High Geometric Fidelity Solar Radiation Pressure Modeling via Graphics Processing Unit

by

P. W. Kenneally

B.Eng., B.Arts., Australian National University, 2010

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Aerospace Engineering Sciences

2016

This thesis entitled:
High Geometric Fidelity Solar Radiation Pressure Modeling via Graphics Processing Unit
written by P. W. Kenneally
has been approved for the Department of Aerospace Engineering Sciences

Hanspeter Schaub

Daniel Kubitschek

Jay McMahon

Date

Kenneally, P. W. (M.S., Aerospace Engineering)

High Geometric Fidelity Solar Radiation Pressure Modeling via Graphics Processing Unit

Thesis directed by Prof. Hanspeter Schaub

Abstract - Solar radiation pressure (SRP), the force imparted on a spacecraft due to impinging solar photons, becomes a dominant dynamic perturbation for both interplanetary and above 1000 km Earth orbit altitude spacecraft missions. This thesis presents a method for the fast computation of spacecraft force and torque due to SRP considering a geometrically complex spacecraft model. The method uses the highly parallel execution capabilities of commodity Graphics Processing Unit (GPU) and the Open Graphics Library (OpenGL) vector graphics software library to render a Computer Aided Design (CAD) generated spacecraft model on the GPU. The SRP forces and torques are resolved per model facet in the custom-developed render pipeline. Using common commercial and open-source 3D mesh modeling tools the material properties are encoded with the CAD model to provide realistic specular, diffuse and absorption surface optical properties. A first order validation is carried out by comparing the method's force result to that provided by the analytic cannonball SRP model. Validation of more complex spacecraft geometry is achieved by comparison of the OpenGL method's computed force value with the force value computed from flight data for the same spacecraft. The method is successfully implemented as a modular component of the Autonomous Vehicle Systems Laboratory's (AVS Lab) spacecraft simulation framework to demonstrate the methods faster than real time online simulation capability. Finally the OpenGL method's online simulation capability is used to demonstrate the methods rapid simulation capability in aid of spacecraft maneuver design.

Dedication

I would like to dedicate this thesis to my family and friends, who have been so supportive of me throughout this process.

Acknowledgements

I would like to thank Dr. Hanspeter Schaub for his continued enthusiasm for his student's educations and pursuit of academic rigor. Additional thanks go to committee members, Dr. Jay McMahon for his insights regarding solar radiation pressure and Dr. Daniel Kubitschek for sharing his knowledge of the Mars Reconnaissance Orbiter program. Finally, I'd like to acknowledge the other researchers in the Autonomous Vehicle Systems Laboratory for their helpful contributions and brainstorming assistance.

Contents

Tables

Figures

Chapter 1

Introduction

1.1 Motivation

Effective orbit determination, maneuver and mission design, and numerical mission simulations require tools that enable accurate modeling of the spacecraft dynamical system. Solar radiation pressure (SRP), the momentum imparted to a body by impinging solar photons, can become the dominant non-conservative force above the Low Earth Orbit (LEO) regime[28]. As the dominant perturbation knowledge of the resultant forces and torque upon a body due to SRP are a primary consideration in the modeling and analysis of spacecraft operating in the high LEO region and above[5, 14].

A variety of methods have been proposed to model the dynamical effect of radiation pressure on the spacecraft motion. Furthermore, the methods combine both analytic and numerical techniques to varying degrees. This thesis investigates the application of a parallel computation methodology to evaluate the resultant force and torque on a spacecraft due to solar radiation pressure. The technique makes use of computer vector graphics and the highly parallel execution characteristics of the commodity Graphics Processing Unit (GPU) to evaluate with high geometric fidelity the perturbing effect of radiation pressure on an arbitrarily shaped spacecraft.

The significance of the dynamic perturbation of radiation pressure above LEO necessitates the need to provide sufficiently accurate dynamic models to the precision orbit determination process. Such a need is exemplified by the GPS satellite constellation where precision orbit determination enables many applications such as, geodesy, determination of Earth reference frames, and low Earth

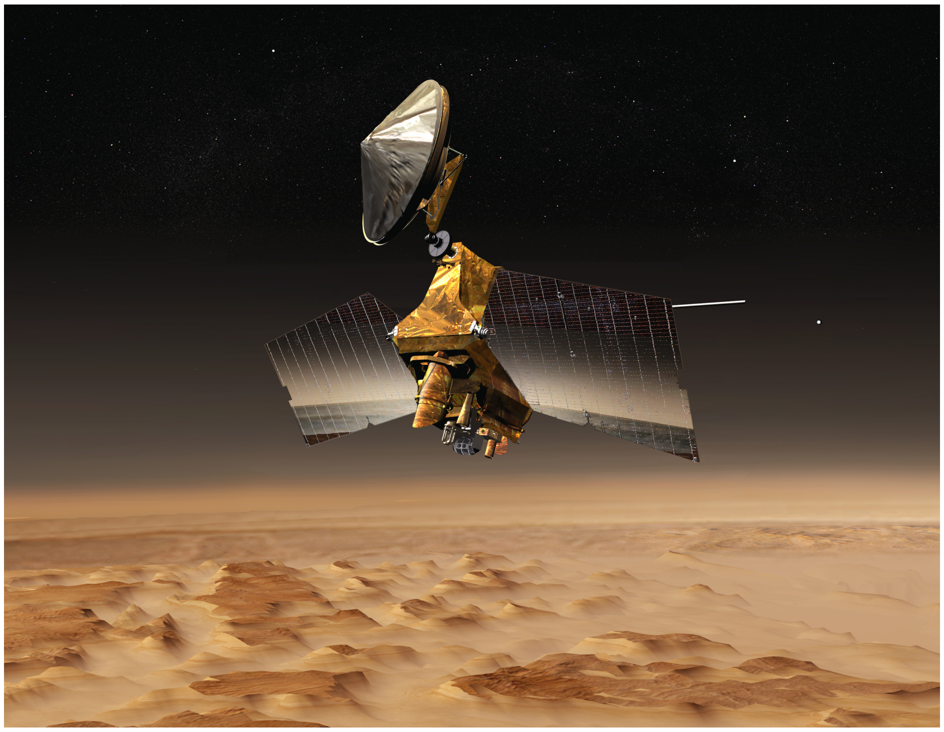


Figure 1.1: Mars Reconnaissance Orbiter (MRO)

orbiter tracking [2].

For deep space missions, after gravitational effects, SRP is a significant perturbation force of the spacecraft's dynamics [25]. The force contribution of SRP imparts a small change in velocity (ΔV) to the spacecraft for which the spacecrafts trajectory needs to account and a continuous torque which must be counteracted in order for the spacecraft to maintain a nominal attitude. For spacecraft which use reaction wheel based attitude control systems the continuous SRP induced torque is counteracted by applying an opposite resultant torque to the reaction wheels. If allowed to persist this exchange of angular momentum will eventually increase the reaction wheel's angular momentum to a saturation point where the reaction wheels are no longer able to counteract further external spacecraft torques. At a considerable cost mission designers and spacecraft engineers analyze the effects of SRP on a spacecraft's angular momentum state [18]. It is therefore, clear that the ability to rapidly assess the dynamics due to SRP for an arbitrary spacecraft geometry presents an opportunity for greater pre-launch insight of spacecraft dynamics and an improvment to pre-launch design team efficiency. In certain missions the small SRP induced ΔV is considered a perturbation, while in others as an actuator or desirable propulsive force. For example the Mars Reconnaissance Orbiter (MRO) mission team treated SRP as a perturbation and characterized the force and torque on the spacecraft so that they may account for its effect during cruise navigation and trajectory planning [31]. Conversely, solar sail spacecraft treat SRP as a desirable actuator to provide spacecraft thrust and control torques [7].

1.2 Current SRP Modeling Approaches

A survey of the current landscape of SRP research reveals a variety of modeling approaches. The most basic approach with regard to its analytic development is referred to as the cannonball model. The cannonball analytic model is given in Eq. (1.1), is computed from the surface area upon which radiation is incident A, solar flux $\Phi_{\odot}$, the spacecraft mass m, speed of light c, heliocentric distance to the spacecraft r, the sun unit direction vector $\hat{\boldsymbol{u}}$ and the reflection, absorption and emission characteristics of the spacecraft surface which are grouped together within the coefficient of reflection C_r. It is often the case that the C_r parameter is continually estimated and updated by an orbit determination effort. This model was most notably used during the LAEGOS missions and continues to prove useful for initial mission analysis[11].

$$a_{\odot} = -C_{\mathrm{r}} \frac{A\Phi_{\odot}}{Mc} \left(\frac{1AU}{r} \right)^2 \hat{u} \tag{1.1}$$

The cannonball model provides a useful, first-order approximation, however, due to its homogeneous material properties and symmetrical shape approximation it is incapable of resolving the SRP generated spacecraft torque. These torques are produced by an offset of the spacecraft's center of pressure from the center of mass (CP-CM). The CP-CM offset is created by a spacecraft's asymmetric geometry and the variation of the spacecraft's optical properties over the spacecraft surface. Inclusion of the induced torque is often achieved by employing an increased fidelity modeling approach to capture the macro effects of spacecraft shape on the CP-CM offset. A macro fidelity is often achieved by defining shape approximations of the spacecraft. A common shape approximation is to model the spacecraft bus and solar panels as multiple box and panels respectively [21]. Additionally the individual reflection, absorption and emission characteristics are kept distinct for each surface and set based on known spacecraft material properties[15].

However, the utility of shape approximation methods extends only so far as to capture the force and torque due to larger spacecraft surfaces. A trait of the shape approximation method is for much of the modeling uncertainty to be accounted for within an estimation process, where spacecraft surface area and optical properties are typically the estimated parameters [25]. While, useful for orbit determination of on-orbit spacecraft, accounting for such modeling uncertainty within an estimator provides little utility for pre-launch spacecraft simulation and design. As a result it is the generation and evaluation of a priori models, in which much work is being done. Notably Zeibart, proposes a SRP evaluation procedure which computes the body forces over all 4π steradian attitude possibilities [33]. Ziebart's approach is also capable of modeling self-shadowing by using ray-tracing techniques and spacecraft re-radiation via a spacecraft reduced thermal model. McMahon and Scheeres extend such an approach by aggregating the resultant SRP forces into a set of Fourier coefficients of a Fourier expansion[15]. The resulting Fourier expansion is available for both online and offline evaluation within a numerical integration process. Evaluation of the Fourier expansion in numerical simulation demonstrates successful prediction of the periodic and secular effects of SRP. Additionally, the Fourier coefficients may replace spacecraft material optical properties estimated during the orbit determination effort.

1.3 Parallel Computation of SRP Models

Prior to the past decade, evaluation of high geometric fidelity radiation pressure models, which included highly faceted models and ray tracing techniques, have been computationally slow. Previously, this slow computation time made such techniques unsuitable for faster than realtime spacecraft simulation [2]. More recently methods have been developed which make use of the parallel processing ability of multicore processors and GPUs. Tanygin and Beatty employ modern

GPU parallel processing techniques to provide a significant reduction in time-to-solution of Ziebart's "pixel array" method[1]. Further and inspiring the methodology presented in this thesis Tichey et al. use OpenGL, a vector graphics GPU software interface common in video games, to dynamically render the spacecraft model and evaluate the force of the incident solar radiation across a spacecraft structure approximated, as in computer aided design (CAD) model, by many thousands of facets [27].

The ability to model and compute, at orders of magnitude faster than real-time, the SRP forces and torques on flexible and time varying spacecraft structures presents compelling opportunities. Current pre-launch SRP evaluation approaches are capable of modeling the resultant dyanmics of an articulated spacecraft where the articulation motion is known prior to evaluation [32]. However, there are many instances in which the articulation motion and the spacecraft state are dependent on the myriad spacecraft control inputs and constraints [10]. Accounting for all possible permutations of the spacecraft dynamical state is further challenged by the inclusion of flexing in the spacecraft structure.

1.4 Application Possibilities of Faster Than Real Time Models

It is evident then that a method of SRP evaluation characterized by an ability to include time varying information of the spacecraft state and faster than realtime evaluation has potential for a wide range of applications. Effective modeling of the SRP induced perturbation of a spacecraft enables mission designers to consider SRP a valuable actuator rather than a disturbance. Such novel use of the SRP force in maneuver and mission design is exemplified by the MErcury Surface, Space ENvironment, GEochemistry and Ranging (MESSENGER) mission. The MESSENGER mission employed six planetary gravity assists during its journey to a Mercury orbit. The MESSENGER mission designers employed a solar sailing technique to perform each trajectory change maneuver (TCM) and accurately target each planetary flyby. Typical methods for performing TCM's use onboard thrusters to impart the required ΔV. However, using SRP as the TCM actuator allowed the MESSENGER team to perform TCM's with more accuracy and finer control due to the smaller magnitude of the SRP induced force[17]. Additionally, the MESSENGER team was able to reduce fuel and related structural accommodations in the spacecraft design to reduce overall mission cost[17]. Additional opportunities exist in the design and simulation of atypical spacecraft maneuvers. For example, large deployable structures, such as the IKAROS solar sail, would gain the ability to iteratively evaluate with greater fidelity the time varying control actuation of the solar sail during and after deployment [7].

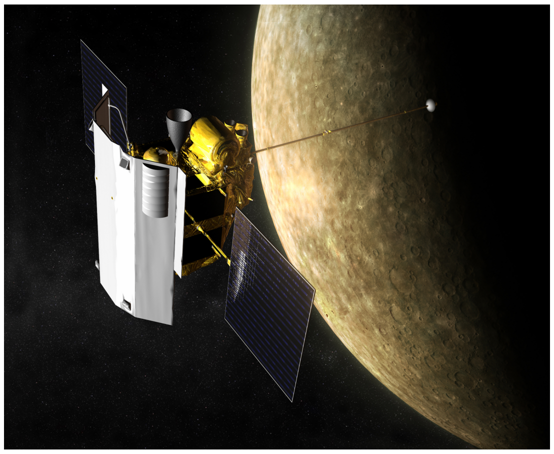


Figure 1.2: MErcury Surface, Space ENvironment, GEochemistry and Ranging (MESSENGER)

Maneuvers which utilize SRP as an actuator are further demonstrated by the design of the rescue maneuver for the Hayabusa spacecraft. Hayabusa, an astroid return mission, lost attitude control due to a failure of the spacecraft's reaction control system. Upon returning to a power positive state ground teams regained attitude control via the electric propulsion system at the expense of valuable fuel reserves. As a result, a cruise maneuver was designed which incorporated SRP to balance the torque induced by the swirling electric thrusters, thus saving fuel for the return journey to Earth[10].

In both the MESSENGER and Hyabusa examples mission designers used a variety of online and offline techniques to simulate and verify the beneficial effect of SRP on the spacecraft's trajectory and attitude. However, this modeling and verification consumes significant human resources to perform at high fidelity. In the case of MESSENGER, TCM maneuver planing began at minimum five weeks prior to the event [17]. Additionally, a-priori SRP models used in these analyses are typically not adjustable or tunable. If modeling parameter inclusion and accuracy requirements change during the lifetime of the mission a costly redesign process is required [2]. In this context a faster than real time SRP evaluation method presents the potential to reduce costs to a mission during it's operations phase.

1.5 Thesis Research Goals

This thesis presents a method for the computation of the spacecraft forces and torques using the highly parallel execution capabilities of GPUs. The method effectively leverages and re-purposes software tools and techniques from the computer graphics discipline. The forces and torques are resolved at a per facet level and summed over the surface of a Computer Aided Design (CAD) generated spacecraft model. Material properties are encoded into the computer model to provide realistic specular, diffuse and absorptive surface radiation interactions.

The primary research goals for this thesis are:

(1) Implement a method of high geometric fidelity SRP modeling using OpenGL and computer vector graphics techniques.

(2) Demonstrate the easy integration and use of the method as a modular component within the Autonomous Vehicle System Laboratory (AVSLab) software simulation framework.

(3) Validate the spacecraft force and torque results provided by the method and characterize the computational performance in regards to time to solution.

(4) Demonstrate that the OpenGL method may be used to design and simulate spacecraft missions in which SRP is used as an actuator rather than treated as a dynamic disturbance.

Beginning with chapter two an overview of relevant SRP modeling methods is presented. Particular modeling methods which range from first order analytic to high-fidelity methods are discussed to exemplify the significant aspects of various modeling approaches. In chapter three the OpenGL modeling method and its software implementation is presented in detail. Chapter four examines the performance and validation of the OpenGL method through comparison with existing models, some of which are presented in chapter two. The final chapter details the development and results of a notional deep space spacecraft utilizing solar radiation pressure torques to reduce the stored angular momentum in the spacecraft's reaction wheel devices. This work presents many opportunities for enhancements to be made to the OpenGL modeling approach and these opportunities and conclusions are discussed in the closing.

Chapter 2

Solar Radiation Pressure Models

In this chapter commonly used pre-launch solar radiation pressure modeling approaches are reviewed. This work is particularly interested in methods which use physical and engineering data to produce a pre-launch dynamic model. In contrast to the a-priori SRP modeling method, the GPS spacecraft focused ROCK and CODE models represent a second, empirical model development approach. These two models are 'tuned' by including significant portions of spacecraft tracking data into the model generation [25]. As a result, models such as ROCK and CODE are not discussed in this work. The models reviewed in this chapter include the analytic cannonball model, volume shape approximations, higher geometric fidelity faceted models and the 'pixel-array' method. Following the model survey, a selection of the evaluation considerations and methods for distilling the results of numerical models into simple and computationally fast data and analytic representations are provided. This chapter is concluded by acknowledging that ultimate model accuracy is achieved by accounting for all spacecraft radiation sources through a complete spacecraft radiation energy balance.

2.1 Cannonball Model

The earliest dynamic models used for solar radiation pressure approximated the body's optical properties as homogeneous and exposed surface area as constant. The model, commonly referred to as the 'cannonball model', was first used in practice during the LAEGOS mission [11]. The LAEGOS satellite was spherical in shape and covered in a repeating homogeneous pattern of mirrors

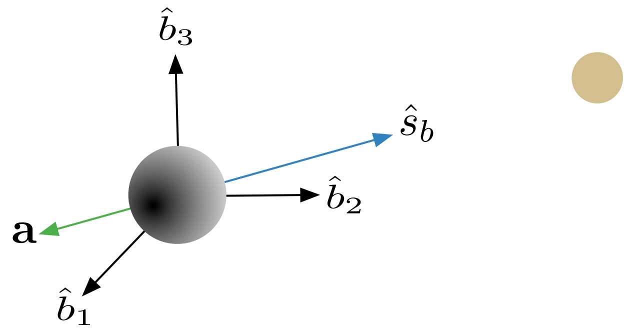


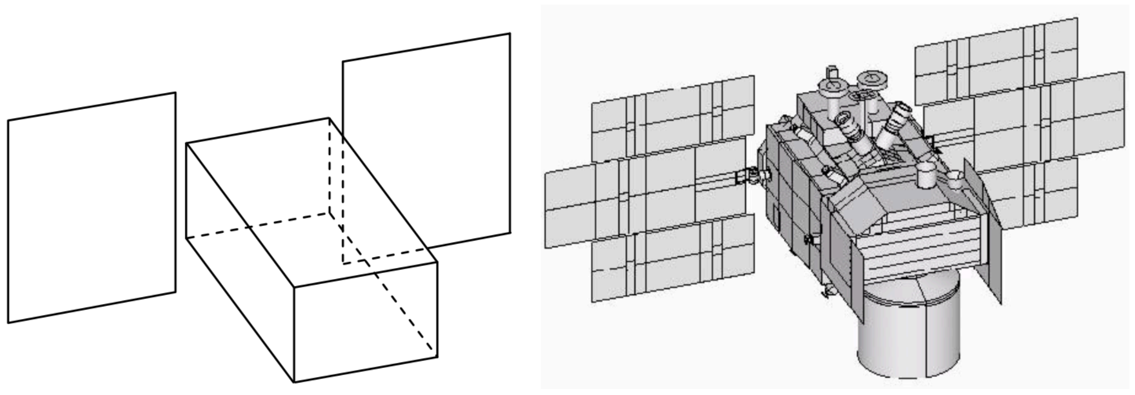
Figure 2.1: Cannonball model with the sun unit direction vector in the body frame $\hat{\boldsymbol{s}}_b$ and the resulting acceleration $\boldsymbol{a}$ in the opposite direction.

The dynamic effects of SRP on the LAGEOS satellite are analogous to the effects a cannonball shaped spacecraft would undergo [28]. The model given at Eq. (2.1) groups the body's optical properties together in a single parameter C_r, where $\Phi_\odot$ is the incident radiation pressure, A the spacecraft area projected into the plane perpendicular to the sun unit direction vector $\hat{\boldsymbol{u}}$, r the distance from the spacecraft to the sun and m the spacecraft mass [28]. The cannonball model approximates the accelerations for complex body shapes, however, it does not account for secondary photon reflections, capture SRP induced spacecraft torques and variation in the projected surface area or surface optical properties. Despite these issues, it is often used as part of first order analysis and, in the case of debris, orbital modeling when a body's shape and optical parameters may be undefined [13]. In situations where a body's optical properties are undefined the area A and coefficient of reflection C_r parameters may be included as estimated parameters in an orbit estimation effort.

$$a_\odot = -C_\mathrm{r}\frac{A\Phi_\odot}{mc}\left(\frac{1AU}{r}\right)^2\hat{\boldsymbol{u}} \tag{2.1}$$

2.2 Shape Approximation and Faceted Model

Increased model fidelity can be attained by generating a model based on shape approximation. Approximating the spacecraft as a combination of primitive volumes such as prisms, cylinders and spheres enables one to capture the macro variations in the spacecraft shape[6]. Furthermore, each



(a) ICESat modeled as a box and wing (b) ICESat spacecraft modeled with facets

Figure 2.2: ICESat modeled with simple shape approximations and with higher geometric fidelity using numerous facets [21]

face of a volume can be assigned individual surface optical properties to capture the variation in the spacecraft surface materials. A rudimentary shape approximation, often called a 'box and wing', is shown in Figure 2.2(a) where the ICESat spacecraft bus is approximated by a rectangular prism and its solar arrays as panels.

Increasing the fidelity beyond the 'box and wing' model can be achieved by departing from primitive volume approximation and defining the spacecraft surface as a collection of facets. A faceted model is capable of capturing an increased number of the variations in spacecraft surface optical properties in addition to more accurately representing the total spacecraft surface area impacted by radiation. A visual comparison of the increase in the physical spacecraft modeling accuracy provided by a faceted model is shown in Figure 2.2(b) where ICESat is modeled with facets rather than the primitive volumes of Figure 2.2(a).

Each facet posses absorptive, diffuse reflection and specular refelction coeficients, a_k, ρ_k and s_k respectively. To ensure a complete energy balance the resulting fractions of absorbed and reflected energy must all satisfy the relationship given at Eq. (2.3a). The coefficient of reflection C_r given in the cannonball model at Eq. (2.1) can be related to the per facet surface material properties, a_k, ρ_k (assumed Lambertian in character) and s_k, as shown at Eq. (2.3b). Evaluating the total force of a faceted model is achieved by summing the force contribution of each k^{th} facet, given at Eq. (2.2), given the facet's diffuse ρ_k and specular s_k material optical properties and θ_k, the solar angle of incidence measured from the facet normal unit direction vector [30]. It is known that secondary photon reflections, the impact of a spacecraft reflected photon onto a second part of the spacecraft, can significantly alter the resulting force direction. The faceted SRP model does not inherently capture secondary reflections. However the application of a ray tracing step during model evaluation can resolve the contributing force and torque due to secondary photon impacts [32].

$$\boldsymbol{F}_{\text{SRP}} = -P(R)\sum_{k=1}^{n} A_k \cos\theta_k [(1-s_k)\hat{\boldsymbol{u}} + 2(\frac{\rho_k}{3} + s_k \cos\theta)\hat{\boldsymbol{n}}_k] \tag{2.2}$$

$$s_k + \rho_k + a_k = 1 \tag{2.3a}$$

$$1 + s_k + \frac{2}{3}\rho_k = C_{\text{r}} \tag{2.3b}$$

2.3 Ray Tracing Methods

Ray tracing is a technique particularly employed in the computer graphics discipline which generates an image by tracing the path of a light ray through points in an image plane and simulating the effects of its interactions with the surface materials of a computer-generated models. Ray tracing SRP modeling methods directly model the projection of an array of solar rays and trace the path of each ray throughout its interactions with the spacecraft. Each solar ray is initiated from a point within an array that is oriented perpendicular to the Sun-spacecraft vector $\hat{\boldsymbol{u}}$ and is of sufficient size to ensure full coverage of the spacecraft model[33].

Each projected ray is tested for intersections with the modeled spacecraft to determine the illuminated and shadowed model facets. Detected intersections will spawn a new ray(s) dependent on the specular and diffuse surface properties, where as absorptive surface will not spawn a new ray. Purely specular surfaces will spawn a single reflected ray, while diffusely reflecting surfaces will spawn multiple rays where the radiation intensity of each new ray is reduced commensurate to the portion of light it represents [20]. The resultant force and torque on the spacecraft due to

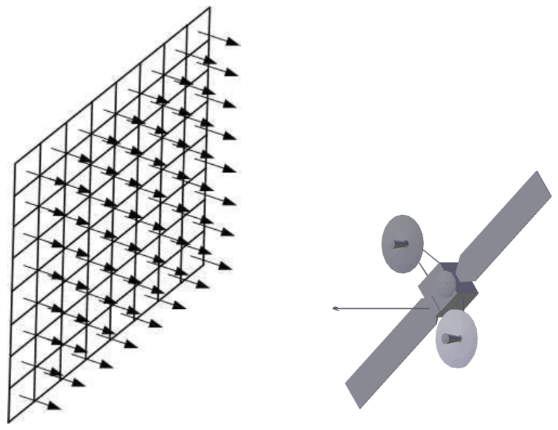


Figure 2.3: Ray tracing methods model the path traced by an array of virtual sun rays given each ray's spacecraft surface interactions.

SRP is computed as the sum of the force contribution due to each absorbed, specular and diffusely reflected traced ray.

2.4 Electromagnetic Energy Balance Model

The highest level of fidelity can be achieved by seeking to account for a complete spacecraft electromagnetic energy balance. Such an energy balance includes, in addition to SRP, is a combination of other spacecraft dynamics models. Zeibart et. al. demonstrates high fidelity radiation pressure modeling by combining a micro faceted model, an anisotropic thermal re-radiation force model, planetary albedo force model and antenna thrust effects [32]. While each of the aforementioned force contributions are significant components of a complete radiation force model, this work does not directly include these contributions. Instead a focus is given to the speed and utility offered by the evaluation methodology.

Chapter 3

OpenGL Solar Radiation Pressure Evaluation Method

The video game and animated video industries have driven the pursuit to create more vivid and realistic artificial worlds. This pursuit has resulted in highly optimized vector graphics software and GPU computer hardware capable of carrying out many thousands of floating point operations in parallel [19]. While these artificial worlds are visually persuasive, their implementation of electromagnetic radiation physics is understandably inaccurate. However, it is the parallel hardware and efficient vector graphics software implementations which may be used to simplify the steps of the SRP computation with great effect.

This chapter details the methodology employed to evaluate SRP induced force and torque vectors with high geometric fidelity. An introduction to the theory and software considerations of the vector graphics and GPU application programming interface (API) called OpenGL is given. It is shown how the OpenGL graphics rendering 'pipeline' is constructed and then how this pipeline is manipulated to process the per facet SRP computation. Noteworthy code listings are discussed to demonstrate the specific stages within the render pipeline which are customized for a unique purpose.

3.1 Vector Graphics

Computer vector graphics systems process sets of points defined in three dimensions. These points, referred to as vertices, are combined into groups of two or more to define lines, triangles and polygon shape primitives [24]. Using a vector graphics based data description, a spacecraft can

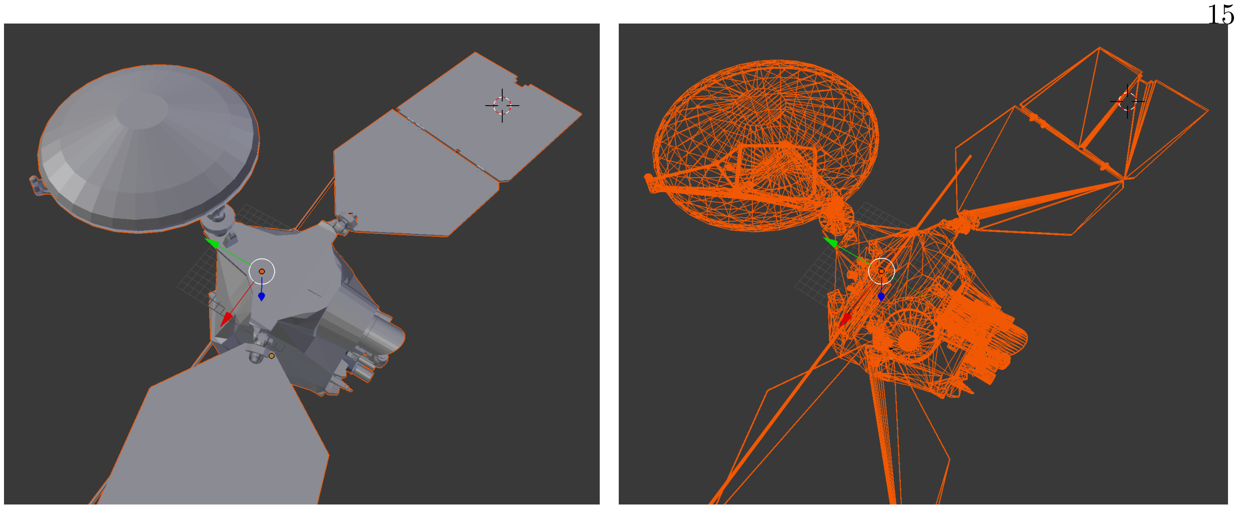

(a) Solid MRO CAD model (b) MRO triangle primitives

Figure 3.1: Model and wire frame views of the Mars Reconnaissance Orbiter computer generated model

be modeled as a polyhedra, an approximation of a system of many thousands of small triangle or polygon shape primitives. This is the same method employed where a CAD model is approximated as a system of many thousands of small triangle primitives. Where a flat plate surface may be modeled by a single rectangle, resolved into two triangles, a curved surface may be approximated by many smaller triangles. Such an approximation is shown in Figures 3.1(a) and 3.1(b). In these figures it is evident that the Mars Reconnaissance Orbiter (MRO) high gain antenna shown in Figure 3.1(b) is approximated by many thousands of primitives while the solar panels are approximated by 10 primitives. The density of primitives used to approximate the spacecraft's structure is what characterizes a model's geometric fidelity.

The representation of each vertex as a vector in $\mathbb{R}^3$ space allows one to map vertices between various coordinate frames. There are three common vector graphics coordinate frames;

- Camera frame: a right handed system with it's origin located at the hypothetical scene view point.
- World frame: a right handed global coordinate frame containing all scene objects.
- Model frame: a right handed frame in which the vertices, forming a specific scene object,

are defined.

Each of these coordinate frames are used to compute the object optical properties, light sources and lighting interactions between scene objects to produce a rendered on-screen scene[24]. The model and world reference frames can be treated analogously to the familiar fundamental frames encountered in the subject of analytic dynamics; those are the body and inertial frames. For the remainder of this presentation the model and world frames are refereed to as the body and inertial frames respectively. While this terminology is unconventional to those working in the computer graphics industries, it is likely to be more within the technical experience of the aerospace-related audience of this thesis.

3.2 Open Graphics Library

As the name suggests, the OpenGL SRP method employs the Open Graphics Library (OpenGL) to facilitate the processing of the spacecraft model vertices and primitives to compute the spacecraft dynamics due to SRP. OpenGL is a language independent API for rendering computer vector graphics [24]. The API provides tools to send, retrieve and process data on OpenGL compliant GPUs. As with any software API, the capability of the software evolves and changes with time. As a result, all references made to the capabilities of the OpenGL API made in this thesis refer specifically to a minimum version of OpenGL 4.0.

A rendered scene is generated by processing each vertex and primitive definitions within the OpenGL pipeline. The OpenGL pipeline allows for various stages to be programmable. These programmable stages are termed shader programs where each shader is a mini-program which serves to process vertices and primitives within the scene. Shader programs are written using the OpenGL Shader Language (GLSL) and each shader stage has a defined set of data types as inputs and outputs, which are passed along the pipeline to subsequent stages.

Each execution of each shader stage operates on a single vertex or a set of vertices that define a shape primitive. Each of the vertices or primitives, are processed in parallel where thousands of shader program instances are executed simultaneously for each stage in the pipeline. It is the highly parallel per vertex/primitive operation (many thousands of evaluations occurring simultaneously) for which GPU devices have been specifically designed. CPUs are designed for general desktop computing where the nature and priority of execution tasks changes often, requiring the processor to be capable of multi-threaded anticipatory code execution. In contrast, a GPU is composed of hundreds of cores and is capable of processing thousands of the same thread simultaneously. A GPU with hundreds of cores can reduce computation time of parallel problems by more than 2 orders of magnitude over a CPU alone [9]. As shown in Figure 3.2, the presence of a CPU's execution cache and the absence of a cache in the GPU is a clear manifestation of the difference in the computational requirement of the two processor architectures.

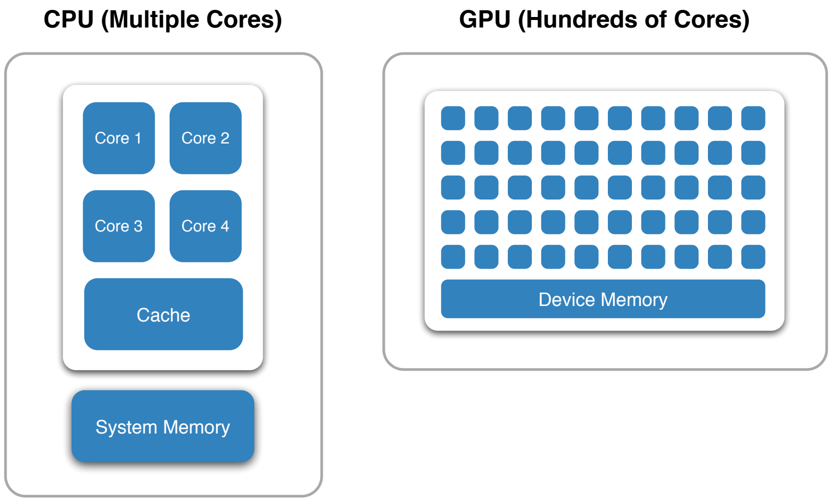


Figure 3.2: A notional comparison of CPU and GPU architectures, where the CPU possess a large on-chip cache and the GPU does not.

3.3 The OpenGL Pipeline

As shown in Figure 3.3 a minimally valid OpenGL pipeline requires the implementation of the vertex and fragment shader stages. The addition of further shader stages allows the software developer to create a custom render pipeline.

To begin, the spacecraft model is defined by reading in the vertex specifications which are

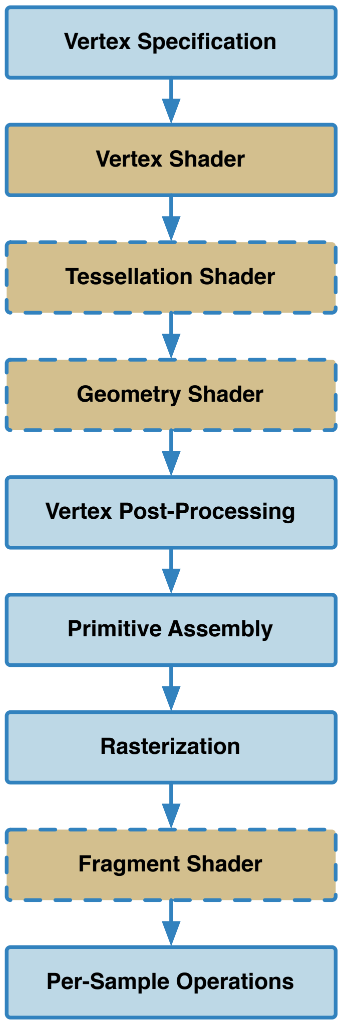


Figure 3.3: The default OpenGL pipeline. Gold fill stages are custom shader stage implementations. Blue full stages are OpenGL built-in shader stages. A Solid border stage indicates a required pipeline stage, while a broken border indicates optional stages.

produced using an external CAD or 3D modeling tool. The vertex specifications are processed and sent to the GPU via OpenGL as a vertex buffer object (VBO). As shown in Figure 3.3, vertex processing, primitive assembly and rasterization are portions of the OpenGL render pipeline which are required (when displaying a final screen image) and may not by customized. The following list describes the operations performed by each of the key vertex, geometry and fragment shader stages.

- Vertex Shader (VS): processes the individual vertices of the model having vertex data as both input and output. The VS is used to perform setup for later shader stages by performing coordinate frame transformations on vertex data by mapping vertices from the model coordinate frame to the world coordinate frame. As shown in Figure 3.3 the vertex shader is a programmable and required stage in the pipeline.

- Geometry Shader (GS): operates on a single primitive and may output zero or more primitives. A GS allows for one primitive to be operated upon and if desired arbitrarily replicated. As notionally demonstrated in Figure 3.4, the GS is able to remove primitives, re-tessellate primitives splitting them and outputting many primitives for a single input and mapping the vertex values of the primitive. Upon completion of processing a Geometry shader outputs zero or more simple primitives.

- Fragment Shader (FS): is executed after the pipeline has rasterized the projected scene. To rasterize the scene, the vertices of each primitive are mapped from $\mathbb{R}^3$ to $\mathbb{R}^2$ screen space samples. Each primitive, as demonstrated in Figure 3.4, is converted to a fragment which can then be mapped to a set of on screen pixels. The output of a FS is a depth value, a possible fragment placement value, and zero or more color values to be written to the buffers in the current frame buffers. The final frame buffer data is ultimately displayed on the screen.

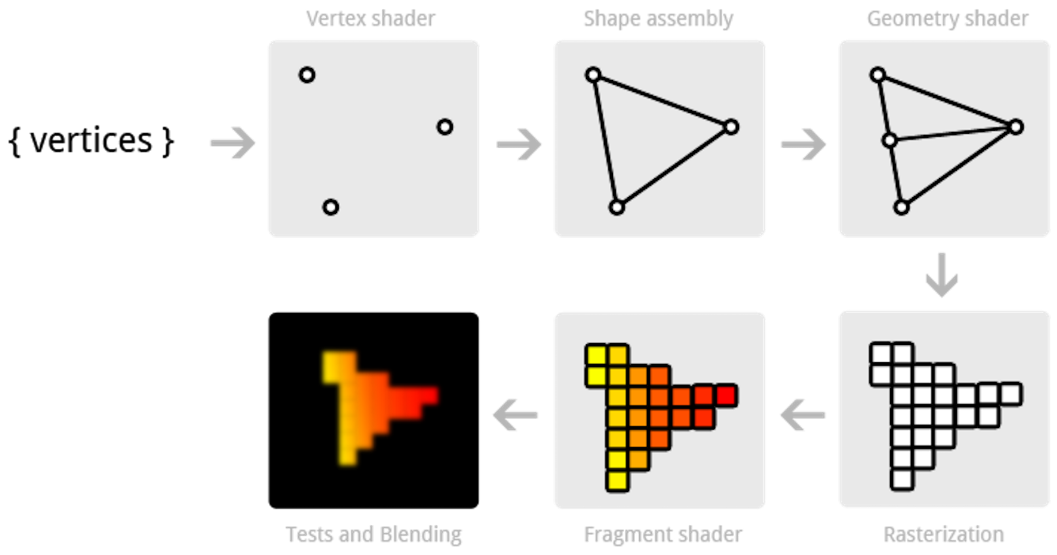


Figure 3.4: OpenGL shader stages and their nominal per vertex, geometry and fragment operation

3.4 Computing SRP via A Custom Pipeline

The OpenGL SRP method utilizes a custom rendering pipeline to compute the SRP forces and torques over the spacecraft surface. Implementing a custom set of shader programs at each pipeline stage allows for the direct manipulation of the render pipeline to support the computation of SRP dynamics.

The OpenGL SRP pipeline employs custom vertex, geometry and fragment shaders. As shown in Figure 3.3 all three of these shader stages are customizable, however, the geometry shader is the critical optional shader stage included in the pipeline. It is the inclusion of the geometry shader stage that allows for the per primitive SRP computations to occur. To capture the per primitive SRP computations the OpenGL specification defines the Transform Feedback functionality. The Transform Feedback is an OpenGL utility enabling the capture of data (primitives, vertices and custom data) generated by the vertex and geometry shader stages into a buffer that may be routed back to a CPU based process. As illustrated in Figure 3.5, the custom geometry shader computes the radiation interaction for each primitive and the per-primitive radiation force and torque data are stored in a buffer. This stored data represents the rendered force and torque for a given spacecraft state and is returned to the CPU as a final SRP computation result where it can be incorporated as a dynamic contribution in a numerical simulation.

3.4.1 Custom Vertex Shader

To present a rendered spacecraft for visual debugging custom vertex and fragment shaders are implemented. The vertex shader prepares the vertex data for the following render stages by mapping vertices defined in the spacecraft body frame to the inertial frame, then camera view frame, and finally screen projection coordinates. Reference frame mappings in OpenGL are defined as homogenous transforms. The homogenous transformation accounts for mapping a point between two frames that have different orientations and origins [23]. It is assumed that, because the OpenGL method implementation renders only the spacecraft, the body frame is taken as coincident with the inertial frame. Therefore, only the body frame orientation relative to the inertial frame is accounted for where the spacecraft-sun distance is computed separately. The body relative to inertial orientation is computed on the CPU given the spacecraft attitude description. In this implementation the spacecraft's attitude dynamics use the three component modified Rodrigues parameters (MRPs) vector $\boldsymbol{\sigma}$, to describe the orientation of the body frame with respect to the inertial frame [23]. The orientation described by $\boldsymbol{\sigma}$ can be expressed as a direction cosine matrix $[BN]$ in terms of the σ vector components σ_1, σ_2 and σ_3. To map a vector expressed in the body frame to the inertial frame, the direction cosine matrix's orthognality property is exploited to yield the inverse mapping $[NB]$ from the transpose $[BN]^T$. The resulting mapping of a body frame defined vertex to an inertial frame defined vertex is shown in Eq. (3.1) [23].

$$^{\mathcal{N}}\boldsymbol{v} = [NB]^{\mathcal{B}}\boldsymbol{v} \tag{3.1}$$

The final 4 × 4 homogeneous transformation matrix is constructed as shown in Eq. (3.2), using the direction cosine matrix $[NB]$ to account for the relative frame orientations and setting all other off-diagonal terms to zero and the final diagonal term to one to produce no translation of the vertex. A value of one is appended as the fourth vertex vector entry to produce a consistent

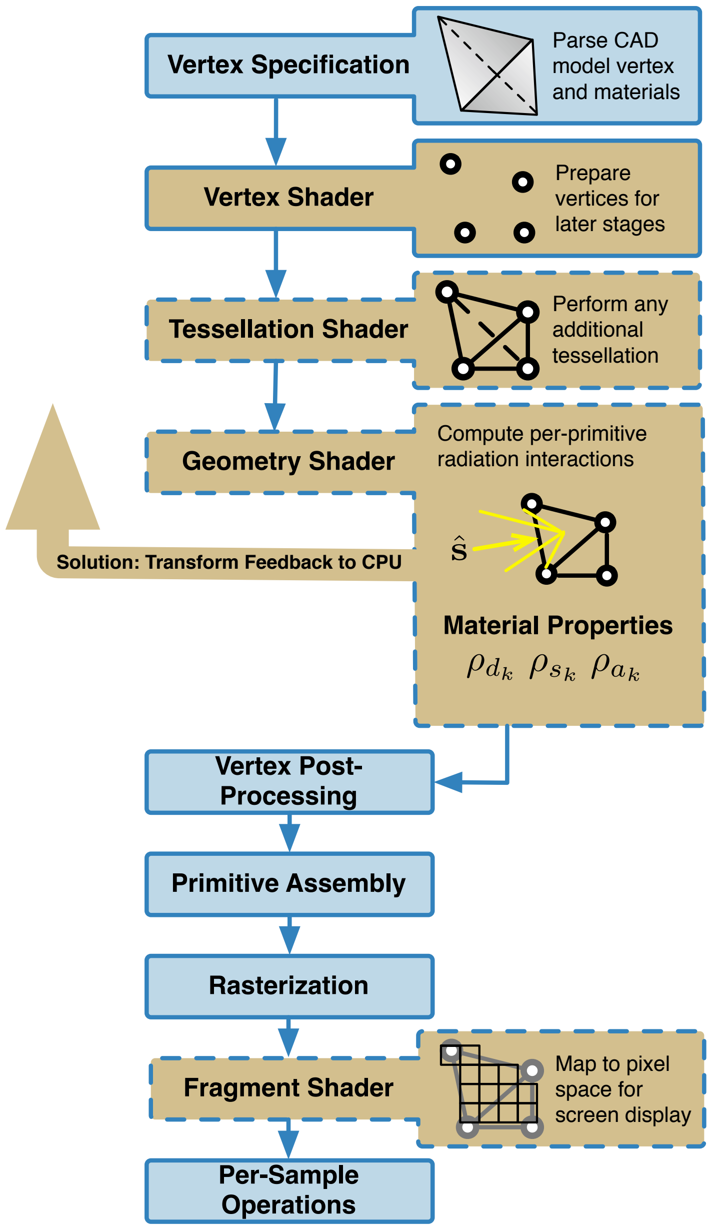

Vertex Specification
Parse CAD model vertex and materials
Vertex Shader
Prepare vertices for later stages
Tessellation Shader
Perform any additional tessellation
Geometry Shader
Compute per-primitive radiation interactions
$\hat{s}$
Material Properties
ρ_{d_k} ρ_{s_k} ρ_{a_k}
Solution: Transform Feedback to CPU
Vertex Post-Processing
Primitive Assembly
Rasterization
Fragment Shader
Map to pixel space for screen display
Per-Sample Operations

Figure 3.5: The custom OpenGL render pipeline demonstrating the discrete steps carried out by shader operations. Gold fill stages are custom shader stage implementations. Blue fill stages are OpenGL built-in shader stages. A Solid border stage indicates a required pipeline stage, while a broken border indicates optional stages.

matrix multiplication expression when performing the homogeneous transform mapping given in Eq. (3.3)[23].

$$[\mathcal{NB}] = \begin{bmatrix} NB & 0_{3\times1} \\ \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} {}^{\mathcal{N}}\boldsymbol{v} \\ \\ 1 \end{bmatrix} = [\mathcal{NB}] \begin{bmatrix} {}^{\mathcal{B}}\boldsymbol{v} \\ \\ 1 \end{bmatrix} \tag{3.3}$$

3.4.2 Custom Geometry Shader

The SRP force and torque calculations are carried out within the geometry shader stage. As shown in Figure 3.5 the geometry shader stage execution follows the vertex shader. Each geometry shader instance receives as its input three vertices ($\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$) defining a single spacecraft model triangle. Vertices are ordered counter-clockwise by default which allows for easy computation of face normals. Additionally, the value for solar flux $\Phi_\odot$ and the sun unit direction vector defined in the body frame $\hat{\boldsymbol{u}}_B$ are provided as constants accessible by each shader instance. The force for the k^{th} triangle primitive, $\boldsymbol{F}_{\odot_k}$, is evaluated in the spacecraft body frame using the expression shown at Eq. (3.4), where $P(|\boldsymbol{r}_\odot|)$ is the solar radiation pressure scaled by the heliocentric distance to the spacecraft [22]. It is important to note that the relationship given at Eq. (3.4) assumes that the energy absorbed by the satellite surfaces is instantaneously re-radiated as heat in a Lambertian dispersion. This assumption holds for a range of spacecraft surface materials modeled as having zero thermal capacity [4]. An example of such a zero thermal capacity material is multi-layer insulation (MLI).

$$\boldsymbol{F}_{\odot_k} = -P(|\boldsymbol{r}_\odot|)A_k\cos(\theta_k)\left\{(\rho_{a_k} + \rho_{d_k})(\hat{\boldsymbol{u}}_B + \frac{2}{3}\hat{\boldsymbol{n}}_k) + 2\rho_{s_k}\cos(\theta_k)\hat{\boldsymbol{n}}_k\right\} \tag{3.4}$$

The triangle primitive area A_k is computed in Eq. (3.5), where vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ define the edges of the primitive defined by the vertices.

$$\boldsymbol{e}_1 = \boldsymbol{v}_2 - \boldsymbol{v}_1 \tag{3.5a}$$

$$\boldsymbol{e}_2 = \boldsymbol{v}_3 - \boldsymbol{v}_1 \tag{3.5b}$$

$$A_k = \frac{1}{2}\|\boldsymbol{e}_1 \times \boldsymbol{e}_2\| \tag{3.5c}$$

The sun angle of incidence θ_k as given by Eq. (3.6), is simply the dot product of the primitive surface normal $\hat{\boldsymbol{n}}_k$ and the sun unit vector $\hat{\boldsymbol{u}}$. A primitive is judged to be illuminated in cases where the inequality at (3.6c) is satisfied.

$$\hat{\boldsymbol{n}}_k = \frac{\boldsymbol{e}_1 \times \boldsymbol{e}_2}{\|\boldsymbol{e}_1 \times \boldsymbol{e}_2\|} \tag{3.6a}$$

$$\theta_k = \hat{\boldsymbol{n}}_k \cdot \hat{\boldsymbol{u}}_B \tag{3.6b}$$

$$0 < \theta_k \leq 1 \tag{3.6c}$$

The parameters ρ_{s_k} and ρ_{d_k} in Eq. (3.4) are respectively the specular and diffuse reflection coefficients of the material definition associated with the primitive. Additional radiation pressure sources such as albedo from planetary bodies may be accounted for through an additional evaluation of Eq. (3.4) for each primitive in where $P(|r|)$ contains the the albedo radiation pressure. Following the force computation, the torque $\boldsymbol{L}_k$ contribution of a single primitive, as given in Eq. (3.7), is computed as the cross product of the vector defined from the body frame origin to the primitive's centroid $\boldsymbol{c}_k$ and the per primitive force $\boldsymbol{F}_{\odot_k}$. The centroid is defined as the geometric center of the triangle primitive and corresponds to the point of action of the force on the primitive.

$$\boldsymbol{L}_k = \boldsymbol{c}_k \times \boldsymbol{F}_{\odot_k} \tag{3.7}$$

A simplified geometry shader code implementation is shown in Listing 3.6. To output the final evaluated primitive the geometry shader must make three calls to the built-in GLSL function

`EmitVertex()`, one for each vertex, and a final call to `EmitPrimitve()` as shown on lines 39 and 42 in listing 3.6. The `EmitVertex()` and `EmitPrimitive()` functions signal to the OpenGL pipeline that geometry shader processing is complete and that a vertex and the final primitive are ready to be passed along the pipeline. The force and torque computations are made when processing the first vertex of a primitive and no operations are executed for the remaining two vertices. The GPU executes many thousands of geometry shader instances in parallel resulting in the force and torque contributions of many thousands of primitive being evaluated simultaneously. To retrieve the resulting force and torque from the GPU a Transform Feedback buffer mechanism is used. The Transform Feedback feature of OpenGL allows for the retrieval of data from the GPU process to the CPU process via an OpenGL buffer [8] . Once the force and torque data is returned to the CPU the values are summed to evaluate the total spacecraft force and torque vectors due to SRP.

3.4.3 Custom Fragment Shader

The fragment shader does not contribute to the force and torque evaluation. It transforms the rendered model into screen space coordinates to provide the final screen space pixel definition. The on-screen view generated by the vertex and fragment shader stages provides important first order confirmation of a correctly implemented custom shader pipeline by displaying the orientation and the sunlit facets of the spacecraft.

The custom pipeline is designed to support both the SRP computation and the final output of the rendered spacecraft for visual debugging and user interface utilities. The fragment shader stage is an optional stage and may be omitted by toggling, true or false, the OpenGL parameter `GL_RASTERIZER_DISCARD`. When only concerned with the SRP computation the fragment shader stage can be ignored and its output discarded. With no image being rendered to the screen a significant speed increase is provided to the final SRP calculation.

```
1  forceSrp = dvec3(0.0,0.0,0.0);
2  torqueSrp = dvec3(0.0,0.0,0.0);
3  cg = dvec3(0.0,0.0,0.0);
4  area = 0.0;
5  cosTheta = 0.0;
6  for (int i = 0; i < 3; i++)
7      {
8          if (i == 0) {
9              dvec3 edge1 = dvec3(gs_in[1].position_modelSpace - gs_in[0].position_modelSpace);
10             dvec3 edge2 = gs_in[2].position_modelSpace - gs_in[0].position_modelSpace;
11             dvec3 faceNormal = cross(edge1,edge2);
12             dvec3 nHat_B = normalize(faceNormal);
13             cosTheta = dot(nHat_B, sHat_B);
14             double P_sun = solarFlux/c; // solar radiation pressure
15
16             // Is the primitive illuminated ?
17             if (cosTheta > 0.0 && cosTheta <= 1.0) {
18                 area = 0.5*length(faceNormal);
19                 V[0] = gs_in[0].position_modelSpace.xyz;
20                 V[1] = gs_in[1].position_modelSpace.xyz;
21                 V[2] = gs_in[2].position_modelSpace.xyz;
22                 cg = (V[0] + V[1] + V[2])/3.0;
23
24             forceSrp = -P_sun*area*cosTheta*((rho_a + rho_d)*(sHat_B + (2.0/3.0)*nHat_B) + 2.0*rho_s*cosTheta*nHat_B);
25                 torqueSrp = cross(cg,forceSrp);
26             } else {
27                 // No processing as the primitive is not illuminated
28             }
29         } else {
30             // No processing for verticies 2 and 3
31         }
32         // Pass through vertex shader per vertex values to the fragment shader
33         gs_out.position_worldSpace = vec3(gs_in[i].position_worldSpace);
34         gs_out.normal_cameraSpace = vec3(gs_in[i].normal_cameraSpace);
35         gs_out.UV = vec2(gs_in[i].UV);
36         gs_out.eyeDir_cameraSpace = vec3(gs_in[i].eyeDir_cameraSpace);
37         gs_out.lightDir_cameraSpace = vec3(gs_in[i].lightDir_cameraSpace);
38         gl_Position = gl_in[i].gl_Position;
39         EmitVertex();
40     }
41
42     EndPrimitive();
```

Figure 3.6: Core OpenGL Shader Language code of the custom geometry shader. The force and torque computations are given at lines 24 and 25.

Chapter 4

OpenGL Method Validation

The focus of this chapter is to validate the OpenGL method by demonstrating agreement with other well proven SRP evaluation methods and models. Firstly, good agreement in the force resolution yielded by the simple analytic cannonball model and the OpenGL method is shown. Secondly, a considerably more geometrically complex MRO interplanetary spacecraft is submitted for processing. The OpenGL method generated force is compared to the force data retrieved by the MRO navigation operations team during Earth to Mars cruise. The remaining half of this chapter describes a demonstration of the OpenGL method as a modular component within the AVS Lab spacecraft simulation framework. As part of the simulation tool a range of time varying data such as orbital elements, SRP force and torque are analyzed..

4.1 Method Verification

The initial validation is performed by comparing results from an analytic cannonball model and a sphere shaped spacecraft model within the OpenGL method. The values for the LAGEOS II spacecraft, given in Table 4.1, are used in both evaluations. A spherical model spacecraft is generated to replicate the LAGEOS II spacecraft parameters. Figure 4.1 illustrates the spherical spacecraft model being evaluated using the OpenGL method. The perturbative acceleration due to SRP as given by the cannonball model and the OpenGL method are given in Table 4.2. The OpenGL method yields a resultant torque of 1.51×10^{-12} Nm. However, it is expected that a perfectly spherical object yields zero torque. The non-zero torque value returned by the OpenGL model is due to the faceted nature of the model not being perfectly spherical. It is observed that by increasing the number of vertices and therefore primitives in the model (better approximating a sphere), the resulting torque value approaches zero. The agreement between the two simple evaluations provides confidence of the methods correctness.

Table 4.1: LAGEOS II spacecraft parameters used for computation of SRP by cannonball model and OpenGL model

LAGEOS II Attribute	Value
mass	405.38 [kg]
area	0.2817 [m^2]
Φ (at 1 AU)	1.38$\times$10^3 [W/m^2]
C_r	1.12

Table 4.2: LAGEOS II spacecraft SRP induced acceleration computed by cannonball and OpenGL models

Model	SRP Acceleration $a_{\odot}$
Cannonball	3.56$\times$10^{-9} [m/s^2]
OpenGL	3.60$\times$10^{-9} [m/s^2]

An example evaluation of a complex spacecraft geometry is shown in Figure 4.2. In this evaluation a 14750 primitive model of the Mars Reconnaissance Orbiter (MRO) is processed at a heliocentric distance of 1AU where the sun vector as defined in the body frame is $\hat{\boldsymbol{u}}_B = [0, -1, 0]^T$. Approximate material optical properties are assigned to the spacecraft bus whereas the solar array emissivity and diffusivity, as quoted by You et. al, are set at 0.12 and 0.05 respectively. The SRP force value as determined by the MRO navigation team post launch of the spacecraft is $a_{\odot} \approx 9\times10^{-11}$ km/s^2 [31]. The OpenGL method computed SRP acceleration is $a_{\odot} = 8.51\times10^{-11}$ km/s^2. The uncertainty bounds on the provided MRO navigation team SRP force value is unknown. However, the small difference between these two values is a promising indication that the OpenGL modeling method can offer a pre-launch SRP force accuracy close to that achieved after on orbit small force calibration exercises. More promising is to acknowledge that this OpenGL method result does not yet include modeling of thermal re-radiation and secondary photon impacts. The MRO navigation team found that thermal re-radiation in particular was a significant contributor

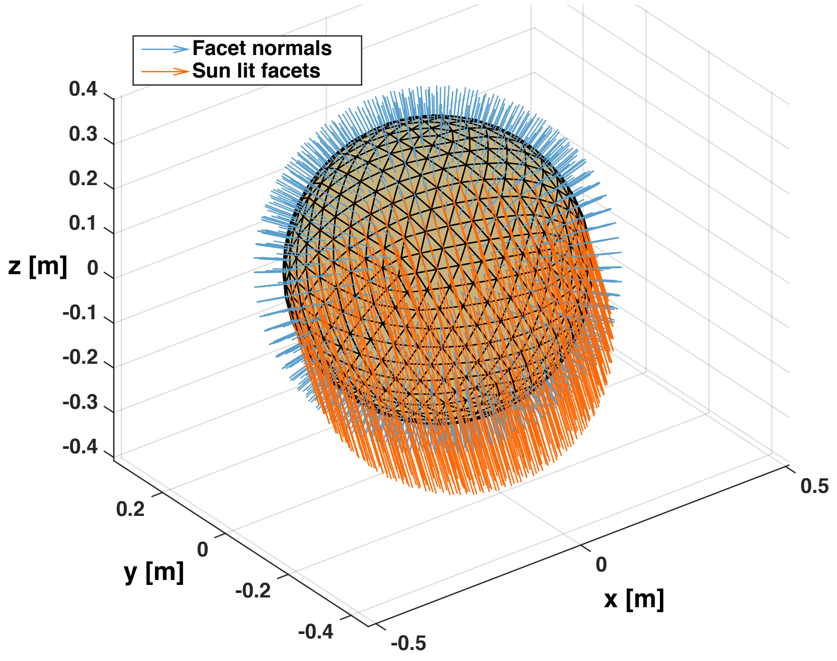


Figure 4.1: A visualized single evaluation of a 1280 primitive, LAGEOS size satellite. Orange vectors indicate $\hat{\boldsymbol{u}}_B$ incident on primitives. Blue vectors indicate primitive face unit normal vector $\hat{\boldsymbol{n}}_B$

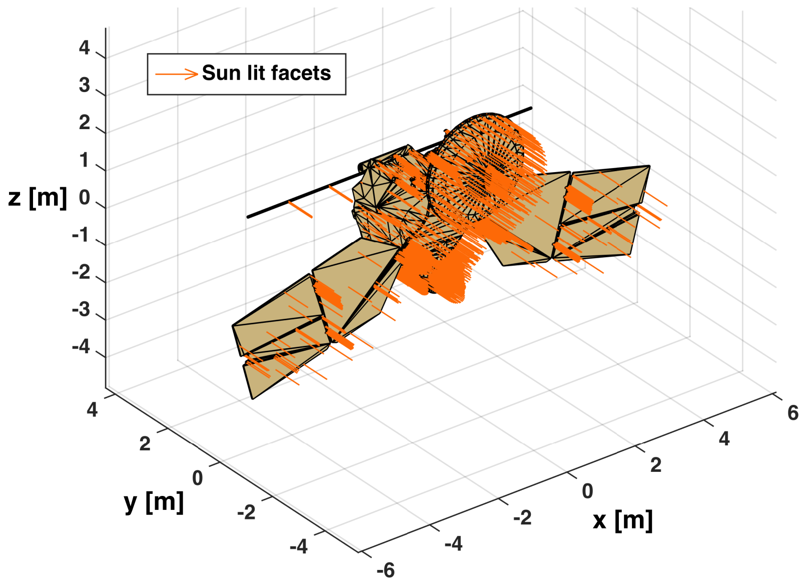


Figure 4.2: A single evaluation of a 14750 primitive, Mars Reconnaissance Orbiter. Orange vectors indicate $\hat{\boldsymbol{u}}_B$ incident on primitives.

to the total observed small forces due to radiation pressure [31]. It is expected that future near term modeling process additions such as thermal re-radiation and secondary photon impacts will allow for an improved fidelity of the spacecraft physical processes.

Additional validation of both of the above results is performed using an equivalent MATLAB based tool set. The MATLAB based tool parses in the vertex, primitive, and face normal data for the generated spacecraft CAD model. The tool set is capable of performing the same per-primitive SRP evaluation as the OpenGL method. However, due to its serial execution CPU based implementation the tool set is only used to compute the force and torque produced upon the spacecraft at a single sun heading.

4.2 Integration of OpenGL Method With AVS Lab Astrodynamics Simulation Framework

A primary goal of this work is to integrate the OpenGL method within a modular spacecraft simulation software framework. To demonstrate the effective inclusion of the OpenGL method within a simulation framework the method's code base was developed to allow for simple integra-

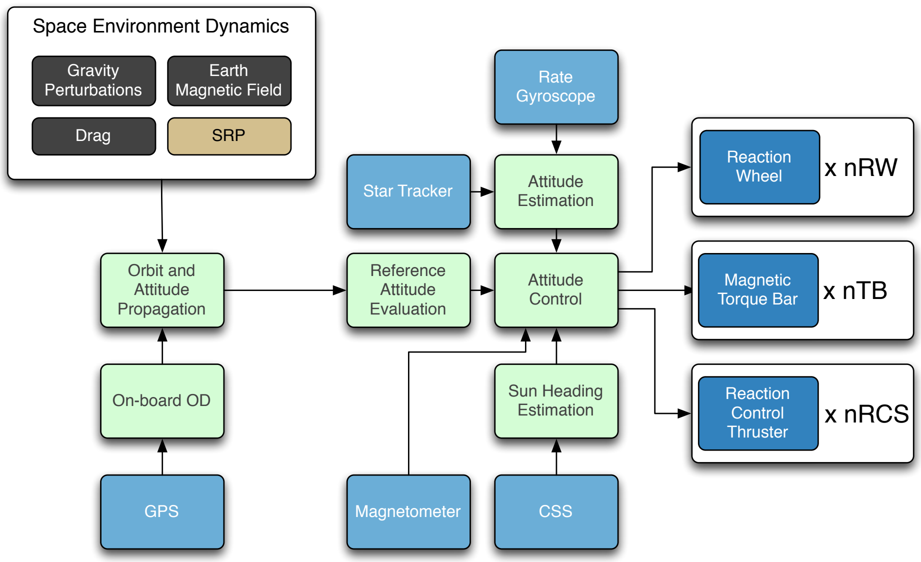
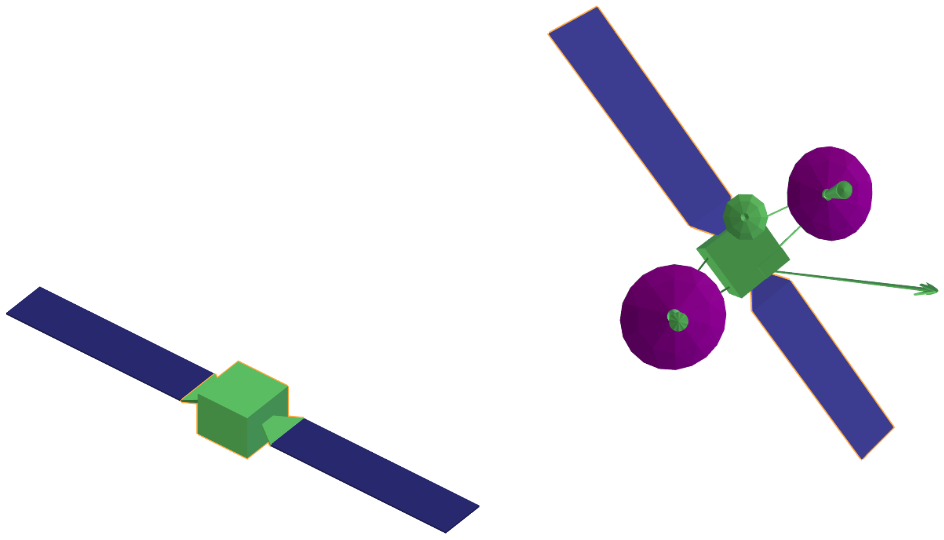


Figure 4.3: AVSLab simulation framework block diagram

tion within the Autonomous Vehicle System Laboratory's spacecraft orbit and attitude simulation framework. The framework used in this work was originally created for the design and analysis of a LEO spacecraft by the Laboratory for Atmospheric and Space Physics (LASP) [16]. The framework, shown as a functional block diagram in Figure 4.3, simulates the dynamics, sensor inputs and processed outputs, of a satellite in Earth orbit. Implemented as a C/C++ code base, the framework contains over 250 options changeable via an input file for altering the simulation behaviour, over half of which are designed to be changed at any time to simulate the failure or fault of any of the 15 different subcomponents modeled [16].

4.2.1 Simulation Integration

To demonstrate the faster than real time capability of the OpenGL method a simulation is configured for a third generation Tracking and Data Relay Satellite (TDRS) satellite. The geosynchronous earth orbiter simulation is configured without spherical harmonic gravity potential terms or multi-body gravity to allow for the effects of the SRP perturbation to be clearly visible. Three simulations are run each utilizing either the cannonball, box and wing or full model geometry



(a) Box and wing with assigned solar array and bus materials (b) Full TDRS model with assigned solar array, bus and antenna materials

Figure 4.4: Assigning material optical properties to a model achieved simply using any 3D modeling tool

as the spacecraft model. As listed in Table 4.3 each model was assigned appropriate material optical properties. The cannonball model C_{r} parameter is computed from an average of the parameters in Table 4.3 using the relationship given at Eq. (2.3b). The placement of these materials on the box and wing and the full model can be seen in Figures 4.4(a) and 4.4(b) respectively.

It is important to note that material properties variations exists at a per-component level and that the optical properties of many materials evolve due to degradation caused by the harsh space environment [20]. Assigning material properties to the box and wing and full model in this simulation is a demonstration of the OpenGL method's facility for easy incorporation of material properties assigned within the analyst's CAD tool of choice (for this work the open source tool Blender was used). At this stage in the development of the OpenGL approach the inclusion of material properties is not intended to provide a high fidelity time evolution of the change in those properties and their corresponding contribution to the radiation perturbation evaluation.

Initialization of the OpenGL model within the simulation requires that the true or false

Table 4.3: TDRS spacecraft model material properties from those given in Ref [12]

Component	Absorptivity	Diffuse Reflectivity	Specular Reflectivity
Bus	0.6	0.2	0.2
Antennas	0.25	0.6	0.15
Solar Arrays	0.8	0.15	0.05

toggle `useOpenglSrp` is set as 'true' and that a configured CAD model file is provided. To evaluate the SRP perturbation effects on a different spacecraft requires the user to simply provide a new spacecraft CAD model. This simplicity of initialization and iterative evaluation capability, coupled with the potential for high geometric fidelity and faster than real time evaluation, is a novel aspect of the OpenGL method not seen in other radiation pressure evaluation approaches.

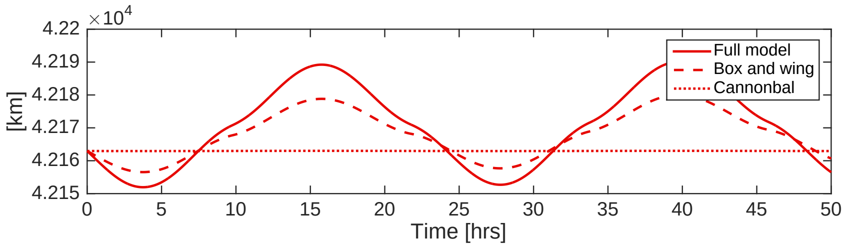
Table 4.4: TDRS spacecraft initial orbital conditions for simulation

Orbit Element	Value
Semi-major Axis	42162.95 [km]
Eccentricity	0.0011145
Inclination	6 [deg]
Ascending Node	329.4411 [deg]
Argument of Periapse	285.5670 [deg]
True Anomaly	1 [deg]

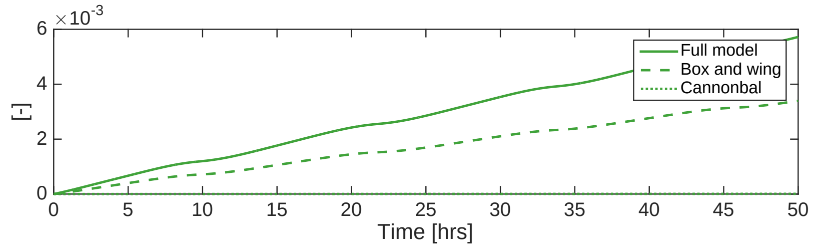
The simulation initial orbital conditions are given in Table 4.4. For the duration of the approximately 50 hour simulation the TDRS satellite is commanded to maintain a nadir pointing attitude which corresponds to maintaining all antenna bore site directions pointed towards the Earth.

4.2.2 Simulation Results

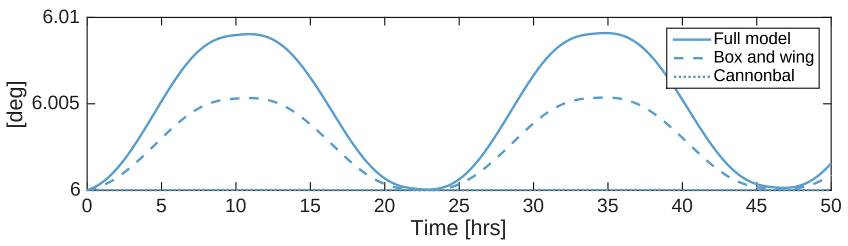
The resulting evolution of the six Keplerian orbital elements is shown in Figure 4.5 and 4.6. As expected the SRP perturbation produces a periodic variation in the semi-major axis for each of the spacecraft models (the cannonball model variation is not visible due to its small variation from its initial condition). Similar periodic variations appear the Right Ascension of Ascending Node (RAAN) and inclination. The evolution of the eccentricity for each simulation shows a secular trend



(a) Semi-Major Axis



(b) Eccentricity

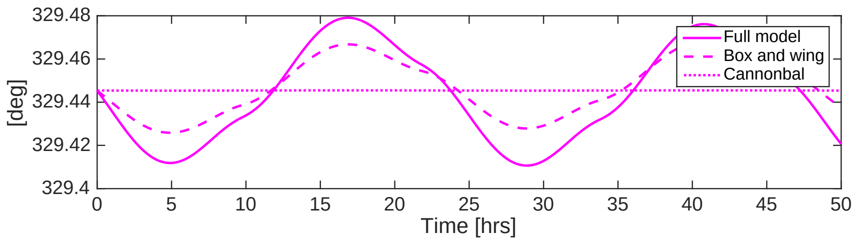


(c) Inclination Angle

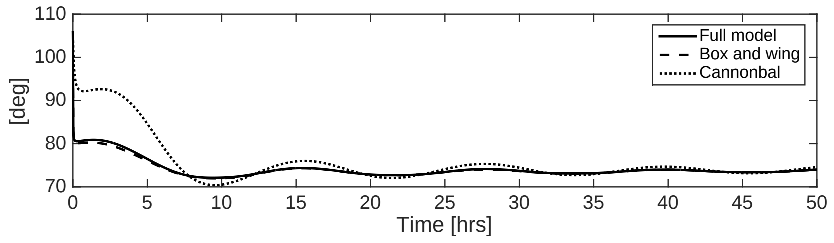
Figure 4.5: Evolution of Keplerian orbital elements for each of the three spacecraft models

for the short simulation duration. However, as shown by Delong et. al. the eccentricity vector of a geosynchronous satellite's orbit is periodic with a period equal to the Earth's yearly orbit around the sun [3]. As a result, a simulation with a one year duration, would be expected that eccentricity variations will exhibit a long term periodic, rather than secular behavior.

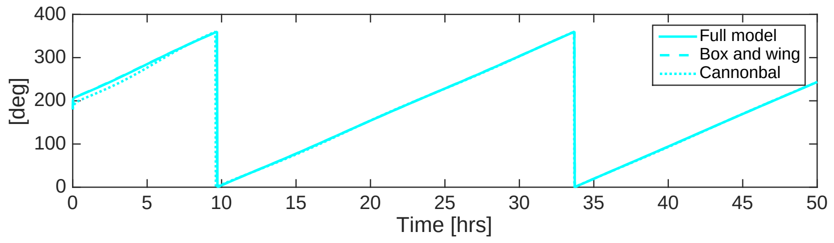
The variation in force and torque for each model is shown in Figures 4.7 and 4.8 respectively. The same periodic behavior, at orbital frequency, seen in the orbital elements is evident in both the force and torque. Examining the force and semi-major axis variations together reveals the expected attitude dependence of the box and wing and full model SRP force evaluations. Over a single orbit period the spacecraft transitions through four notable attitudes where a particular body frame axis will be primarily oriented with the sun direction $\hat{\boldsymbol{u}}_B$. Two sun relative attitudes occur with the +z and -z axes and two more with the +y and -y. When +z and -z axes are primarily in the sun



(a) RAAN



(b) Argument of Perigee



(c) True Anomaly

Figure 4.6: Evolution of Keplerian orbital elements for each of the three spacecraft models

direction the spacecraft experiences a maximum SRP force whereas when the +y and -y axis are sun pointing the spacecraft experiences a minimum force and torque. These variations in force can be matched to variations in the semi-major axis evolution of the two models.

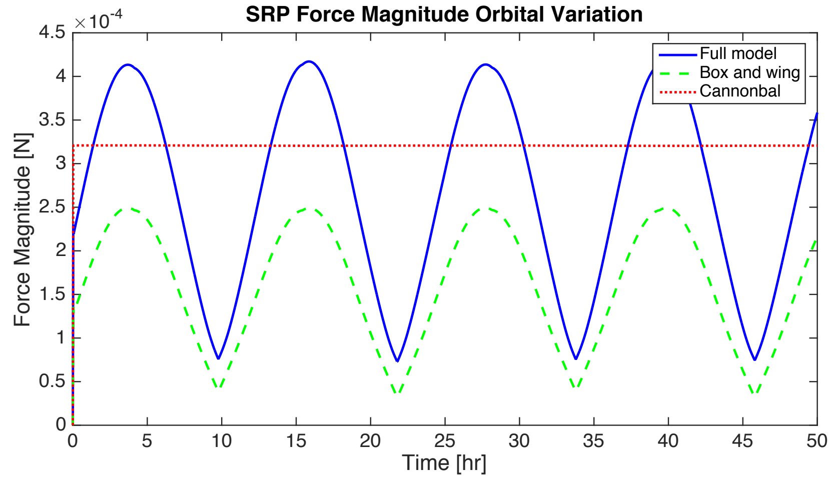


Figure 4.7: Evolution of force due to SRP for each model

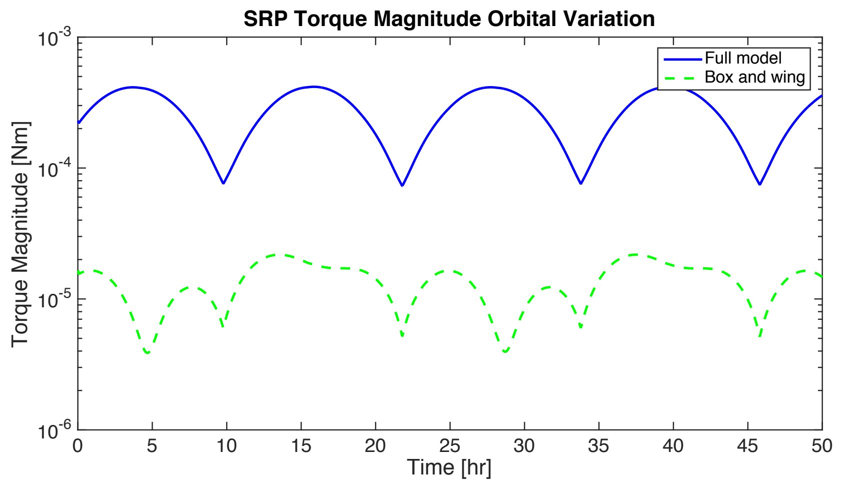


Figure 4.8: Evolution of torque due to SRP for each model

4.3 Computational Performance

A low time to solution for a high geometric fidelity spacecraft model requires the OpenGL method to demonstrate high computational performance. To provide a crude metric of the increase in evaluation speed afforded by the OpenGL method, a single evaluation of the MRO model shown in Figure 4.2 is carried out using the MATLAB verification tool and the OpenGL method. This comparison is termed 'crude' as the same serial evaluation implemented in MATLAB would undoubtedly execute in less time with a change in programming language and implementation. However, this crude demonstration aims to highlight the vast speed increase of the OpenGL GPU method compared to the same serial evaluation on the CPU. The result of the two evaluations are given in Table 4.5. While it is clear that the highly parallel OpenGL method evaluates the model up to three orders of magnitude faster than the serial MATLAB implementation, much work remains to optimize and further decrease the OpenGL method execution time. Primary areas of optimization include moving the final per primitive force and torque summation process on to the GPU and performing analysis as to which computations do and do not require double precision variables in the shader code. Further, and most importantly, the potential to utilize a general computing shader stage called "compute shader" in which to execute the entire radiation modeling process is under active development.

Table 4.5: Execution time for single evaluation of MRO model using serial MATLAB verification tool and the OpenGL method. (OpenGL 4.1 on 2015 MacBook Pro 3.1 GHz Intel Core i7, 8GB Ram, Intel Iris 6100 1536 MB).

Model Implementation	Execution Time [sec] 14750 Primitives
MATLAB	5.7
OpenGL Method	0.002

Chapter 5

Momentum Management Strategy Using OpenGL Method

During the cruise phase of an interplanetary mission it is common for ground controllers to command the spacecraft to maintain a number of nominal attitudes specified by the mission's concept of operations (CONOPS). Two often commanded CONOPS attitudes are sun-pointing (supporting power generation) and earth-pointing (to ensure a high availability of communication links with ground stations)[26]. Maintaining such attitudes requires the spacecraft's momentum management system to continually absorb the torques imparted to the spacecraft by the SRP perturbation.

This chapter presents an application of the OpenGL methods utility in design and implementation of a control strategy which alternates between periods of absorbing angular momentum in alternate total angular momentum vector directions such that a spacecraft's momentum state remains below some nominal level. A simulation of the developed control algorithm is carried out and analysis shows that passive SRP momentum management achieved in a restricted range of attitudes.

5.1 Spacecraft Torque Characterization

Using the OpenGL method integrated with the AVS Lab simulation framework the angular momentum storage in the spacecraft's reaction wheel devices is characterized during a hypothetical interplanetary cruise. The spacecraft model is a simple box and wing structure, constructed using CAD modeling software. The model's faceted construction can be seen in Figure 5.1. The simple

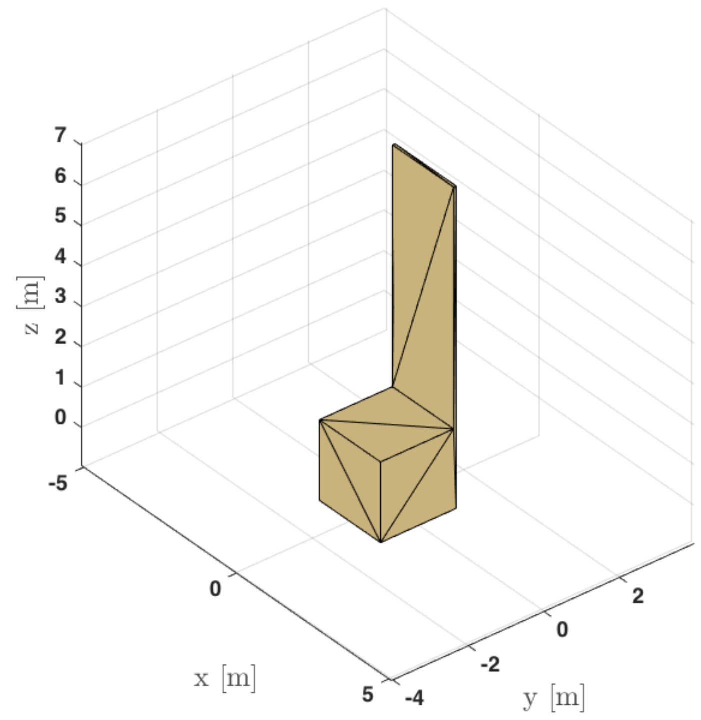


Figure 5.1: Box and wing spacecraft shown in faceted geometry

geometry of a box and wing geometry provides a significant torque generating capability, which is ideal for demonstrating the following momentum management strategy.

Two methods are used to characterize the reaction wheel angular momenta induced by the SRP torque. The first method is a static model evaluation using the MATLAB developed SRP tool set. The SRP induced spacecraft torque is computed at a single instant in time. For this initial static characterization the spacecraft attitude is set such that the solar panel normal vector is parallel to the spacecraft-sun unit vector $\hat{\boldsymbol{u}}_B$. As shown in Figure 5.2 this results in the entire top side of the spacecraft being illuminated. The resulting torque for this static evaluation about each of the spacecraft body frame axes is given at Eq. (5.1).

$$\tau_{SRP1} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0.2966 \times 10^{-3} \\ 0.0 \\ 0.0 \end{pmatrix} [\text{N.m}] \tag{5.1}$$

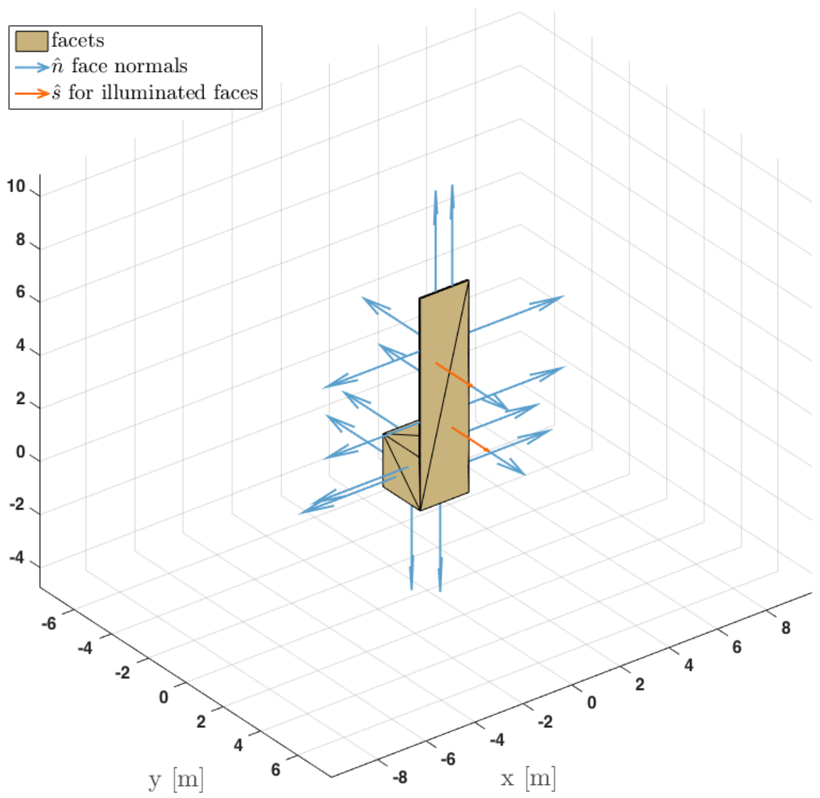


Figure 5.2: Spacecraft showing the impinging solar radiation

The second characterization method is to employ the AVS Lab simulation to characterize the time evolution of wheel momentum accumulation. The box and wing spacecraft is configured with four reaction wheel control devices configured in a pyramid arrangement where the wheel spin axes and the wheel inertia about the spin axis are given in Table 5.1. The spacecraft's center of mass is located at the box's center which results in an uncharacteristically large center of mass (CM) to center of pressure (CP) offset of 4 meters. The spacecraft mass moment of inertia is given at Eq. (5.2). The spacecraft was placed on an Earth to Mars Hohmann transfer orbit with the initial heliocentric orbital elements given in Table 5.2. The spacecraft model used in the analysis presents an extreme case where the spacecraft's CP-CM offset is sufficiently large to clearly produce a SRP induced torque when aligning $\hat{\boldsymbol{c}}_B$ with $\hat{\boldsymbol{u}}_B$.

Table 5.1: Reaction wheel spin axis unit vectors and spin axis inertia

Device	Spin Axis $\hat{\boldsymbol{g}}_{\mathrm{s}}$	Inertia J_{s} about $\hat{\boldsymbol{g}}_s$ [kg.m^2]
RW_1	$(0.0, \cos\pi/4, \sin\pi/4)$	0.159
RW_2	$(0.0, \sin\pi/4, -\cos\pi/4)$	0.159
RW_3	$(\cos\pi/4, -\sin\pi/4, 0.0)$	0.159
RW_4	$(-\cos\pi/4, -\sin\pi/4, 0.0)$	0.159

$$I = \begin{bmatrix} 1000.0 & 0.0 & 0.0 \\ 0.0 & 800.0 & 0.0 \\ 0.0 & 0.0 & 800.0 \end{bmatrix} [\mathrm{kg.m}^2] \tag{5.2}$$

Table 5.2: Initial conditions for and Earth to Mars Hohmann transfer heliocentric orbit

Description	Value
Semi-major axis a	1.26 [AU]
Eccentricity e	0.207501
Inclination i	0 [deg]
Ascending Node Ω	0 [deg]
Argument of Periapse ω	0 [deg]
True Anomaly f	70 [deg]

During the 30 hour simulation time the spacecraft's controller was commanded to maintain the solar panel normal vector pointing directly towards the sun. The resulting evolution of the reaction wheel speeds can be seen in Figure 5.3. The demonstrated increasing wheel rates corresponds to a constant SRP torque shown at Eq. (5.3). A comparison of the SRP generated torque in Eq. (5.1) with Eq. (5.3) shows close agreement in the values computed by the two evaluation methods. An extrapolation of this rate of momentum accumulation yields reaction wheel rates above nominal operational limits of 1500 [rpm] in approximately 10 days and rates at a common manufacturer limit of 6000 [rpm] in approximately 24 days of cruise. However, it is common to prevent wheel speeds from reaching above a nominal rate lower than the manufacturer limit through momentum dumping maneuvers [29]. This lower threshold provides significant control safety margin to the spacecraft.

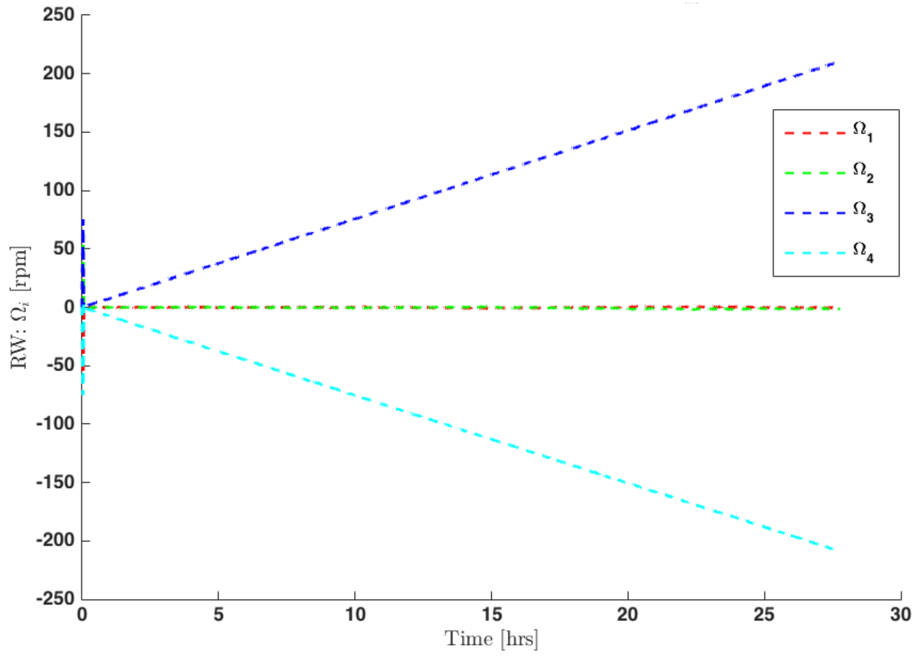


Figure 5.3: Increase in reaction wheel speeds due to persistent worst case SRP torque

$$\tau_{SRP2} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0.2966 \times 10^{-3} \\ 0.000001 \times 10^{-3} \\ 0.000001 \times 10^{-3} \end{pmatrix} \text{[N.m]} \tag{5.3}$$

5.2 Determining The Reference Attitude

To reduce the reaction wheel angular momenta the new spacecraft attitude must provide a SRP induced torque vector which has a direction as close to parallel to that of the sum of the wheel angular momenta vectors. To provide this vector a look up table for all SRP torque possibilities is generated. The table is generated offline by sweeping $\hat{\boldsymbol{u}}_B$ over the 4π steradian attitude possibilities at uniformly distributed intervals to collect 600 torque attitudes. For each new $\hat{\boldsymbol{u}}_B$ the torque on the spacecraft is evaluated using the OpenGL method and recorded in the lookup table structure shown in Table 5.3. The lookup table is composed of an index, a sun heading vector $\hat{\boldsymbol{u}}_B$ and a normalized torque vector $\hat{\boldsymbol{\tau}}_B$. The generated lookup table is loaded into the spacecraft flight

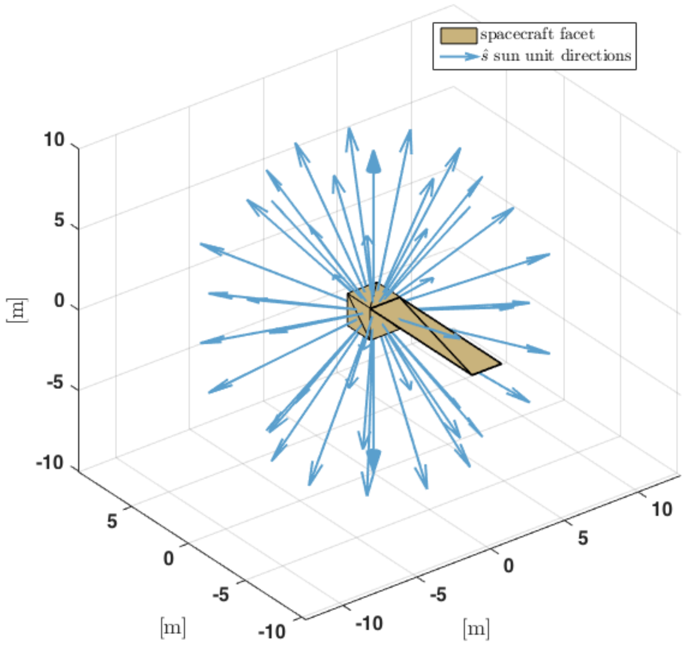


Figure 5.4: An example of generating the reference attitude lookup table by sweeping the $\hat{\boldsymbol{u}}_B$ unit direction vector through the attitude sphere at 30° increments.

software to be executed software-in-the-loop within the AVS Lab simulation framework.

Table 5.3: Reference attitude lookup table

Index	$\hat{\boldsymbol{u}}_b$	$\hat{\boldsymbol{\tau}}_b$
0	$(u_{b_1}, u_{b_2}, u_{b_3})$	$(\tau_{b_1}, \tau_{b_2}, \tau_{b_3})$
...	...	...
N	$(u_{b_1}, u_{b_2}, u_{b_3})$	$(\tau_{b_1}, \tau_{b_2}, \tau_{b_3})$

The spacecraft control algorithm determines that a new reference attitude is required when a single reaction wheel reaches a nominal maximum angular rate. The new reference attitude must serve to reduce the wheel rates and maintain a power positive attitude. The first step of the new attitude reference generation is to evaluate the sum of the angular momentum contributions of each reaction wheel as given by Eq. (5.4). A unit direction vector for the sum of the angular momentum of all wheels is computed at Eq. (5.5). A search through the previously generated lookup table is performed to select a new reference attitude which best reduces the current wheel momenta.

The lookup table search is a linear ($\mathcal{O}(n)$) process. Optimization of this lookup problem is clearly possible, however, its use and implementation are not addressed in this work. An example lookup table for a box spacecraft model, generated at 12° increments, can be viewed in the Appendix.

Reference attitude selection is a three-step process. The first step is to compare $\hat{\boldsymbol{H}}_{RW}$ and a candidate $\hat{\boldsymbol{\tau}}_B$ to determine if they are more parallel than a previous $\hat{\boldsymbol{\tau}}_B$ candidate. The second step is to determine if the potential $\hat{\boldsymbol{u}}_B$ lies within the angular offset from the spacecraft-sun line, $\theta_{\text{constraint}}$, as permitted by the mission CONOPS. Torque possibilities outside $\theta_{\text{constraint}}$ are to be rejected in support of ensuring adequate solar panel power generation. The third step is is to check if the potential torque magnitude of $\hat{\boldsymbol{\tau}}_B$ is greater than a previously found attitude torque vector candidate. If greater, the new $\hat{\boldsymbol{u}}_B$ is saved as the candidate reference heading. The final candidate reference vector will be used to form the new reference attitude. This selection process is described in Algorithm listing 1. The reference attitude is generated using the resulting $\hat{\boldsymbol{u}}_B$ from the lookup table search. A right handed body fixed frame is constructed for use as the controller reference frame where the remaining axes of the frame are arbitrarily constrained.

$$^{\mathcal{B}}\boldsymbol{H}_{RW} = \sum_{i=1}^{N} {}^{\mathcal{B}}I_{RW_i}\Omega_i \tag{5.4}$$

$$\hat{\boldsymbol{H}}_{RW} = \frac{^{\mathcal{B}}\boldsymbol{H}_{RW}}{\|^{\mathcal{B}}\boldsymbol{H}_{RW}\|} \tag{5.5}$$

```
Data: $\hat{\boldsymbol{H}}_{RW}$, $\hat{\boldsymbol{c}}_s, \theta_{\text{constraint}}$
Result: $\hat{\boldsymbol{u}}_B$
initialization;
idx = 0;
i = 1;
i_max = length of lookup table;
while i < i_max do
    $\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}_i$
    if $\hat{\boldsymbol{\tau}}$ is more parallel to $\hat{\boldsymbol{H}}_{RW}$ than $\hat{\boldsymbol{\tau}}_{idx}$ then
        // Ensure solar panel's normal vector 'sees' sun
        if $0 < \hat{\boldsymbol{u}}_B \cdot \hat{\boldsymbol{c}}_s \leq \cos(\theta_{\textbf{constraint}})$ then
            // Is there greater torque potential for this reference?
            if $\|\boldsymbol{\tau}\| > \|\boldsymbol{\tau}_{\textbf{sol\_index}}\|$ then
                // Lookup table index of ideal torque and sun heading
                idx = i;
            end
        end
    end
end
```

Algorithm 1: Selection of new reference attitude to reduce reaction wheel speeds

5.3 Controlling To The Reference Attitude

The control law development proceeds as given in Schaub and Junkins [23]. The equations of motion for a system of N reaction wheels is given at Eq. (5.6).

$$[I_{RW}]\dot{\boldsymbol{\omega}} = -[\tilde{\boldsymbol{\omega}}]([I_{RW}]\boldsymbol{\omega} + [G_s]\boldsymbol{h}_s) - [G_s]\boldsymbol{u}_s + \boldsymbol{L} \tag{5.6}$$

where the components of $\boldsymbol{h}_s$, the wheel spin axis angular momenta, are given as

$$h_{s_i} = J_{s_i}(\dot{\Omega}_i + \hat{\boldsymbol{g}}_{s_i}^T\dot{\boldsymbol{\omega}}) \tag{5.7}$$

and the wheel angular rate error as

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_r \tag{5.8}$$

To ensure a stable control law a Lyapunov function is sought which gives a positive definite result for attitude and rate errors and at least negative semi-definite first time derivative. The Lyapunov function given at Eq. (5.9) is constructed using the elemental position and velocity Lyapunov functions for σ and $\delta\omega$.

$$V(\boldsymbol{\sigma}, \delta\boldsymbol{\omega}) = \frac{1}{2}\delta\boldsymbol{\omega}^T[I_{RW}]\delta\boldsymbol{\omega} + 2K\ln(1+\boldsymbol{\sigma}^T\boldsymbol{\sigma}) \tag{5.9}$$

Taking the first time derivative of Eq. (5.9) results in the following rate potential where the initial inclusion of K provides an attitude error feedback gain.

$$\dot{V}(\boldsymbol{\sigma}, \delta\boldsymbol{\omega}) = \delta\boldsymbol{\omega}^T(K\sigma) + \delta\boldsymbol{\omega}^T[I_{RW}]\delta\dot{\boldsymbol{\omega}} \tag{5.10}$$

Equation (5.10) is set equal to the negative semi-definite function $\dot{V} = -\delta\boldsymbol{\omega}^T[P]\delta\boldsymbol{\omega}$. Here $[P]$ represents a positive controller gain matrix. The equations of motion given at Eq. (5.6) are substituted into Eq. (5.10) to provide the required control torque at Eq. (5.11).

$$[G_s]\boldsymbol{u}_s = K\boldsymbol{\sigma} + [P]\,\delta\boldsymbol{\omega} \tag{5.11}$$

If the system possess more than a minimal set of three reaction wheels a minimum norm inverse can be used to solve for the torque $\boldsymbol{u}_s$

$$\boldsymbol{u}_s = [G_s]^T([G_s][G_s]^T)^{-1}(K\boldsymbol{\sigma} + [P]\,\delta\boldsymbol{\omega}) \tag{5.12}$$

5.4 Simulation Results

Initial results of the simulation revealed a fundamental error in the assumption that an attitude would exit for which a momentum reducing torque and satisfied the CONOPS constraint could be found. In retrospect, it is a simple thought experiment to determine that for the sun heading in Figure 5.2 the SRP induced torque vector will be oriented in the $-\hat{\boldsymbol{x}}$ axis and the angular momentum accumulation in the $+\hat{\boldsymbol{x}}$ axis. To remove the accumulated momentum the spacecraft will need to control to the opposite, $-\hat{\boldsymbol{y}}$ axis, sun heading direction to produce a momentum reducing $+\hat{\boldsymbol{x}}$ axis torque vector so the reaction wheels accumulate momentum in the $-\hat{\boldsymbol{x}}$. A $-\hat{\boldsymbol{y}}$ axis sun heading orientation clearly violate the sun pointing CONOPS constraint. To proceed with this analysis, the CONOPS constraint is removed to allow for a complete demonstration of momentum management.

The spacecraft control flight software is set to trigger a desaturation maneuver when any reaction wheel spin rate reaches 500 [rpm]. This moderate wheel rate is chosen to allow for all spacecraft behavior to be visible in a reasonably short simulation duration. Inspecting Figures 5.5 through to Figure 5.8 it is clear that the desaturation maneuver is triggered at approximately 15 hours. As shown in Figure 5.7 the spacecraft selects a new attitude which orients the $-\hat{\boldsymbol{y}}$ axis towards the sun. The spacecraft SRP torque magnitudes shown in Figure 5.8 before and after the maneuver are approximately equal due to the symmetry in the sun facing geometry of a $-\hat{\boldsymbol{y}}$ and $+\hat{\boldsymbol{y}}$ axis sun pointing spacecraft. The small 'wiggle' in the SRP torque magnitude before the maneuver is due to the spacecraft cruising in a sun pointing control mode which utilizes a control dead band about the sun point attitude. Additionally, the variation in SRP torque during the maneuver is visible as a sharp dip in the SRP torque magnitude at the maneuver time. Finally, following the maneuver, Figure 5.6 shows that individual reaction wheel momenta trend back towards the zero point before the simulation ending at 28 hours.

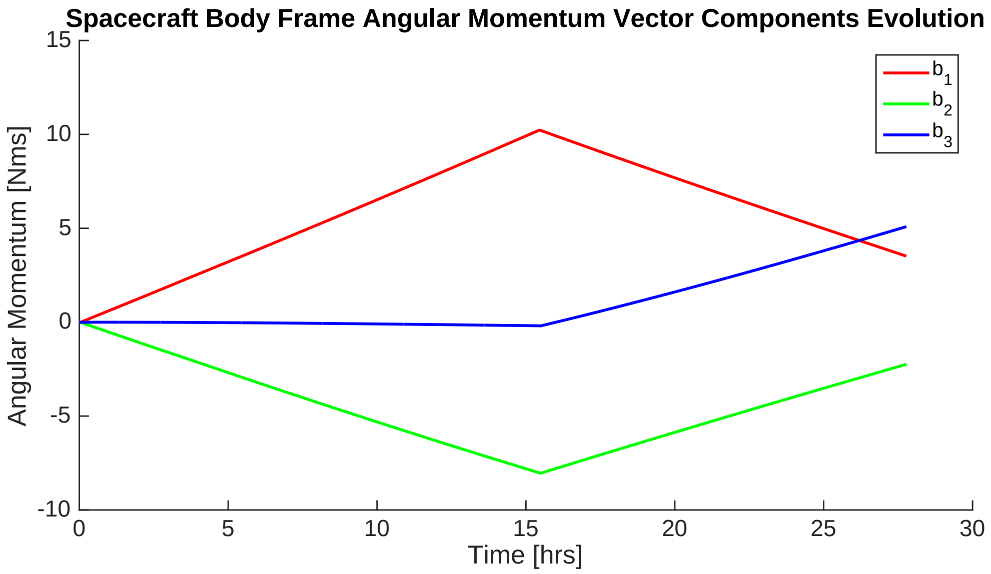


Figure 5.5: Box and wing body angular momentum

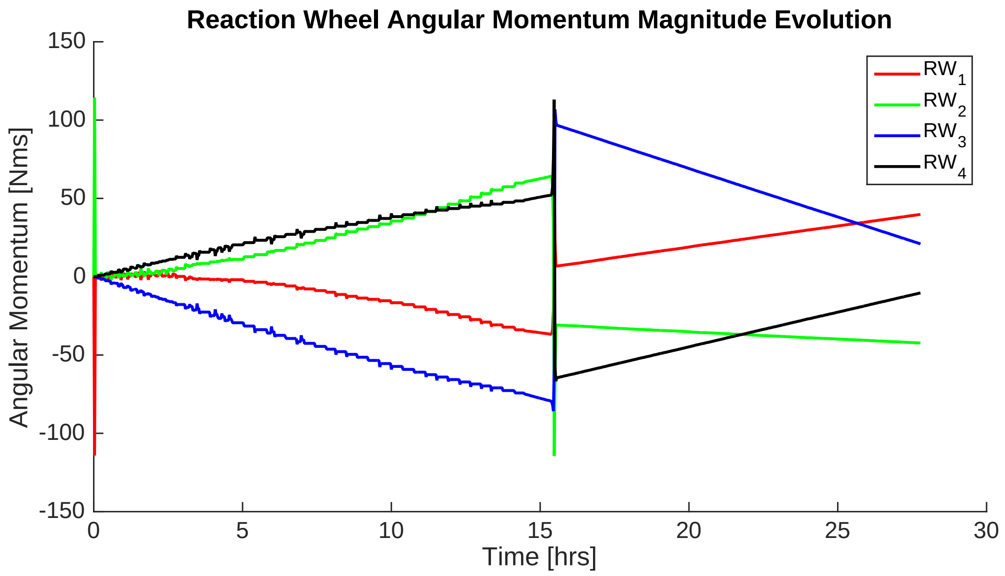


Figure 5.6: Reaction wheel angular momentum

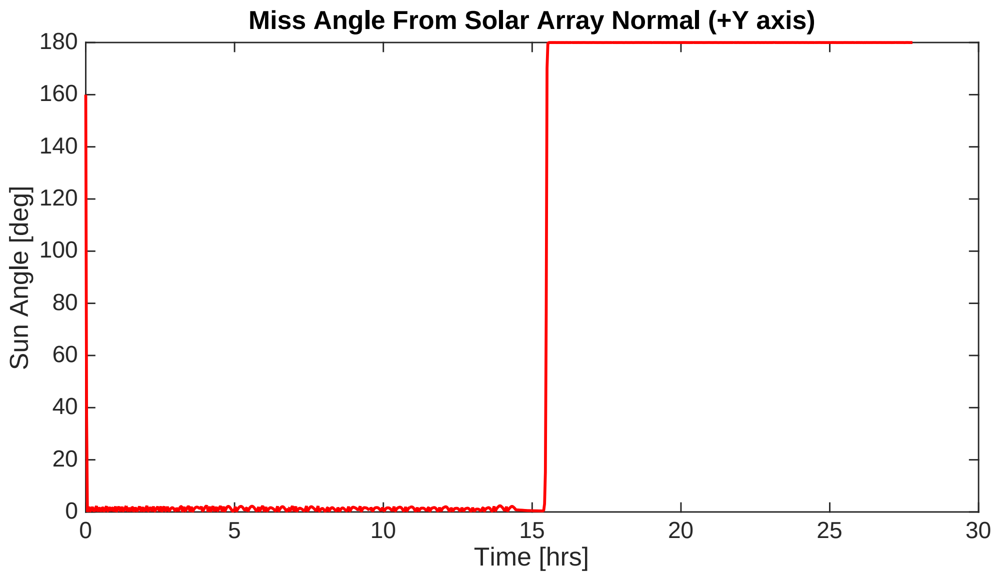


Figure 5.7: Solar array bore sight miss angle

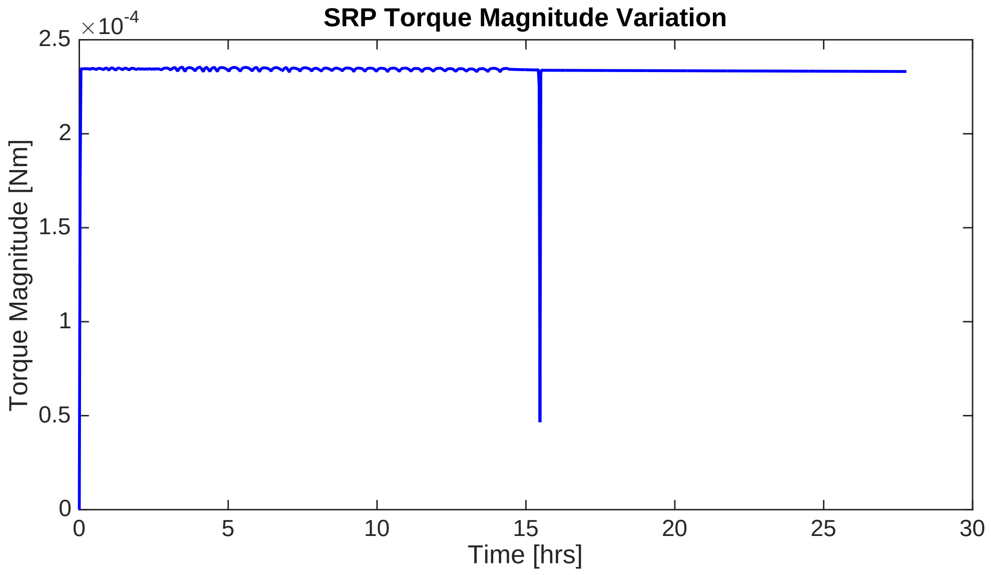


Figure 5.8: SRP torque magnitude on the box and wing spacecraft

Chapter 6

Conclusions and Further Work

The cannonball and shape approximation models provide useful first order assessments of the effect of SRP on a spacecraft during its mission. However, deficiencies exist in both of these models which limit their utility in further mission and spacecraft design efforts. The cannonball model does not resolve spacecraft torques and its use is limited to perturbations in the spacecraft's transnational motion. Additionally, the cannonball and shape approximation models are unable to capture the time varying articulation of spacecraft appendages and the micro variations in material properties across the spacecraft surface. In contrast because the OpenGL method leverages techniques from the vector graphics discipline it is developed with the inherent ability to resolve arbitrary spacecraft articulation appendages, variations in material properties and

The tools and techniques developed by the computer vector graphics disciplines are ideally suited to evaluating the per-facet SRP expression due to its' similarly geometric nature. Implementation of a high geometric fidelity SRP modeling approach using the existing OpenGL software tool set provides an optimized API for data processing on commodity GPUs. in contrast to the commodity CPU, the GPU is designed to have many thousands of compute cores on a single chip. The OpenGL method makes use of the GPU execution environment and the same highly parallel processing approach as applied to rendering computer graphics. The OpenGL pipeline architecture is customized for the SRP computation at the vertex, geometry and fragment shader stages. This customization yields a highly parallel vector math processing strategy which is capable of processing up to thousands of spacecraft model facets simultaneously.

By leveraging established vector graphics techniques and highly parallel GPU computing the OpenGL SRP method provides benefit to mission designers of reduced computation time for a significantly higher geometric fidelity spacecraft model. The OpenGL method provides facility for the inclusion of per facet surface material optical properties and definition of arbitrary spacecraft articulation. In doing so the method provides engineers with a tool capable of capturing and making use of significantly more geometric detail and pre-launch engineering data.

Easy integration and use of the OpenGL method as a modular component within the Autonomous Vehicle System Laboratory (AVSLab) software simulation framework demonstrates the method's suitability as a module within a faster than real time spacecraft simulation. The method's simple integration makes possible rapid iteration of the spacecraft models and engineering data which influence the SRP forces and torques within a simulation. Rapid iteration enables mission designers and spacecraft engineers to perform fast "what if?" type simulations with minimal setup time and a greater fidelity than currently available with a faceted or cannonball model. Furthermore, the methods reliance on only existing and familiar 3D modeling tools reduces model preparation time and increases the opportunity to include a wealth of engineering data already present in spacecraft CAD models.

Verification of the OpenGL method force and torque results against existing SRP models and data sources provides confidence in the validity of the method's computation. Using the initial validation as a foundation, further simulations are run to demonstrate that using a higher geometric fidelity spacecraft model provides greater resolution of a spacecraft's SRP induced rotational and translational motion throughout orbit, when compared with the box and wing and cannonball modeling approaches.

The OpenGL method provides a novel first step towards a more complete pre-launch radiation force model. The ultimate pursuit in developing a pre-launch radiation force model is to produce a model which accounts for a full energy balance of radiation sources. Primary components needed to achieve a full energy balance include a reduced spacecraft surface thermal model, accommodations for high power antenna radiation, resolving of secondary photon impacts on the spacecraft and further use of existing spacecraft engineering data to model the degradation of spacecraft surface materials [32]. Of the aforementioned energy balance components, the detection and resolution of secondary photon impacts and spacecraft self-shadowing presents the greatest challenge. The challenge is greatest due to the fact that current algorithms which resolve secondary impacts use ray tracing techniques which even in parallel computing environments require significant computational effort to execute. This increased computational effort risks increasing the OpenGL method's computation time and thus negating one of the method's computational speed advantage. As a result there is significant further work required in the formulation of computational efficient resolution of secondary impacts.

The verification and validation of the OpenGL method would benefit from further work to complete a comparative analysis of the method with other existing model types. This thesis studies pre-launch SRP modeling approaches which use only a-priori engineering data to generate a model. To place the OpenGL model more firmly in a context with other SRP modeling approaches a comparison of the various approaches performance, accuracy and abilities to aid spacecraft and mission design would prove instructive. Such analysis should characterize the utility of both a-priori and flight data based modeling methods across a variety of mission types and mission phases (pre-launch, operations and end of life) where SRP modeling requirements vary.

Finally, further enhancement of the current OpenGL method implementation is required to allow for the demonstration of articulated spacecraft components. This relatively simple enhancement, when compared with the addition of secondary photon impact capability, will complete the OpenGL method as a tool that may be used to design and simulate spacecraft missions in which SRP is used as an actuator rather than treated as a dynamic disturbance.

Putting OpenGL and vector graphics techniques to work on modeling the SRP effect on a spacecraft has yielded a modeling approach characterized as simple to use and providing fast computation of high geometric fidelity SRP forces and torques on an arbitrary spacecraft geometry. The results presented here demonstrate that the method provides good insight into spacecraft dynamics. The modeling approach holds great promise and that with the development of additional model components the OpenGL SRP method could eventually be considered a completely high fidelity model.

Bibliography

[1] AAS/AIAA Astrodynamics Specialist Conference, At Vail, CO. GPU-Accelerated Computation of SRP Forces with Graphical Encoding of Surface Normals, number AUGUST, 2015.

[2] Y E. Bar-Sever. New and improved solar radiation models for gps satellites based on flight data final report. Technical report, Jet Propulsion Laboratory, 1997.

[3] N. Delong and C. Frémeaux. Eccentricity management for geostationary satellites during end of life operations. In D. Danesy, editor, Proceedings of the 4th European Conference on Space Debris, number ESA SP-587, page 297, 2005.

[4] H. F. Fliegel, T. E. Gallini, and E. R. Swift. Global Positioning System Radiation Force Model for geodetic applications. Journal of Geophysical Research, 97(B1):559, 1992.

[5] Henry F. Fliegel and Thomas E. Gallini. Solar force modeling of block IIR Global Positioning System satellites. Journal of Spacecraft and Rockets, 33(6):863–866, 1996.

[6] Bent Fritsche and Heiner Klinkrad. Accurate Prediction of Non-Gravitational Forces for Precise Orbit Determination - Part I: Principles of the Computation of Coefficients of Force and Torque. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, (August):1–13, 2004.

[7] Ryu Funase, Yoji Shirasawa, Yuya Mimasu, Osamu Mori, Yuichi Tsuda, Takanao Saiki, and Jun'ichiro Kawaguchi. On-orbit verification of fuel-free attitude control system for spinning solar sail utilizing solar radiation pressure. Advances in Space Research, 48(11):1740 – 1746, 2011. Solar Sailing: Concepts, Technology And Missions.

[8] Khronos Group. Opengl documentation. https://www.opengl.org, 10 2015.

[9] K Krewell. What's the difference between a cpu and a gpu? https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/, April 2016.

[10] Hitoshi Kuninaka and Jun'ichiro Kawaguchi. Lessons learned from round trip of Hayabusa asteroid explorer in deep space. 2011 Aerospace Conference, pages 1–8, 2011.

[11] David M. Lucchesi. Reassessment of the error modelling of non-gravitational perturbations on LAGEOS II and their impact in the Lense–Thirring derivation—Part II. Planetary and Space Science, 50(10-11):1067–1100, 2002.

[12] S.B Luthcke, J.A Marshall, S.C Rowton, K.E Rachlin, C.M Cox, and R.G Williamson. Enhanced radiative force modeling of the tracking and data relay satellites. The Journal of the Astronautical Sciences, 45(3):349–370, July-September 1997.

[13] K. Luu and C. Sabol. Effects of perturbations on space debris in supersynchronous storage orbits. Technical Report AFRL-VS-PS-TR-1998-1093, Air Force Research Labs, Space Vehicles Directorate, 1998.

[14] J A Marshall, S B Luthcke, P G Antreasian, and G W Rosborough. Modeling Radiation Forces Acting on TOPEX/Poseidon for Precision Orbit Determination. Technical report, 1992.

[15] Jay W. McMahon and Daniel J. Scheeres. New solar radiation pressure force model for navigation. Journal of Guidance, Control, and Dynamics, 2010.

[16] Stephen A OKeefe. Autonomous Sun-Direction Estimation Using Partially Underdetermined Coarse Sun Sensor Configurations by. PhD thesis, University of Colorado, Boulder, 2015.

[17] Daniel J. O'Shaughnessy, James V. McAdams, Peter D Bedini, Andrew B Calloway, Kenneth E Williams, and Brian R Page. Messenger's use of solar sailing for cost and risk reduction. Acta Astronautica, 93:483–489, jan 2014.

[18] Daniel J. O'Shaughnessy, James V. McAdams, Kenneth E Williams, and Brian R Page. Fire sail: Messenger's use of solar radiation pressure for accurate mercury flybys. pages 1–16, 2011.

[19] J.D. D Owens, M. Houston, D. Luebke, S. Green, J.E. E Stone, and J.C. C Phillips. GPU Computing. Proceedings of the IEEE, 96(5), 2008.

[20] Benny Rievers. High precision modelling of thermal perturbations with application to Pioneer 10 and Rosetta. PhD thesis, University of Bremen, 2012.

[21] Hyung-Jin Rim, Charles Webb, Sungpil Yoon, and Bob Schutz. Radiation Pressure Modeling for ICESat Precision Orbit Determination. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, (August):1–7, 2006.

[22] C. J. Rodriguez-Solano, U. Hugentobler, and P. Steigenberger. Adjustable box-wing model for solar radiation pressure impacting GPS satellites. Advances in Space Research, 49(7):1113–1128, 2012.

[23] Hanspeter Schaub and John L. Junkins. Analytical Mechanics of Space Systems. AIAA Education Series, Reston, VA, 3rd edition, 2014.

[24] Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane. OpenGL programming guide : the official guide to learning OpenGL, version 4.3. Addison-Wesley, Upper Saddle River, NJ, 2013.

[25] T. A. Springer, G. Beutler, and M. Rothacher. A new solar radiation pressure model for gps satellites. GPS Solutions, 2(3):50–62, 1999.

[26] Jim Taylor, editor. Deep Space Communications. DEEP SPACE COMMUNICATIONS AND NAVIGATION SERIES. NASA Jet Propulsion Laboratory, California Institute of Technology, October 2014.

[27] Jason Tichy, Abel Brown, Michael Demoret, Benjamin Schilling, and David Raleigh. Fast finite solar radiation pressure model integration using opengl. 2014.

[28] David Vallado. Fundamentals of astrodynamics and applications. Springer, New York, 2007.

[29] J.R. Wertz, D.F. Everett, and J.J. Puschell. Space Mission Engineering: The New SMAD. Space technology library. Microcosm, 2011.

[30] Bong Wie. Space Vehicle Dynamics and Control. American Institute of Aeronautics and Astronautics, Reston, VA, 2008.

[31] Tung-Han You, Eric Graat, Allen Halsell, Dolan Highsmith, Stacia Long, Ram Bhat, Stuart Demcak, Earl Higa, Neil Mottinger, and Moriba Jah. Mars reconnaissance orbiter interplanetary cruise navigation. In 20th International Symposium on Space Flight Dynamics, 2007.

[32] M Ziebart, S Adhya, a Sibthorpe, S Edwards, and P Cross. Combined radiation pressure and thermal modelling of complex satellites: Algorithms and on-orbit tests. Advances in Space Research, 36(3):424–430, 2005.

[33] Marek Ziebart. High Precision Analytical Solar Radiation Pressure Modelling for GNSS Spacecraft. PhD thesis, University of East London, 2001.