

Reflected Albedo Flux on a Drag-Free Satellite

David Jackson*

August 7, 2015

Introduction

A zero-drag satellite is a satellite engineered only to experience gravitational forces. All other force and torques (like drag, magnetic, or any other non-gravitational) are analyzed so their influence can be nullified or avoided. A typical setup when building a drag-free satellite involves placing an outer “shell” around an inner payload. The inner payload might contain an object used for incredibly sensitive measurements, like a gravity wave sensor. Any external force or torque on the inner payload would ruin any measurement. Therefore, when maneuvering the outer shell, analysis of every potential external influence is desired. The word “external influence” is used in this report to mean both external torques and forces on a system.

One force on the outer shell is reflected sun radiation, called “albedo” radiation. This is electromagnetic radiation from the sun that bounces off the earth’s surface and hits the satellite. Under the particle model of physics, electromagnetic radiation possesses all the typical kinematic properties of moving objects, and thus is capable of exerting an impulse on the satellite. Figure 1, taken from a paper from the University of Austin¹, gives a diagram of the setup:

*daj17@stanford.edu, Stanford University, B.S. Mechanical Engineering Candidate, rising Junior

¹EARTH RADIATION PRESSURE EFFECTS ON SATELLITES. P. C. Knocke *, J. C. Ries, and B. D. Tapley. 11 pages. Copyright 1988. Accessed 7 August 2015.

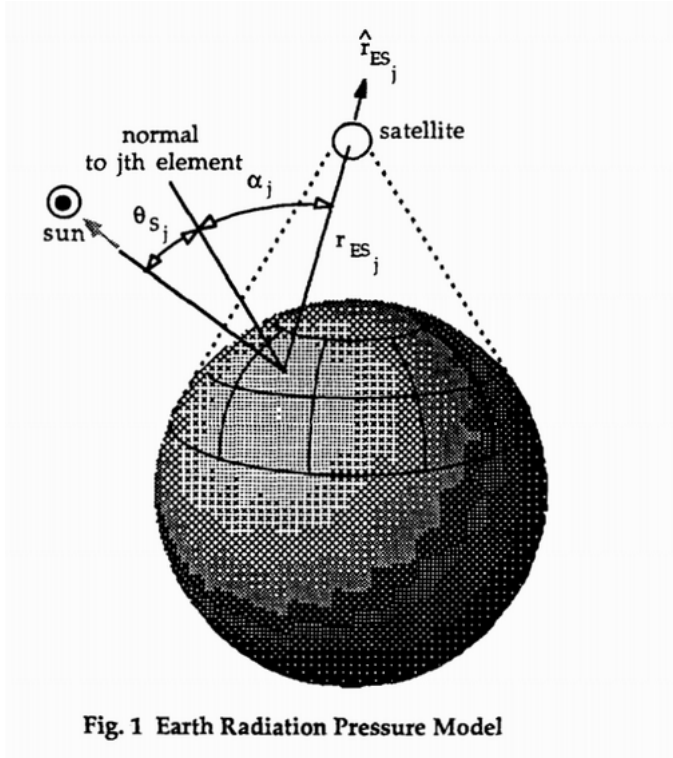


Fig. 1 Earth Radiation Pressure Model

The same paper provides a derivation of the flux a single earth element reflects upon a satellite orbiting the earth at any position, yielding:

$$d\phi = aE_s \cos(\theta_s) + eM \cos(\alpha) A_c / (r^2 \pi)$$

where $d\phi$ is the flux hitting the satellite,

a is the ratio of radiation power immediately reflected,

e is 1- a (ratio of radiation power absorbed and then re-emitted later),

E_s is the sun's intensity,

θ_s is the angle between the earth element normal vector and a vector from the earth element to the Sun,

M_b is the Sun's radiant flux divided by the surface area of the earth, equal to $E_s/4$,

α is the iteration angle,

A_c is the area of the earth element,

and r is the distance from the earth element to the satellite.

Objective

For our project we wished to know the influence of each discretized element on the satellite. Satellites can have complicated geometries, and each earth element is therefore capable of exerting not only a force but also a torque on the satellite. A Matlab function was written to iterate over common visibility area and return a list of unit vectors from each earth element to the satellite as well as the corresponding flux from each element. This information is sufficient to describe all resulting forces and torques on the satellite. Calculation of this external influence is achieved by using two additional tools: the wave model of radiation and simply trigonometry. This is all detailed in the derivation of the University of Austin paper (see footnote 1). Here we describe how use the results of that paper to achieve our objective.

Method

Reorientation, also called rotation, of the axes was the first step in our method. Reorienting our axes allows us particular advantage algebraically. Specifically, we chose an axes rotation that places the satellite, sun and x axis in the x-z plane (see Figure 2).

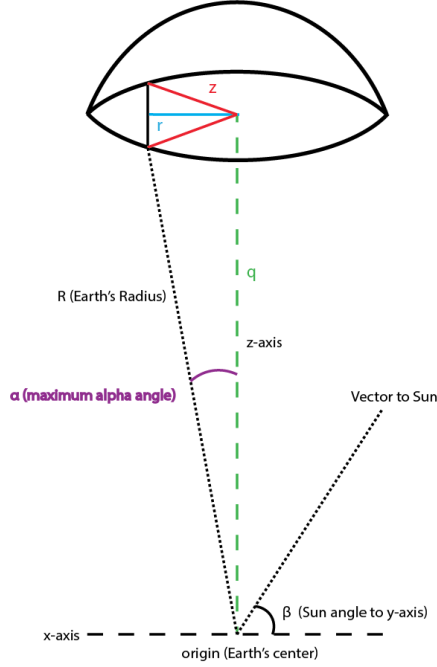


Figure 2: Satellite Cone of Visibility

From this figure we can derive an analytical formulation for the limits of theta (the typical spherical coordinates ‘theta’), which is our parameter for iteration about the z-axis.

Let’s define alpha hereafter not as the maximum alpha, but instead as any arbitrary alpha which goes from 0 to the maximum alpha, in other words an iteration alpha. An alpha of 0 describes the earth element on the z-axis. Thus:

$$q = R \cos(\alpha)$$

$$r = q \sin(\beta) = R \sin(\beta) (1 - \cos(\alpha))$$

$$z = R \sin(\alpha)$$

$$\psi = \pi - \arccos(r/z)$$

If the z coordinate of the Sun is positive, and the Sun is in the first quadrant, then ψ is our upper limit for θ . Small checks can handle variations in the sun position. For example if the Sun is in a different quadrant or either the Sun or satellite is exactly on any axis, the program elegantly handles edge cases. See comments in the Matlab code for explanations of specific edge cases.

Because the Sun is in the x - z plane, it shines equal amounts of light on both sides of the y -axis; thus our problem is symmetrical. We can use this convenience to reduce the number of area elements that need to be iterated over in half and for every element iterated return an identical result (the same unit vector and flux) except the y coordinate of the unit vector has been multiplied by -1 .

We desire to construct an accurate model in both the micro-state and macro-state. Although for our purposes here we are not interested in the net flux hitting the satellite, we still want to ensure that we choose a number of steps where the derivative of the net flux with respect to the time steps is small, in other words, our micro model still accurately reflects the macro state. The user of this program is strongly suggested to check this derivative of net flux for a large number of points in their model to ensure that they maintain this accuracy. In other words, don't choose a number of θ steps or α steps small enough that slight variation of either of these parameters introduces significant change in the net flux. An intelligent balance of efficiency versus accuracy will need to be chosen by the user of this program.

The final result of this method is a Matlab function which takes in albedo, a vector from the center of the earth to the Satellite, an identical one to the sun, and returns unit vectors and fluxes which correspond to the elements in the order they were iterated over.