**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 simulation was made utilizing MATLAB software. The purpose is to predict the lifespan of a satellite, experiencing drag from perpendicular free molecular flow, given three different geometric shapes. These shapes represent various nose cones attached to the satellite. The tested shapes are known as a flat plate (or no nose cone), a half cylinder, and a double wedge. The theorized best shape, the double wedge, was confirmed by the simulation analysis. With the conclusion of the double wedge resulting in the longest lifespan, a fourth scenario was added to the simulation. This scenario represents the satellite as having four double wedge packages, one on each side. However, for this test the free molecular flow representation will interact with the satellite at non perpendicular angle.

# Walkthrough:

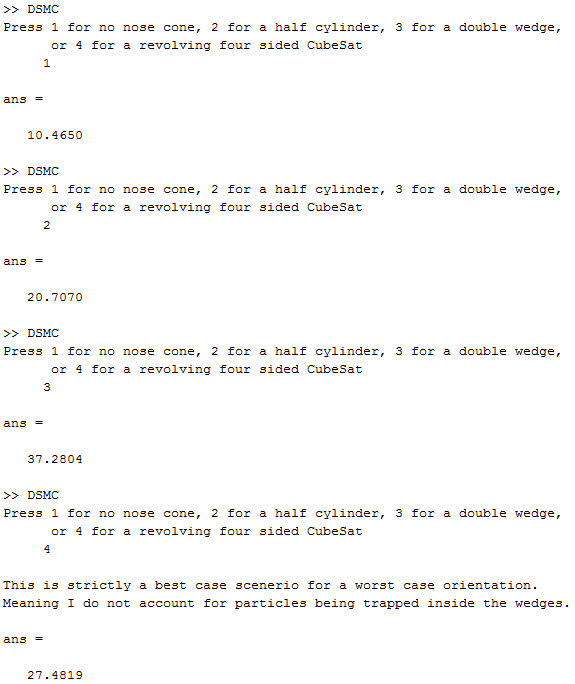
The code for this simulation can be found in Appendix A. The program is first told information about the projected orbit and atmosphere of the satellite at 250 km above Earth’s sea level. Everything from the speed of the satellite to the density of the free molecular flow particles is taken into account and provided to the software simulation. All information was obtained using mathematic formulas or using case studies in collaboration with Team 99 – A Windless Wind Tunnel.

Next the program asks for a user input, based off of this input four different scenarios take place. As mentioned previously, there are the flat plate, half cylinder, double wedge, and a four sided double wedge. From these inputs different shapes are taken into account using a reference angle with the x-axis. For no nose cone this angle is 90 degrees. With a half cylinder this angle is randomly generated between 0 and 90 degrees. This is due to the angle of deflection being dependent on where the particle strikes the half cylinder, thus being random. The double wedge angle is 32 degrees. This was obtained using our proposed design specifications for the double wedge. The four sided design was taken apart geometrically to find an average reference angle of 24.5 degrees.

Theoretically, for free molecular flow the smallest angle of deflection would provide the best aerodynamic properties. This is a direct result of conservation of momentum. However, in the case of the four sided design, since we are adding four double wedges instead of one, the area is increased by a factor of 2.8. This increase in area will affect the aerodynamic efficiency of the four sided design because there is more area for particles to interact with the satellite.

Now that the shape of the satellite has been chosen, the simulation provides free molecular flow to the satellite for a set period of time. This free molecular flow can be thought of as tennis balls flying towards an object and deflecting in various directions. After calculating how many impacts will occur during the set time period, a loop is created to calculate the force of each impact with the satellite. The loop stops with a summation of the total force of the impacts against the satellite. This is achieved using rotational matrices and I = FΔt = mΔv.

The simulation then goes through a final loop which calculates the change in the altitude of the satellite. The loop runs as long as the altitude of the orbit is above the reentry point – 100km. For every time this loop is run the total energy of the impacts is calculated. Using the fact of energy is equal to force times distance, and taking into account the current velocity of the satellite, the calculated energy of the impacts is applied to the satellite. Applying gravitational physics, this change in energy can be related to a change in altitude. When this altitude is found to be below 100 km the loop stops. The lifespan of the satellite, t, is then converted from seconds to days. The lifespan of the proposed satellite is then printed on the screen, as can be seen in Figure 1.



***Figure 1:*** *Results for DSMC.m*

# Conclusion:

As seen, the double wedge is best followed by the four sided design. Due to satellite stabilization technology currently costing an estimated $3,000, it may provide beneficial to implement the four sided design. That design again is a double wedge installed on the four sides of the CubeSat, not including the top and bottom. Until other technologies can prove themselves and become readily available this is a plausible solution.

# 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 scenario 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

%becames 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.

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

dVsat = Vsati - Vsat;

%% Caluclating the lifespan of the satellite

dx = SatDist/s;

t = 0;

while R > 100

dE = (Ftot)\*(dx); %Calculates the energy of the sat after one interval

E = E - dE; %Find the new over energy

V = Vsati - dVsat; %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