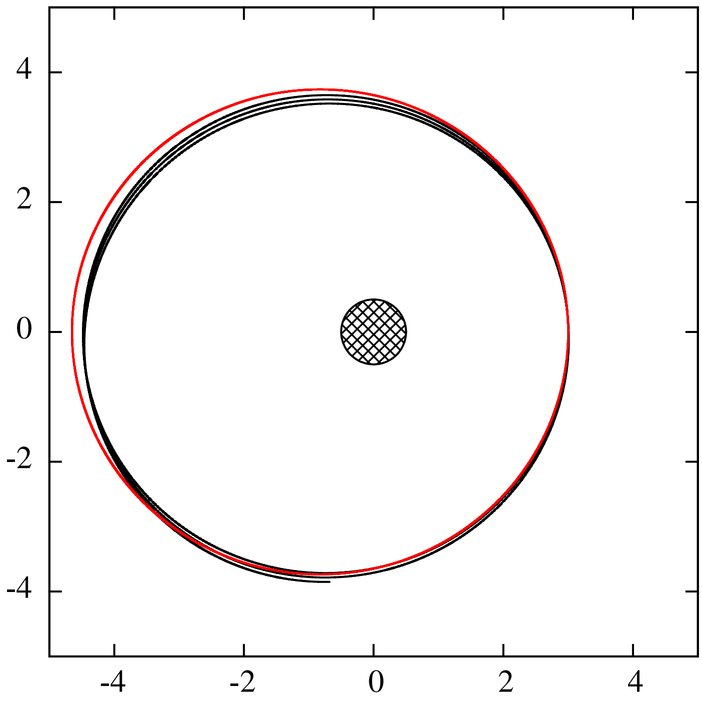


Figure 4.2: Far-attractor orbit comparison. Upon closer inspection, it is seen that the red orbit closes, as expected, and the black orbit does not.

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

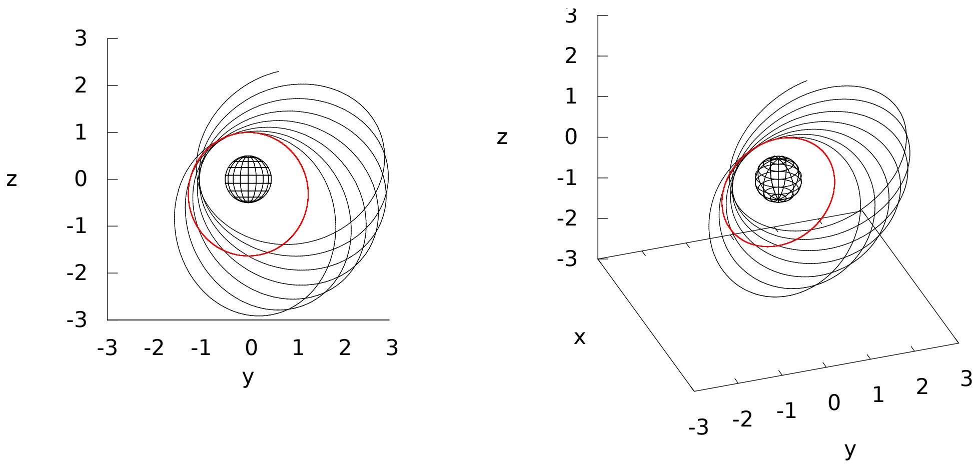


Figure 4.3: Given an initial condition where the initial position is at the z-axis, and the initial velocity lies only on the azimuthal direction, we get an orbit restricted in a polar plane.

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.

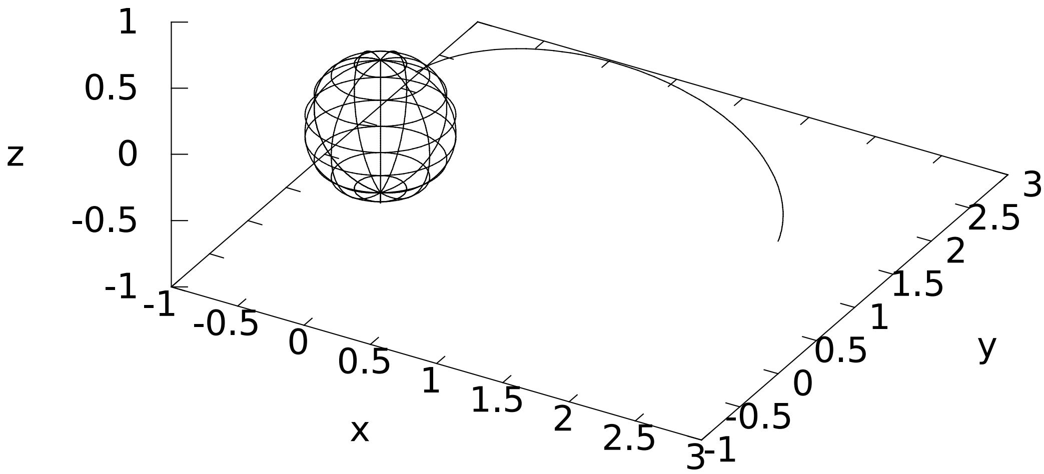


Figure 4.4: Given a low enough velocity, the orbiting particle will fall into the attractor.

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 condtions are $\mathbf{r}_0 = (1,0,0)$ and $\mathbf{v}_0 = (0,12,0)$.

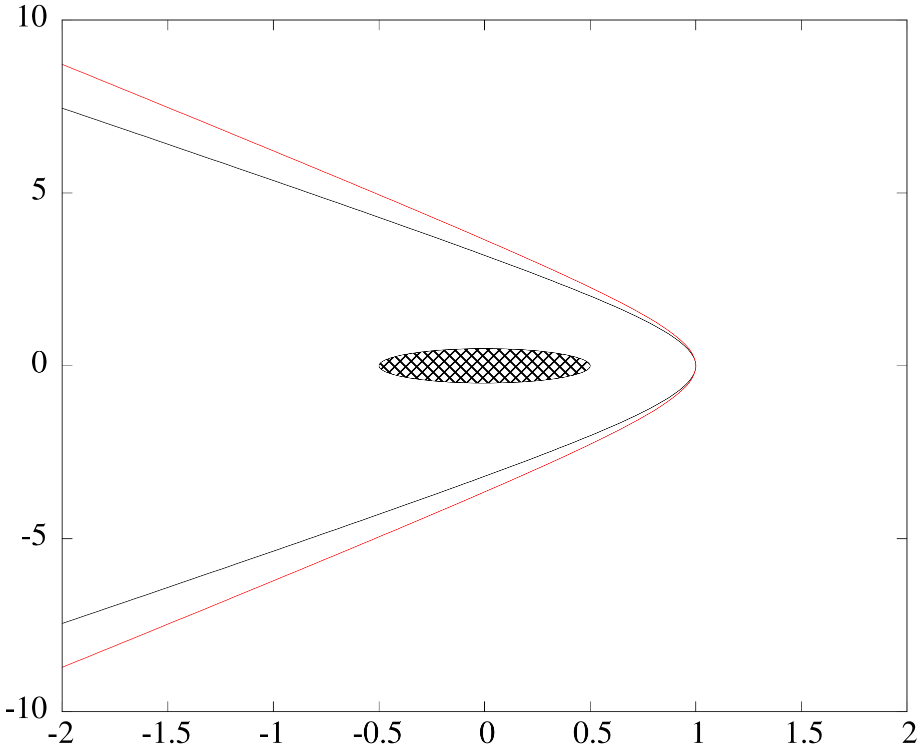


Figure 4.5: An unbound orbit that approaches the attractor at $r = 1$

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.

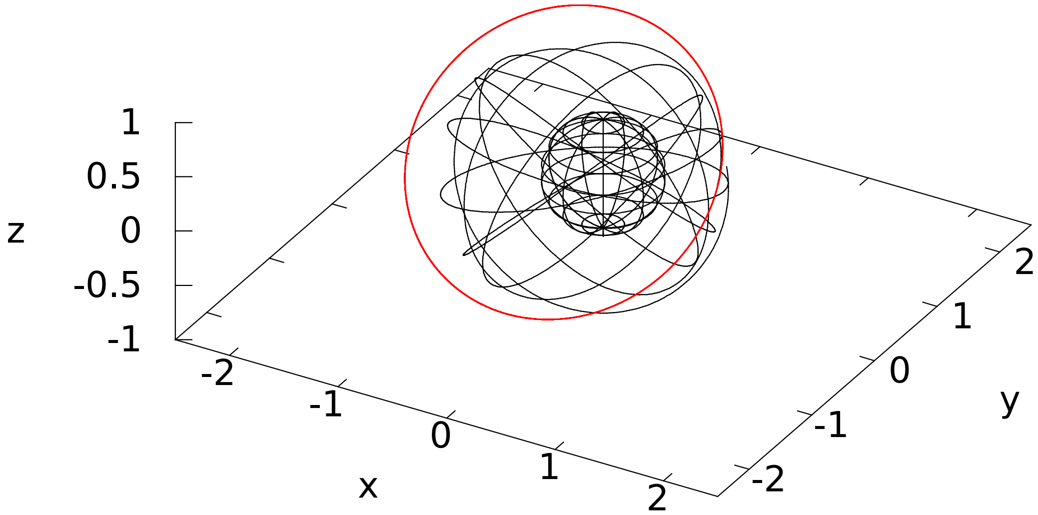


Figure 4.6: Given the proper initial conditions, we can get an orbit that is confined in a topological torus outside the attractor.

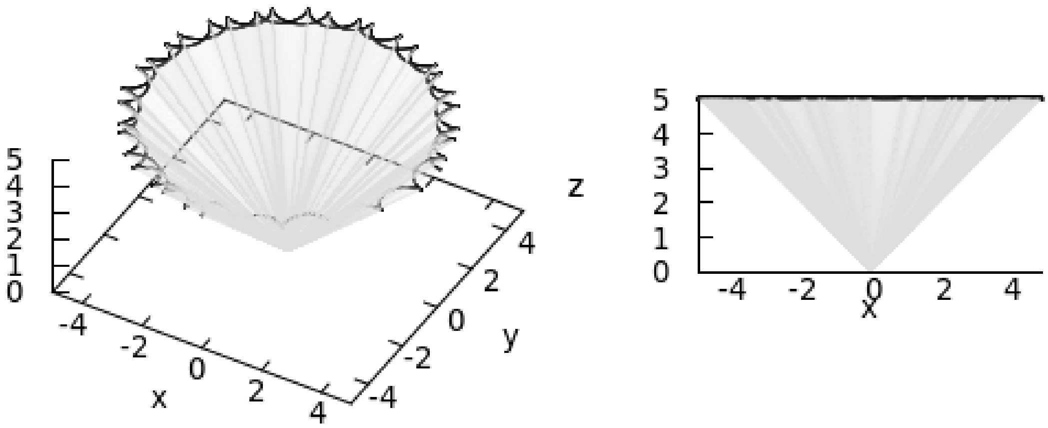
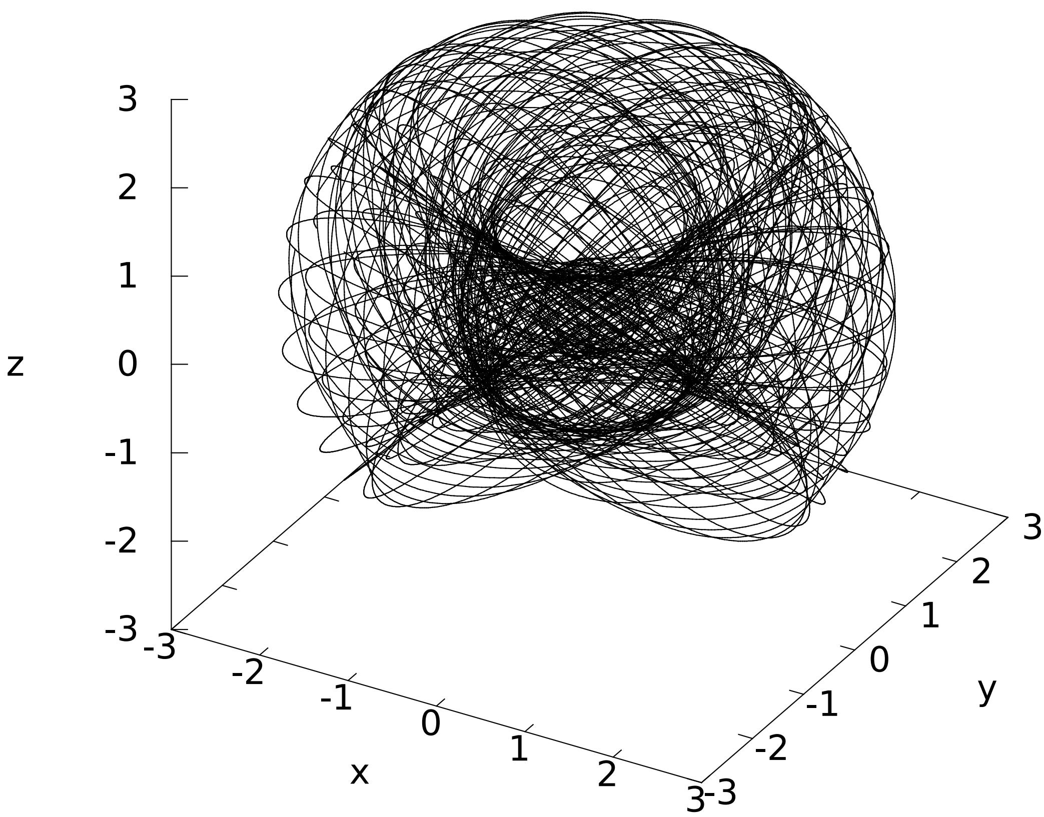


Figure 4.7: This is the cone generated by the angular momentum vectors. We see here that the cone of constant height indicates that the angular momentum along the z-direction is constant.

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 $\mathbf{v}_0 = (4,3,0)$.

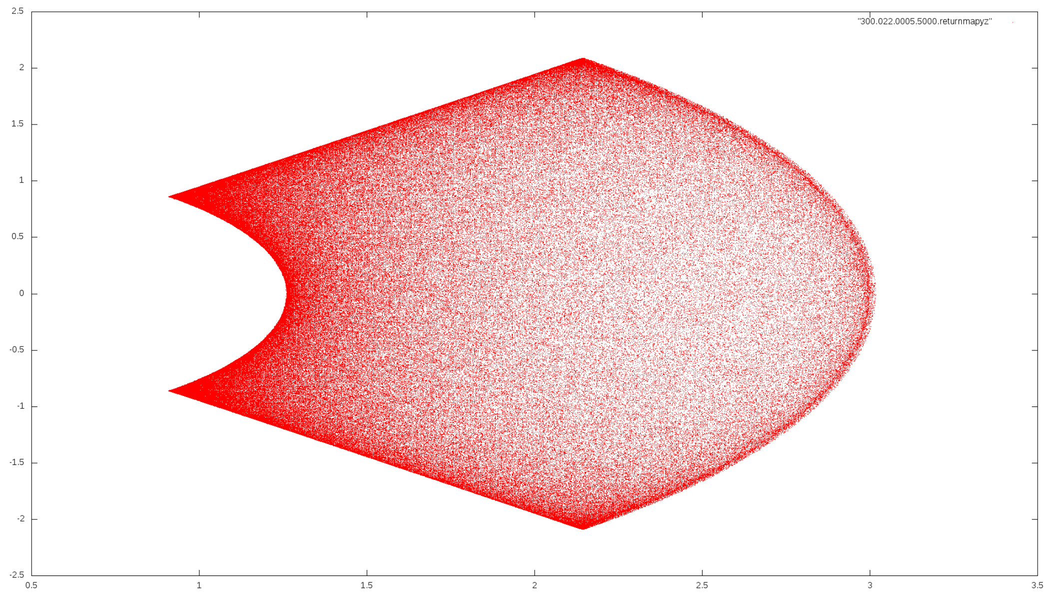


Figure 4.8: Return Map for $\mathbf{r}_0 = (3, 0, 0)$

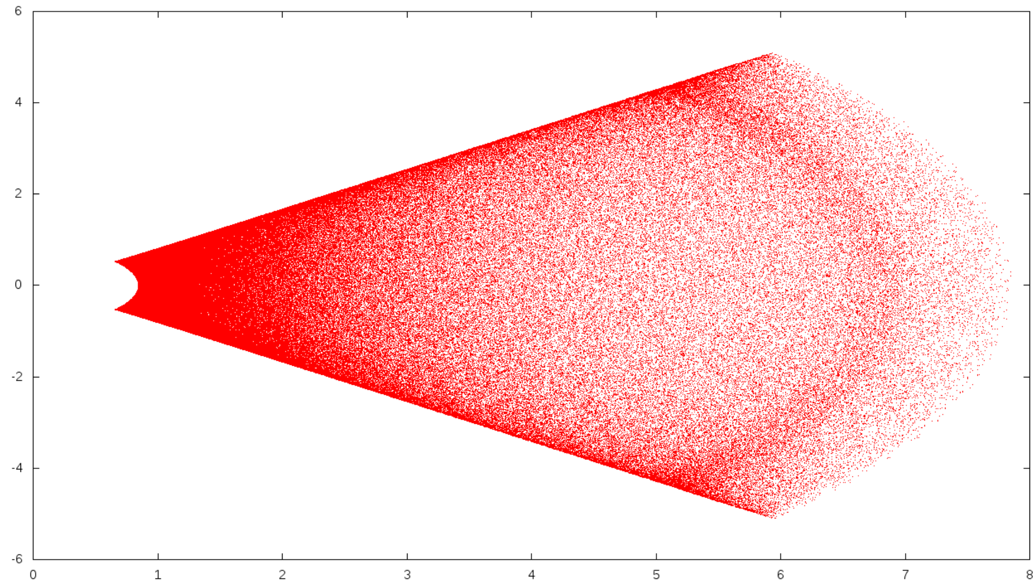


Figure 4.9: Return Map for $\mathbf{r}_0 = (2, 0, 1)$

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

$$Y_{l,-m}(\theta,\varphi) = (-1)^m Y^*_{l,m}(\theta,\varphi), \tag{A.1}$$

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

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

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

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

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

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

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

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

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

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

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

```
SUBROUTINE qromb(func,a,b,ss,r,radius)
INTEGER :: jmax, jmaxp, k, km
REAL :: a, b, func, ss, eps
REAL :: r, radius
EXTERNAL :: func
PARAMETER (eps=1.e-6, jmax=20, jmaxp=jmax+1, k=5, km=k-1)
INTEGER :: j
REAL :: dss, h(jmaxp), s(jmaxp)
h(1) = 1
DO j=1, jmax
  CALL trapzd(func,a,b,s(j),j,r,radius)
  IF (j .GE. k) THEN
    CALL polint(h(j-km),s(j-km),k,0.,ss,dss)
    IF (ABS(dss).LE.eps*ABS(ss)) RETURN
  END IF
  s(j+1) = s(j)
  h(j+1) = 0.25*h(j)
END DO
STOP !'too many steps in qromb'
END SUBROUTINE qromb
```

Source Code B.2: Trapezoid Rule

```
SUBROUTINE trapzd(func,a,b,s,n,r,radius)
INTEGER :: n
REAL :: a,b,s,func
REAL :: r, radius
EXTERNAL func
INTEGER :: it,j
REAL :: del,sum,tnm,x
IF (n.EQ.1) THEN
  s=0.5*(b-a)*(func(r,a,radius)+func(r,b,radius))
ELSE
  it = 2**(n-2)
  tnm=it
  del=(b-a)/tnm
  x=a+0.5*del
  sum = 0.0
  DO j=1, it
    sum=sum+func(r,x,radius)
    x=x+del
  END DO
  s=0.5*(s+(b-a)*sum/tnm)
ENDIF
RETURN
END SUBROUTINE trapzd
```

Source Code B.3: Polynomial Interpolation

```
SUBROUTINE polint(xa,ya,n,x,y,dy)
INTEGER n,nmax
REAL dy,x,y,xa(n),ya(n)
PARAMETER (nmax=10)
INTEGER i,m,ns
REAL den,dif,dift,ho,hp,w,c(nmax),d(nmax)
ns=1
dif=ABS(x-xa(1))
DO i=1,n
  dift=ABS(x-xa(i))
  IF (dift.LT.dif) THEN
    ns=i
    dif=dift
  ENDIF
  c(i)=ya(i)
  d(i)=ya(i)
END DO
y=ya(ns)
ns=ns-1
DO m=1,n-1
  DO i=1,n-m
    ho=xa(i)-x
    hp=xa(i+m)-x
    w=c(i+1)-d(i)
    den=ho-hp
    IF(den.EQ.0) STOP !'failure in polint'
    den=w/den
```

```
      d(i)=hp*den
      c(i)=ho*den
    END DO
    IF (2*ns.LT.n-m) THEN
      dy=c(ns+1)
    ELSE
      dy=d(ns)
      ns=ns-1
    END IF
    y=y+dy
END DO
RETURN
END SUBROUTINE polint
```

Source Code B.4: Exact Potential Integrand

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

Source Code B.5: Multipole Potential Function

```
FUNCTION multipolev(r,th,a,mo)
IMPLICIT NONE
REAL :: r,th,a,multipolev
integer :: mo
REAL, PARAMETER :: PI = 3.14159265
REAL, PARAMETER :: GM = 39.4784176
if (mo == 0) then
  IF (r>a) THEN
    multipolev = -GM/r
  ELSE IF (r<a) THEN
    multipolev = -GM/a
  ELSE
     WRITE(*,*) "r!=a"
     STOP
  END IF
else if (mo == 1) then
  IF (r>a) THEN
    multipolev = -GM/r
  ELSE IF (r<a) THEN
    multipolev = -GM/a
  ELSE
     WRITE(*,*) "r!=a"
     STOP
  END IF
else if (mo == 2) then
  IF (r>a) THEN
    multipolev = -GM/r + ((GM*a**2)/(4.0*r**3))*((3*(COS(th)**2))-1)
  ELSE IF (r<a) THEN
    multipolev = -GM/a + ((GM*r**2)/(4.0*a**3))*((3*(COS(th)**2))-1)
  ELSE
```

```
    WRITE(*,*) "r!=a"
    STOP
  END IF
else
  write(*,*) "moment unavailable"
end if

RETURN
END FUNCTION multipolev
```

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

```
PROGRAM main
IMPLICIT NONE
REAL, PARAMETER :: PI = 3.14159265
REAL, PARAMETER :: GM = 39.4784176
REAL, PARAMETER :: m = 3e-3
REAL, DIMENSION(:), ALLOCATABLE :: r
REAL :: rstep
REAL :: phi, res
INTEGER :: n,i
REAL :: r0,rf,radius
REAL, EXTERNAL :: func, multipolev
character(len=20) :: filename

! Ask the user for input
WRITE(*,*) "This program calculates the exact potential"
WRITE(*,*) "of the ring on the equatorial plane."
WRITE(*,*) "radius"
READ(*,*) radius
WRITE(*,*) "Set the range of the R"
READ(*,*) r0, rf
WRITE(*,*) "Step size of R"
READ(*,*) rstep
WRITE(*,*) "Filename of output"
READ(*,*) filename

n = INT( (rf-r0)/rstep )
ALLOCATE(r(n))

r(1) = r0
DO i = 1,n-1
   r(i+1) = r(i) + rstep
END DO

! This program generates five files:
! filename.exact where the program will create two-columns
!    of data, the first for r and the second for the exact potential
OPEN ( unit = 101, file = trim(filename)//".exact", status = ’replace’, &
    & action = ’write’)
! filename.mon, used to generate a r vs monopole potential at r
OPEN ( unit = 103, file = trim(filename)//".mon", status = ’replace’, &
    & action = ’write’)
! filename.monquad, used to generate a r vs monopole and quadrupole potential
OPEN ( unit = 104, file = trim(filename)//".monquad", status = ’replace’, &
```

```fortran
    & action = 'write')
! filename.monerr, used to generate a r vs the error assuming that
!   we only use the monopole potential only
OPEN ( unit = 105, file = trim(filename)//".monerr", status = 'replace', &
    & action = 'write')
! filename.mqerr, used to generate a r vs the error assuming that
!   we only use the monopole and quadrupole potential
OPEN ( unit = 106, file = trim(filename)//".mqerr", status = 'replace', &
    & action = 'write')

DO i=1,n

! given a certain r(i), the subroutine qromb integrates the potential
!  and stores it in the variable res. this is then written to file
  CALL qromb(func,0.0,2*PI,res,r(i),radius)
  WRITE(101,200) r(i), -(res*GM)/(2*PI)

  WRITE(103,200) r(i), multipolev(r(i),PI/2,radius, 0)
  WRITE(104,200) r(i), multipolev(r(i),PI/2,radius, 2)
  WRITE(105,200) r(i), &
     & ABS(-(res*GM)/(2*PI)) - ABS(multipolev(r(i),PI/2,radius, 0))
  WRITE(106,200) r(i), &
     & ABS(-(res*GM)/(2*PI)) - ABS(multipolev(r(i),PI/2,radius, 2))
END DO

200 FORMAT (f12.9, 3x, f12.3)
END PROGRAM
```

B.2 Ring Orbit Simulation

Source Code B.7: RK2 Subroutine for Orbit Solution

```
SUBROUTINE planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, error)
!------------------------------------------------------------------
!  SUBROUTINE planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, error)
!     input: integer n (number of elements in the following arrays)
!           number GM (product of attractor mass and grav. const.)
!           array x (of n elements)
!           array y (of n elements)
!           array z (of n elements)
!           array vx (of n elements)
!           array vy (of n elements)
!           array vz (of n elements)
!           array t (of n elements)
!     output: arrays x y z vx vy vz t
!           integer error (returns either zero or one)
!           integer num (if n!=num, this indicates that the planet stopped)
!     note: this program allows numbers up to 10 decimal places
!        : all arrays must already have a set initial value at n=1
!        : as outlined in Numerical Methods, page 704,
!        expanded to accomodate three independent variables.
!------------------------------------------------------------------

IMPLICIT NONE
INTEGER, PARAMETER :: dbl = selected_real_kind(p=10)
INTEGER, INTENT(in) :: n
INTEGER, INTENT(out) :: error, num
REAL(dbl), INTENT(in) :: radius, GM, dt
REAL(dbl), INTENT(out), DIMENSION(n) :: t
REAL(dbl), INTENT(out), DIMENSION(n) :: x, y, z
REAL(dbl), INTENT(out), DIMENSION(n) :: vx, vy, vz
INTEGER :: i, j
REAL(dbl) :: r, rpri, p, ppri, o, opri, q, qpri, m, mpri

REAL(dbl) :: xpri, ypri, zpri
REAL(dbl) :: vxpri, vypri, vzpri
REAL(dbl) :: vmag

num = 0

DO i = 1, n-1
  t(i+1) = t(i) + dt

  r = SQRT(x(i)**2 + y(i)**2 + z(i)**2)
  xpri = x(i) + ( ( dt*vx(i) ) / 2.0 )
  ypri = y(i) + ( ( dt*vy(i) ) / 2.0 )
  zpri = z(i) + ( ( dt*vz(i) ) / 2.0 )

  m = GM/(r**3)
  o = -3.75*( (GM*radius**2)/(r**7) )*(z(i)**2)
  p = 0.75*( (GM*radius**2)/(r**5) )
```

```
  q = 1.50*( (GM*radius**2)/(r**5) )
  vxpri = vx(i) - (dt*x(i)/2.0)*( m + o + p )
  vypri = vy(i) - (dt*y(i)/2.0)*( m + o + p )
  vzpri = vz(i) - (dt*z(i)/2.0)*( m + o + p + q )

  rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
  x(i+1) = x(i) + (dt*vxpri)
  y(i+1) = y(i) + (dt*vypri)
  z(i+1) = z(i) + (dt*vzpri)

  mpri = GM/(rpri**3)
  opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
  ppri = 0.75*( (GM*radius**2)/(rpri**5) )
  qpri = 1.50*( (GM*radius**2)/(rpri**5) )
  vx(i+1) = vx(i) - (dt*xpri)*( mpri + opri + ppri )
  vy(i+1) = vy(i) - (dt*ypri)*( mpri + opri + ppri )
  vz(i+1) = vz(i) - (dt*zpri)*( mpri + opri + ppri + qpri )

  IF ( r < radius ) THEN
    num = i
    error = 1                                      !particle stopped

    DO j = num, n-1
      x(j+1) = x(i)
      y(j+1) = y(i)
      z(j+1) = z(i)
      vx(j+1) = 0.0
      vy(j+1) = 0.0
      vz(j+1) = 0.0
    END DO

    EXIT
  ELSE
    num = i
    IF (num == n-1) THEN
      error = 0
    END IF
  END IF

END DO

RETURN
END SUBROUTINE planetrk2
```

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

```
PROGRAM planeorbit
!---------------------------------------------------------------
!  PROGRAM planeorbit
!    generates the numerical solution to the orbit
!    given the monopole and quadrupole potential
!---------------------------------------------------------------
IMPLICIT NONE

INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
```

```
INTEGER :: n, i, num
INTEGER :: outputerr, allocerr, rk2error
CHARACTER(len=20) :: filename

REAL(dbl), PARAMETER :: PI = 3.14159265359
REAL(dbl) :: GM

REAL(dbl), DIMENSION(:), ALLOCATABLE :: t
REAL(dbl), DIMENSION(:), ALLOCATABLE :: x, y, z
REAL(dbl), DIMENSION(:), ALLOCATABLE :: vx, vy, vz

REAL(dbl) :: dt, tf, r, vmag
REAL(dbl) :: x0, y0, z0, vx0, vy0, vz0
REAL(dbl) :: radius

REAL(dbl) :: E, KE, PH
!    energies

REAL(dbl), PARAMETER :: mass = 1.65956463e-7
!    uses the ratio of mercury and the sun as the mass of the attractor
!    this is only relevant in the computation of the
!    energies of the problem

WRITE(*,*) "This program generates the actual trajectory of a planet "
WRITE(*,*) "orbiting about a ring of mass on the equatorial plane"
WRITE(*,*) "using the multipole approximation and RK2."
WRITE(*,*) "This uses a unit system scaled such that G*M_sun = 4pi^2
    (default)"
WRITE(*,*)
WRITE(*,*) "Input the following information:"
WRITE(*,*) "initial position (cartesian coordinates)"
READ(*,*) x0, y0, z0
WRITE(*,*) "Initial velocity"
READ(*,*) vx0, vy0, vz0
WRITE(*,*) "Attractor radius"
READ(*,*) radius
WRITE(*,*) "Final time, timestep"
READ(*,*) tf, dt
WRITE(*,*) "(debug) value for GM. enter 0 for default"
READ(*,*) GM
WRITE(*,*) "Filename for output"
READ(*,*) filename

IF ( GM == 0 ) THEN
  GM = 4.0*(PI**2)
END IF

n = ABS(INT(tf/dt))
WRITE(*,*) "Plotting ", n, "points..."

ALLOCATE( t(n) , x(n) , y(n) , z(n) , vx(n) , vy(n) , vz(n) , stat=allocerr)
IF ( allocerr /= 0 ) THEN
  WRITE(*,*) "Error in allocation. Exiting..."
  STOP
END IF

```

```
!     set initial conditions
t(1) = 0.0
x(1) = x0
y(1) = y0
z(1) = z0
vx(1) = vx0
vy(1) = vy0
vz(1) = vz0

OPEN ( unit = 100, file = TRIM(filename)//".gp", status = 'replace', &
    & action = 'write', iostat=outputerr)
IF ( outputerr /= 0 ) THEN
  WRITE(*,*) "Error in", TRIM(filename)//".gp", ". Exiting..."
  STOP
ELSE
  WRITE(100,*) "# echo input data"
  WRITE(100,*) "# Initial Position: ", x0, y0, z0
  WRITE(100,*) "# Initial velocity: ", vx0, vy0, vz0

  r = SQRT(x0**2 + y0**2 + z0**2)
  vmag = SQRT(vx0**2 + vy0**2 + vz0**2)
  WRITE(100,*) "# Initial velocity magnitude: ", vmag
  KE = 0.5*mass*vmag**2
  PH = -GM/r + ((0.75*GM*(radius**2)*(z0**2))/r**5) -
      ((0.25*GM*radius**2)/r**3)
  E = KE + mass*PH
  WRITE(100,*) "# Initial kinetic energy: ", KE
  WRITE(100,*) "# Initial potential energy: ", mass*PH
  WRITE(100,*) "# Initial total energy: ", E
  WRITE(100,*) "# Attractor Radius"
  WRITE(100,*) "r = ", radius
  WRITE(100,*) "# GM =", GM

  IF ( dt > 0 ) THEN
   WRITE(100,*) "# Initial time: ", 0.0
   WRITE(100,*) "# Final time: ", tf
  ELSE IF ( dt < 0 ) THEN
   WRITE(100,*) "# Initial time: ", -tf
   WRITE(100,*) "# Final time: ", 0.0
  END IF

  WRITE(100,*) "# Timestep: ", ABS(dt)
END IF

IF ( dt > 0 ) THEN
 CALL planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, rk2error)
ELSE IF ( dt < 0 ) THEN
 CALL planetrk2(n, num, radius, GM, dt, t, x, y, z, vx, vy, vz, rk2error)

 CALL reversearray(x, n)
 CALL reversearray(y, n)
 CALL reversearray(z, n)
 CALL reversearray(vx, n)
 CALL reversearray(vy, n)
 CALL reversearray(vz, n)
END IF
```

```
WRITE(100,*) "# Points plotted: "
WRITE(100,*) "n = ", num

WRITE(*,*) "rk2error", rk2error
WRITE(*,*)

WRITE(100,*) "plotfilename = ", "'"//TRIM(filename)//".plot'"
WRITE(100,*) "lvecfilename = ", "'"//TRIM(filename)//".lvec'"
WRITE(100,*) "outtmpl = ", "'plot/"//TRIM(filename)//"%07d.png'"

WRITE(100,*) "# > bounding box"
WRITE(100,*) "# x", MINVAL(x), MAXVAL(x)
WRITE(100,*) "# y", MINVAL(y), MAXVAL(y)
WRITE(100,*) "# z", MINVAL(z), MAXVAL(z)
WRITE(100,*) "set xrange [", MINVAL(x)-0.5, ":", MAXVAL(x)+0.5, "]"
WRITE(100,*) "set yrange [", MINVAL(y)-0.5, ":", MAXVAL(y)+0.5, "]"
WRITE(100,*) "set zrange [", MINVAL(z)-0.5, ":", MAXVAL(z)+0.5, "]"

WRITE(100,*) 'set label "attractor radius:', radius, &
  & '" font "sans,8" right at screen 1, character 6'
WRITE(100,203) 'set label "initial position:', x0, y0, z0, &
  & '" font "sans,8" right at screen 1, character 5'
WRITE(100,203) 'set label "initial velocity:', vx0, vy0, vz0, &
  & '" font "sans,8" right at screen 1, character 4'

CALL printxyzarray(x, y, z, n, filename)
CALL printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)

203 FORMAT (a, f6.2, f6.2, f6.2, a)
DEALLOCATE(t, x, y, z, vx, vy, vz)
CLOSE(100)
END PROGRAM
```

Source Code B.9: Print XYZ plot to file

```
SUBROUTINE printxyzarray(x, y, z, n, filename)
!-----------------------------------------------------------------
!  SUBROUTINE printxyzarray(x, y, z, n, filename)
!    input: integer n (number of elements in the following arrays)
!           array x (of n elements)
!           array y (of n elements)
!           array z (of n elements)
!           string (20 chars max) filename
!    output: file "filename.plot"
!              three columns: x y z
!    note: this program allows numbers up to 10 decimal places
!-----------------------------------------------------------------

IMPLICIT NONE
INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
REAL(dbl), DIMENSION(n), INTENT(in) :: x, y, z
INTEGER, INTENT(in) :: n
CHARACTER(len=20), INTENT(in) :: filename
```

```
INTEGER :: outputerr, i

OPEN ( unit = 111, file = TRIM(filename)//".plot", status = 'replace', &
    & action = 'write', iostat=outputerr )

IF ( outputerr == 0 ) THEN
  DO i = 1, n
    WRITE(111,201) x(i), y(i), z(i)
  END DO
END IF

CLOSE(111)
201 FORMAT (f30.10, 3x, f30.10, 3x, f30.10)
RETURN
END SUBROUTINE printxyzarray
```

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

```
SUBROUTINE printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)
!----------------------------------------------------------------------
!  SUBROUTINE printlmag(x, vx, y, vy, z, vz, t, n, mass, filename)
!    input: integer n (number of elements in the following arrays)
!           array x (of n elements)
!           array y (of n elements)
!           array z (of n elements)
!           array vx (of n elements)
!           array vy (of n elements)
!           array vz (of n elements)
!           array t (of n elements)
!           real mass (for debugging purposes)
!           string (20 chars max) filename
!    output : file "filename.lmag"
!              two columns: t lmag
!    output : file "filename.lvec"
!              three columns: lx ly lz
!    note: this program allows numbers up to 10 decimal places
!              to generate the surface of angular momentum vectors,
!              run "splot 'filename.plot' with vectors nohead" in gnuplot
!          this program prints out the angular momentum per unit mass
!----------------------------------------------------------------------

IMPLICIT NONE
INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
REAL(dbl), DIMENSION(n), INTENT(in) :: t
REAL(dbl), DIMENSION(n), INTENT(in) :: x,y,z
REAL(dbl), DIMENSION(n), INTENT(in) :: vx,vy,vz
REAL(dbl), INTENT(in) :: mass
INTEGER, INTENT(in) :: n
CHARACTER(len=20), INTENT(in) :: filename
INTEGER :: outputerr, i
REAL(dbl) :: lx, ly, lz, lmag

OPEN ( unit = 107, file = TRIM(filename)//".lmag", status = 'replace', &
    & action = 'write', iostat=outputerr)
```

```
OPEN ( unit = 108, file = TRIM(filename)//".lvec", status = 'replace', &
    & action = 'write', iostat=outputerr)

IF ( outputerr == 0 ) THEN
  DO i = 1, n
    lx = ((y(i)*vz(i)) - (vy(i)*z(i)))
    ly = ((z(i)*vx(i)) - (x(i)*vz(i)))
    lz = ((x(i)*vy(i)) - (y(i)*vx(i)))
    lmag = SQRT(lx**2 + ly**2 + lz**2)
    WRITE(107,200) t(i), lmag
    WRITE(108,202) 0.0, 0.0, 0.0, lx, ly, lz
  END DO
END IF

CLOSE(107)
CLOSE(108)
200 FORMAT (f30.10, 3x, f30.10)
202 FORMAT (f30.10, 3x, f30.10, 3x, f30.10, 3x, f30.10, 3x, f30.10, 3x,
   f30.10)
END SUBROUTINE printlmag
```

Source Code B.11: Reverse Input Array

```
SUBROUTINE reversearray(array, n)
!----------------------------------------------------------------------
!  SUBROUTINE reversearray(array, n)
!    description: this subroutine takes the input array and
!                 outputs the same array in reverse order
!    input: integer n (number of elements in the following array)
!           array array (of n elements)
!    output: array array
!    note: this program allows numbers up to 10 decimal places
!----------------------------------------------------------------------

IMPLICIT NONE

INTEGER, PARAMETER :: dbl = SELECTED_REAL_KIND(p=10)
INTEGER, INTENT(in) :: n
REAL(dbl), INTENT(out), DIMENSION(n) :: array
REAL(dbl), DIMENSION(n) :: temp
INTEGER :: i
REAL(dbl) :: othertemp

DO i = 1, n
 temp(i) = array(n-i+1)
END DO

DO i = 1, n
 array(i) = temp(i)
END DO

RETURN
END SUBROUTINE reversearray
```

B.3 Return Map

Source Code B.12: Return Map

```
PROGRAM returnmap
!----------------------------------------------------------------------
!  PROGRAM returnmap
!     description: used to generate the return map on the yz plane
!                  given a set of initial conditions
!     note: this program does not allow modifiable attractor mass
!----------------------------------------------------------------------

IMPLICIT NONE
REAL, PARAMETER :: PI = 3.14159265359
REAL, PARAMETER :: GM = 4.0*PI**2
REAL :: radius, dt
REAL :: x1, x2, y1, y2, z1, z2
REAL :: vx1, vx2, vy1, vy2, vz1, vz2
real, dimension(100) :: ycept, zcept

INTEGER :: i=0, j, n=1, repcount=0, num, rep, outputerr, forbac=1
REAL :: r, rpri, p, ppri, vmag, o, opri, q, qpri, m, mpri
real :: temp, xc, yc, zc

REAL :: xpri, ypri, zpri, vxpri, vypri, vzpri

write(*,*) "generate yz return map of orbit"
write(*,*) "warning: not for unbound and invalid orbits"
write(*,*) "Initial Position"
read(*,*) x1, y1, z1
write(*,*) "Initial Velocity"
read(*,*) vx1, vy1, vz1
write(*,*) "timestep, radius"
read(*,*) dt, radius
write(*,*) "repeat"
read(*,*) rep

! this is simply a marker indicating that the program is running
write(*,*) "1"

OPEN ( unit = 100, file = "returnmapyz", status = 'replace', &
    & action = 'write', iostat=outputerr)

! this is a marker indicating that the file is sucessfully opened
write(*,*) "2", outputerr

write(100,*) "# return map using initial position"
write(100,*) "#", x1, y1, z1
write(100,*) "#initial velocity"
write(100,*) "#", vx1, vy1, vz1
write(100,*) "# timestep", dt
write(100,*) "# radius", radius
write(100,*)
```

```

! this is a marker indicating that the main program is ready
write(*,*) "3"

DO
! this loop only generates one point in the orbit after another
! no unused data is stored
  if (forbac == 1) then
    forbac = 2

! generate the next point using the initial conditions
  r = SQRT(x1**2 + y1**2 + z1**2)
  xpri = x1 + ( ( dt*vx1 ) / 2.0 )
  ypri = y1 + ( ( dt*vy1 ) / 2.0 )
  zpri = z1 + ( ( dt*vz1 ) / 2.0 )

  m = GM/(r**3)
  o = -3.75*( (GM*radius**2)/(r**7) )*(z1**2)
  p = 0.75*( (GM*radius**2)/(r**5) )
  q = 1.50*( (GM*radius**2)/(r**5) )
  vxpri = vx1 - (dt*x1/2.0)*( m + o + p )
  vypri = vy1 - (dt*y1/2.0)*( m + o + p )
  vzpri = vz1 - (dt*z1/2.0)*( m + o + p + q )

  rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
  x2 = x1 + (dt*vxpri)
  y2 = y1 + (dt*vypri)
  z2 = z1 + (dt*vzpri)

  mpri = GM/(rpri**3)
  opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
  ppri = 0.75*( (GM*radius**2)/(rpri**5) )
  qpri = 1.50*( (GM*radius**2)/(rpri**5) )
  vx2 = vx1 - (dt*xpri)*( mpri + opri + ppri )
  vy2 = vy1 - (dt*ypri)*( mpri + opri + ppri )
  vz2 = vz1 - (dt*zpri)*( mpri + opri + ppri + qpri )

! if the generated point is a match on the conditions
! the intercept is found and stored in the array
! n counts how many times a match has been found
    IF ( (x1 > 0.0) .AND. (x2 < 0.0) ) THEN
      temp = -x1/(x2 - x1)

      yc = y1 + temp*(y2-y1)
      zc = z1 + temp*(z2-z1)

      if ( (yc > 0.0) ) then
        ycept(n) = yc
        zcept(n) = zc
        n = n + 1
      end if
    ELSE IF ( (x1 < 0.0) .AND. (x2 > 0.0) ) THEN
      temp = -x1/(x2 - x1)

      yc = y1 + temp*(y2-y1)
      zc = z1 + temp*(z2-z1)
```

```
      if ( (yc > 0.0) ) then
        ycept(n) = yc
        zcept(n) = zc
        n = n + 1
      end if
    ELSE IF ( x1 == 0.0 ) THEN
      if ( (y1 > 0.0) ) then
        ycept(n) = y1
        zcept(n) = z1
        n = n + 1
      end if
    END IF
  else if ( forbac == 2 ) then
! this process repeats the
   forbac = 1

  r = SQRT(x2**2 + y2**2 + z2**2)
  xpri = x2 + ( ( dt*vx2 ) / 2.0 )
  ypri = y2 + ( ( dt*vy2 ) / 2.0 )
  zpri = z2 + ( ( dt*vz2 ) / 2.0 )

  m = GM/(r**3)
  o = -3.75*( (GM*radius**2)/(r**7) )*(z2**2)
  p = 0.75*( (GM*radius**2)/(r**5) )
  q = 1.50*( (GM*radius**2)/(r**5) )
  vxpri = vx2 - (dt*x2/2.0)*( m + o + p )
  vypri = vy2 - (dt*y2/2.0)*( m + o + p )
  vzpri = vz2 - (dt*z2/2.0)*( m + o + p + q )

  rpri = SQRT(xpri**2 + ypri**2 + zpri**2)
  x1 = x2 + (dt*vxpri)
  y1 = y2 + (dt*vypri)
  z1 = z2 + (dt*vzpri)

  mpri = GM/(rpri**3)
  opri = -3.75*( (GM*radius**2)/(rpri**7) )*(zpri**2)
  ppri = 0.75*( (GM*radius**2)/(rpri**5) )
  qpri = 1.50*( (GM*radius**2)/(rpri**5) )
  vx1 = vx2 - (dt*xpri)*( mpri + opri + ppri )
  vy1 = vy2 - (dt*ypri)*( mpri + opri + ppri )
  vz1 = vz2 - (dt*zpri)*( mpri + opri + ppri + qpri )


    IF ( (x2 > 0.0) .AND. (x1 < 0.0) ) THEN
      temp = -x2/(x1 - x2)

      yc = y2 + temp*(y1-y2)
      zc = z2 + temp*(z1-z2)

      if ( (yc > 0.0) ) then
        ycept(n) = yc
        zcept(n) = zc
        n = n + 1
      end if
    ELSE IF ( (x2 < 0.0) .AND. (x1 > 0.0) ) THEN
```

```
      temp = -x2/(x1 - x2)

      yc = y2 + temp*(y1-y2)
      zc = z2 + temp*(z1-z2)

      if ( (yc > 0.0) ) then
       ycept(n) = yc
       zcept(n) = zc
       n = n + 1 !n counts how many times a match has been found
      end if
     END IF
  end if

!   i = i + 1 ! i counts how many times that the entire code has been run
   if ( n == 100 ) then ! once we have filled up the 100 element array...
     repcount = repcount + 1 ! so now the array is filled and we write to file
     write(*,*) repcount
     do j = 1, 100

      if ( (repcount /= rep) .and. (j /= 100) ) write(100,200) ycept(j),
         zcept(j)

     end do
     n = 1 !start over
   end if

   if (repcount == rep) exit

END DO

close(100)
200 FORMAT (f12.3, 3x, f12.3)
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*. Cambrdige 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.