

Aerodynamic Package for Low Altitude Orbit CubeSat

Documentation of DSMC.m

Senior Design Team 02

MEM 492



By:

Frank Arute
Jasen Carroll
Kelly Meighan
Daniel Zettler

Submitted to:

Dr. Jin Kang

3/2/2012

Introduction:

This is a program made using MATLAB. The purpose of the program is to predict the lifespan of the satellite given three different geometrical shapes representing various nose cones attached to the satellite. Those shapes would be known as a flat plate (in other words no nose cone), a half cylinder and a double wedge. It is fairly accurate and the relationship between the shapes is as predicted, furthermore the theorized best shape being the double wedge was confirmed by this analysis. Using this software the team was able to fully understand what shape the nose cone should be and why. It was then asked later to implement a fourth scenario. In this case the satellite has four double wedge packages attached to it, but if the flow comes in at an awkward angle, it was determined to know that lifespan.

Walkthrough:

The entire code can be found in Appendix A. First the program is told a lot of information about the projected orbit and atmosphere of the satellite at 250 km above Earth's crust. Everything is taken into account from the speed of the satellite to the speed of the free particles, to the density of the particles and back to the area of the satellite. All information was obtained using mathematical formulas or using case studies.

Next the program asks for a user input and based off of this input four different scenarios take place. As mentioned previously, there are 'no nose cone,' 'half cylinder,' 'double wedge,' and 'a revolving four sided CubeSat.' From these inputs different shapes are taken into account using a reference angle with the x-axis. For no nose cone the angle is 90 degrees, with a half cylinder the angle is randomly generated due to the fact that the angle is constantly changing tangent to a circle and thus when a particle strikes the circle it is very hard to predict what angle it will create, thus being random. The double wedge angle is 32 degrees which is how our package for a double wedge has been designed. The rotating CubeSat was taken apart mathematically to find an average reference angle of 24.5 degrees for the proposed worst case scenario.

For this application in free molecular flow, theoretically the smaller the angle the better the aerodynamic properties of the satellite. However, in the case of the rotating CubeSat, since we are adding 4 double wedge's instead of one, the area is increased by a factor of 2.8 thus reducing its lifespan, the math of which can be seen in Appendix B. Below in Figure 1 you will see the results of each scenario.

Now, the program knows everything about the atmosphere and the proposed satellite, for a set amount time, in this case $T/100$ where T is the period of your satellite orbit, the satellite is "thrown" particles at it. After the calculated number of particles impacting the satellite the loop stops with a calculation for the total force of the impacts with the satellite. This is achieved using rotational matrices and $I = F\Delta t = m\Delta v$.

Finally, the program goes through one more loop which takes a change in distance and a time of 0 and runs the loop as long as the range of the orbit is above reentry range - 100km. For every time this loop is run the total energy of the impacts is calculated, because energy is equal to force times distance, then taking into account the current velocity of the satellite it recalculates the range when this range is found if it is below 100 km the loop stops. t is then converted from seconds to days and the lifespan of the proposed satellite is printed on the screen, as also seen in Figure 1.

```
>> DSMC
Press 1 for no nose cone, 2 for a half cylinder, 3 for a double wedge,
    or 4 for a revolving four sided CubeSat
    1

ans =

    10.4650

>> DSMC
Press 1 for no nose cone, 2 for a half cylinder, 3 for a double wedge,
    or 4 for a revolving four sided CubeSat
    2

ans =

    20.7070

>> DSMC
Press 1 for no nose cone, 2 for a half cylinder, 3 for a double wedge,
    or 4 for a revolving four sided CubeSat
    3

ans =

    37.2804

>> DSMC
Press 1 for no nose cone, 2 for a half cylinder, 3 for a double wedge,
    or 4 for a revolving four sided CubeSat
    4

This is strictly a best case scenerio for a worst case orientation.
Meaning I do not account for particles being trapped inside the wedges.

ans =

    27.4819
```

Figure 1: Results for the simulation

Conclusion:

As we can see, the double wedge is best followed by the rotating CubeSat. This may be a viable option for ensuring increased lifespan if there is no other technology readily available for over \$3,000 to stabilize the satellite in one direction the entire time. Until other technologies can prove themselves this is a plausible answer to the solutions.

Appendix A:

```
%% DSMC Simulation for LEO CubeSat using MATLAB
clear

%% Constants
R = 250000; %m altitude of orbit
G = 6.67e-11; %m^3/(kg*s^2) gravitational constant
Me = 5.9742e24; %kg mass of earth
Re = 6387100; %m radius of earth
MEWe = G*Me; % standard graviational parameter for earth
Msat = 1.33; %kg mass of satellite
Rsat = Re + R; %m radius of CubeSat orbit from center of earth
Vsat = sqrt(MEWe/Rsat); %Initial orbital velocity
E = Vsat^2/2-MEWe/Rsat; %Energy of the satellite
P = 2*pi*sqrt((Rsat)^3/MEWe); %Period of the satellite for one orbit
deltaT = 0.001; %s change in time for calculation of impulse equation
rho = (5.94e-08); %g/m^3 mass of particles found at 250km
PartTot = 1.7e15; %number of particles found at 250km per m^3
Mpart = rho/PartTot; %average mass of a particle at 250km
AreaS = 100/10000; %area of typical CubeSat in m^3
MperS = rho*(AreaS)*Vsat; %mass of particles hitting satellite per second
MdT = MperS*deltaT; %mass of particles hitting satellite per deltaT
SatDist = 2*pi*Rsat; %circumference of the satellite's orbit
s = 100; %sampling rate

%% Nose Cones
reply = input('Press 1 for no nose cone, 2 for a half cylinder, 3 for a
double wedge,\n          or 4 for a revolving four sided CubeSat \n          ');
if reply == 1
    alpha = 90*(pi/180); %radians angle between sat and x-axis
elseif reply == 2
    alpha = 0;
elseif reply == 3
    alpha = 32*(pi/180); %radians angle between sat and x-axis
elseif reply == 4
    disp(' ')
    disp('This is strictly a best case scenerio for a worst case
orientation.')
```

disp('Meaning I do not account for particles being trapped inside the wedges.')

AreaS = 2.8*AreaS; %This design has 2.8 times more surface area
%because there are now four double wedges, they stick out on either
%side. Because we are looking at this from a corner, and the wedge's
%are 10cm long by 10 cm wide, the overall width of the satellite
%becomes sqrt(200)+sqrt(200) = 28.28 whereas the single nose cone is
%simply 10cm wide.

MperS = rho*(AreaS)*Vsat;
MdT = MperS*deltaT;

%So now we need to have this question "If a particle hits a 30 degree
%slope at a 45 degree angle, what angle does the particle experience?"
%Well, we know that the angle of the wedge is 30 degrees and we know
%that the angle the particle is incoming is 45 degrees, so if we
%multiply 30*sin(45) we get the angle the particle will experience

```

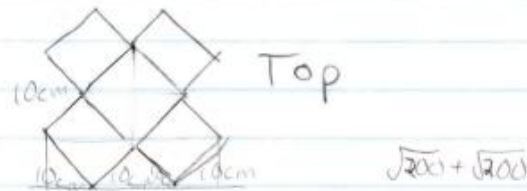
    %along the axis parallel to the flow of the free molecules.
    alpha = ((45/2+2*(30*sin(45)))/3); %Averages the corner angle along
    %with the two angles from the wedges that are now in view. Need to fix
    %the 64
    alpha = alpha*(pi/180); %average radians angle between sat and x-axis
end
%% Calculations
i = 0;
Ftot = 0;
Vsati = Vsat;
dT = P/s; %Period divided by sampling rate
while i < dT
    i = i + deltaT;
    if reply == 2
        beta = normrnd(0,3.3333333,1)*(pi/180); %radians randomly generated
        %number from a normal distribution to calculate beta - the angle
        %between the incoming particle and the x-axis
        alpha = randi(90)*(pi/180);
    else
        beta = normrnd(0,3.3333333,1)*(pi/180); %radians randomly generated
        %number from a normal distribution to calculate beta - the angle
        %between the incoming particle and the x-axis
    end
    Vpart = 1150; %m/s randomly generated number
    %from a normal distribution to calculate the velocity of the
    %incoming particle
    v1 = [Vpart*cos(beta);Vpart*sin(beta)]; % Vpart needs to be randomized
    theta = alpha + beta; %radians angle of incidence
    phi = 2*pi - 2*theta;
    Rot = [cos(phi) -sin(phi);sin(phi) cos(phi)]; %rotational matrix
    v2 = Rot*v1;
    deltaVpart = v2-v1; %m/s change in velocity of particle from impact
    deltaV = deltaVpart(1);
    F = (MdT*abs(deltaV))/deltaT; % force from impulse
    deltaVsat = (F*deltaT)/(Msat); % change in velocity from impact
    % figure out two dimensional perfectly elastic collision
    Vsat = Vsat - abs(deltaVsat); %m/s new sat velocity
    Ftot = Ftot + F; %the total force over the given period
end
dVsati = Vsati - Vsat;

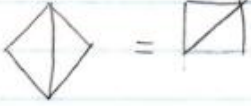

%% Calculating the lifespan of the satellite
dx = SatDist/s;
t = 0;
while R > 100
    dE = (Ftot)*(dx); %Calculates the energy of the sate after one interval
    E = E - dE; %Find the new over energy
    V = Vsati - dVsati; %the current velocity
    R = MEWe/(V^2/2-E); %the new range
    t = t + dT; % the time the satellite lasted in seconds
end
t/(3600*24) %prints the number of days to the screen



```

Appendix B:

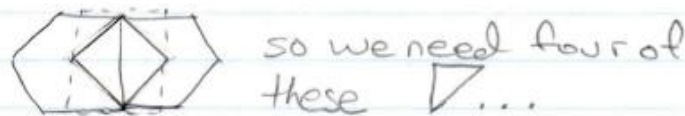
KANG'S REVOLVING CUBESAT



So, we have  =  And because the wedge area is the same on each Bingo!


 = 

Now we were looking at this...



$$\begin{aligned} \theta &= 45^\circ & y &= 5\text{cm} \\ \tan^{-1}\left(\frac{y}{x}\right) &= \theta \\ \tan \theta &= \frac{y}{x} \\ x &= \frac{y}{\tan \theta} & x &= 5 \end{aligned}$$

umm, yea it's a 45° angle... obvi...
 so... triangle area = $\frac{1}{2}bh = \frac{1}{2}xy = (\frac{1}{2})(5)(5)$
 $A = 12.5\text{cm}$. Nose cone area = 10×10
 $= 100\text{cm}$. We need $4 \times 12.5\text{cm}$ of 100cm
 $= 50\%$ of one nose cone.
 Alright... finally.

Turn the page if still interested 

Here we go...

We need the deflection rate of the nose cone & the edge of the satellite.

But

cone + edge = answer? NO!

Con & edge need to be averaged... How?

$$\frac{[1 \times (\text{cone angle}) + 0.5 \times (\text{edge angle})]}{1.5} = \text{answer equals new angle of}$$
$$\frac{[1 \times (45^\circ) + 2 \times (60 \sin 45^\circ)]}{1.5} = 43.3^\circ \text{ equal nose cone!}$$
$$\theta = 43.3^\circ$$



$$\theta = 2\alpha \Rightarrow \alpha = \boxed{21.64^\circ}$$

B.T.W. α for cone = 32° ...

By this comparison "Dr. Kang's Revolving CubeSat" will outlast the aeropack alone.

Ok, more math.

Current design of Aeropack:

$$\theta = 64^\circ \Rightarrow \alpha = 32^\circ$$

Cost: $\approx \$1000$ a unit, Materials & Productions

DKRC design:

4 Aeropacks = $\approx \$4000$ but... might be cheaper than other alternatives for stabilization. While $A_{\text{new}} = 2.8$ times old A_{old}