Developing a Comprehensive Power Simulation Model for the MEMESat-1 CubeSat using Orbital Dynamics

Mission for Education and Multimedia Engagement Satellite (MEMESat-1)
Small Satellite Research Laboratory, University of Georgia

Matthew Knauss, Ali Batu Ozdener, Vedika Ghildyal, Sydney Whilden, Deepak R. Mishra
mknauss@uga.edu, ali.ozdener@uga.edu, vedika.ghildyal@uga.edu, sydney.whilden25@uga.edu, dmishra@uga.edu

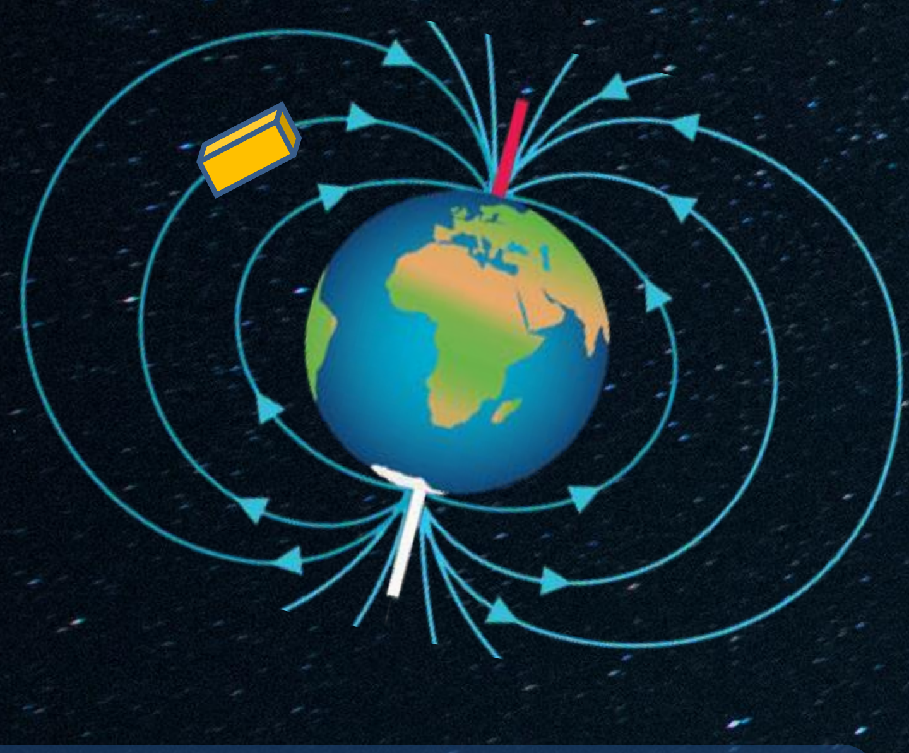
Abstract

The University of Georgia's Small Satellite Research Lab's Mission for Education and Multimedia Engagement Satellite (MEMESat-1) requires the use of variables such as power generation, power draw, orbital path, packet size, and data processing times. As power generation and charge varies, MEMESat-1 will automatically transition through three operational modes to prevent battery depletion and halt system processes in case of anomalies.

Taking these variables and operational modes into account, the MEMESat-1 Mission Operations (MOPS) team will use FreeFlyer software to analyze power generation and draw during MEMESat-1's orbital cycle. The power limitations of MEMESat-1 are budgeted based on battery and solar cell specifications implying the necessity of power simulations by MOPS.

Figure 1

A Passive Magnetic Attitude Control (PMAC) system utilizing hysteresis rods, nutation dampers, and bar magnets is considered as MEMESat-1 will adjust according to Earth's magnetic field lines.

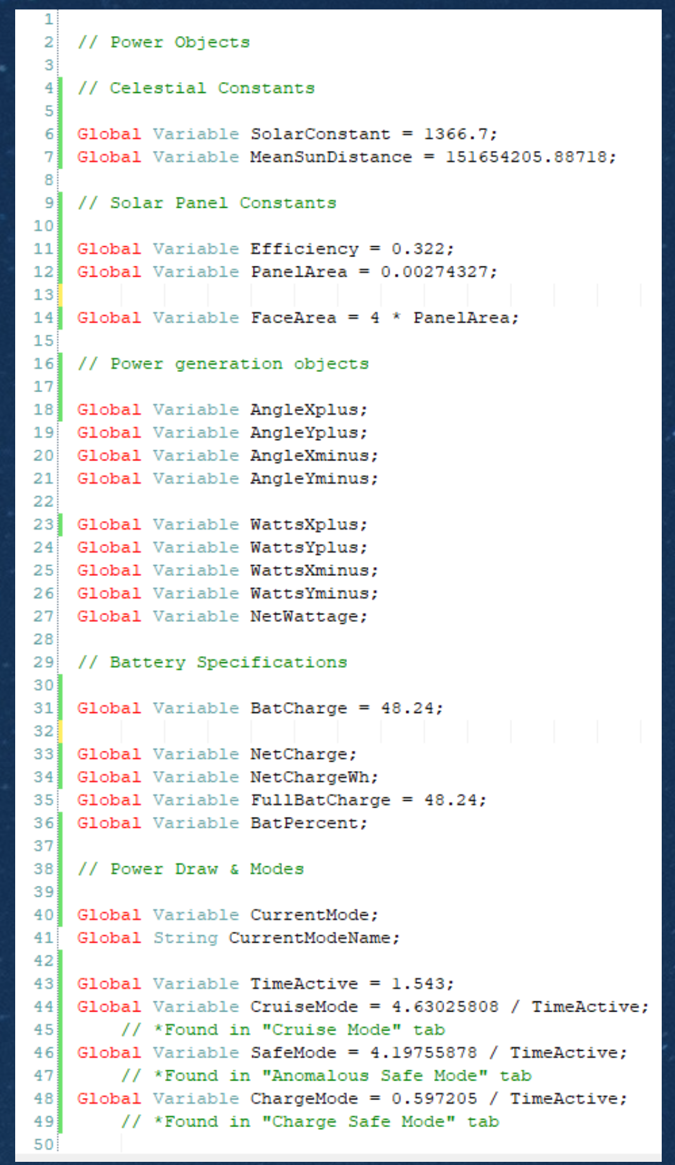


Introduction

A FreeFlyer simulation utilizing a 30 second time step was created to analyze any issues present within the parameters of MEMESat-1 over a five-day run time. The simulation considers MEMESat-1's orbital parameters (ISS values until given launch parameters), attitude behavior (Figure 1), satellite to sun distance and the angle between them, and 4 XTE-SF solar panels for each 2U face.

The simulation uses given parameters to calculate solar power generation and determine any operational mode changes with each time step. At the end of the run time, a power generation and mode changes are analyzed. For an ideal mission plan, MEMESat-1 will stay in Cruise Mode with consistent power generation, and only transition to safe modes when an anomaly is encountered.

Defining Angles & Power Generation

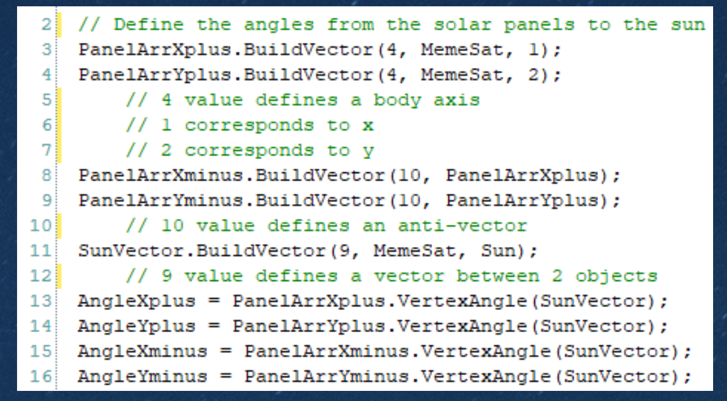


```

// Power Objects

// Celestial Constants

Global Variable SolarConstant = 1366.7;
Global Variable MeanSunDistance = 151654205.88718;

// Solar Panel Constants

Global Variable Efficiency = 0.322;
Global Variable PanelArea = 0.00274327;

Global Variable FaceArea = 4 * PanelArea;

// Power generation objects

Global Variable AngleXplus;
Global Variable AngleYplus;
Global Variable AngleXminus;
Global Variable AngleYminus;

Global Variable WattsXplus;
Global Variable WattsYplus;
Global Variable WattsXminus;
Global Variable WattsYminus;
Global Variable NetWattage;

// Battery Specifications

Global Variable BatCharge = 48.24;

Global Variable NetCharge;
Global Variable NetChargeWh;
Global Variable FullBatCharge = 48.24;
Global Variable BatPercent;

// Power Draw & Modes

Global Variable CurrentMode;
Global String CurrentModeName;

Global Variable TimeActive = 1.543;
Global Variable CruiseMode = 4.63025808 / TimeActive;
    // *Found in "Cruise Mode" tab
Global Variable SafeMode = 4.19755878 / TimeActive;
    // *Found in "Anomalous Safe Mode" tab
Global Variable ChargeMode = 0.597205 / TimeActive;
    // *Found in "Charge Safe Mode" tab

```

While orbital parameters can be easily set through spacecraft properties in FreeFlyer, other variables used in power generation must be created or defined in the script. The power simulations model creates all variables for MEMESat-1's 2U solar faces in a single subsection (Figure 2.1). Celestial and solar panel constants are defined while power generation objects are created for future use. The battery specifications and power draw for each operational mode are specified for calculating net charge later.

Figure 2.1 (left)
Variables used in the script are created and organized before the time step loop, allowing for a neat power generation calculation.

To define the relative angles between each of the solar faces and the sun, vectors are created for the 2U faces. An additional reference vector is created between the sun and MEMEsat-1. Finally, the relative angles for each face are found by taking the angle between the face vectors and the sun vector.



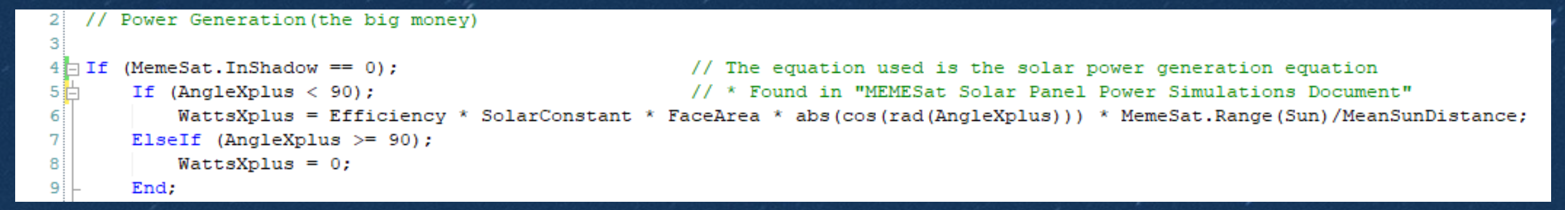
```
// Define the angles from the solar panels to the sun
PanelArrXplus.BuildVector(4, MemeSat, 1);
PanelArrYplus.BuildVector(4, MemeSat, 2);
    // 4 value defines a body axis
    // 1 corresponds to x
    // 2 corresponds to y
PanelArrXminus.BuildVector(10, PanelArrXplus);
PanelArrYminus.BuildVector(10, PanelArrYplus);
    // 10 value defines an anti-vector
SunVector.BuildVector(9, MemeSat, Sun);
    // 9 value defines a vector between 2 objects
AngleXplus = PanelArrXplus.VertexAngle(SunVector);
AngleYplus = PanelArrYplus.VertexAngle(SunVector);
AngleXminus = PanelArrXminus.VertexAngle(SunVector);
AngleYminus = PanelArrYminus.VertexAngle(SunVector);
```

Figure 2.2
A subsection of the script was created solely for determining panel angles. The BuildVector and Vertex Angle commands were used for simplicity.

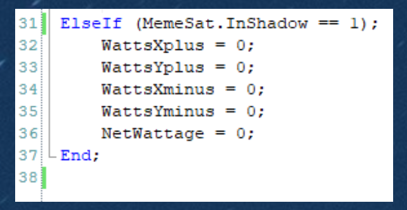
Since all relevant variables have been input or determined, power generation can be calculated using a solar power generation equation (right). Power generation is only to be determined for a 2U face when MEMESat-1 is not in Earth's shadow, and the face is not in the shadow produced by the satellite (Figure 2.3 & 2.4).

$$Power = \eta \cdot S_o \cdot A \cdot \cos\alpha \cdot \frac{R_0}{\overline{R}_0}$$

where: η = Solar Panel Efficiency
S_o = Solar Constant
A = Solar Panel Area
α = Solar Panel Angle
R_0 = Spacecraft Sun Distance
$\overline{R}_0$ = Spacecraft Mean Sun Distance.



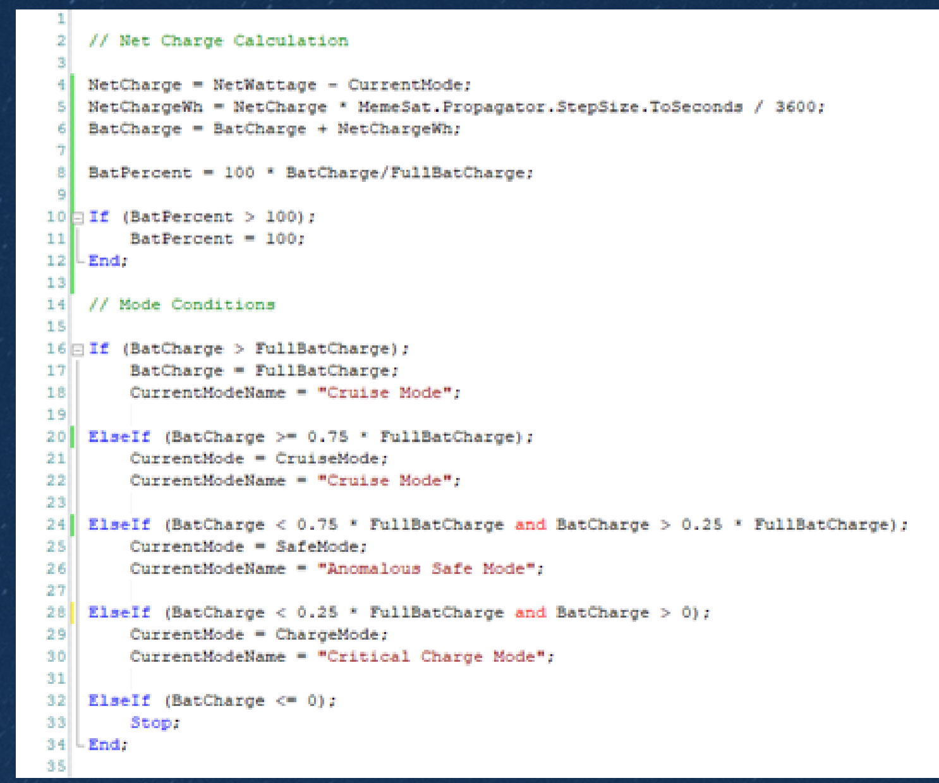
```
// Power Generation(the big money)

If (MemeSat.InShadow == 0);                          // The equation used is the solar power generation equation
    If (AngleXplus < 90);                            // * Found in "MEMESat Solar Panel Power Simulations Document"
        WattsXplus = Efficiency * SolarConstant * FaceArea * abs(cos(rad(AngleXplus))) * MemeSat.Range(Sun)/MeanSunDistance;
    ElseIf (AngleXplus >= 90);
        WattsXplus = 0;
    End;
```



```
ElseIf (MemeSat.InShadow == 1);
    WattsXplus = 0;
    WattsYplus = 0;
    WattsXminus = 0;
    WattsYminus = 0;
    NetWattage = 0;
End;

```

Figures 2.3 & 2.4
This subsection of the script contains a nested if-else loop which calculates the output of each solar face while in sunlight. When MEMESat-1 is in shadow, the net output is automatically zero.

Calculating Net Charge & Mode Conditions



```

// Net Charge Calculation

NetCharge = NetWattage - CurrentMode;
NetChargeWh = NetCharge * MemeSat.Propagator.StepSize.ToSeconds / 3600;
BatCharge = BatCharge + NetChargeWh;

BatPercent = 100 * BatCharge/FullBatCharge;

If (BatPercent > 100);
    BatPercent = 100;
End;

// Mode Conditions

If (BatCharge > FullBatCharge);
    BatCharge = FullBatCharge;
    CurrentModeName = "Cruise Mode";

ElseIf (BatCharge >= 0.75 * FullBatCharge);
    CurrentMode = CruiseMode;
    CurrentModeName = "Cruise Mode";

ElseIf (BatCharge < 0.75 * FullBatCharge and BatCharge > 0.25 * FullBatCharge);
    CurrentMode = SafeMode;
    CurrentModeName = "Anomalous Safe Mode";

ElseIf (BatCharge < 0.25 * FullBatCharge and BatCharge > 0);
    CurrentMode = ChargeMode;
    CurrentModeName = "Critical Charge Mode";

ElseIf (BatCharge <= 0);
    Stop;
End;

```

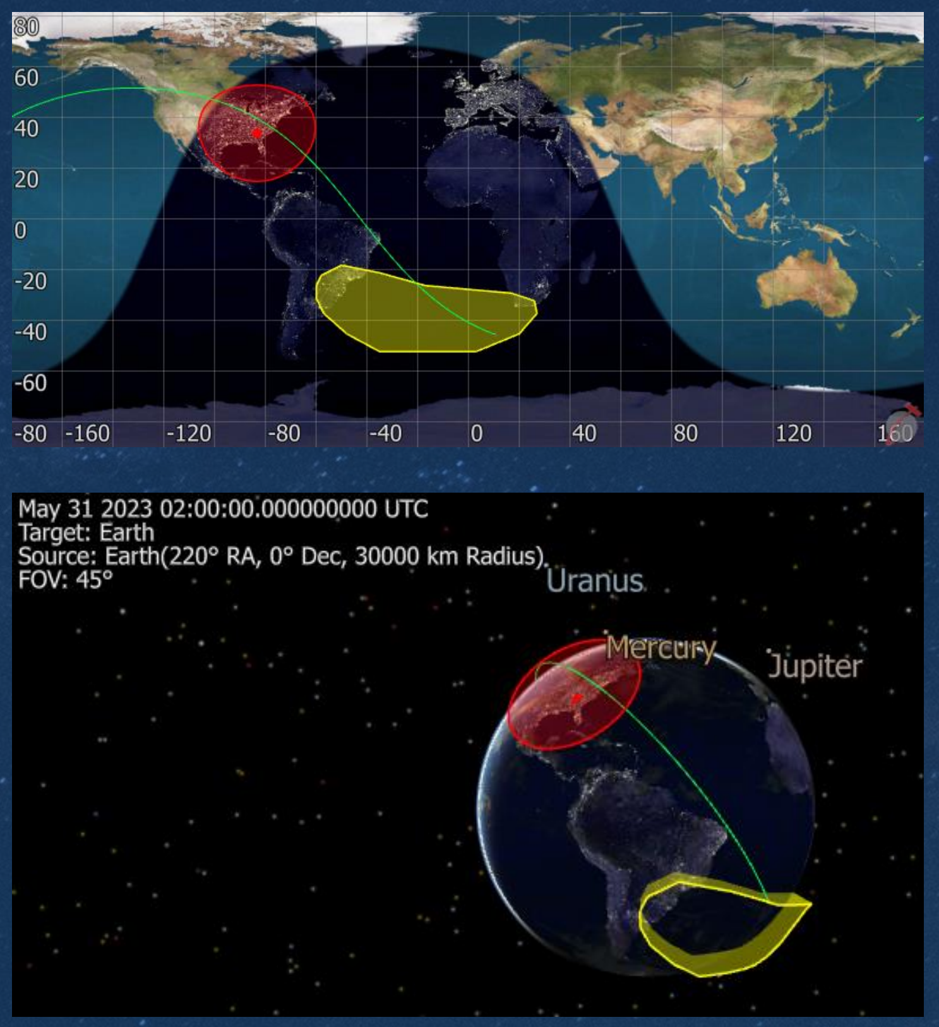
Figure 3
This subsection of the code demonstrates the calculation of battery charge and the determination of the current operational mode.

The net charge to the battery is calculated by subtracting the current mode's power draw from the wattage produced by solar panels. This is converted to watt-hours by multiplying by the number of time steps per hour. The net charge in watt-hours is added to the current battery charge to produce the new charge (These table values are updated with each progression of the time step loop.

Operational mode conditions are set to mimic MEMESat-1 transitioning modes once net battery charge dips below a certain threshold. Right now, the simulations model observes a transition to Anomalous Safe and Critical Charge Mode at 75% and 25% battery charge, respectively. However, this is subject to change with the mode conditions set by MEMESat-1's State Machine.

Crossing Times & Anomalous Mode Changes

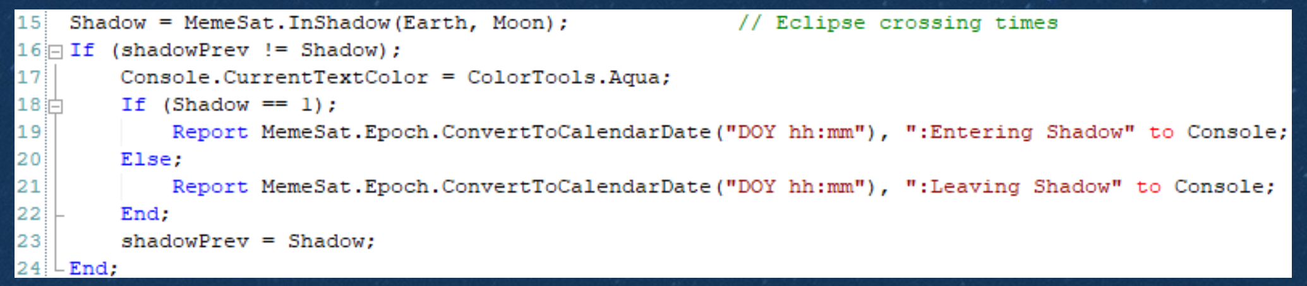
The power simulations model also observes times when power generation and draw changes dramatically for MEMESat-1. Crossing times for areas such as the South Atlantic Anomaly (SAA) and the UGA ground station are reported (Figures 4.1 & 4.2). In addition, shadow times are used to realize net charge transitions from negative to positive (Figure 4.3).



MOPS will consider a transition to an anomalous safe mode upon crossing into the SAA. Upon transitioning to this mode MEMESat-1 will continue charging and only beacon health checks upon request. It will cease payload activity to prevent damages caused by the higher ionizing radiation levels in this region.

Figure 4.3 (right)
This sample loop reports crossing times for regions to the console and allows us to analyze the average amount of time spent in each region.

The average time spent in the UGA ground station region will help determine the number of transmissions MOPS can consider while within range.

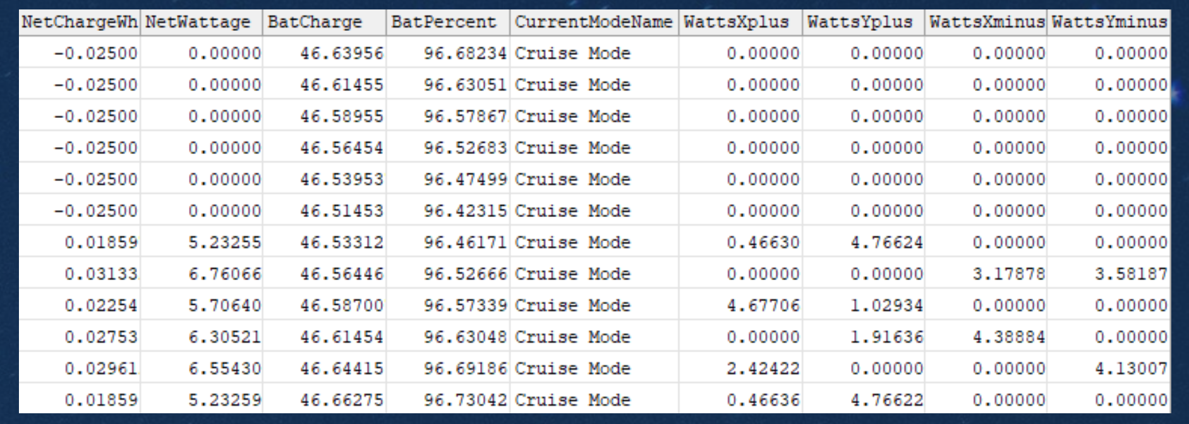


```
Shadow = MemeSat.InShadow(Earth, Moon);               // Eclipse crossing times
If (shadowPrev != Shadow);
    Console.CurrentTextColor = ColorTools.Aqua;
    If (Shadow == 1);
        Report MemeSat.Epoch.ConvertToCalendarDate("DOY hh:mm"), ":Entering Shadow" to Console;
    Else;
        Report MemeSat.Epoch.ConvertToCalendarDate("DOY hh:mm"), ":Leaving Shadow" to Console;
    End;
    shadowPrev = Shadow;
End;
```

Figures 4.1 & 4.2 (above)
The ViewWindow command can be utilized in FreeFlyer to to display a satellite's simulated orbit path in real time. These can be used to verify the accuracy of time spent in shadow and other regions.

Results

The DataTableWindow command was used to report power variables in FreeFlyer during the simulation (Figure 5). This allows MOPS to look at power generation throughout orbit and determine if it meets system requirements. In addition, the validity of gathered data can be assessed by analyzing solar panel face outputs throughout orbit since only two faces will be lit at once.



NetChargeWh	NetWattage	BatCharge	BatPercent	CurrentModeName	WattsXplus	WattsYplus	WattsXminus	WattsYminus
-0.02500	0.00000	46.63956	96.68234	Cruise Mode	0.00000	0.00000	0.00000	0.00000
-0.02500	0.00000	46.61455	96.63051	Cruise Mode	0.00000	0.00000	0.00000	0.00000
-0.02500	0.00000	46.58955	96.57867	Cruise Mode	0.00000	0.00000	0.00000	0.00000
-0.02500	0.00000	46.56454	96.52683	Cruise Mode	0.00000	0.00000	0.00000	0.00000
-0.02500	0.00000	46.53953	96.47499	Cruise Mode	0.00000	0.00000	0.00000	0.00000
-0.02500	0.00000	46.51453	96.42315	Cruise Mode	0.00000	0.00000	0.00000	0.00000
0.01859	5.23255	46.53312	96.46171	Cruise Mode	0.46630	4.76624	0.00000	0.00000
0.03133	6.76066	46.56446	96.52666	Cruise Mode	0.00000	0.00000	3.17878	3.58187
0.02254	5.70640	46.58700	96.57339	Cruise Mode	4.67706	1.02934	0.00000	0.00000
0.02753	6.30521	46.61454	96.63048	Cruise Mode	0.00000	1.91636	4.38884	0.00000
0.02961	6.55430	46.64415	96.69186	Cruise Mode	2.42422	0.00000	0.00000	4.13007
0.01859	5.23259	46.66275	96.73042	Cruise Mode	0.46636	4.76622	0.00000	0.00000

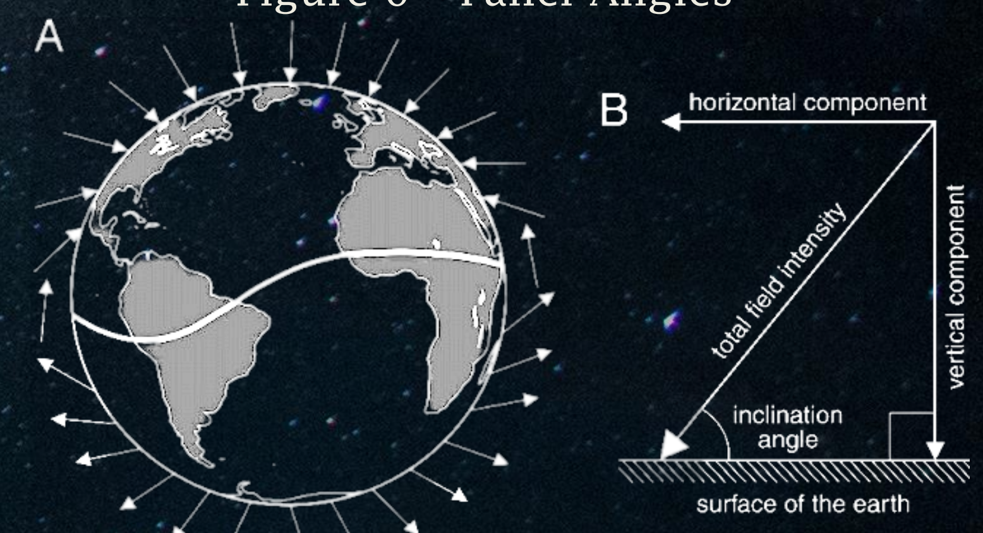
Figure 5
The table updates values with each progression of the time step loop. This section displays the transition into sunlight and the subsequent generation and charge values.

The simulation concludes that MEMESat-1 will stay in cruise mode throughout orbit due to adequate power generation. MEMESat-1 will not require any hardware changes and should only transition to a safe mode if it experiences a foreign anomaly or unprecedented power draw.

Future Considerations

Currently, the model uses a precessing spin to simulate the relative motion of MEMESat-1's PMAC system. These parameters were used in previous STK simulations; however, MOPS will update this behavior once ADCS simulations are complete. The future model will consider magnetic inclination angles (Figure 6) from the World Magnetic Model to determine panel angles.

Figure 6 – Panel Angles



Source: https://www.researchgate.net/figure/The-Earths-magnetic-field-A-Diagram-illustrating-how-field-lines-represented-by_fig3_23627259

Acknowledgments: Clay Reece (clayreece@uga.edu), FreeFlyer University Guide