Numerical Integration of the n-Body Problem

Harrison Brown*

University of Colorado Boulder, Boulder, Colorado, 80309, USA

This paper is intended to present a numerical integrator of the n-body problem created in the GNU Octave language. This paper begins with an overview of the equations behind the n-body problem and a demonstration of how they are written in the case where n=3, transitions into a demonstration of the n-body solver for some theoretical systems of 2, 3, and 4 bodies, and concludes with a demonstration of a real-world application of the solver by modeling the Earth, Sun, Luna system.

Nomenclature

$\vec{f_i}$	Force, N
G	Universal Gravitation Constant, $6.673 \times 10^{-11} \frac{Nm^2}{kg^2}$
m_i	Mass, kg
$\vec{r_i}$	Position Vector, m
r_{ij}	Radius, m
$\dot{\vec{r}}_i$	Velocity Vector, $\frac{m}{s}$
$\ddot{\vec{r}}_i$	Acceleration Vector, $\frac{m}{s^2}$
$\vec{y}_k$	Substitution Vector
y_k	Substitution Scalar
G_A	Astronomical Universal Gravitation Constant, $1.18555535802194 \times 10^{-4} \frac{AstronomicalUnit^3}{(Year^2)(EarthMass)}$
Subscript	
i	Body Number
j	Body Number
k	Substitution Index

I. Introduction

This paper intends to present an n-body solver developed in GNU Octave. Starting with an overview of the mathematics involved in the n-body problem, the paper will then show 2, 3, and 4-body solutions for bodies of the same mass as a demonstration of the capabilities of the solver. The paper will conclude with an example of a real-world application of the solver, in the form of a model of the Earth, Sun, Luna system developed using the solver.

II. The n-Body Mathematical Basis

This section provides the equations behind the n-body problem and a demonstration of how to rewrite the 3-body case for solving via a technique such as Octave's ode45.

*Graduate Student, Department of Aerospace Engineering Sciences, University of Colorado Boulder.

II.A. General n-Body Equations

It is known from Newton's Law of Gravitation that

$$\vec{f}_i = G\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{m_i m_j}{r_{ij}^3}(\vec{r}_j - \vec{r}_i) \tag{1}$$

where G is the universal gravitation constant, m_i and m_j are the two masses involved, and the remaining components are described by Eq. (2) and Eq. (4), with Eq. (3) describing the velocity.

$$\vec{r}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix} \tag{2}$$

$$\dot{\vec{r}}_i = \begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \\ \dot{x}_{i3} \end{bmatrix} \tag{3}$$

$$r_{ij} = |\vec{r}_j - \vec{r}_i| \tag{4}$$

Because

$$\vec{f}_i = m_i \ddot{\vec{r}}_i \tag{5}$$

it is possible to transform Newton's Law of Gravitation into a description of the i^{th} body's acceleration as Eq. (6).

$$\ddot{\vec{r}}_i = G\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{m_j}{r_{ij}^3}(\vec{r}_j - \vec{r}_i) \tag{6}$$

For a system of n bodies interacting only via gravity it is thus possible to take Eq. (6) from i = 1 to n and integrate to obtain the motion of each body due to the effects of the initial conditions and the other bodies.

II.B. The 3-Body Example

To integrate Eq. (6) over 3 bodies using ode45 (or a similar technique), it is necessary to rewrite the second order differential equations as a set of first order equations. If we start by saying that

$$\vec{y}_1 = \vec{r}_1 \tag{7}$$
$$\vec{y}_2 = \vec{r}_2 \tag{8}$$
$$\vec{y}_3 = \vec{r}_3 \tag{9}$$
$$\vec{y}_4 = \dot{\vec{r}}_1 \tag{10}$$
$$\vec{y}_5 = \dot{\vec{r}}_2 \tag{11}$$
$$\vec{y}_6 = \dot{\vec{r}}_3 \tag{12}$$

then the substitution needed to integrate the equations is created. By taking advantage of the fact that the x_1, x_2, and x_3 components of each equation are independent of each other, it is possible to build the set of 18 first-order equations that ode45 will be given to solve the 3-body problem.

Writing out the 18 first-order equations associated with the 3-body problem gives the velocities as

$$\dot{y}_{11} = y_{41} \tag{13}$$
$$\dot{y}_{12} = y_{42} \tag{14}$$
$$\dot{y}_{13} = y_{43} \tag{15}$$
$$\dot{y}_{21} = y_{51} \tag{16}$$
$$\dot{y}_{22} = y_{52} \tag{17}$$
$$\dot{y}_{23} = y_{53} \tag{18}$$
$$\dot{y}_{31} = y_{61} \tag{19}$$
$$\dot{y}_{32} = y_{62} \tag{20}$$
$$\dot{y}_{33} = y_{63} \tag{21}$$

and the accelerations as

$$\dot{y}_{41} = \ddot{y}_{11} = G(\frac{m_2}{r_{12}^3}(x_{21} - x_{11}) + \frac{m_3}{r_{13}^3}(x_{31} - x_{11})) \tag{22}$$

$$\dot{y}_{42} = \ddot{y}_{12} = G(\frac{m_2}{r_{12}^3}(x_{22} - x_{12}) + \frac{m_3}{r_{13}^3}(x_{32} - x_{12})) \tag{23}$$

$$\dot{y}_{43} = \ddot{y}_{13} = G(\frac{m_2}{r_{12}^3}(x_{23} - x_{13}) + \frac{m_3}{r_{13}^3}(x_{33} - x_{13})) \tag{24}$$

$$\dot{y}_{51} = \ddot{y}_{21} = G(\frac{m_1}{r_{21}^3}(x_{11} - x_{21}) + \frac{m_3}{r_{23}^3}(x_{31} - x_{21})) \tag{25}$$

$$\dot{y}_{52} = \ddot{y}_{22} = G(\frac{m_1}{r_{21}^3}(x_{12} - x_{22}) + \frac{m_3}{r_{23}^3}(x_{32} - x_{22})) \tag{26}$$

$$\dot{y}_{53} = \ddot{y}_{23} = G(\frac{m_1}{r_{21}^3}(x_{13} - x_{23}) + \frac{m_3}{r_{23}^3}(x_{33} - x_{23})) \tag{27}$$

$$\dot{y}_{61} = \ddot{y}_{31} = G(\frac{m_1}{r_{31}^3}(x_{11} - x_{31}) + \frac{m_2}{r_{32}^3}(x_{21} - x_{31})) \tag{28}$$

$$\dot{y}_{62} = \ddot{y}_{32} = G(\frac{m_1}{r_{31}^3}(x_{12} - x_{32}) + \frac{m_2}{r_{32}^3}(x_{22} - x_{32})) \tag{29}$$

$$\dot{y}_{63} = \ddot{y}_{33} = G(\frac{m_1}{r_{31}^3}(x_{13} - x_{33}) + \frac{m_2}{r_{32}^3}(x_{23} - x_{33})) \tag{30}$$

which is a set of 18 equations that can be integrated by ode45 and similar solvers to obtain the desired solution to the 3-body problem.

III. The 2, 3, and 4-Body Universes

This section examines universes containing n particles of equal mass, starting from n=2, moving to n=3, and finishing with an examination of n=4. All of the problems here, along with the Sun, Earth, Luna system presented later in this paper, depend upon the n-Body solver provided in appendix subsection nBody.m.

III.A. A 2-Body Universe

The first universe to explore with the n-body solver is a 2-body universe with both particles having equal mass. The goal is to get both particles moving on a circular orbit around their shared barycenter. The mass is set at an arbitrary 2 kg for each body and they are placed at ± 1 m. Using the known solution to the 2-body problem the needed starting velocity was set as $\pm 5.775 \times 10^{-6}$ $\frac{m}{s}$. Running the script designed to generate this model (found in appendix subsection twoBodyScript.m) for one orbital period generated figure 1. This figure shows success in creating the desired model - both bodies are on the same orbital track

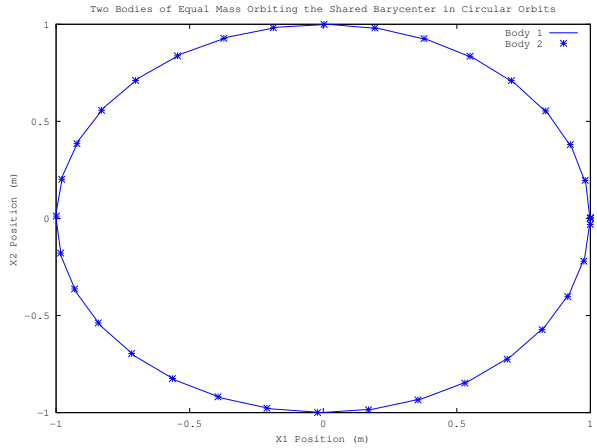

Figure 1. Two bodies of equal mass orbiting their shared barycenter on the same orbital track.

about the barycenter and the orbit is circular.

III.B. A 3-Body Universe

The next universe explored with the n-body solver is one in which the system from the 2-body universe is altered by the addition of a third body of the same mass placed at the barycenter and having no velocity. No other changes are made to the system. Running the script designed to generate this model (found in appendix subsection threeBodyScript.m) for one orbital period generated figure 2. Looking at the result,

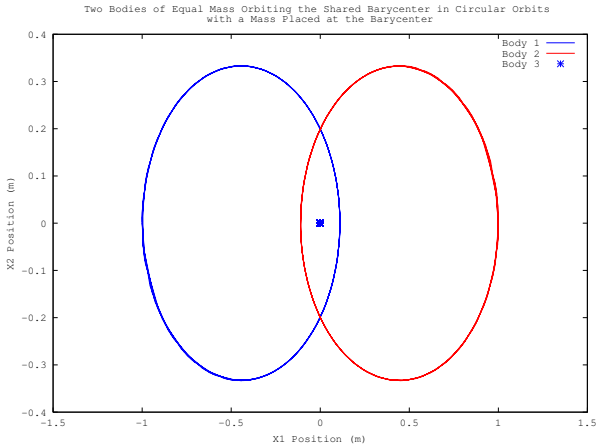

Figure 2. Two bodies of equal mass orbiting their shared barycenter with a third body placed at the barycenter.

it is observed that the addition of the third mass has caused the orbits of the other two masses to become elliptical, so that they no longer move on the same track. The third mass, however, is stationary due to the symmetry of this universe.

III.C. A 4-Body Universe

The final arbitrary universe explored with the n-body solver was a 4-body universe created by taking the system from the 2-body universe and combining it with itself rotated into the x_1 - x_3 plane. Running the script designed to generate this model (found in appendix subsection fourBodyScript.m) for one orbital period of the 2-body system generated figure 3. Looking at this result, it can be seen that without modifications,

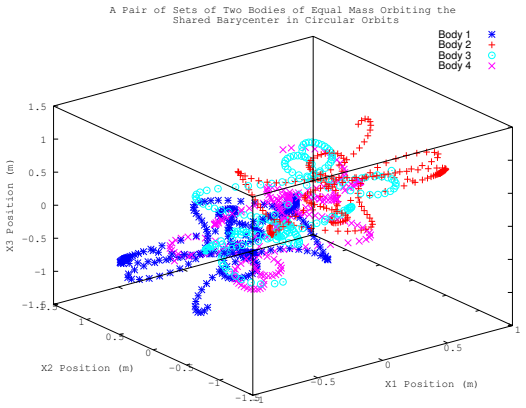

Figure 3. A pair of the 2-body systems oriented at 90° to each other.

the addition of two new moving bodies causes the motion of the system to differ greatly from the 2-body system from which it is derived.

IV. The Sun, Earth, Luna System

Using the n-body simulator demonstrated previously, the Sun, Earth, Luna system can be modeled as a 3-body system where only the gravitational effects are considered. The average orbital radii for this 3-body problem along with the other relevant pieces of information are in table 1 which is taken from Vallado.[1] Note the use here of the Astronomical Unit (AU), which is 149,597,870 km and the Earth Mass, which is

Table 1. Physical Details of the Sun, Earth, Luna System

Body	Average Orbit Radii (AU)	Mass (Earth Masses)	Average Velocity($\frac{AU}{yr}$)
Sun	0	332946	0
Earth	1	1	6.286156439
Luna	0.0025700	0.012303192	0.2148058584

5.9742×10^{24} kg.[1] In order to reduce the differences in magnitude on the Solar System scale, the tropical year (yr) is employed as the unit of time for this problem (1 tropical year is 3.15569×10^7).[1] When operating using these units as the basic units (in place of the m-kg-s system) G_A is used in the calculations instead of G. Note also that the Luna averages are with respect to the Earth, not the Sun. With these initial conditions a script can be created (as seen in appendix subsection SELSystem.m) that integrates the equations to model the Sun, Earth, Luna system for a period of time - here chosen as one tropical year. Running the script generates the result seen in figure 4. Looking at this result, a few things demonstrate the success of the

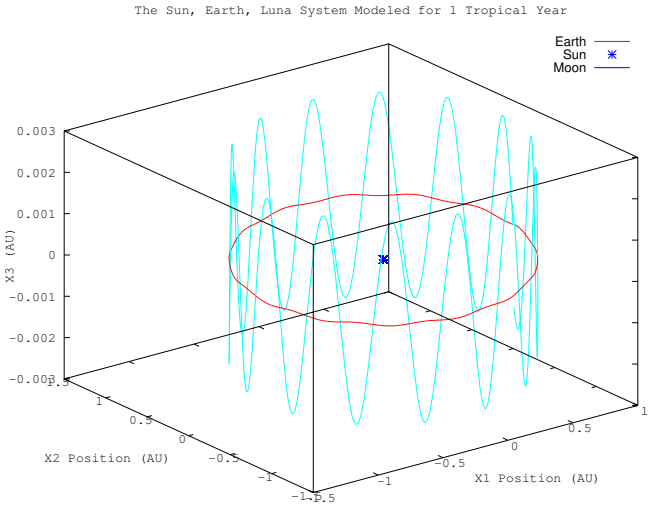

Figure 4. The simulated Sun, Earth, Luna system over the course of one year showing one orbit of the Earth about the Sun and a bit over 13 orbits of the Moon around the Earth.

script in creating a model of the Sun, Earth, Luna system. The Earth orbits the sun once over the course of one orbital period, returning to approximately the place it started. The Moon orbits the Earth a little over 13 times in the same period (expected, as the circular Moon orbit has a period a little below a 13^{th} of the period of the circular Earth orbit). The Sun moves negligibly, as is expected given the relative smallness of the Moon and the Earth compared to the Sun. Therefore, it is possible to use the n-Body equations and a selection of the system's physical properties to create a model of the Sun, Earth, Luna system.

Appendix

All scripts included here were written for GNU Octave version 3.6.4.

nBody.m

```
function [dx] = nBody(t,x,m,G)
# This is the function that the ode solver needs to solve the n-body problem
# Written for GNU Octave version 3.6.4

# Setting up things
n = length(m); # The number of bodies
dx = zeros(6*n,1); # Holder for the dx
half = 3*n;

# Creating the r matrix
# The idea employed here is that precomputing the repeated values of the
# radius magnitudes cubed saves computation later - and additionally the
# symmetry of the solution may be taken of advantage to reduce the number
# of computations used as well.
r = zeros(n,n);
for i = 1:n
for j = 1:n
if i == j || r(i,j) != 0
continue; # The radius between a body and itself is 0
else
r(i,j) = (((x((3*(j-1))+1)-x((3*(i-1))+1)).^2+(x((3*(j-1))+2)-
   x((3*(i-1))+2)).^2+(x(3*j)-x(3*i)).^2).^(3/2));
r(j,i) = r(i,j);
endif
endfor
endfor

# Filling in the velocities
for i = 1:half
dx(i) = x(half+i);
endfor

# Building the accelerations
for i = 1:n
temp = 0; # resetting the holder

# building the x-component
for j = 1:n
if j == i
continue; # The body has no gravitational effect on itself
else
temp = temp + (m(j)/(r(i,j)))*(x((3*(j-1))+1)-x((3*(i-1))+1));
endif
endfor
temp = G*temp; # adjusting for the gravitational constant
dx(half+(3*(i-1))+1) = temp; # setting the dx
temp = 0; # resetting the temp holder

# building the y-component
for j = 1:n
```

```
if j == i
continue; # The body has no gravitational effect on itself
else
temp = temp + (m(j)/(r(i,j)))*(x((3*(j-1))+2)-x((3*(i-1))+2));
endif
endfor
temp = G*temp; # adjusting for the gravitational constant
dx(half+(3*(i-1))+2) = temp; # setting the dx
temp = 0; # resetting the temp holder

# building the z-component
for j = 1:n
if j == i
continue; # The body has no gravitational effect on itself
else
temp = temp + (m(j)/(r(i,j)))*(x(3*j)-x(3*i));
endif
endfor
temp = G*temp; # adjusting for the gravitational constant
dx(half+(3*i)) = temp; # setting the dx
endfor
endfunction
```

twoBodyScript.m

```
# This is a script that simulates two bodies of the same mass orbiting each
# other.

format long;
G = 6.673e-11; # Universal gravitation constant
m = [2, 2]; # Setting both masses as 2 kg
time = [0 1087763]; # running for 1 period worth of seconds
x0 = [-1;0;0;1;0;0;0;-5.775e-6;0;0;5.775e-6;0];
# Starting positions and velocities
[t,x] = ode45(@nBody,time,x0,m,G);

figure();
hold on;
plot(x(:,1),x(:,2));
plot(x(:,4),x(:,5),'*');
```

threeBodyScript.m

```
# This is a script that simulates two bodies of the same mass orbiting each
# other, with a third mass at the barycenter.

format long;
G = 6.673e-11; # Universal gravitation constant
m = [2, 2, 2]; # Setting the masses as 2 kg
time = [0 1087763]; # running for 1 period worth of seconds
x0 = [-1;0;0;1;0;0;0;0;0;0;-5.775e-6;0;0;5.775e-6;0;0;0;0];
# Starting positions and velocities
[t,x] = ode45(@nBody,time,x0,m,G);

figure();
hold on;
```

```
plot(x(:,1),x(:,2));
plot(x(:,4),x(:,5),'r');
plot(x(:,7),x(:,8),'*');
```

fourBodyScript.m

```
# This is a script that simulates the interaction between a pair of the
# system created in the 2-body universe.

format long;
G = 6.673e-11; # Universal gravitation constant
m = [2, 2, 2, 2]; # Setting the masses as 2 kg
time = [0 1087763]; # running for 1 period worth of seconds
x0 = [-1;0;0;1;0;0;0;0;-1;0;0;1;0;-5.775e-6;0;0;5.775e-6;0;-5.775e-6;0;0;5.775e-6;0;0];
# Starting positions and velocities
[t,x] = ode45(@nBody,time,x0,m,G);

figure();
hold on;
plot3(x(:,1),x(:,2),x(:,3),'b');
plot3(x(:,4),x(:,5),x(:,6),'r');
plot3(x(:,7),x(:,8),x(:,9),'c');
plot3(x(:,10),x(:,11),x(:,12),'m');
```

SELSystem.m

```
# This is a script that simulates the Sun, Earth, Luna system over the
# period of one tropical year
G = 1.18555535802194e-04; # Gravitational Constant in astronomical units
time = [0 1]; # From 0 to 1 year
x0 = [1;0;0;0;0;0;0.99743;0;0;0;6.286156439;0;0;0;0;0;6.286156439;0.2148058584];
# Positions of the Earth-Sun-Moon, then the Velocities of the Earth-Sun-Moon
# AU and (AU/yr)
m = [1;332946;0.012303192]; # The masses of the Earth, Sun, and Moon (EM)
[t,x] = ode45(@nBody,time,x0,m,G);

# Plotting Functions
figure
hold on;
plot3(x(:,1),x(:,2),x(:,3),'r-'); # Earth
plot3(x(:,4),x(:,5),x(:,6),'*'); # Sun
plot3(x(:,7),x(:,8),x(:,9),'b-'); # Moon
```

Acknowledgments

The author of this paper would like to thank Dr. Brian Argrow for introducing this topic and for his help in developing the n-Body solver, as well his suggestions for further work on this problem and its applications.

References

[1]Vallado, D.A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press, Hawthorne, CA, 2013.