Astrophysical N-Body Simulations of Star Clusters

New Mexico
Supercomputing Challenge
Final Report
April 1, 2009

Team Number 61
Los Alamos Middle School

Team Members:
Sam Baty
Peter Armijo

Teacher:
Robert Dryja, LAMS

Project Mentors:
Christopher Fryer, Ph.D, LANL
Stephen Diehl, Ph.D, LANL

Table of Contents

Executive Summary

This project is focused on creating computer simulations of bodies in space and the gravitational forces they exert upon each other. Depending on the mass and position of each body, the bodies exert varying amounts of gravitational force upon each other. This gravitational force then affects the orbits of the various bodies.

We plan to ultimately create a simulation of a globular star cluster containing between 100,000 and 500,000 bodies. In this first phase of the project, we developed a computer code to simulate small groups of bodies in space and time. The computer used Newton's second law of mechanics and the universal law of gravitation.

We started first with a two-body problem and tested the code to simulate the orbit of the Moon about the Earth. We then moved to a five-body problem and modeled the orbits of the four major moons of Jupiter. The five-body code was extended to a code for an arbitrary number of bodies and then applied to simulate the Sun and nine planets of our Solar System. We extended our simulation by adding Jupiter's four major moons and the Earth's moon to the model, making a 15-body model.

We used a MacIntosh Computer with a 10.4 operating system, a compiler for the C programming language and Gnuplot for all plotting. We inserted the mass, velocity and position of each body at an initial time into the computer code and produced orbits of the bodies including the positions and velocities of the bodies relative to each other, over time. All simulations used a fourth-order Runge-Kutta integrator. For shorter simulations, we applied 18-second time steps in the integrator. For longer simulations with more bodies, we used time steps up to two-hours.

The first phase of the project was successful, in that we were able to create working models of various actual systems. These models closely reflect the actual orbits of the bodies in space. The next steps will entail simulations with more bodies, so that we can ultimately create a model of a globular star cluster.

Introduction

N-Body simulations model the gravitational force between masses such as stars or planets. The goal of this project is, ultimately, to create a simulation of a globular star cluster. Such a simulation would need to incorporate the following variables for each body: mass, velocity, and position. It would then calculate the interactions of the bodies based upon those variables and simulate their motion over time.

During the first phase of this project, which was completed this year, we created a series of simulations that incrementally added more bodies in order to begin solving the problem of modeling a star cluster with thousands of bodies. By the end of this year, we had a working simulation of the major bodies in our solar system. With the basic simulation method in place, we can work toward parallelizing our code and applying it to a globular cluster as a continuing project in the following years.

We chose this project because we are interested in space and wanted to understand more about how the universe works. It seemed like the type of project that was particularly well suited to a super-computer project. And in working with our mentors, it was a project that they felt had significant potential for original research and applications of interest to the astrophysics community.

We expect that our work on large n-body simulations, along with the work of others, will increase scientific knowledge of how our universe exists and works. This is an incredible opportunity because, in most situations, we cannot observe the workings of a globular cluster in nature. The timescale for human observation is simply too short. By creating a simulation on a computer, we can make time pass more quickly than in nature, allowing researchers to obtain a general understanding of how gravity and globular clusters might evolve over time.

Description

This project began with a presentation given by Dr. Stephen Diehl on Galactic Simulations. One of our team members attended the presentation and became interested in the idea of using this type of simulation for a supercomputing project. We began, then, by talking with Dr. Diehl about the project and doing some very basic research on the physics of bodies in space. The sources we used are listed in the references at the end of this report in the section on Gravity. Dr. Diehl also referred us to other references about the specifics of simulations and globular clusters. These references are also cited in references under the section on Gravity.

Our n-body simulations were performed by solving Newton's second law of mechanics,

$$\boldsymbol{F_j} = m_j \, \boldsymbol{a_j},$$

where $\boldsymbol{F_j}$ is the force on the j^{th} body given by the universal law of gravitation

$$\boldsymbol{a_j} = -G \sum m_j \, [\boldsymbol{r_i}-\boldsymbol{r_j}] \, / \, |\boldsymbol{r_i}-\boldsymbol{r_j}|^3.$$

The law of gravitation is added over all the bodies $\boldsymbol{r_i} \neq \boldsymbol{r_j}$. In the second equation, G is the gravitation constant, $\boldsymbol{r_i}$ is the position vector of the i^{th} body and $\boldsymbol{a_j} = d^2\boldsymbol{r_j} /dt^2$ is the acceleration vector of the j^{th} body. For an n-body simulation the above equations are integrated numerically. We applied a fourth-order Runge-Kutta integrator. Each simulation was started by fixing the mass (m_j), velocity ($\boldsymbol{v_j^0} = d\boldsymbol{r_j} /dt$), and position ($\boldsymbol{r_i^0}$) of every body at the initial time ($\boldsymbol{t = 0}$).

As a first step, we modeled a system of two bodies with identical masses and created a working simulation of the gravitational forces they exerted on each other. For the model two-body problem, escape velocities were discovered. We used a MacIntosh computer with a 10.4 operating system. We also used the computer language C for the coding and Gnuplot for all the graphics. We are developing our own code, rather than beginning with an existing code, so that we can learn the process from the earliest step and so that our final research results will be based upon our own work.

We then modeled a system of two bodies with different masses and used this formulation to model the Earth and Moon. We worked with the Earth and Moon simulation until we achieved a model that correctly represented the observed orbits of the Earth and Moon. After the Earth and Moon simulation, the code was modified to run a five-body problem of Jupiter and its four major moons: Io, Ganymede, Europa, and Callisto. When we used the physical mass, velocity, and position of each of these bodies we were able to produce a realistic model that accurately represents the orbits of these moons around Jupiter. Our code was then written in vector form for an arbitrary number of bodies. The present version of the n-body code is given in Section 6. We then moved on to simulate the solar system. As a first step, we modeled the Sun and nine planets (including Pluto). Once we had obtained a working model with these bodies, the four major moons of Jupiter and the Earth's Moon were added. Pluto and Mercury were modeled with orbits that do not move in the same planes as the other planets.

All of the orbital simulations were run in center of mass coordinates so that the calculations could be checked. The center of mass coordinates satisfy

$$\sum m_j \boldsymbol{r_j} \sim small,$$

and

$$\sum m_j \, d\boldsymbol{r_j}/dt \sim small,$$

for all simulations presented in this report.

Results

Our Project resulted in a working model of our Solar System, including the Sun, nine planets, the four major moons of Jupiter, and the Earth's Moon. This model accurately reflects the orbits of these bodies in relation to each other over time when compared to published NASA data. The results for the two-body and five-body simulations were calculated using 18-second time steps. The results for the complete Solar System were calculated using larger time steps, ranging up to two-hours, because the orbits of some of the planets have very long orbital periods around the sun. The code was used to simulate time periods of up to approximately 300 years. All simulations assumed physical units of Moon masses (7.350e+23 kg), hours and kilometers.

In modeling the orbits, we had to account for several physical differences in the geometry of the orbits of the planets. For instance, the orbits of both Mercury and Pluto are at an angle relative to the orbits of the other planets. Mercury's orbit is at a 7° angle and Pluto's is at a 17° angle. We modeled these angles to simulate the geometric differences in Mercury's and Pluto's orbits, so the calculations reflect these geometric effects correctly.

Pluto's orbit actually intersects the orbit of Neptune at two points, because Pluto's orbit is at an angle. We first did simulations that did not include the angle in Pluto's orbit, and the simulation did not predict an intersection with Neptune's orbit. We then adjusted the model to accurately reflect Pluto's out-of-plane angular variation and the simulations predicted the intersections.

We plotted the results for various simulations to show the position of the bodies in space and time on two-dimensional plots. Graphs included in the report are as given in the Figures section below.

Figures

A. Two-Body Simulations of the Earth and Moon:

Figure 1. The Moon's orbit around the Earth. The code simulated orbits of 720 hours.

Figure 2. The Earth's motion in a center of mass coordinate system. The Moon's pull on the Earth is seen by the Earth's motion.

Figure 3. The velocity components of the Moon.

B. Five-Body Simulations of Jupiter and its Moons:

Figure 4. The orbits of Jupiter's four major moons. The moons simulated were: Io, Ganymede, Europa, and Callisto.

Figure 5. The orbital periods of Jupiter's four major moons.

Figure 6. Jupiter's motion in a center of mass coordinate system. The code simulated 720 hours. Jupiter's motion is caused by the pull of its four major moons.

Figure 7. Jupiter's motion in a center of mass coordinate system. The code simulated 250 years. Since Jupiter's motion is contained in a small region, the numerical simulation is stable.

C. 10-Body and 15-Body Simulations of the Solar System:

Figure 8. The orbits of Jupiter's four major moons in a Sun centered coordinate system. This was a 15-body simulation to show that Jupiter's moons continue to orbit Jupiter as Jupiter orbits the sun.

Figure 9. The orbits of the inner planets. This was a 15-body simulation to compute the orbits of Mercury, Venus, Earth/Moon, and Mars. The results simulate approximately 3 years.

Figure 10. The orbits of the outer planets (with Pluto). This was a 15-body simulation. The results simulate approximately 250 years. This simulation predicted the intersection of Pluto's orbit with Neptune's orbit.

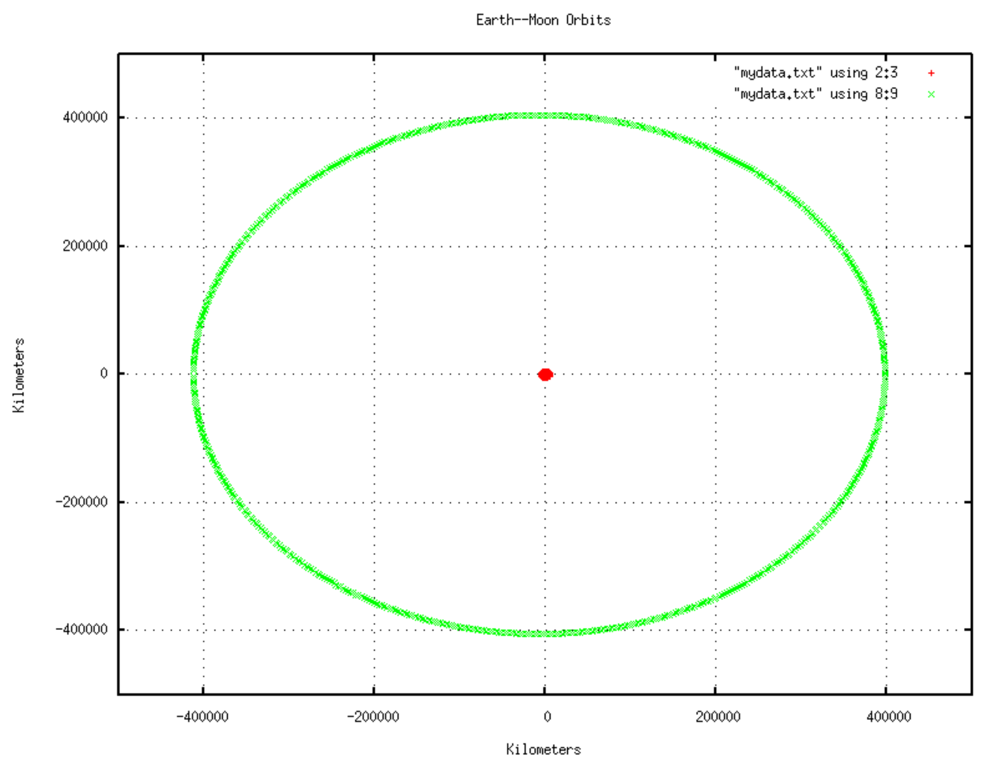

Figure 1. The Moon's orbit around the Earth.

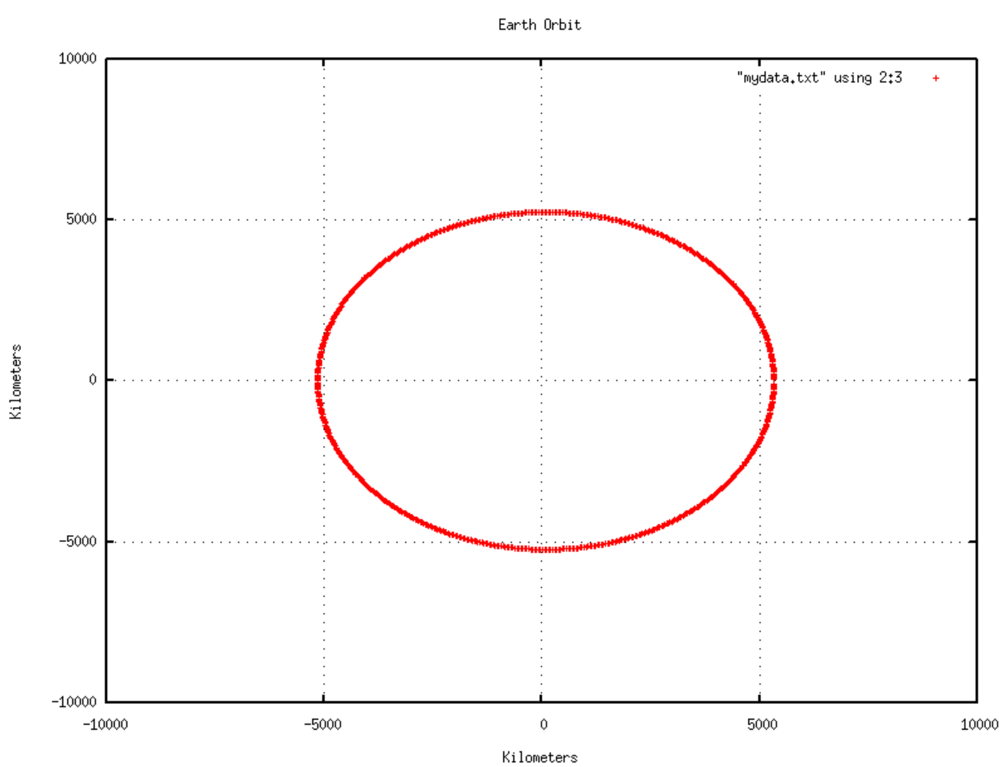

Figure 2. The Earth's motion in a center of mass coordinate system.

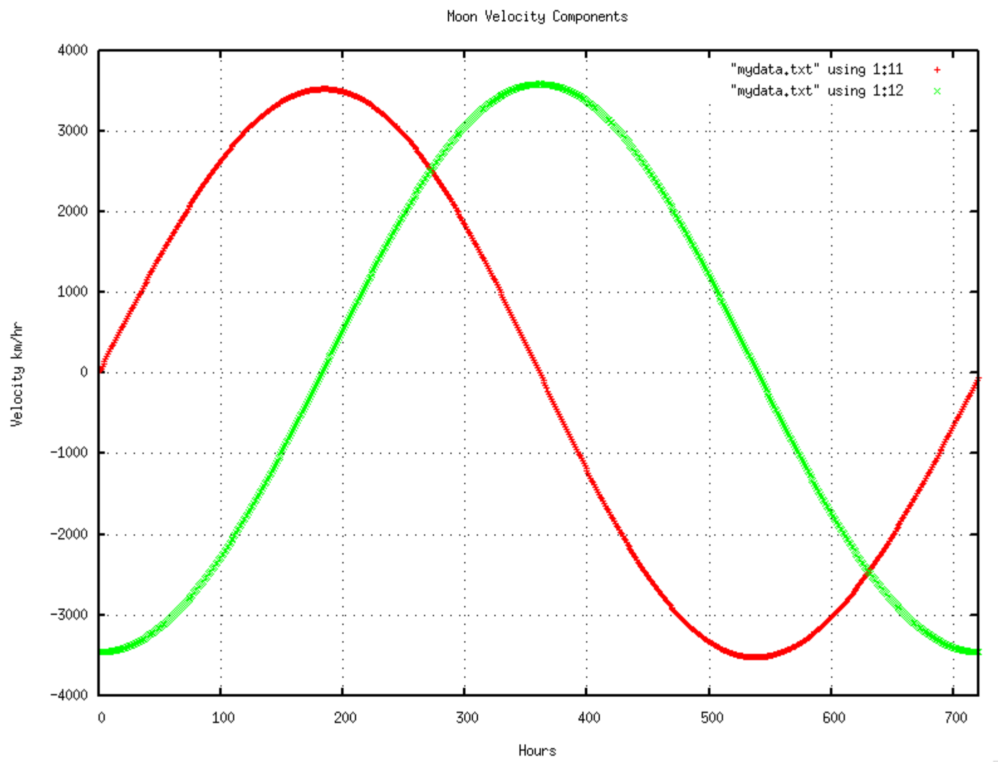

Figure 3. The velocity components of the Moon.

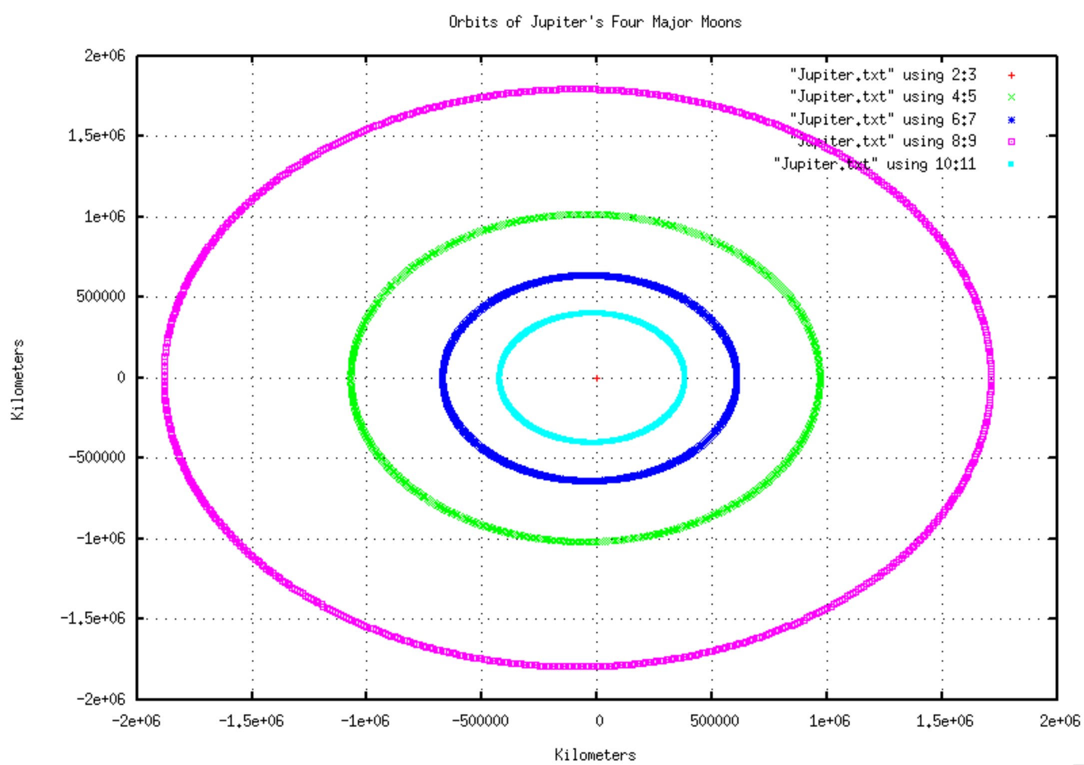

Figure 4. The orbits of Jupiter's four major moons.

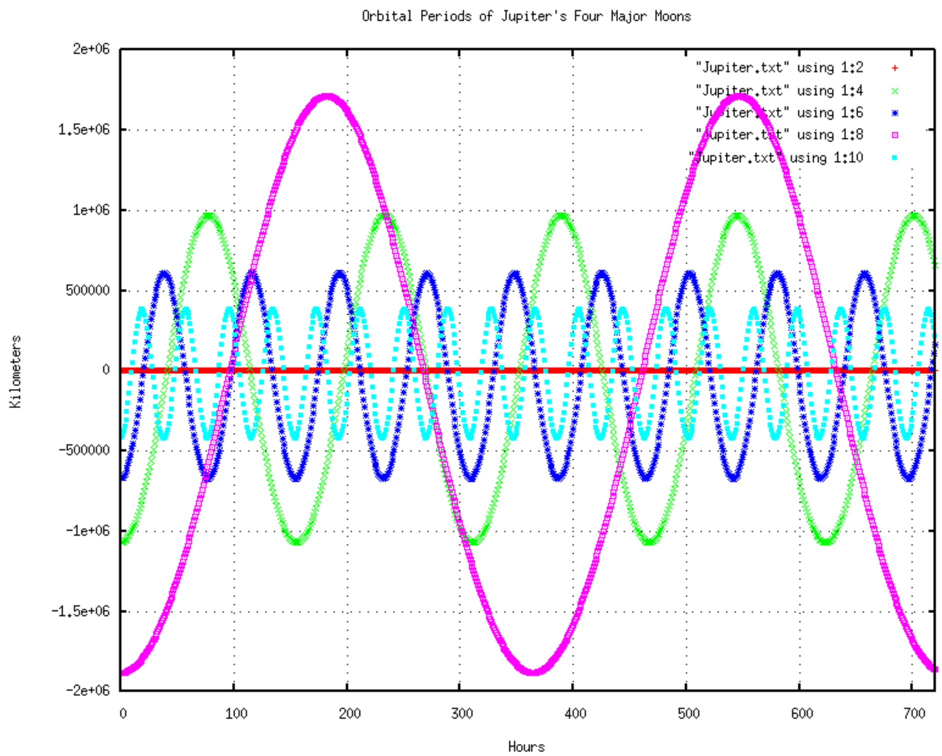

Figure 5. Orbtial periods of Jupiter's four major moons.

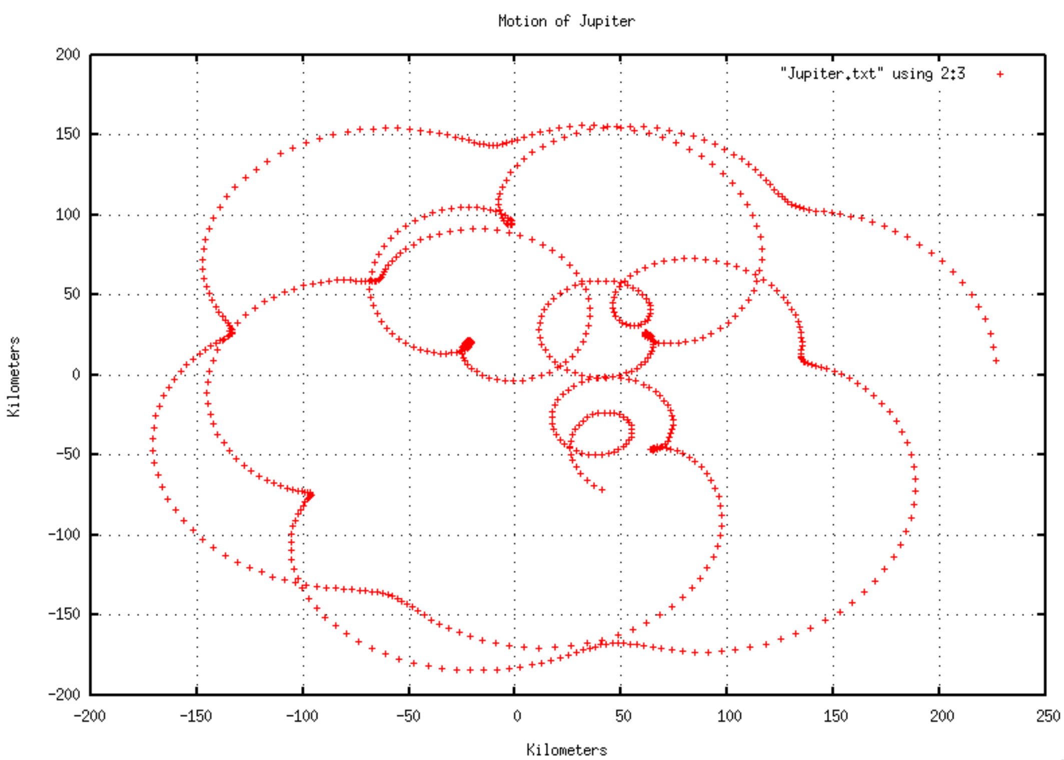

Figure 6. Jupiter's motion in a center of mass coordinate system for 720 hours.

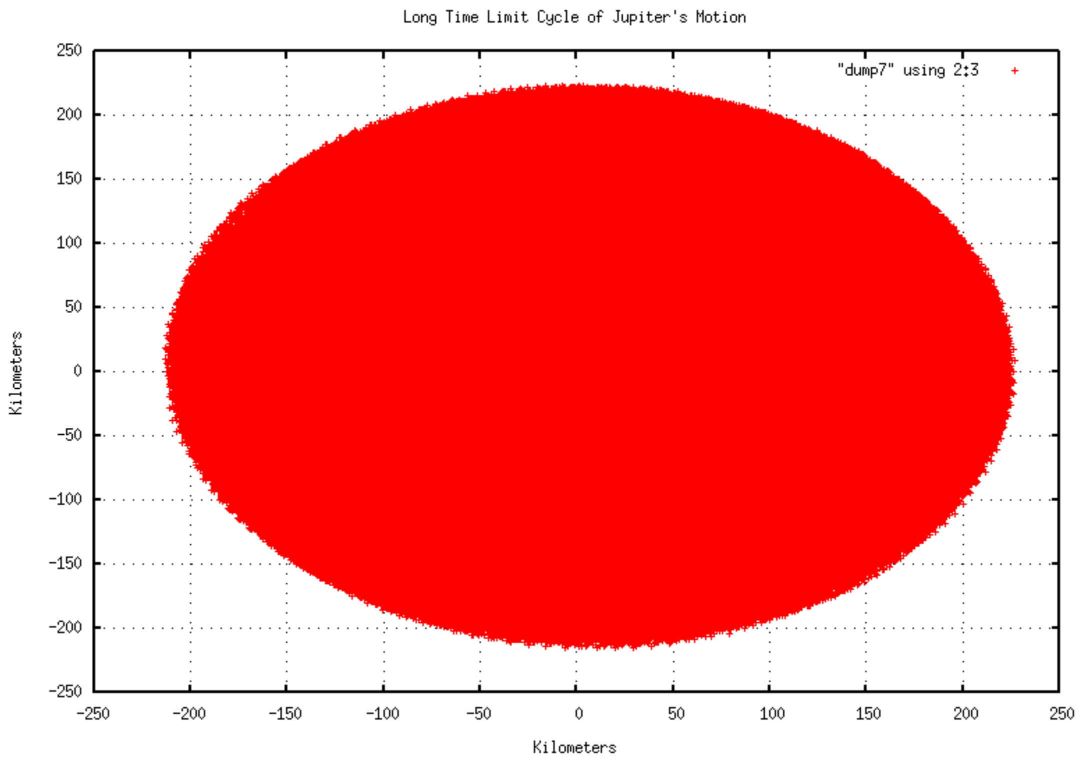

Figure 7. Jupiter's motion in a center of mass coordinate system for 250 years.

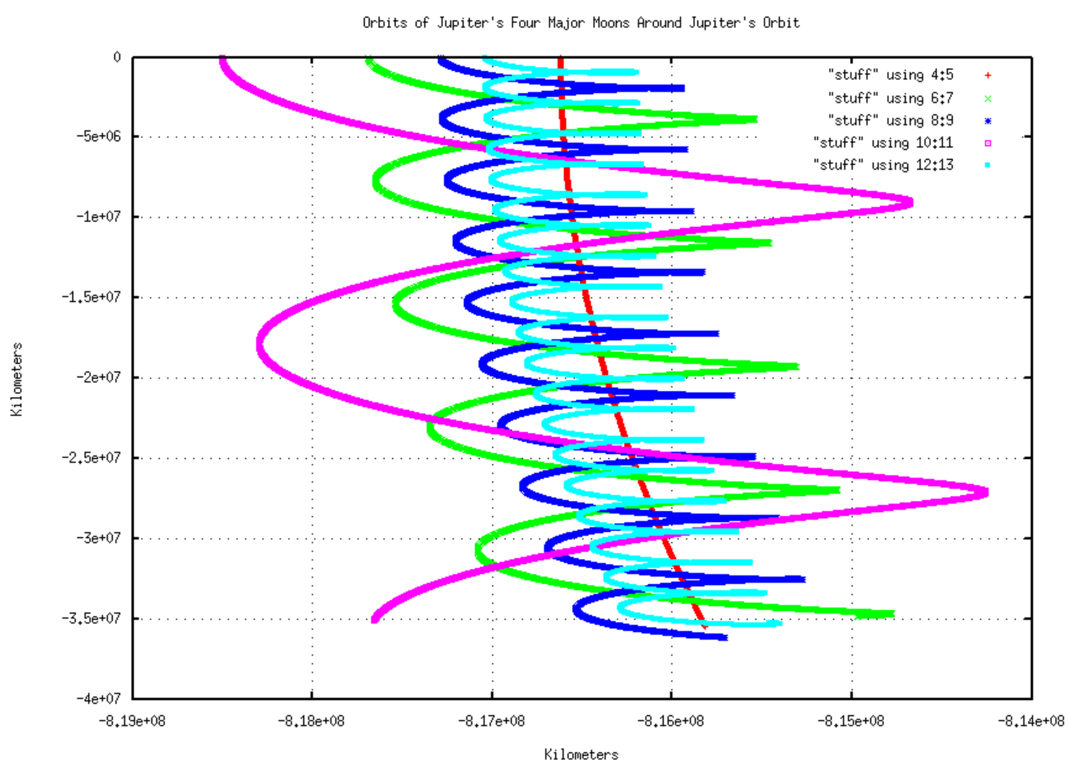

Figure 8. The orbits of Jupiter's four major moons in a Sun centered coordinates.

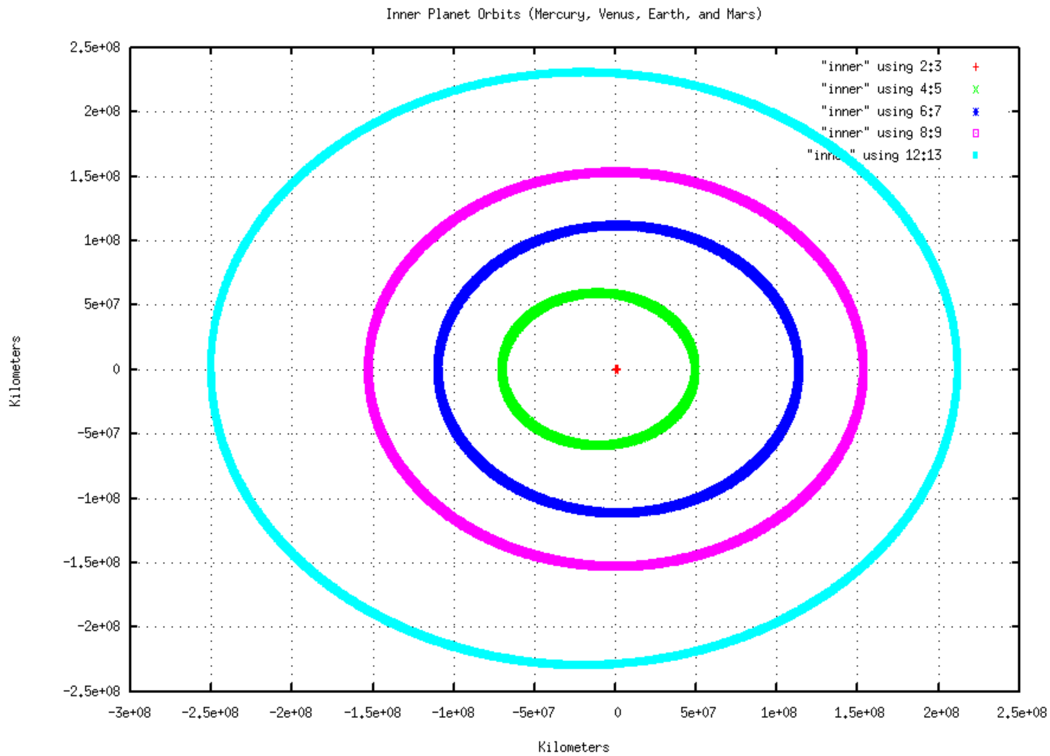

Figure 9. The orbits of the inner planets.

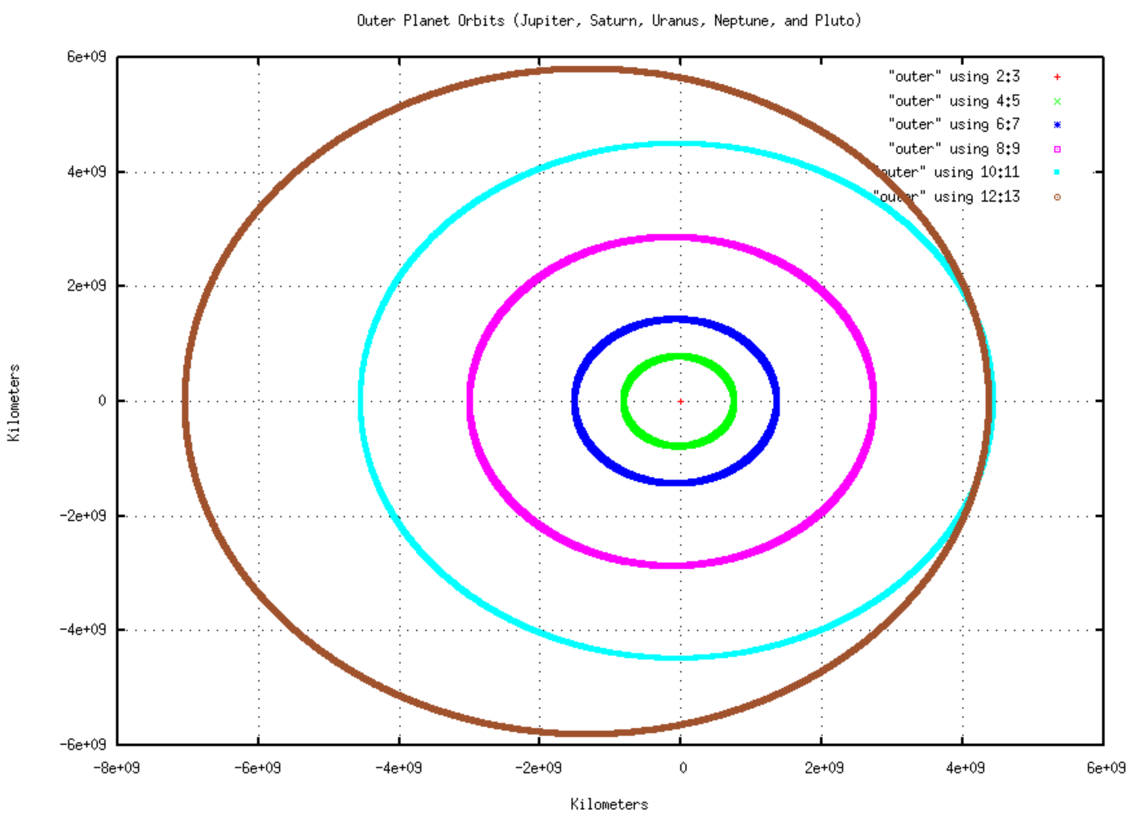

Figure 10. The orbits of the outer planets (with Pluto).

Conclusions

For our Project this year, we completed computer coding and numerical experiments for simple n-body problems in space and time. The simplest simulation was for a two-body problem, and the most complex simulation was for a 15-body problem. With a general version of our code, we were able to create a working model of the Solar System, including the Sun, the nine planets, Jupiter's four major moons and the Earth's Moon. These simulations accurately predicted data from NASA on the actual orbits of these bodies in our Solar System. We simulated the three-dimensional orbits of Pluto and Mercury and the intersection of the orbits of Pluto and Neptune.

We plan to continue the project over the next two years. During the next academic year, we hope to complete a model of a small globular cluster and run the simulation on a single computer. The following year, we plan to expand the Project to include a large globular cluster of tens of thousands of bodies and run the simulation on a large parallel computer.

N-Body Computer Program

```
/* Basic RK4 Integrator */

#include <stdio.h>
#include <math.h>

double fctn(int j, double m[], double G, double t, double x[], double v[]);
double labels1(int j, double x[], double xd[], double v[], double vd[], double k1[]);
double labels2(int j, double x[], double xd[], double v[], double vd[], double k1[]);
double labels3(int j, double x[], double v[], double kd[]);
double fmod(double c1, double c2);

int main()
{
  int i, i2, ii, j, k, l, N, p1, p2, p3;
  double c1,c2,c3;
  double m[15], G, nrm;
  double h,hh,t,t0;
  double x[45], v[45];
  double kd[90], k1[90], k2[90], k3[90], k4[90];
  double xd[90],vd[90];
  double cc1,cc2,cc3,cc4,cc5,cc6;

  /*  Initialize Physical Values for the Masses and the Constant G  */

  m[1] = 27066267.52;  /* Sun */
  m[2] = 4.49;      /* Mercury */
  m[3] = 66.27;      /* Venus */
  m[4] = 81.28;      /* Earth */
  m[5] = 1.00;      /* Moon */
  m[6] = 8.73;      /* Mars */
  m[7] = 25834.81;       /* Jupiter */
  m[8] = 2.01;      /* Jupiter Moon 1 2.01 */
  m[9] = 0.65;      /* Jupiter Moon 2 0.65 */
  m[10] = 1.47;     /* Jupiter Moon 3 1.47 */
  m[11] = 1.21;     /* Jupiter Moon 4 1.21 */
  m[12] = 7735.20;     /* Saturn */
  m[13] = 1181.55;     /* Uranus */
  m[14] = 1393.80;     /* Neptune */
  m[15] = 0.17;     /* Pluto */

  G=63460000000.0;

  /*  Initialize Parameters for RK4 Integrator h, N and t0  */

  h=1.0/20.0;
  l=4800*30;

  hh=0.5*h;
  t0=0.0;
  t=t0;

  /*  Specify Initial Position and Velocity for the N-Bodies */

  /*  Position of the Sun m1 x[1], x[2], z[3]  */
  /*  Velocity of the Sun m1 v[1], v[2], v[3]  */

  x[1]=1578520.0151093653+0.23166959;    /* position in the x direction of mass m1 */
  x[2]=0.0;           /* position in the y direction of mass m1 */
  x[3]=-15.080004084419837;           /* position in the z direction of mass m1 */

  v[1]=0.0;           /* velocity in the x direction of mass m1 */
  v[2]=54.79881428143069+0.008620769;    /* velocity in the y direction of mass m1 */
  v[3]=0.0;           /* velocity in the z direction of mass m1 */

  /*  Position of Mercury m2 x[4], x[5], x[6]  */
  /*  Velocity of Mercury m2 v[4], v[5], v[6]  */

  x[4]=-69299572.31;    /* position in the x direction of mass m2 69820000.0*/
  x[5]=0.0;           /* position in the y direction of mass m2 */
  x[6]=8508917.56;           /* position in the z direction of mass m2 */

  v[4]=0.0;           /* velocity in the x direction of mass m2 */
  v[5]=-38.86*3600;     /* velocity in the y direction of mass m2 */
  v[6]=0.0;           /* velocity in the z direction of mass m2 */
```

```
/*  Position of Venus m3 x[7], x[8], x[9] */
/*  Velocity of Venus m3 v[7], v[8], v[9] */

x[7]=-108940000.0;    /* position in the x direction of mass m3 */
x[8]=0.0;           /* position in the y direction of mass m3 */
x[9]=0.0;           /* position in the z direction of mass m3 */

v[7]=0.0;           /* velocity in the x direction of mass m3 */
v[8]=-34.79*3600;     /* velocity in the y direction of mass m3 */
v[9]=0.0;           /* velocity in the z direction of mass m3 */

/*  Position of the Earth m4 x[10], x[11], x[12]  */
/*  Velocity of the Earth m4 v[10], v[11], v[12]  */

x[10]=-152100000.0;   /* position in the x direction of mass m4 */
x[11]=0.0;          /* position in the y direction of mass m4 */
x[12]=0.0;          /* position in the z direction of mass m4 */

v[10]=0.0;          /* velocity in the x direction of mass m4 */
v[11]=-29.29*3600;    /* velocity in the y direction of mass m4 */
v[12]=0.0;          /* velocity in the z direction of mass m4 */

 /*  Position of the Moon m5 x[13], x[14], x[15] */
 /*  Velocity of the Moon m5 v[13], v[14], v[15] */

x[13]=x[10]-405500.0;   /* position in the x direction of mass m5 */
x[14]=0.0;          /* position in the y direction of mass m5 */
x[15]=0.0;          /* position in the z direction of mass m5 */

v[13]=0.0;          /* velocity in the x direction of mass m5 */
v[14]=v[11]-0.964*3600;    /* velocity in the y direction of mass m5 */
v[15]=0.0;          /* velocity in the z direction of mass m5 */

/*  Position of Mars m6 x[16], x[17], z[18]  */
/*  Velocity of Mars m6 v[16], v[17], v[18]  */

x[16]=-249230000.0;  /* position in the x direction of mass m6 */
x[17]=0.0;           /* position in the y direction of mass m6 */
x[18]=0.0;           /* position in the z direction of mass m6 */

v[16]=0.0;           /* velocity in the x direction of mass m6 */
v[17]=-21.97*3600;    /* velocity in the y direction of mass m6 */
v[18]=0.0;           /* velocity in the z direction of mass m6 */

/*  Position of Jupiter m7 x[19], x[20], x[21]  */
/*  Velocity of Jupiter m7 v[19], v[20], v[21]  */

x[19]=-816620000.0;    /* position in the x direction of mass m7 */
x[20]=0.0;           /* position in the y direction of mass m7 */
x[21]=0.0;           /* position in the z direction of mass m7 */

v[19]=0.0;           /* velocity in the x direction of mass m7 */
v[20]=-12.44*3600;     /* velocity in the y direction of mass m7 */
v[21]=0.0;           /* velocity in the z direction of mass m7 */

/*  Position of Jupiter Moon #1 m8 x[22], x[23], x[24] */
/*  Velocity of Jupiter Moon #1 m8 v[22], v[23], v[24] */

x[22]=x[19]-1070000.0;    /* position in the x direction of mass m8 */
x[23]=0.0;           /* position in the y direction of mass m8 */
x[24]=0.0;           /* position in the z direction of mass m8 */

v[22]=0.0;           /* velocity in the x direction of mass m8 */
v[23]=v[20]-39169.0;     /* velocity in the y direction of mass m8 */
v[24]=0.0;           /* velocity in the z direction of mass m8 */

/*  Position of Jupiter Moon #2 m9 x[25], x[26], x[27]  */
/*  Velocity of Jupiter Moon #2 m9 v[25], v[26], v[27]  */
x[25]=x[19]-670900.0;   /* position in the x direction of mass m9 */
x[26]=0.0;          /* position in the y direction of mass m9 */
x[27]=0.0;          /* position in the z direction of mass m9 */

v[25]=0.0;          /* velocity in the x direction of mass m9 */
v[26]=v[20]-49464.0;    /* velocity in the y direction of mass m9 */
v[27]=0.0;          /* velocity in the z direction of mass m9 */

 /*  Position of Jupiter Moon 3 m10 x[28], x[29], x[30] */
 /*  Velocity of Jupiter Moon 3 m10 v[28], v[29], v[30] */

x[28]=x[19]-1883000.0;   /* position in the x direction of mass m10 */
```

```
  x[29]=0.0;          /* position in the y direction of mass m10 */
  x[30]=0.0;          /* position in the z direction of mass m10 */

  v[28]=0.0;          /* velocity in the x direction of mass m10 */
  v[29]=v[20]-29556.0;    /* velocity in the y direction of mass m10 */
  v[30]=0.0;          /* velocity in the z direction of mass m10 */

  /*  Position of Jupiter Moon 4 m11 x[31], x[32], z[33]  */
  /*  Velocity of Jupiter Moon 4 m11 v[31], v[32], v[33]  */

  x[31]=x[19]-421600.0;  /* position in the x direction of mass m11 */
  x[32]=0.0;           /* position in the y direction of mass m11 */
  x[33]=0.0;           /* position in the z direction of mass m11 */

  v[31]=0.0;           /* velocity in the x direction of mass m11 */
  v[32]=v[20]-62424.0;    /* velocity in the y direction of mass m11 */
  v[33]=0.0;           /* velocity in the z direction of mass m11 */

  /*  Position of Saturn m12 x[34], x[35], x[36]  */
  /*  Velocity of Saturn m12 v[34], v[35], v[36]  */

  x[34]=-1514500000.0;    /* position in the x direction of mass m12 */
  x[35]=0.0;           /* position in the y direction of mass m12 */
  x[36]=0.0;           /* position in the z direction of mass m12 */

  v[34]=0.0;           /* velocity in the x direction of mass m12 */
  v[35]=-9.09*3600;     /* velocity in the y direction of mass m12 */
  v[36]=0.0;           /* velocity in the z direction of mass m12 */

  /*  Position of Uranus m13 x[37], x[38], x[39] */
  /*  Velocity of Uranus m13 v[37], v[38], v[39] */

  x[37]=-3003620000.0;    /* position in the x direction of mass m13 */
  x[38]=0.0;           /* position in the y direction of mass m13 */
  x[39]=0.0;           /* position in the z direction of mass m13 */

  v[37]=0.0;           /* velocity in the x direction of mass m13 */
  v[38]=-6.49*3600;     /* velocity in the y direction of mass m13 */
  v[39]=0.0;           /* velocity in the z direction of mass m13 */

  /*  Position of Neptune m14 x[40], x[41], x[42]  */
  /*  Velocity of Nepture m14 v[40], v[41], v[42]  */

  x[40]=-4545670000.0;   /* position in the x direction of mass m14 */
  x[41]=0.0;          /* position in the y direction of mass m14 */
  x[42]=0.0;          /* position in the z direction of mass m14 */

  v[40]=0.0;          /* velocity in the x direction of mass m14 */
  v[41]=-5.37*3600;    /* velocity in the y direction of mass m14 */
  v[42]=0.0;          /* velocity in the z direction of mass m14 */

   /*  Position of Pluto m15 x[43], x[44], x[45] */
   /*  Velocity of Pluto m15 v[43], v[44], v[45] */

  x[43]=-7047587322.65;   /* position in the x direction of mass m15 Ref:
7,375,930,000.0*/
  x[44]=0.0;          /* position in the y direction of mass m15 */
  x[45]=2176202264.16;          /* position in the z direction of mass m15 */

  v[43]=0.0;          /* velocity in the x direction of mass m15 */
  v[44]=-3.71*3600;    /* velocity in the y direction of mass m15 */
  v[45]=0.0;          /* velocity in the z direction of mass m15 */

  /*  Initialize Computational Arrays */

  for (j=1;j<=90;++j)
    {
      k1[j]=0.0;
      k2[j]=0.0;
      k3[j]=0.0;
      k4[j]=0.0;
    }

  for(j=1;j<=45;++j)
    {
      xd[j]=0.0;
      vd[j]=0.0;
    }

  /*  Loop for Integration */
```

```
  for (ii=1;ii<=l;++ii)
  {
    for (j=1;j<=90;++j){k1[j]=h*fctn(j,m,G,t,x,v);}

    t +=hh;
    labels1(j, x, xd, v, vd, k1);
    for(j=1;j<=90;++j){k2[j]=h*fctn(j,m,G,t,xd,vd);}
    /*    for(j=1;j<=90;++j){printf("%i %12.6g\n", j, k1[j]);} */

    labels1(j, x, xd, v, vd, k2);
    for(j=1;j<=90;++j){k3[j]=h*fctn(j,m,G,t,xd,vd);}

    t +=hh;
    labels2(j, x, xd, v, vd, k3);
    for(j=1;j<=90;++j){k4[j]=h*fctn(j,m,G,t,xd,vd);}

    for(j=1;j<=90;++j){kd[j]=(k1[j]+2.0*k2[j]+2.0*k3[j]+k4[j])/6.0;}
    labels3(j, x, v, kd);

    /* Check Center-of-Mass Results x[-] position, v[-] velocity */

    /*    cc1=0.0; */
    /*    cc2=0.0; */
    /*    cc3=0.0; */
    /*    for(j=1;j<=15;++j){p1=1+3*(j-1);cc1+=m[j]*x[p1];}  */
    /*    for(j=1;j<=15;++j){p2=2+3*(j-1);cc2+=m[j]*x[p2];}  */
    /*    for(j=1;j<=15;++j){p3=3+3*(j-1);cc3+=m[j]*x[p3];}  */

    c1=ii;
    c2=20;
    c3=fmod(c1,c2);

    /* Print Results for Inner Planets: Mercury, Venus, Earth, Earth's Moon, and Mars */
    if (c3 == 0.0){printf("%12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g
%12.8g %12.8g %12.8g %12.8g\n",t, x[1], x[2], x[4], x[5], x[7], x[8], x[10], x[11],
x[13], x[14], x[16], x[17]);}

    /* Print Results of Outer Planets: Jupiter, Saturn, Uranus, Neptune, and Pluto */
    /*    if (c3 == 0.0){printf("%12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g
%12.8g %12.8g %12.8g %12.8g %12.8g\n",t, x[1], x[2], x[19], x[20], x[34], x[35], x[37],
x[38], x[40], x[41], x[43], x[44]);} */

    /* Print Results of Jupiter: Jupiter, and Jupiter's Four Major Moons */
    /*    if (c3 == 0.0){printf("%12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g %12.8g
%12.8g %12.8g %12.8g %12.8g %12.8g\n",t, x[1], x[2], x[19], x[20], x[22], x[23], x[25],
x[26], x[28], x[29], x[31], x[32]);} */

    /* Print Results of Center-of-Mass Check */
    /*    if (c3 == 0.0){printf("%12.8g %12.8g %12.8g %12.8g\n",t, cc1, cc2, cc3);} */
  }

  return 0;
}

double fctn(int j, double m[], double G, double t, double x[], double v[])
{
  int i1, j1, j2, j3, j4, jb, k1, k2, k3, k4;
  double c1, c2, c3, eps, f, nrm;

    eps=0.001;

    c1=j;
    c2=2.0;
    c3=fmod(c1,c2);

  if (c3 != 0.0) {
    for(i1=1;i1<1+j/2;++i1){}
    f=v[i1];
  }

  if (c3 == 0.0) {
    for(i1=1;i1<j/2;++i1){}
    f=0.0;

    if (i1 <= 3) {j1=1;}

    jb=91;
    j2=i1;

     if(i1 > 3) {
       for(j4=i1;j4 < jb;++j4){
```

```
           j2=j2-3;

           if(j2 <= 3){jb=j2;}
        }
      }
       for(j3=1;j3<=15;++j3){
         k1=1+3*(j1-1);
         k2=1+3*(j3-1);
          nrm=(x[k1]-x[k2])*(x[k1]-x[k2])+(x[1+k1]-x[1+k2])*(x[1+k1]-x[1+k2])+(x[2+k1]-
x[2+k2])*(x[2+k1]-x[2+k2]);
          nrm=sqrt(nrm);
          k3=j2+3*(j1-1);
          k4=j2+3*(j3-1);
          f+=-1.0*G*m[j3]*(x[k3]-x[k4])/(nrm*nrm*nrm+eps);
        }
       /*     printf("%i %i %i %i\n",j,i1,j1,j2); */
      if(j2 == 3){j1+=1;}
  }
  return f;
}


double labels1(int j, double x[], double xd[], double v[], double vd[], double k1[])
{
  int i2;
  double c1, c2, c3;

     for (j=1;j<=90;++j)
    {
    c1=j;
    c2=2.0;
    c3=fmod(c1,c2);

    if (c3 != 0.0) {
    for(i2=1;i2<1+j/2;++i2){}
    xd[i2]=x[i2]+0.5*k1[j];
    }
    if (c3 == 0.0) {
    for(i2=1;i2<j/2;++i2){}
    vd[i2]=v[i2]+0.5*k1[j];
    }
    }
 return;
}

double labels2(int j, double x[], double xd[], double v[], double vd[], double k1[])
{
  int i2;
  double c1, c2, c3;

     for (j=1;j<=90;++j)
    {
    c1=j;
    c2=2.0;
    c3=fmod(c1,c2);

    if (c3 != 0.0) {
    for(i2=1;i2<1+j/2;++i2){}
    xd[i2]=x[i2]+k1[j];
    }
    if (c3 == 0.0) {
    for(i2=1;i2<j/2;++i2){}
    vd[i2]=v[i2]+k1[j];
    }
    }
 return;
}

double labels3(int j, double x[], double v[], double kd[])
{
  int i2;
  double c1, c2, c3;

     for (j=1;j<=90;++j)
    {
    c1=j;
    c2=2.0;
    c3=fmod(c1,c2);

    if (c3 != 0.0) {
    for(i2=1;i2<1+j/2;++i2){}
```

```
    x[i2]+=kd[j];
    }
    if (c3 == 0.0) {
    for(i2=1;i2<j/2;++i2){}
    v[i2]+=kd[j];
    }
    }
 return;
}
```

Significant Achievements

Over the course of the year, we accomplished our goal of creating a working model of the Solar System, which was not a simple project. We accounted for various differences in the orbits of the planets and obtained a model that accurately reflects the basic features of the planetary orbits and matches NASA data. While working on this model, we learned some of the basics of computer modeling of physical problems, as well as some basic computer skills. We hope to expand our knowledge in these areas in the upcoming years of this Project. Our greatest achievement was in the area of gravitational physics. We have gained an understanding of the concepts of how gravity and gravitational interactions work by running our numerical simulations. We will be able to apply this understanding in our future educational work.

Acknowledgements

We would especially like to thank Dr. Stephen Diehl, Dr. Christopher Fryer, Mr. Robert Dryja, and Dr. Roy Baty for helping us with all of the aspects of this project. They spent a great deal of time with us, explaining all aspects of the project, and proved to be excellent at explaining difficult concepts at a level we could understand.

References

A. Gravity:

1. Hewitt, P. G., **Conceptual Physics**, Prentice Hall, 2006.

2. Kittel, C., Knight, W. D., and Ruderman, M. A., **Mechanics, Berkeley Physics Course, Volume 1, 2nd Edition**, McGraw-Hill, 1973.

3. Shu, F. H., **The Physical Universe: An Introduction to Astronomy**, University Science Books, 1982.

4. Aarseth, Sverre J., **Gravitational N-Body Simulations**, Cambridge University Press, 2003.

5. Binney, J. and Tremaine, S., **Galactic Dynamics, 2nd Edition**, Princeton University Press, 2008.

B. Orbital Data:

6. Williams, D. R., "NASA Planetary Fact Sheets," September 2004, nssdc.gsfc.nasa.gov/planetary/factsheet.

C. Runge-Kutta Integrator:

7. Burden, R. L., and Faires, J. D., **Numerical Analysis, 3rd Edition**, Prindle, Weber & Schmidt, 1985.

D. Computing References:

8. Crawford, D., "gnuplot: An Interactive Plotting Program," December 1998, www.gnuplot.info/docs_4.3/gnuplot.pdf.

9. Kernighan, B. W. and Ritchie, D. M., **The C Programming Language, 2nd Edition**, Prentice Hall, 1988.