Ballistic Entry: Monte Carlo Simulation

Justin Gaumer, Chirag Gokani, Morgan Yost

December 1, 2019

1 Abstract

The entry of objects into Earth's atmosphere is a significant problem whose solution dictates our understanding of solar system formation and the structure and evolution of the atmosphere. In addition to being a natural phenomenon that occurs on the order of 10^2 times every day (in the form of meteorites) [2], ballistic entry is a serious problem in aerospace engineering as retrievable satellite missions in both scientific and commercial worlds grow in sophistication.

This report provides insights into both contexts as we estimate the landing zone of an object entering the Earth's atmosphere given some estimate of initial conditions.[1] This is precisely the hypothetical problem of interest for engineers wanting to know where their retrievable satellite would land, or for astronomers interested in knowing the zone of impact due to an approaching asteroid. It is reasonable to assume that in the former case, the mass of the satellite is precisely known, and that in the latter case, the mass of the asteroid is estimated with low uncertainty using existing data from solar system models [1]. Further, in both hypothetical cases, the scientists have a few measurements of the object's position, providing a set of points which can be numerically differentiated to estimate the object's velocity and flight angle.

We use a MATLAB-based Monte Carlo simulation to calculate an object's trajectory as it enters Earth's atmosphere at an estimated initial velocity and flight angle, thereby arriving at a corresponding set of landing positions. The object's "landing zone" is the range of these positions. Our results corroborate those found in well-reputed sources [5].

[1] The initial conditions are namely the initial velocity and flight angle.

2 Introduction

The Monte Carlo simulation is a numerical method used to understand the impact of distinct variables on a system. We are often interested in the uncertainty of those variables and the way this uncertainty effects the probability of arriving at particular solutions. Forecasting models are developed by running through a set of commands thousands of times, each time randomly selecting a new value within a certain bound. The simulation results in thousands of solutions, each based on a set of random inputs. These results are then used to describe the probability of reaching particular solutions.

2.1 Approximations

To simplify the physical complexity of an object entering the atmosphere, we treat its trajectory as ballistic.[2] To facilitate this assumption, we choose an object that generates a relatively little lift: a spinning spherical mass [7].[3] This geometry allows us to set a the lift-to-drag coefficient, denoted `LD` in our code, to a value close to 0 (we chose 10^{-3}).

While our model stands for any spherical macroscopic particle of density ρ and volume V such that mass $m = \rho V << M_{\text{Earth}}$ (i.e., having a negligible moment of inertia), we choose the object's density ρ (notated `density`) to be that of a typical iron meteorite for their relative abundance. [4] Thus we choose $\rho = 7873$ kg/m^3 and radius (denoted `r`) $r = 2$ meters. We assume that these parameters are constant for the duration of its trajectory.[5] For ease of use, we employ the following user-defined function to calculate the object's mass given its density and radius:[6]

```
function m = mass_entry(density,r)
V = (4/3)*pi*r^3;
m = density*V;
end
```

[2]Note the distinction between "ballistic reentry" and "aerodynamic reentry." The former accounts for only gravitational and frictional forces, while the latter also considers the effects of aerodynamical lift.

[3]It is reasonable to assume that the object accumulates spin as there exist density fluctuations in the atmosphere.

[4]Iron meteorites only account for 3-5% of all meteorites catalogued [3][8].However, we feel justified using their density for the objects we discuss in this report. The user of the code should feel free to experiment with different values of `density`.

[5]This is one of the biggest assumptions we make in the project; we comment on this in the conclusion.

[6]$V = \frac{4}{3}\pi r^3$ since the object is spherical.

We also write a the following user-defined function to calculate the maximal cross-sectional area of the spherical object:[7]

```
function S = entry_S(r)
S = pi*r^2;
end
```

Although Earth is realistically in the shape of an oblate spheroid, we assume a perfectly spherical shape, setting the Earth's mean radius of 6371000 meters as its spherically approximated radius (denoted `R`) [9]. Since we set the initial height of the object to be only 1.9% of an Earth radius beyond the Earth's surface, we assume that the acceleration due to gravity does not change as a function of distance from the Earth's center and set $g = 9.81$ m/s uniformly.

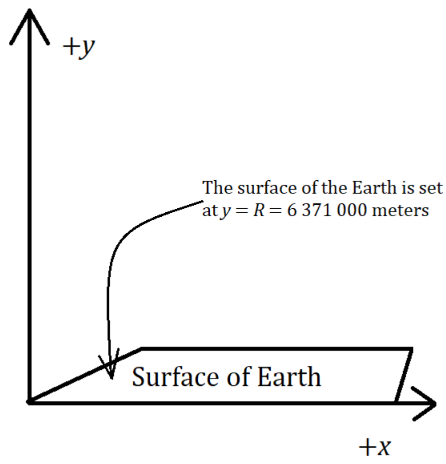
As discussed in the project proposal, atmospheric drag will be treated to be proportional to $v(t)^2\rho(r)$, where $\rho(r)$ (the atmospheric density) falls off exponentially as r grows, and where $v(t)^2$ is the square of the object's speed as a function of time. This is a major simplification of the structure of the Earth's atmosphere, but a more accurate $\rho(r)$ would require a piecewise definition that would add complexities to solving the equations of motion described in the following section. In defining the density of Earth's atmosphere, we set $\rho(R) \equiv \rho_0$ (denoted `p0`) equal to 1.225, the atmospheric density at sea level [9]. We also treat the atmosphere to be isothermal [10, 11], so the scale height parameter `H` is constant.

2.2 Coordinate System

To prevent considerations of noninertial reference frames, we assume that the object does not have any component of motion along the direction of the axis pointing into the page (see figure below). That is, the object is confined to the longitude over which it is released. This implies that the total trajectory time is much less than an hour (a time over which the Earth has rotated 0.26 radians). This assumption helps establish a lower limit to the object's initial velocity.

While we were considering using the relativistic $G = c = 1$ convention for ease of sight when writing the code, we will use SI units throughout the project to preserve the physically meaningful nature of the solutions.

[7] Of course, the maximal cross sectional area of a sphere of radius r is a circle of area πr^2



3 Discussion

The Monte Carlo simulation for this project only considers the uncertainty in initial velocity and initial entry angle. That is, we assume that all other parameters are as stated in the *Introduction*, and that we are confident in their values. The uncertainty of the initial velocity and initial entry angle values are determined by the ± error in the measurement.

3.1 Initial Conditions

The parameter `c` equals the number of sample trajectories run by the Monte Carlo simulation. A variety of sources, as well as some trial-and-error experimentation, resulted in the following ranges for the initial velocity and initial flight angle. Note that initial values that deviated significantly from this range "bounced off" the atmosphere. The authors invite the reader to change initial values so as to reproduce this phenomenon (one that is indeed observed in nature).

$$v_{min} = 16300 \text{ meters/second} \tag{1}$$
$$v_{max} = 17700 \text{ meters/second} \tag{2}$$
$$\theta_{min} = 0.26 \text{ radians} \tag{3}$$
$$\theta_{max} = 0.34 \text{ radians} \tag{4}$$

Hard-coded, the above statements are denoted:

```
vmin = 16300;
vmax = 17700;
Omin = .26;
Omax = .34;
```

At this juncture, the random parameterization of the initial velocity and flight angle is introduced. We emphatically note that an identical set of solutions would be achieved if we solved the equations of motion for every combination of incremental values of initial velocity & flight angle respectively. However, the Monte Carlo approach is significantly easier as it prevents us from having to deal with the combinatorics of matching every possible initial velocity value to every possible flight angle value.

To simulate the uncertainty in initial velocity and flight angle, the random parameters `vdist` and `Odist` are respectively defined by the following user-defined functions. We employ the `rand` function, where `rand(1,c)` generates a vector of length `c` consisting of random values between 0 and 1.

```
function vdist = vdist(vmin,vmax,c)
rav = rand(1,c);
vdist = rav*(vmax-vmin)+vmin;
end
```

```
function Odist = Odist(Omin,Omax,c)
raO = rand(1,c);
Odist = raO*(Omax-Omin)+Omin;
end
```

We set the initial position to be at the point $(x_i, y_i) = (0, 121000 + R)$ (respectively denoted `xi` and `yi`. We also define an anonymous function for the magnitude of the displacement vector, $d = \sqrt{x^2 + y^2}$, where $\vec{d}$ points from center of earth to object's center of gravity:

```
d = @(x,y) sqrt(x^2+y^2)
```

From the initial condition $x_i = 0$, the initial distance from the center of earth is $d_i = y_i$.

3.2 Equations of Motion

We now consider the equations of motion, common throughout the literature [10, 11], which are treated as anonymous functions of the variables `vdist`, `Odist`, and `d`. Note that $B1 \equiv \frac{m}{\mathrm{Cd}*S}$ where Cd is the drag coefficient $\equiv 0.47$.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v\cos\theta \tag{5}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -v\sin\theta \tag{6}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{1}{2}\rho_0 \exp\left(-(d-R)/H\right) v^2 \frac{1}{\mathrm{B1}} + g\sin\theta \tag{7}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\frac{1}{2}\rho_0 \exp\left(-(d-R)/H\right) v \frac{\mathrm{LD}}{\mathrm{B1}} + \frac{g\cos\theta}{v} - \frac{v\cos\theta}{R} \tag{8}$$

3.3 Numerical Solution

The function `RK4 System NumMethProj MC` is a generalization of the 4th order Runge-Kutta method we learned in class. It is generalized to a system of ODEs in a manner similar to how Euler's method is applied to a system, with one caveat: the increment functions must be applied in a specific order. Specifically, the increments at the beginning of the interval `(k1, l1, h1, ...)` must be applied before the increments at the midpoint and endpoint can be applied.

The code below takes the initial conditions of *one* scenario and numerically solves for its trajectory using the fourth-order Runge–Kutta approach[8].

```
function [t,x,y,v,O,r] =
RK4_System_NumMethProj_MC(i1,fcn1,fcn2,fcn3,fcn4,fcn5,a,b,c,xi,yi,vi,Oi,ri,N)

t = linspace(a,b,N+1);
s = (b-a)/N;
h = zeros(c,length(t));

x = zeros(size(h));
y = zeros(size(h));
v = zeros(size(h));
O = zeros(size(h));
r = zeros(size(h));
```

[8]The function r is placed in the loop since the values of r must be incremented alongside the other functions.

```
y(i1,1) = yi;
x(i1,1) = xi;
v(i1,1) = vi;
O(i1,1) = Oi;
r(i1,1) = ri;

for n = 1:N

    k1 = fcn1(v(i1,n),O(i1,n));
    l1 = fcn2(v(i1,n),O(i1,n));
    h1 = fcn3(v(i1,n),O(i1,n),r(i1,n));
    g1 = fcn4(v(i1,n),O(i1,n),r(i1,n));

    k2 = fcn1(v(i1,n)+.5*h1*s,O(i1,n)+.5*g1*s);
    l2 = fcn2(v(i1,n)+.5*h1*s,O(i1,n)+.5*g1*s);
    h2 = fcn3(v(i1,n)+.5*h1*s,O(i1,n)+.5*g1*s,r(i1,n)+.5*l1*s);
    g2 = fcn4(v(i1,n)+.5*h1*s,O(i1,n)+.5*g1*s,r(i1,n)+.5*l1*s);

    k3 = fcn1(v(i1,n)+.5*h2*s,O(i1,n)+.5*g2*s);
    l3 = fcn2(v(i1,n)+.5*h2*s,O(i1,n)+.5*g2*s);
    h3 = fcn3(v(i1,n)+.5*h2*s,O(i1,n)+.5*g2*s,r(i1,n)+.5*l2*s);
    g3 = fcn4(v(i1,n)+.5*h2*s,O(i1,n)+.5*g2*s,r(i1,n)+.5*l2*s);

    k4 = fcn1(v(i1,n)+h3*s,O(i1,n)+g3*s);
    l4 = fcn2(v(i1,n)+h3*s,O(i1,n)+g3*s);
    h4 = fcn3(v(i1,n)+h3*s,O(i1,n)+g3*s,r(i1,n)+l3*s);
    g4 = fcn4(v(i1,n)+h3*s,O(i1,n)+g3*s,r(i1,n)+l3*s);

    x(i1,n+1) = x(i1,n)+(k1+2*k2+2*k3+k4)*s/6;
    y(i1,n+1) = y(i1,n)+(l1+2*l2+2*l3+l4)*s/6;
    v(i1,n+1) = v(i1,n)+(h1+2*h2+2*h3+h4)*s/6;
    O(i1,n+1) = O(i1,n)+(g1+2*g2+2*g3+g4)*s/6;
    r(i1,n+1) = fcn5(x(i1,n+1),y(i1,n+1));

 end

end
```

3.4 Monte Carlo Implementation

By marrying the above code to a Monte Carlo code, we are able to repeat the Runge–Kutta method for thousands of simulations, where c is the number of simulations.

```
function [t,xmat,ymat,vmat,Omat,rmat,tsol,Isol] =
MonteCarloSim(fcn1,fcn2,fcn3,fcn4,fcn5,a,b,c,R,xi,yi,vdist,Odist,di,N)

t = linspace(a,b,N+1);
h = zeros(c,length(t));
tsol = zeros(c,1);
Isol = zeros(c,1);

x = zeros(size(h));
y = zeros(size(h));
v = zeros(size(h));
O = zeros(size(h));
r = zeros(size(h));

xmat = zeros(size(h));
ymat = zeros(size(h));
vmat = zeros(size(h));
Omat = zeros(size(h));
rmat = zeros(size(h));

for i1=1:c
[t,x,y,v,O,r] = RK4_System_NumMethProj_MC(
i1,fcn1,fcn2,fcn3,fcn4,fcn5,a,b,c,xi,yi,vdist(i1),Odist(i1),di,N);

h = r-R; %height above surface of earth

subplot(2,2,1),plot(x(i1,:),h(i1,:),'-k');
title('Altitude vs Range'); ylabel('Height(m)'); xlabel('Range(m)');
hold on;

subplot(2,2,2),plot(t,h(i1,:),'-r');
title('Altitude vs Time'); ylabel('Height(m)'); xlabel('Time(s)');
hold on;

subplot(2,2,3),plot(t,v(i1,:),'-b');
title('Velocity vs Time'); ylabel('Velocity(m/s)'); xlabel('Time(s)');
```

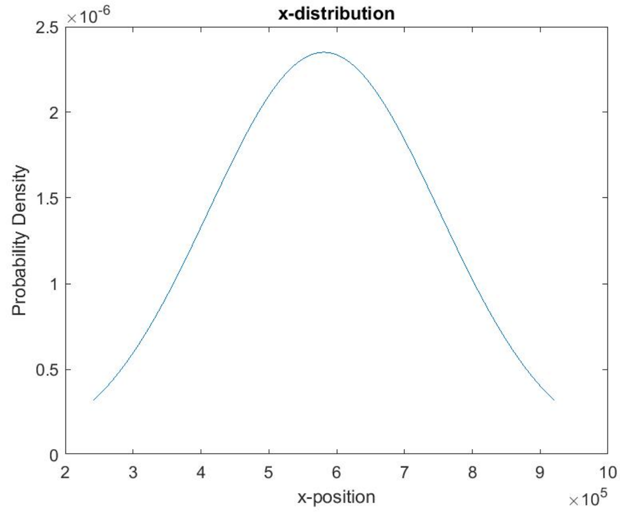
```
hold on;

subplot(2,2,4),plot(t,O(i1,:),'-g');
title('Flight Angle vs Time'); ylabel('Flight Angle(rads)'); xlabel('Time(s)');
hold on;

[h(i1,:),I] = min(h(h>=0)); %gives best estimate for h=0 and its index
tsol(i1) = t(I);  %gives time for h=0
Isol(i1) = I;

xmat = xmat+x;
ymat = ymat+y;
vmat = vmat+v;
Omat = Omat+O;
rmat = rmat+r;

end
end
```

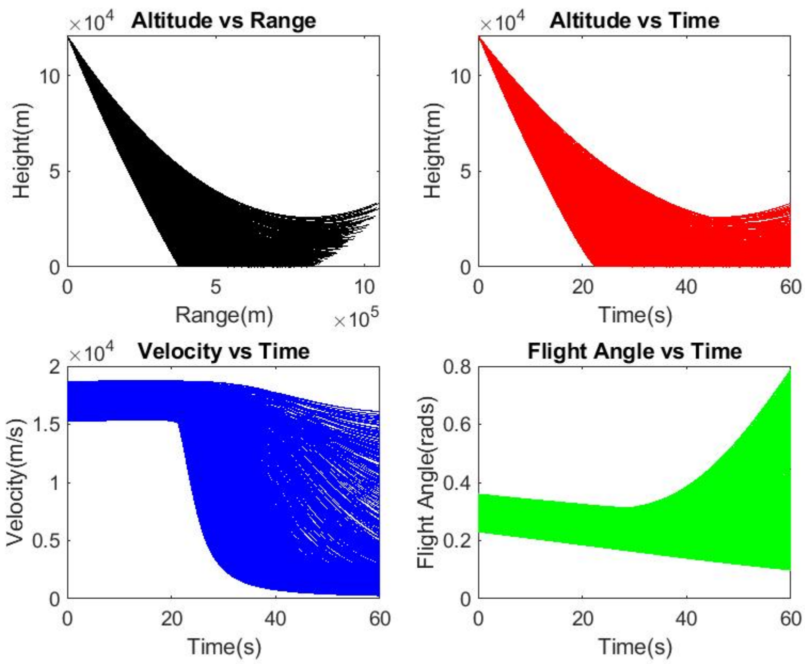
4 Results

The results from `c = 1000` iterations of two scenarios—high and low uncertainty about initial conditions—are shown below.

4.1 High Uncertainty

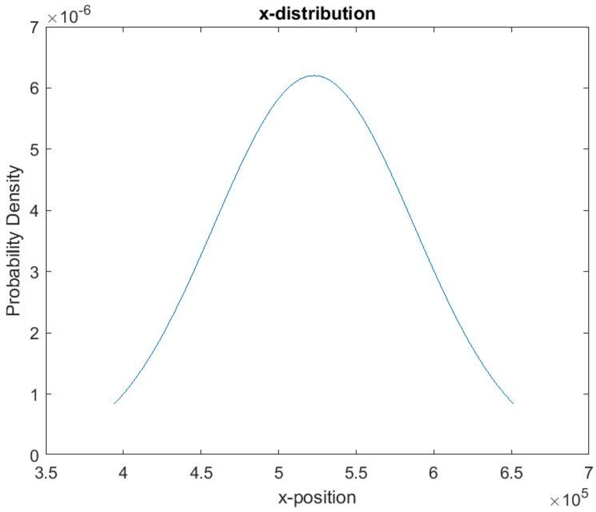
vi = 17000 +/- 1700, Oi = 0.3 +/- 0.09

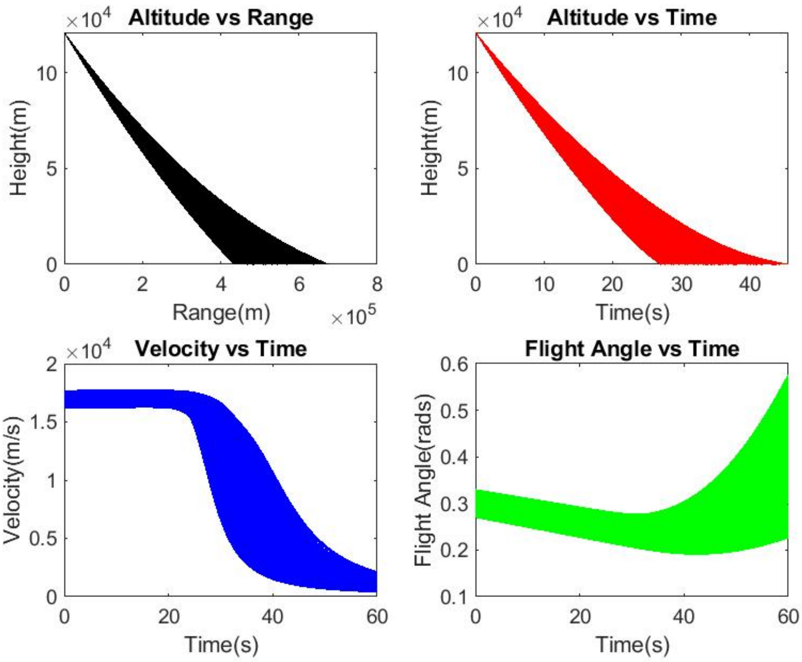




4.2 Low Uncertainty

vi = 17000 +/- 700, Oi = 0.3 +/- 0.03





5 Conclusion

We successfully employed a Monte Carlo simulation to calculate the range of trajectories traversed by a spherical object of negligible moment of inertia as it enters Earth's atmosphere at an estimated initial velocity and flight angle. The figures above match the general trends of those in Apollo-era papers [5, 11].

Our code can be improved to become more realistic. For example, it was assumed that both the size of our object and its density were constant throughout the trajectory. This assumption is far from true if attempting to model the entry of a meteorite. In reality, a significant fraction of a meteorite's mass is lost as it enters the atmosphere. Also the meteorite heats up as it passes through the atmosphere, making its density a function of its path.
Another area in which the code could be improved is in our model of the atmosphere. We assumed the density of the atmosphere falls off exponentially as the distance from the Earth's center increases, while in reality, atmospheric density is defined in a piecewise fashion [4] over specific altitudes. Further, the atmosphere is not isothermal, so the scale height parameter `H` can be improved to be a function of r.

References

[1] Carry, B. *Density of Asteroids*. European Space Astronomy Centre: arXiv:1203.4336v1, 2012.

[2] Carter, Lynn. *How many meteorites hit Earth each year? (Intermediate).* "Ask an Astronomer," Cornell Astrophysics, 2015.

[3] Darling, David J. *The Universal Book of Astronomy: From the Andromeda Galaxy to the Zone of Avoidance.* Wiley. (260) 2004.

[4] *Geometric altitude vs. temperature, pressure, density, and the speed of sound derived from the U.S. Standard Atmosphere* NASA, 1962.

[5] Goossens, Michel; Frank Mittelbach; and Alexander Samarin. *A Study of Ballistic Reentry Trajectories at a Velocity of 50000 feet per second.* Langley Research Center, NASA, 1967.

[6] James, A.M. and M.P. Lord. *Macmillan's Chemical and Physical Data* Macmillan, London, UK, 1992.

[7] Hall, Nancy. *Ideal Lift of a Spinning Ball.* Glenn Research Center, NASA, 2008.

[8] Meteorites and Their Properties. *The Structure and Composition of Meteorites* Universities Space Research Association.

[9] "Equatorial Radius of the Earth." *Numerical Standards for Fundamental Astronomy: Astronomical Constants : Current Best Estimates (CBEs).* IAU Division I Working Group. 2012.

[10] Tetzman, Derrick G. *Simulation and Optimization of Spacecraft Reentry Trajectories.* University of Minnesota, 2010.

[11] White, Jack A. and Johnson, Katherine G. *APPROXIMATE SOLUTIONS FOR FLIGHT-PATH ANGLE OF A REENTRY VEHICLE IN THE UPPER ATMOSPHERE* Langley Research Center, July 1964.

This report was created using LaTeX.