

# **Chaos in the Motion of Asteroids**

New Mexico Adventures in  
Supercomputing Challenge  
Final Report  
April 7, 2004

Team 66  
Santa Fe High School

Team Members:

Taryn Flock  
Bruce Miano

Teacher:

Anita Gerlach

Mentors

David Stupin  
Doyne Farmer

Table of Contence:	
Executive Summary:	3
Report:	
Method:	4
Results:	5
Figure 1: Graph of Asteroids Motion	5
Conclusions:	5
Recommendations:	6
Acknowledgements:	6
Appendixes:	
A: Equations	7
B: Data:	9
C: Program Code:	13
Bibliography of Sources and Softwear:	19

## Executive Summary:

In this project, we attempted to gain some insight into the nature of the three-body problem and its relation to celestial mechanics. The task was to compute, given initial values for mass, velocity, and position, any of these three values at some future point for any of the bodies. For the sake of simplicity, we chose to work with the restricted three-body problem in which one body, the most massive, remains at rest while the other two bodies orbit the first within a plane.

The solutions of the “three-body” problem are generally chaotic, meaning that slight variations in initial values produce vastly different results over time, and, hence, they are difficult to calculate. To predict values over successive time intervals, we used a computer program based on symplectic<sup>1</sup> integration because the solutions are not provided by a simple equation. Using a plotting routine, we analyzed our predicted values and how they fluctuated with time. Our results indicate the chaotic nature of the problem, and the perturbed motion of both outer bodies.

---

<sup>1</sup> Symplectic’ indicates conservation of energy through Hamiltonian, energy conserving, structure.

The problem of “three bodies” has been at the forefront of physics for many centuries. Now, however, it has become particularly accessible to study, with the development of chaos theory and the supercomputing capabilities of modern computers. In order to tie our interest in astrophysics with the background of one of our mentors in nonlinear dynamics (i.e., chaotic motion), we decided to explore this classic problem. The goal was to provide generalizations about the behavior of the three bodies, because their positions and velocities can not be computed either directly or accurately. They behave chaotically and by definition are highly dependent on initial conditions. In particular, we concentrated on the irregular motion exhibited by asteroids.

Methods:

We approached a restricted “three-body” problem in which the largest mass, significantly larger than the other two, is fixed in the center and the two outer masses are free to orbit within a plane. The center mass represents our Sun and rests at origin on the Cartesian plane. The outer mass represents Jupiter, with a mass and an orbital radius assigned correspondingly. The middle mass, the smallest, represents an asteroid, with a mass and an orbital radius corresponding to a typical asteroid in the asteroid belt.

We began by describing the system with equations developed in Newtonian Dynamics (see Appendix A). From these equations, we derived the Hamiltonian for the system. The Hamiltonian is an equation for the total energy of a system and is commonly used for such systems as the solar system, i.e., those that are volume-preserving (meaning that solutions that are based on the same initial values remain, even when chaotic, within a reasonable range [Lorenz: 61]). We then used a symplectic<sup>2</sup> integrator to obtain values for the position and velocity of the two outer bodies over successive time intervals, by integrating the derivative of our Hamiltonian. In our program, the initial velocities are the velocities necessary to maintain a stable orbit, if the system considered consisted only of the body whose velocity we are calculating and the sun. Our program (see Appendix

---

<sup>2</sup> ‘Symplectic’ indicates conservation of energy through Hamiltonian structure. Because we needed to integrate the derivatives of the Hamiltonian, an integration routine specially designed for use with Hamiltonian structure seemed particularly appropriate.

C) was compiled and run in the Borland C++ Compiler and the data was plotted with Gnuplot.

Results:

Our data does indicate that the orbits of both Jupiter and the Asteroid were affected and the motion was chaotic. Our data shows perturbed, altered slightly in a non-linear fashion, orbits for both bodies, . The asteroid, the smaller body, moves slowly from away from Jupiter, more slowly than it would if Jupiter were not present because of Jupiter's gravitational attraction. Jupiter similarly moves inward infinitesimally more quickly than it would with out the presence of the asteroid. The gravitational forces between the bodies alter the orbits of each. In addition, the changes in both position and momentum are not linear; hence, the idea that the data is chaotic is supported (see tables in Appendix B).

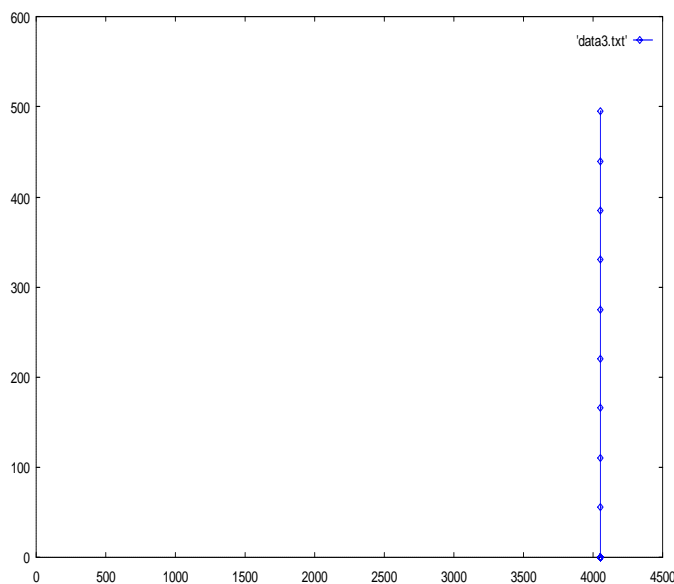


Figure 1: Graph of Asteroid's Motion

--In this graph, we see orbit of the asteroid in blue. **The linear appearance is due to the small time interval:  $9 \times 10^{-2}$  s**

--Range and Domain have been distorted to make graphing possible, x is in ten-million kilometers and y is in million kilometers.

Conclusions:

As expected, the motion of each body is affected by the presence of the other. Jupiter moving slightly inward, while the asteroid stays farther out than it ordinarily would. The solution of the three-body problem for the motion of the sun, Jupiter, and an asteroid is not a periodic solution, as seen in the perturbation of the orbits, of the three-body problem; rather the solution appears chaotic.

### Recommendations:

The next step is to carry the integration over a larger time interval. Pluto was the first planet proved to have a chaotic orbit; the calculations giving this proof use a time interval of 845 million years. [Sussman; 1]. To prove Jupiter's orbit chaotic a similar or perhaps longer time interval would be needed. Also, it would be beneficial to explore what happens as the mass of the asteroid is varied. Is there some ranges of masses for which the perturbations of the orbit could lead to collisions with Earth? Could the same happen as initial velocity is varied?

### Acknowledgements:

Our mentors David Stupin and J. Dooyne Farmer, and teacher Ms. Gerlach were an invaluable source of motivation and aid. A special thanks, as well, to Yuzuru Sato who coached us through the integration routine. This project happened because of their assistance.

## Appendix A: Equations

Newtonian:

$$F_{12} = Gm_1m_2 * L_{12} / (r_2-r_1)^2$$

$$F_{13} = Gm_1m_3 * L_{13} / (r_3-r_1)^2$$

Where  $L_{12}$  is a unit vector parallel to  $(r_2-r_1)$  such that:

$$L_{12} = \frac{r_2 - r_1}{|r_2 - r_1|} = \frac{(x_2-x_1)i + (y_2-y_1)j}{\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}}$$

Where  $i$  is a unit vector in the x-direction and  $j$  is a unit vector in the y-direction.

$$= \frac{F_{12} + F_{13}}{M_1} = (m_2 + m_3) G \left( \frac{L_{12}}{(r_2-r_1)^2} + \frac{L_{13}}{(r_3-r_1)^2} \right)$$

Hamiltonian:

$$H = KE + PE$$

$$H = \frac{p_{m1}^2}{m_1} + \frac{p_{m2}^2}{m_2} + \frac{Gm_1m_2}{\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}} + \frac{Gm_1m_3}{\sqrt{((x_3-x_1)^2 + (y_3-y_1)^2)}} + \frac{Gm_2m_1}{\sqrt{((x_1-x_2)^2 + (y_1-y_2)^2)}} + \frac{Gm_2m_3}{\sqrt{((x_3-x_2)^2 + (y_3-y_2)^2)}}$$

Where  $p_{m1} = \sqrt{(p_{x1}^2 + p_{y1}^2)}$  and  $p_{m2} = \sqrt{(p_{x2}^2 + p_{y2}^2)}$

Subscripts of 1 or  $m_1$  indicate that the value is related to the first body, 2 or  $m_2$  the second, and 3 the third

$$dH / dp_{x1} = p_{x1} / m_1$$

$$dH / dp_{y1} = p_{y1} / m_1$$

$$dH / dp_{x2} = p_{x2} / m_2$$

$$dH / dp_{y2} = p_{y2} / m_2$$

$$dH / dx_1 = \frac{Gm_1m_2(x_2-x_1)}{((x_2-x_1)^2 + (y_2-y_1)^2)^{3/2}} + \frac{Gm_1m_3(x_3-x_1)}{((x_3-x_1)^2 + (y_3-y_1)^2)^{3/2}}$$

$$dH/dy_1 = \frac{Gm_1m_2(y_2-y_1)}{((x_2-x_1)^2+(y_2-y_1)^2)^{2/3}} + \frac{Gm_1m_3(y_3-y_1)}{\sqrt{((x_3-x_1)^2+(y_3-y_1)^2)^{2/3}}$$

$$dH/dx_2 = \frac{Gm_2m_1(x_1-x_2)}{((x_2-x_1)^2+(y_2-y_1)^2)^{2/3}} + \frac{Gm_2m_3(x_2-x_1)}{\sqrt{((x_3-x_1)^2+(y_3-y_1)^2)^{2/3}}$$

$$dH/dy_2 = \frac{Gm_2m_1(y_1-y_2)}{((x_2-x_1)^2+(y_2-y_1)^2)^{2/3}} + \frac{Gm_2m_3(y_2-y_1)}{\sqrt{((x_3-x_1)^2+(y_3-y_1)^2)^{2/3}}$$

Appendix B: Data

**Data for Solution:**

First Integration:

Asteroid:

Position x:404999999999999.938000, y:5760001612809.960940  
momentum: x:-0.000001 y:806399999.999989

Jupiter:

Position x:52668379430663086100.000000,  
y:4.21000117880000072000000000000000000000e+54  
momentum: x:0.000010 y:79935270000000009000000000000.000000

After Second Time Step:

Asteroid:

Position x:404999999999999.875000, y:11520003225609.841800  
momentum: x:-0.000013 y:806399999.999977

Jupiter:

Position x:105336008861326148000.000000,  
y:8.42000235760000143000000000000000000000e+54  
momentum: x:0.000010 y:79935270000000009000000000000.000000

After Third Time Step:

Asteroid:

Position x:404999999999999.750000, y:17280004838409.642600  
momentum: x:-0.000024 y:806399999.999966

Jupiter:

Position x:158003638291989234000.000000,  
y:1.26300035364000015000000000000000000000e+55  
momentum: x:0.000010 y:79935270000000009000000000000.000000

After Fourth Time Step:

Asteroid:

Position x:404999999999999.500000, y:23040006451209.359400  
momentum: x:-0.000035 y:806399999.999955

Jupiter:

Position x:210671267722652320000.000000,  
y:1.68400047152000029000000000000000000000e+55  
momentum: x:0.000010 y:79935270000000009000000000000.000000

After Fifth Time Step:

Asteroid:

Position x:404999999999999.250000, y:28800008064009.000000  
momentum: x:-0.000046 y:806399999.999944

Jupiter:

Position x:263338897153315406000.000000,  
y:2.10500058940000043000000000000000000000e+55

momentum: x:0.000010 y:79935270000000009000000000000.000000

After Sixth Time Step:

Asteroid:

Position x:40499999999998.875000, y:34560009676808.562500

momentum: x:-0.000058 y:806399999.999933

Jupiter:

Position x:316006526583978459000.000000,

y:2.52600070728000084000000000000000000000e+55

momentum: x:0.000010 y:79935270000000009000000000000.000000

After Seventh Time Step:

Asteroid:

Position x:40499999999998.375000, y:40320011289608.054700

momentum: x:-0.000069 y:806399999.999922

Jupiter:

Position x:368674156014641611000.000000,

y:2.94700082516000098000000000000000000000e+55

momentum: x:0.000010 y:79935270000000009000000000000.000000

After Eighth Time Step:

Asteroid:

Position x:40499999999997.750000, y:46080012902407.453100

momentum: x:-0.000080 y:806399999.999910

Jupiter:

Position x:421341785445304762000.000000,

y:3.36800094304000112000000000000000000000e+55

momentum: x:0.000010 y:79935270000000009000000000000.000000

After Ninth Time Step:

Asteroid:

Position x:40499999999997.062000, y:51840014515206.773400

momentum: x:-0.000091 y:806399999.999899

Jupiter:

Position x:474009414875967914000.000000,

y:3.78900106092000126000000000000000000000e+55

Momentum: x: 0.000010 y: 79935270000000009000000000000.000000

**Data for Isolated Asteroid:**

Asteroid:

Position x:405000000000000.062000, y:5500001540009.960940

momentum: x:0.000001 y:643499999.999991

Asteroid:

Position x:405000000000000.125000, y:11000003080009.843800

momentum: x:-0.000009 y:643499999.999981

Asteroid:

Position x:40500000000000.062000, y:16500004620009.644500  
momentum: x:-0.000018 y:643499999.999972

Asteroid:

Position x:40499999999999.938000, y:22000006160009.359400  
momentum: x:-0.000028 y:643499999.999962

Asteroid:

Position x:40499999999999.750000, y:27500007700008.992200  
momentum: x:-0.000037 y:643499999.999953

Asteroid:

Position x:40499999999999.375000, y:33000009240008.546900  
momentum: x:-0.000047 y:643499999.999943

Asteroid:

Position x:40499999999998.938000, y:38500010780008.023400  
momentum: x:-0.000056 y:643499999.999933

Asteroid:

Position x:40499999999998.375000, y:44000012320007.414100  
momentum: x:-0.000065 y:643499999.999924:

### **Data For Isolated Jupiter**

Jupiter:

Position x:52668379430663086100.000000,  
y:4.000001120000000210000000000000000000000000e+54  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:

Position x:105336008861326148000.000000,  
y:8.000002240000000420000000000000000000000000e+54  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:

Position x:158003638291989234000.000000,  
y:1.200000336000000000000000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:

Position x:210671267722652320000.000000,  
y:1.600000448000000036000000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:

Position x:263338897153315406000.000000,  
y:2.000000560000000085000000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:316006526583978459000.000000,  
y:2.40000067200000135000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:368674156014641611000.000000,  
y:2.80000078400000185000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:421341785445304762000.000000,  
y:3.20000089600000235000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:474009414875967914000.000000,  
y:3.60000100800000284000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:526677044306631066000.000000,  
y:4.00000112000000334000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:579344673737294217000.000000,  
y:4.40000123200000384000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:632012303167957238000.000000,  
y:4.80000134400000434000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:684679932598620258000.000000,  
y:5.20000145600000538000000000000000000000e+55  
momentum: x:0.000010 y:75948000000000001500000000000.000000

Jupiter:  
Position x:737347562029283279000.000000,  
y:5.60000156800000696000000000000000000000e+55  
Momentum: x: 0.000010 y: 75948000000000001500000000000.000000

## Appendix C: Code

```
/* File: sympl3.cpp
* -----
* This program integrates the derivatives Hamiltonian providing values for the
* position and momentum of three bodies. Symplectic integration is used. The P
* values are momentum, the Q are position.
* Variables : Body 1, Asteroid          Body 2, Jupiter
* momentum x   p1          p3
*              momentum y       p2          p4
*              position x   q1          q3
*              position y       q2          q4
*/

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

/*Function Prototypes*/
void symplectic(int run);
void introduce (void);
double dhdp1(double p1);
double dhdp2(double p2);
double dhdp3(double p3);
double dhdp4(double p4);
double dhdq1(double q1,double q2,double q3,double q4);
double dhdq2(double q1,double q2,double q3,double q4);
double dhdq3(double q1,double q2,double q3,double q4);
double dhdq4(double q1,double q2,double q3,double q4);

/*Mass Definitions*/
#define m1 1.17*pow(10,6)
#define m2 1.8987*pow(10,27)
#define m3 1.98892*pow(10,30)
/*Gravitational Constant*/
#define G 6.673*pow(10,-11)
/* Position of greatest Mass*/
#define m3x 0
#define m3y 0

/* Global Variable Declarations*/
double q1, q2, q3, q4, p1, p2, p3, p4;

/* PROGRAM */

void main ()
```

```

{
    int i;
    introduce ();
    /* initializing values*/
    q1=4.05*pow(10,14);
    q2=10 ;
    q3=7.5*pow(10,14);
    q4=10 ;
    p1=.00001;
    p2=550*m1;
    p3=.00001;
    p4= 400*m2;
    /*integrating*/
    for (i=1;i<15;i++)
    {
        symplectic (i);
    }
}

/*Differential Equations For Hamiltonian*/
double dhdp1(double p1)
{
    return((double)p1/m1);
}

double dhdp2(double p2)
{
    return((double)p2/m1);
}

double dhdp3(double p3)
{
    return((double)p3/m2);
}

double dhdp4(double p4)
{
    return((double)p4/m2);
}

double dhdq1(double q1,double q2,double q3,double q4)
{
    double  xq12, yq12, xq13, yq13;
    xq12= q1-q3;
    yq12= q2-q4;
    xq13= q1-m3x;
}

```

```

        yq13= q2-m3y;
        return((double) (G*m1*m2*xq12*pow(pow(sqrt(xq12*xq12+yq12*yq12),3),-
1)))+
            G*m1*m3*xq13*pow(pow(sqrt(xq13*xq13+yq13*yq13), 3), -1));
    }

double dhdq2(double q1,double q2,double q3,double q4)
{
    double xq12, yq12, xq13, yq13;
    xq12= q1-q3;
    yq12= q2-q4;
    xq13= q1-m3x;
    yq13= q2-m3y;
    return((double) (G*m1*m2*yq12*pow(pow(sqrt(xq12*xq12+yq12*yq12), 3), -
1)))+
        G*m1*m3*xq13*pow(pow(sqrt(xq13*xq13+yq13*yq13), 3), -1));
}

double dhdq3(double q1,double q2,double q3,double q4)
{
    double xq21, yq21, xq23, yq23;
    xq21= q3-q1;
    yq21= q4-q2;
    xq23= q3-m3x;
    yq23= q4-m3y;
    return((double) (G*m2*m1*xq21*pow(pow(sqrt(xq21*xq21+yq21*yq21), 3), -
1)))+
        G*m3*m2*xq23*pow(pow(sqrt(xq23*xq23+yq23*yq23), 3), -1));
}

double dhdq4(double q1,double q2,double q3,double q4)
{
    double xq21, yq21, xq23, yq23;
    xq21= q3-q1;
    yq21= q4-q2;
    xq23= q3-m3x;
    yq23= q4-m3y;
    return((double) (G*m2*m1*yq21*pow(pow(sqrt(xq21*xq21+yq21*yq21), 3), -
1)))+
        G*m2*m3*xq23*pow(pow(sqrt(xq23*xq23+yq23*yq23), 3), -1));;
}

/* Integration Routine */
void symplectic(int run)
{

```

```

double c1,c2,c3,c4,c5,c6,c7,c8,d1,d2,d3,d4,d5,d6,d7,d8;
double q1dum,q2dum,q3dum;
double p1dum,p2dum,p3dum;
double delta;
FILE *outfile;
/* To save data to be plotted to a different file than 3bodydata.txt, simply
*change name within the quotes*/
outfile= fopen ("3bodydata.txt", "a+");
if (outfile==NULL) printf ("error, bad file");
delta= pow(10,-2); /*Time step*/

/* Definitions of constants*/
d1=0.78451361;
d2=0.23557321;
d3=-1.17767984;
d4=1.31518632;
d5=-1.17767984;
d6=0.23557321;
d7=0.78451361;
d8=0.0;
c1=0.5*d1;
c2=0.5*(d1+d2);
c3=0.5*(d2+d3);
c4=0.5*(d3+d4);
c5=c4;
c6=c3;
c7=c2;
c8=c1;

q1dum =q1+c1*delta*dhdp1(p1);
q2dum =q2+c1*delta*dhdp2(p2);
q3dum =q3+c1*delta*dhdp3(p3);
q4 =q4+c1*delta*dhdp4(p4);
q1=q1dum;
q2=q2dum;
q3=q3dum;

p1dum =p1-d1*delta*dhdq1(q1,q2,q3,q4);
p2dum =p2-d1*delta*dhdq2(q1,q2,q3,q4);
p3dum =p3-d1*delta*dhdq3(q1,q2,q3,q4);
p4 =p4-d1*delta*dhdq4(q1,q2,q3,q4);
p1=p1dum;
p2=p2dum;
p3=p3dum;

```

$q1dum = q1 + c2 * \text{delta} * \text{dhdp1}(p1);$   
 $q2dum = q2 + c2 * \text{delta} * \text{dhdp2}(p2);$   
 $q3dum = q3 + c2 * \text{delta} * \text{dhdp3}(p3);$   
 $q4 = q4 + c2 * \text{delta} * \text{dhdp4}(p4);$   
 $q1 = q1dum;$   
 $q2 = q2dum;$   
 $q3 = q3dum;$

$p1dum = p1 - d2 * \text{delta} * \text{dhdq1}(q1, q2, q3, q4);$   
 $p2dum = p2 - d2 * \text{delta} * \text{dhdq2}(q1, q2, q3, q4);$   
 $p3dum = p3 - d2 * \text{delta} * \text{dhdq3}(q1, q2, q3, q4);$   
 $p4 = p4 - d2 * \text{delta} * \text{dhdq4}(q1, q2, q3, q4);$   
 $p1 = p1dum;$   
 $p2 = p2dum;$   
 $p3 = p3dum;$

$q1dum = q1 + c3 * \text{delta} * \text{dhdp1}(p1);$   
 $q2dum = q2 + c3 * \text{delta} * \text{dhdp2}(p2);$   
 $q3dum = q3 + c3 * \text{delta} * \text{dhdp3}(p3);$   
 $q4 = q4 + c3 * \text{delta} * \text{dhdp4}(p4);$   
 $q1 = q1dum;$   
 $q2 = q2dum;$   
 $q3 = q3dum;$

$p1dum = p1 - d3 * \text{delta} * \text{dhdq1}(q1, q2, q3, q4);$   
 $p2dum = p2 - d3 * \text{delta} * \text{dhdq2}(q1, q2, q3, q4);$   
 $p3dum = p3 - d3 * \text{delta} * \text{dhdq3}(q1, q2, q3, q4);$   
 $p4 = p4 - d3 * \text{delta} * \text{dhdq4}(q1, q2, q3, q4);$   
 $p1 = p1dum;$   
 $p2 = p2dum;$   
 $p3 = p3dum;$

$q1dum = q1 + c4 * \text{delta} * \text{dhdp1}(p1);$   
 $q2dum = q2 + c4 * \text{delta} * \text{dhdp2}(p2);$   
 $q3dum = q3 + c4 * \text{delta} * \text{dhdp3}(p3);$   
 $q4 = q4 + c4 * \text{delta} * \text{dhdp4}(p4);$   
 $q1 = q1dum;$   
 $q2 = q2dum;$   
 $q3 = q3dum;$

$p1dum = p1 - d4 * \text{delta} * \text{dhdq1}(q1, q2, q3, q4);$   
 $p2dum = p2 - d4 * \text{delta} * \text{dhdq2}(q1, q2, q3, q4);$   
 $p3dum = p3 - d4 * \text{delta} * \text{dhdq3}(q1, q2, q3, q4);$   
 $p4 = p4 - d4 * \text{delta} * \text{dhdq4}(q1, q2, q3, q4);$   
 $p1 = p1dum;$   
 $p2 = p2dum;$

$p3=p3dum;$

$q1dum =q1+c5*\delta*dhdp1(p1);$   
 $q2dum =q2+c5*\delta*dhdp2(p2);$   
 $q3dum =q3+c5*\delta*dhdp3(p3);$   
 $q4 =q4+c5*\delta*dhdp4(p4);$   
 $q1=q1dum;$   
 $q2=q2dum;$   
 $q3=q3dum;$

$p1dum =p1-d5*\delta*dhdq1(q1,q2,q3,q4);$   
 $p2dum =p2-d5*\delta*dhdq2(q1,q2,q3,q4);$   
 $p3dum =p3-d5*\delta*dhdq3(q1,q2,q3,q4);$   
 $p4 =p4-d5*\delta*dhdq4(q1,q2,q3,q4);$   
 $p1=p1dum;$   
 $p2=p2dum;$   
 $p3=p3dum;$

$q1dum =q1+c6*\delta*dhdp1(p1);$   
 $q2dum =q2+c6*\delta*dhdp2(p2);$   
 $q3dum =q3+c6*\delta*dhdp3(p3);$   
 $q4 =q4+c6*\delta*dhdp4(p4);$   
 $q1=q1dum;$   
 $q2=q2dum;$   
 $q3=q3dum;$

$p1dum =p1-d6*\delta*dhdq1(q1,q2,q3,q4);$   
 $p2dum =p2-d6*\delta*dhdq2(q1,q2,q3,q4);$   
 $p3dum =p3-d6*\delta*dhdq3(q1,q2,q3,q4);$   
 $p4 =p4-d6*\delta*dhdq4(q1,q2,q3,q4);$   
 $p1=p1dum;$   
 $p2=p2dum;$   
 $p3=p3dum;$

$q1dum =q1+c7*\delta*dhdp1(p1);$   
 $q2dum =q2+c7*\delta*dhdp2(p2);$   
 $q3dum =q3+c7*\delta*dhdp3(p3);$   
 $q4 =q4+c7*\delta*dhdp4(p4);$   
 $q1=q1dum;$   
 $q2=q2dum;$   
 $q3=q3dum;$

$p1dum =p1-d7*\delta*dhdq1(q1,q2,q3,q4);$   
 $p2dum =p2-d7*\delta*dhdq2(q1,q2,q3,q4);$   
 $p3dum =p3-d7*\delta*dhdq3(q1,q2,q3,q4);$   
 $p4 =p4-d7*\delta*dhdq4(q1,q2,q3,q4);$

```

p1=p1dum;
p2=p2dum;
p3=p3dum;

q1dum =q1+c8*delta*dhdp1(p1);
q2dum =q2+c8*delta*dhdp2(p2);
q3dum =q3+c8*delta*dhdp3(p3);
q4 =q4+c8*delta*dhdp4(p4);
q1=q1dum;
q2=q2dum;
q3=q3dum;

p1dum =p1-d8*delta*dhdq1(q1,q2,q3,q4);
p2dum =p2-d8*delta*dhdq2(q1,q2,q3,q4);
p3dum =p3-d8*delta*dhdq3(q1,q2,q3,q4);
p4 =p4-d8*delta*dhdq4(q1,q2,q3,q4);
p1=p1dum;
p2=p2dum;
p3=p3dum;

/*displaying values*/
printf(" Run %d:\n", run);
printf("Asteroid:\n Position x:%lf, y:%lf\n momentum: x:%lf y:%lf\n", q1, q2,
      p1, p2);
printf("Jupiter:\n Position x:%lf, y:%lf\n momentum: x:%lf y:%lf\n\n", q3, q4,
      p3, p4);
/*Saving positions of Asteroid to a file to be plotted, division by constant
*necessary to make data plotable, using gnuplot*/
fprintf (outfile, "%lf %lf\n", q1/pow(10,11), q2/pow(10,9));

    }
/*Function to introduce program to user.*/
void introduce (void)
{
    printf ("Description of the orbits of two bodies about another of much greater
mass.\n");
}

```

Bibliography of Sources and Softwear:

Borland Turbo C++. Version 4.5. Copyright 1995.

G N U P L O T. MS-Windows version 3.5 patchlevel 3.50.1.17, 27 Aug 93  
last modified Fri Aug 27 05:21:33 GMT 1993 Copyright(C) 1986 - 1993 Colin  
Kelley, Thomas

Lorenz, Edward N. The Essence of Chaos. The University of Washington Press. Seattle.  
1993.

Sussman, and Wisdom. Numerical Evidence that the motion of Pluto is chaotic.  
[http://adabs.harvard.edu/cgi-bin/nph-bib\\_query?bibcode=1988Sci...241..433S&db\\_key=AST&high=3f8dbb998322818](http://adabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1988Sci...241..433S&db_key=AST&high=3f8dbb998322818)