NASA/TM—2014-218318

A Raster Based Approach to Solar Pressure Modeling

Theodore W. Wright
Glenn Research Center, Cleveland, Ohio

June 2014

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to *help@sti.nasa.gov*

- Fax your question to the NASA STI Information Desk at 443–757–5803

- Phone the NASA STI Information Desk at 443–757–5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM—2014-218318

A Raster Based Approach to Solar Pressure Modeling

Theodore W. Wright
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

June 2014

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

A Raster Based Approach to Solar Pressure Modeling

Theodore W. Wright
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

The impact of photons upon a spacecraft introduces small forces and moments. The magnitude and direction of the forces depend on the material properties of the spacecraft components being illuminated. Which components are being lit depends on the orientation of the craft with respect to the Sun as well as the gimbal angles for any significant moving external parts (solar arrays, typically). Some components may shield others from the Sun.

To determine solar pressure in the presence overlapping components, a 3D model can be used to determine which components are illuminated. A view (image) of the model as seen from the Sun shows the only contributors to solar pressure. This image can be decomposed into pixels, each of which can be treated as a non-overlapping flat plate as far as solar pressure calculations are concerned. The sums of the pressures and moments on these plates approximate the solar pressure and moments on the entire vehicle.

The image rasterization technique can also be used to compute other spacecraft attributes that are dependent on attitude and geometry, including solar array power generation capability and free molecular flow drag.

Contents

1 Introduction

The impact of photons upon a spacecraft introduces small forces and moments, on the order of hundreds of micro-Newtons for NASA's Orion Vehicle (also known as CEV and MPCV) in the vicinity of Earth. For long duration missions, such as station keeping in a lunar orbit, these small forces can accumulate enough to affect subsystems such as the attitude control system. Modeling these forces may allow mission designers to account for items such as orbit perturbations or the additional fuel required to maintain a desired attitude.

Some simulation frameworks include the effects of solar pressure for simple convex non-overlapping shapes (flat plates, cylinders, etc.), but few take into account the shading of spacecraft components due to other components of the craft. The large gimbaling solar arrays of the CEV tend to shade other CEV components from the Sun, depending on the attitude of the craft and the gimbal angles of the arrays.

This paper describes a *raster* based approach to calculating solar pressure. A 3D model of the CEV is rotated to show a 2D image of the craft as seen from the Sun. The pixels in this image show the only parts of the craft that contribute to solar pressure. Each pixel is then interpreted as a non-overlapping flat plate. The vector sum of all the solar pressure forces on these flat plates is used to find the total force and moment on the craft due to solar pressure.

2 Model Requirements

One advantage of the raster approach is that the algorithm is relatively independent of the model. Models can be simple or complex, but that does not change the amount of information that is passed to the solar pressure calculations.

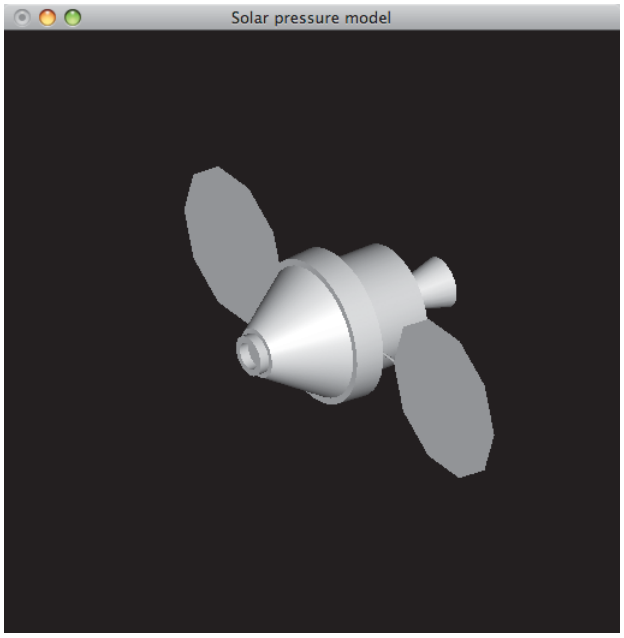
The model must minimally support:

- generating a 2D view (an image) at any orientation (the view from Sun)
- gimbaling external components through their range of motion
- calibrating the size of a pixel with the model size
- extracting the X,Y (image) and Z (depth) model locations for all viewed pixels
- determining which spacecraft component is associated with each pixel

The first two requirements are needed to determine which components of the craft are illuminated. Given a particular view, the last three requirements are the only information that the model must provide to the solar pressure algorithm.

A knowledge of which spacecraft component is associated with each pixel is required so that material properties can be looked up in a table. Surface normal information is also required, but can

Figure 1: CEV model



be approximated from the three dimensional pixel coordinates.

3 Proof-of-concept Implementation

For a proof-of-concept demonstration, a simple CEV model and the code necessary to implement raster-based solar pressure calculations was developed using the OpenGL modeling library. The shapes in the model are built programatically using OpenGL calls to draw simple shapes (cylinders, cones, etc). The program is less than 2000 lines of C language code.

Figure 1 shows a typical view of the CEV in the program, and Figure 2 shows another view with the solar array gimbals optimized to point the arrays in the Sun direction (which is always assumed to be opposite of the view direction). The shading in the figures is not accurate and is only used to help visualize the surface orientations. Figure 3 shows a view of the CEV with the surfaces color coded, as used in the solar pressure calculations. Unlike the human eye, the solar pressure calculations can extract depth information from the model, and so do not need to rely on shading to find surface normals.

This model uses accurate CEV dimensions[**?**], but is limited by the simple geometric shapes it uses to model the CEV. For the most accurate CEV calculations, a 3D mesh model should be generated by a CAD system. However, the accuracy of the model is probably not the current limiting factor in the overall accuracy of the program.

Figure 2: CEV model with solar arrays oriented to face the Sun

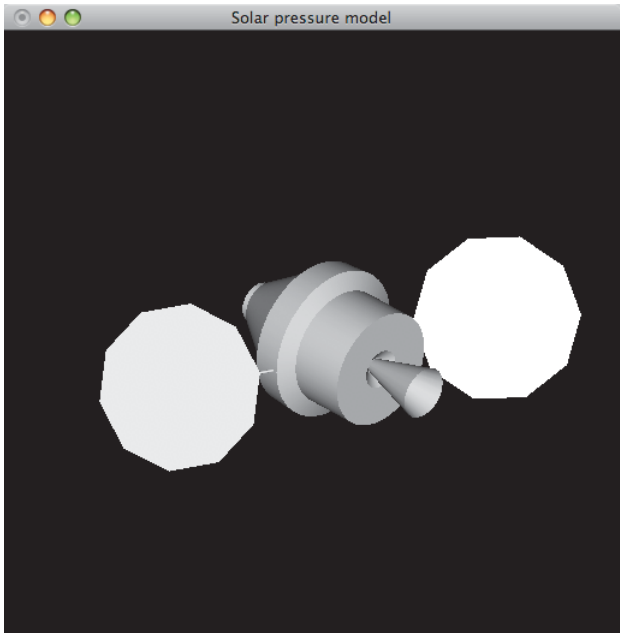
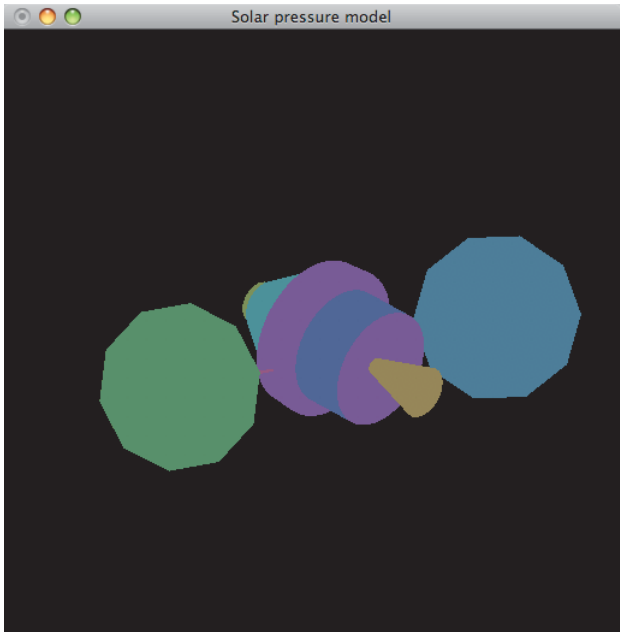


Figure 3: CEV model with components color coded for identification



Additionally, an alternative flat plate model is included in the code for debugging and performance measurement. The flat plate model is used because the solution to its solar pressure calculation is easily calculated.

The proof-of-concept program runs on Unix based computers (Apple OS X and Linux have been demonstrated). Simplicity and clarity of code were prioritized over computational speed, so performance improvements are certainly possible. One of the few optimizations included is a memoizing of the results of the screen-to-model coordinate transformation, and the resulting order of magnitude increase in code size demonstrates the tradeoff between optimization and code clarity. A second optimization (that also greatly increases the code size) is a triangular gridding routine (see Appendix A) to evenly space pitch and yaw attitudes over the surface of a sphere when stepping through all attitudes (gridding is use to generate tables of pre-calculated solar pressure results for incorporation into vehicle simulation programs). Triangular gridding reduces the number of solar pressure calculations by half compared to linearly distributing pitch and yaw angles, and makes it easier to estimate the accuracy change when the grid spacing is varied. However, linear distribution is usually preferred for easier interpolation between results. Some simple features have also been added to split the program into multiple worker processes to take advantage of computers with multiple processor cores.

The OpenGL *pick* functionality was originally used to determine the model component that is visible at each pixel, but was found to be very slow. An alternative implementation was developed that color codes the model components so that identifying the hue associated with each pixel is enough to determine the component. The OpenGL *gluUnProject* function is use to determine the model $[x, y, z]$ coordinate associated with each pixel. The relationship between the model size and pixel size is determined from the current view angle and window height.

3.1 Coordinate Systems and Angles

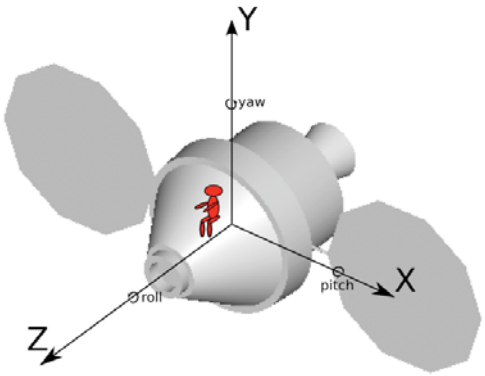
Two coordinate systems are needed in the proof of concept program: one for describing the relationship between the spacecraft components (*model* coordinates) and one that describes the view of the spacecraft from the Sun (*view* coordinates)[1]. The rotations between these coordinate systems define the spacecraft's attitude. The coordinate systems share the same origin, and with no spacecraft attitude rotations (0 degrees pitch, yaw, and roll), the coordinate systems are coincident.

The model origin point was chosen at a convenient spot on the CEV model (the axial center at the widest visible part of the back shell: 3.803 m from the apex of the back shell —the cone shaped part of the vehicle), and components were modeled relative to this point.

The coordinate system of the view from the Sun is fixed to the users view of the computer screen. The positive Z axis was chosen to point in the Sun direction (which is toward the viewer, or out of the computer monitor). The positive X axis is to the screen right. The positive Y axis is to the

[1] An additional optional conversion from the model coordinate system to the Orion CM structural reference coordinate system is described in Appendix B

Figure 4: View coordinate system used in the proof-of-concept program



screen top. With no spacecraft attitude rotations, the model is oriented with its nose pointing along the positive Z axis (toward the Sun). The port solar array boom points along the positive X axis, and the starboard boom points along the negative X axis.

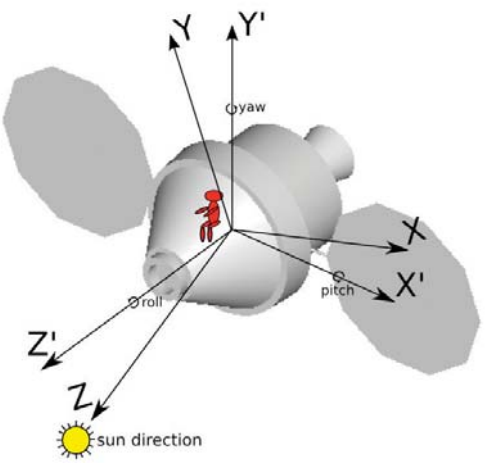
Figure 4 shows the coincident coordinate systems. With no spacecraft attitude rotations and 0 degree solar array gimbal angles, the solar arrays and nose of the vehicle are facing the Sun (positive Z direction). This is the default configuration when the program starts. The direction of a vector $[x, y, z]$ in the Orion body coordinate system corresponds to a vector $[-y, -z, x]$ in the solar pressure model coordinate system.

Spacecraft attitude changes (from model to view) are characterized by 3 Euler angles: pitch, yaw, and roll, applied in that order. Note that these are the program's view angles, not pitch, yaw, and roll as a vehicle pilot would interpret the words. Pitch is a right-handed rotation about the model's positive X axis. A small positive pitch from the default (0 degrees pitch, yaw, and roll) attitude will point the spacecraft nose more toward the bottom of the screen. Yaw is a right-handed rotation about the model's positive Y axis (which may have been rotated already by a pitch). A small positive yaw from the default attitude will point the spacecraft nose more toward the right of the screen. Roll is a right-handed rotation about the model's positive Z axis (which may have been rotated already by a pitch and/or a yaw). A positive roll from the default attitude will rotate the spacecraft counterclockwise as seen on the screen. Pitch is understood by the program to be an angle between -90 and +90 degrees. Yaw and roll angles describe a full circle (0 to 360 degrees). Figure 5 shows the CEV model rotated from being coincident with the view coordinate system $[x, y, z]$ to an new arbitrary attitude $[x', y', z']$ by rotations about its pitch, then yaw, then roll axes.

The two solar arrays each have two gimbal angles. The inner gimbal angle describes the motion of the solar array boom. The boom can be swept forward (positive rotation) towards the spacecraft nose by up to 25 degrees, or back toward the spacecraft engine by up to 40 degrees (negative rotation). The inner gimbal angle's total range is -40 to +25 degrees. The outer gimbal angle describes the rotation of the solar array boom, which ranges from 0 to 360 degrees.

In the spacecraft model coordinate system, the roll rotation is about the vector from the craft to

Figure 5: CEV model rotated to arbitrary attitude $[x', y', z']$



the Sun, so the roll angle does not affect the apparent direction of solar flux. This is useful for reducing the size of the table needed for looking up solar pressure based on pre-calculated values. If the solar flux direction and resulting forces and torques are expressed in model coordinates, the flux direction can be converted to the equivalent pitch and yaw for a two dimensional table lookup instead of three.

The relationship between the pitch and yaw attitude angles and a solar flux direction vector in the model coordinate system $[x_m, y_m, z_m]$ is:

$$
\begin{aligned}
\textit{pitch} &= arcsin(-y_m) \\
\textit{yaw} &= arctan\left(\frac{x_m}{-z_m}\right) \\
\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} &= \begin{bmatrix} sin(\textit{yaw})cos(\textit{pitch}) \\ -sin(\textit{pitch}) \\ -cos(\textit{yaw})cos(\textit{pitch}) \end{bmatrix}
\end{aligned}
$$

4 Optical Properties

Each spacecraft component in the model needs two associated optical properties. The first is the percent of incident light that is reflected (the remainder is absorbed). The second is the percent of reflected light that is specular (the remainder is diffuse). Once the component represented by a pixel is identified, its optical properties can be looked up in a table.

The proof-of-concept program implements the table lookup, and the optical property coefficients are based on current *Beginning Of Life* (BOL) CEV estimates[**?**] (which assume that reflection is

fully diffuse, so the percent of specular reflection is zero). Coefficients for eleven CEV components are modeled. For the flat plate model used for accuracy measurements, coefficients are set to have equal amounts of absorption, diffuse reflection, and specular reflections.

All spacecraft materials are assumed to be opaque. A translucency property could be added to the model, but the raster technique would not identify the surfaces behind a translucent material. The raster technique also cannot account for reflected light between surfaces of the vehicle. The optical properties are assumed to be averages over the entire object being modeled. Decomposing an object model into smaller piece with different properties is possible to increase fidelity.

5 Surface Normals

Surface normal vectors are required to calculate solar pressure. These can be estimated for each pixel by looking at the 3D locations of the surrounding pixels and fitting a plane to the set using a least squares approach[3]. The surface normal is perpendicular to the fitted plane.

There will be a residual error in the plane fitting algorithm if the nine points under consideration are not in the same plane (such as pixels on the edges of the vehicle where a near component obscures a more distant component). If the residual is large enough, the proof-of-concept program concludes that it does not know the surface normal for that particular pixel, and therefore cannot compute the direction of the solar pressure force at that point. If the pressure direction is unknown, the pixel is omitted from further calculations.

6 Solar Pressure Calculation

Once the position, material properties, and the normal vector for each pixel are known, and the size of the apparent area represented by each pixel is calculated, then each pixel can be treated as an independent non-overlapping flat plate. Forces and moments can be calculated for each plate, and summed to find the combined effect on the spacecraft.

6.1 Flat Plate Solar Pressure Calculation

The solar pressure force acting on a flat plate normal to the solar flux is proportional to the plate's area and inversely proportional to its distance from the Sun:

$$f_N = \left(\frac{r_0}{r}\right)^2 \frac{J_0}{c}\, area$$

where r_0 is the Earth's distance from the Sun, r is the space craft's distance from the Sun, J_0 is the average solar energy flux at r_0, and c is the speed of light. The direction of this force is along

the solar flux vector. This assumes a purely absorptive plate (no reflection). A purely specularly reflective plate would double this force, and a purely diffusely reflective plate would increase this force by $2/3$.

The solar pressure force vector acting on a flat plate inclined at angle θ from normal to the solar flux is decomposed into two parts[1, 2]: the part of the force perpendicular to the plate $F_\perp$, and the part of the force tangent to the plate $F_\parallel$.

The magnitude of the two force components is dependent on the material properties of the plate. A coefficient γ represents the percent of reflected photons for a particular plate material, and $(1-\gamma)$ is the percent of absorbed photons. Another coefficient β represents the fraction of reflected photons that are specular, so $\beta\gamma$ is the percent of photons reflected specularly, and $(1-\beta)\gamma$ is the percent of photons reflected diffusely. The percent of absorbed photons, percent of reflected photons that are specular, and percent of photons reflected diffusely sum to 100%.

The magnitude of the force perpendicular to the plate is:

$$f_\perp = f_N \left[(1+\beta\gamma)\cos^2\theta + \frac{2\gamma}{3}(1-\beta)\cos\theta \right]$$

The magnitude of the force tangent to the plate is:

$$f_\parallel = f_N (1-\beta\gamma)\sin\theta\cos\theta$$

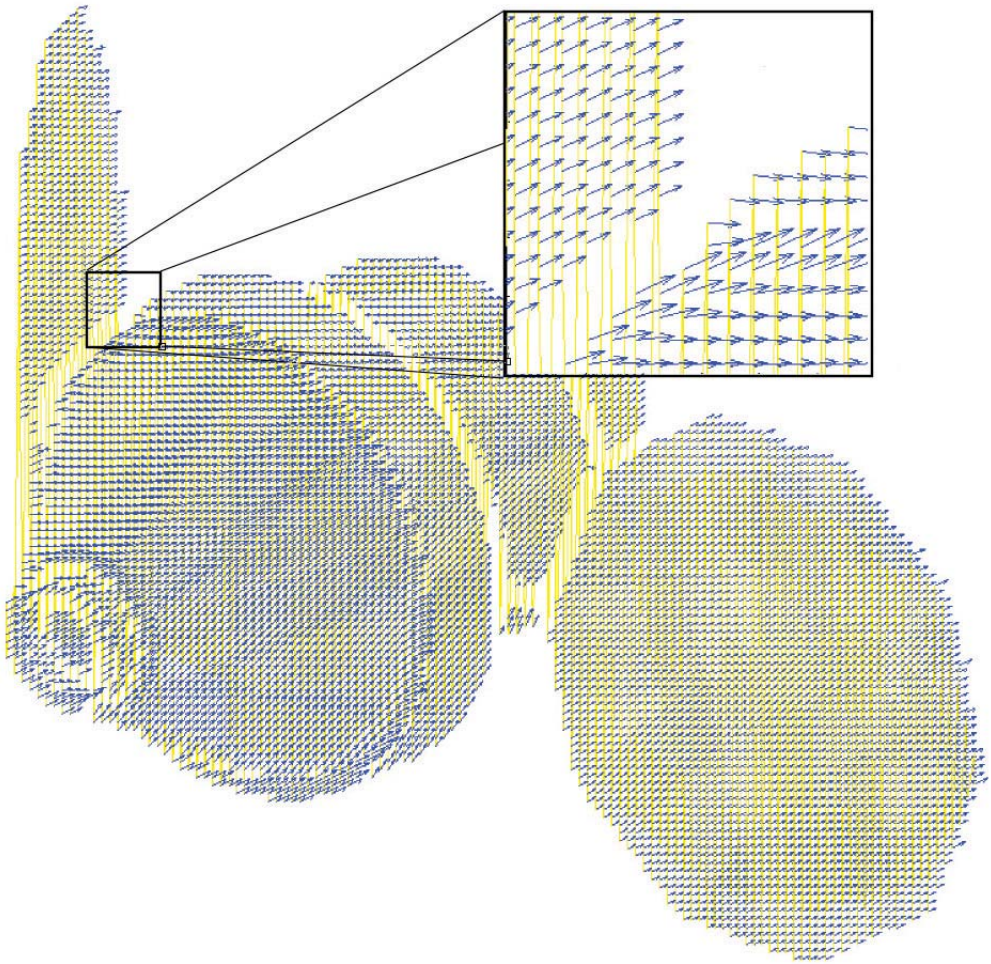
The total force is the vector sum of these two force components, and is not necessarily in the same direction as the solar flux vector. The direction of the force on a purely absorptive plate is the same as the solar flux vector (inclined at angle θ from normal to the plate). The direction of the force on a purely specularly reflective plate is entirely perpendicular to the plate. The direction of the force on a purely diffusely reflective plate is somewhere between the purely absorptive and purely specular directions. In general, the resultant force differs from the solar flux direction by angle:

$$\theta_R = \theta - \arctan\left[\frac{(1-\beta\gamma)\sin\theta}{\frac{2\gamma}{3}(1-\beta) + (1+\beta\gamma)\cos\theta} \right]$$

Figure 6 shows a (very low resolution, but otherwise typical) view of the CEV model with the position and magnitude of the individual flat plate forces shown as arrows. The largest component of the forces is in the solar flux vector direction (perpendicular to the view). The arrows show that the attitudes of the force components perpendicular to the solar flux vary in different parts of the vehicle. The low resolution data used to make the arrows more clear in this figure greatly exaggerates the gaps in the data where components overlap (due to residuals in the surface normal algorithm).

In the proof-of-concept program, the area occupied by one pixel is actually the *apparent area* (cross section area as seen from the Sun). If the flat plate associated with a pixel is inclined from normal to the solar flux at angle θ, the actual area represented by that pixel is the apparent area divided by $\cos\theta$. θ will always be less than 90 degrees, but if θ approaches 90, the actual area represented by a pixel becomes very large.

Figure 6: Force components perpendicular to the solar flux vector



6.2 Combined Forces And Moments

Rasterizing an image to find solar pressure generates a large number (typically tens of thousands) of independent flat plate pressures f_i and locations where those pressures are applied r_i. This information can be combined to find the major results of the algorithm: a single equivalent force and moment (calculated about the model origin):

$$\begin{aligned} f_{total} &= \sum f_i \\ m_{total} &= \sum (r_i \times f_i) \end{aligned}$$

If a center of mass r_{cm} is known (in model coordinates), the vehicle's moment about the center of mass can be calculated from the totals:

$$m_{cm} = m_{total} - (r_{cm} \times f_{total})$$

A different approach for summarizing moments was considered that reported an equivalent *center of pressure* location instead of the total moment. At a center of pressure, all of the individual moments $(r_i \times f_i)$ should sum to zero. However, because the forces f_i are not constrained to the same direction, no such location exists in 3 dimensions.

6.3 Hand Calculation For Rough Estimate

A hand calculation of solar pressure force is useful to show that raster based solar pressure calculation is yielding results in the right neighborhood.

Consider the solar pressure on the CEV, at one Earth radius from the Sun (so $r_0/r = 1$), with its nose and solar arrays facing directly towards the Sun. The illuminated area for this case can be approximated by 3 circles: one for each solar array, and one for the combined back shell and avionics ring (the largest diameter part of the vehicle). The radius of each solar array is roughly 3.0 m, and the radius of the avionics ring is about 2.7 m. So the total illuminated area is about:

$$\begin{aligned} \textit{area} &= 2\pi r_{array}^2 + \pi r_{shell}^2 \\ &= 2\pi(3.0\ [m])^2 + \pi(2.7\ [m])^2 \\ &= 79\ [m^2] \end{aligned}$$

The solar energy flux at one Earth radius J_0 is 1367 $J/m^2 s$. The equivalent momentum flux is found by dividing by c, the speed of light, $2.998 \cdot 10^8\ m/s$:

$$\begin{aligned} \textit{flux} &= J_0/c \\ &= \frac{1367\ [J/m^2 s]}{2.998 \cdot 10^8\ [m/s]} \\ &= 4.56 \cdot 10^{-6}\ [N/m^2] \end{aligned}$$

Multiplying the momentum flux by the illuminated area yields the solar pressure force:

$$\begin{aligned} \textit{force} &= \textit{flux} \cdot \textit{area} \\ &= 4.56 \cdot 10^{-6}\ [N/m^2] \cdot 79\ [m^2] \\ &= 0.000360\ [N] \end{aligned}$$

This rough estimate ignores small components such as the docking mechanism and solar array booms, the non-circular geometry of the solar arrays, the slope of the back shell, and all the material coefficients. Taking the slope of the back shell into account will decrease the solar pressure. This calculation also assumes purely absorptive materials; the reflectivity of the actual components will increase the solar pressure.

7 Combinatorial Explosion

An Orion Guidance Navigation and Control simulation has an interface for looking up solar pressure from a pre-computed table. For full generality, Orion spacecraft orientation (3 angles) and the gimbal angles for the solar arrays (4 angles) need to be supplied to a table lookup routine, and the resulting solar pressure is returned. This raster-based procedure could be used to generate the lookup table by sweeping through every possible combination of angles and storing the results. Small changes in angle will not make much difference in the accuracy of the result, so there is a tradeoff between angular resolution (and therefore, table size) and accuracy. Visually, it appears that a 5 degree angular resolution is approximately adequate (more accurate estimates are shown in Table 3 and Table 4).

Table size quickly becomes infeasible with small angular resolutions. For example, if the angle for pitch can vary between -90 and +90 degrees, and the four angles for vehicle yaw, roll and the two outer solar array gimbals can range from 0-360 degrees, and the two inner solar array gimbals range from -40 to +25 degrees, and angular resolution linearly distributed in 5 degrees increments, there will be $(180/5+1)\cdot(360/5)^4\cdot(65/5+1)^2 = 194,889,203,712$ entries in the table. If the table stores x, y, and z values of force and moment using 8 byte double precision floating point numbers, each table entry will require $3\cdot 2\cdot 8 = 48$ bytes of space. The table will require a total of 8.5 TB of memory. If a 512x512 resolution solar pressure calculation takes 0.3 seconds to compute, over 1800 years would be required to create this table. Clearly, a different approach is necessary.

One alternative is to assume that the solar arrays will always be doing their best to be pointing at the Sun. So, at any given spacecraft attitude (roll, pitch, yaw), there is only one configuration of the solar array gimbals that needs to be examined. With an angular resolution of 5 degrees, this results in a reduction to $(180/5+1)\cdot(360/5)^2 = 191,808$ entries in the table, requiring 4.4 MB of space and 16 hours to compute. Finding the optimal array pointing for each spacecraft orientation involves additional calculations, but if "optimal" is restricted to mean just finding the gimbal orientations that point closest to the Sun (ignoring potential shading by other spacecraft components), the calculations do not take much time.

Another simplification is to leave roll variations out of the table. Roll changes have no effect if the force and torque results are reported in model coordinates. Some other code will be necessary to convert from model coordinates to inertial coordinates, but that code is probably already present in a simulation. This allows 1.0 degree pitch and yaw attitude variations with optimal solar array gimbal angles to result in $(180 + 1) \cdot 360 = 65,160$ entries in the table, requiring 3.0 MB of space and 5 hours to compute.

Another alternative is to abandon table based pre-calculations and calculate solar pressure for only the actual configurations that are seen. This approach could be combined with a table that assumes optimal solar array pointing for nominal configurations, but uses on-the-fly calculations for the exceptional configurations when the solar arrays are not pointing at the Sun.

8 Performance

8.1 Image resolution trade-offs

The accuracy of the result is dependent on the resolution of the image being processed. A resolution that is too low will be inaccurate, but a resolution that is too high will take too long to calculate. A range of resolutions was examined to determine the best resolution trade off between adequate accuracy and computation speed.

An alternative to reducing the resolution of the image is to use a larger image but only process every Nth pixel. This would probably have more accurate surface normal information, but might also include aliasing effects due to sub-sampling without applying a low pass filter. Maybe an appropriate filter can be developed.

Table 1 shows how resolution and step size can be adjusted to control run time and accuracy. The error was calculated by subtracting the calculated force (and moment) from the ideal solution for a single flat plate model with a known area of exactly 4 square meters, oriented so that its normal vector coincides with the view vector. This allows the ideal force numbers to be calculated (2.835e-05 Newtons entirely in the -Z direction). Ideally, the torque should be zero. The "Plates" number in the table is the number of pixels (independent flat plate models) contributing to the calculation.

The resultant force calculated by summing the pixels in the solar pressure model is in exactly the right direction (the -Z axis), but its magnitude has some error. This error is completely proportional to the estimated area error, [2] so it can be attributed to the inaccuracies of estimating the size of an object by measuring how many pixels it occupies in an image.

Based on the results shown in Table 1, the default resolution for the proof-of-concept program was chosen to be 512x512 pixels, with a step size of 1. This choice results in a run time of about .3 seconds per solar pressure calculation, and overestimates the total force by .3%.

[2] The force error in the table is always reported as positive due to a Root-Mean-Square operation, but its magnitude matches the signed area error. The sign of the area error indicates an overestimate vs. an underestimate.

Table 1: Performance vs. resolution

Step Size	Window Resolution	Run Time (s)	Area (m^2)	Plates	Force Error (N)	Moment Error ($N \cdot m$)
5	64x64	0.0009	3.77 (-5.74%)	30	1.63e-06 (5.74%)	3.42e-06
4	64x64	0.0009	3.94 (-1.46%)	49	4.15e-07 (1.46%)	2.80e-06
3	64x64	0.0012	4.07 (1.80%)	90	5.12e-07 (1.80%)	2.29e-06
2	64x64	0.0018	3.94 (-1.46%)	196	4.15e-07 (1.46%)	8.55e-22
1	64x64	0.0048	3.94 (-1.46%)	784	4.15e-07 (1.46%)	1.40e-06
5	128x128	0.0025	4.15 (3.69%)	132	1.05e-06 (3.69%)	2.15e-06
4	128x128	0.0026	3.94 (-1.46%)	196	4.15e-07 (1.46%)	1.40e-06
3	128x128	0.0040	4.08 (2.09%)	361	5.92e-07 (2.09%)	1.64e-22
2	128x128	0.0062	3.94 (-1.46%)	784	4.15e-07 (1.46%)	3.16e-22
1	128x128	0.0186	3.94 (-1.46%)	3136	4.15e-07 (1.46%)	7.00e-07
5	256x256	0.0099	3.80 (-4.95%)	484	1.40e-06 (4.95%)	3.38e-07
4	256x256	0.0099	3.94 (-1.46%)	784	4.15e-07 (1.46%)	7.00e-07
3	256x256	0.0150	3.98 (-0.60%)	1406	1.70e-07 (0.60%)	5.59e-07
2	256x256	0.0247	3.94 (-1.46%)	3136	4.15e-07 (1.46%)	4.47e-22
1	256x256	0.0762	3.94 (-1.46%)	12544	4.15e-07 (1.46%)	3.50e-07
5	512x512	0.0301	3.98 (-0.58%)	2025	1.65e-07 (0.58%)	1.46e-21
4	512x512	0.0409	4.01 (0.30%)	3192	8.41e-08 (0.30%)	3.56e-07
3	512x512	0.0634	3.98 (-0.58%)	5625	1.65e-07 (0.58%)	1.14e-21
2	512x512	0.1026	4.01 (0.30%)	12769	8.63e-08 (0.30%)	3.56e-07
1	512x512	0.3133	4.01 (0.30%)	51076	8.63e-08 (0.30%)	1.78e-07
5	1024x1024	0.1226	4.02 (0.52%)	8190	1.48e-07 (0.52%)	2.28e-07
4	1024x1024	0.1628	4.01 (0.30%)	12769	8.63e-08 (0.30%)	1.78e-07
3	1024x1024	0.2567	4.00 (0.08%)	22650	2.31e-08 (0.08%)	1.41e-07
2	1024x1024	0.4163	4.01 (0.30%)	51076	8.63e-08 (0.30%)	2.68e-22
1	1024x1024	1.2551	4.01 (0.30%)	204304	8.63e-08 (0.30%)	8.91e-08

8.2 Accuracy trade-offs

There are inherent inaccuracies in using an image base approach. A single pixel only approximates the area it represents. There may be quantization or edge effects introduced where the model does not line up properly with the pixel grid.

Table 2 shows some measurements of the effects of spatial jitter. Ideally, slightly moving a model in the plane perpendicular to the solar flux should have no effect on the solar pressure. However, the solar pressure model output can change with small position offsets due to the way the image pixels are gridded onto the view. These measurement are for a 512 by 512 resolution image of a flat plate, with a step size of 1 (no sub-sampling). One hundred measurements were taken with the model shifted by a slight random amount in the X and Y planes (perpendicular to the view vector).

The average area error (and equivalently, force error) underestimates the true value by 0.3%, with a standard deviation of 0.32%.

Table 2: Jitter measurements

Test	Time	Area	Area Error	Plates	Force Error	Moment Error
1	0.314	4.012	0.30%	51076	8.63e-08	1.78e-07
2	0.312	3.994	-0.14%	50850	3.95e-08	3.51e-07
3	0.314	4.012	0.30%	51076	8.63e-08	2.82e-07
4	0.312	3.994	-0.14%	50850	3.95e-08	3.58e-07
...	...	...	...	...	...	...
97	0.312	3.994	-0.14%	50850	3.95e-08	3.75e-07
98	0.313	4.012	0.30%	51076	8.63e-08	2.46e-07
99	0.313	3.994	-0.14%	50850	3.95e-08	2.21e-07
100	0.311	3.977	-0.58%	50625	1.65e-07	4.88e-07
Mean	0.319	3.999	-0.30%	50906	8.05e-08	2.58e-07
Std Dev	0.00121	0.0129	0.32%	164.6	4.39e-08	1.11e-07

8.3 Angular resolution trade-offs

To estimate error dues to insufficient angular resolution in a solar pressure lookup table, a sweep of all possible CEV model attitudes was run at approximately[3] 2.5 degree intervals. The results of this sweep were considered to be "truth", and then compared to the results obtained by interpolating the values at 2.5 degree intervals from calculations made at 5, 10, and 20 degree intervals. This provides an estimate of the error introduced by sampling attitude at too high of an angular resolution.

[3] For sweeping through roll angles, roll was stepped from 0 to 360 by 2.5 degrees. However, pitch and yaw were distributed using a triangular gridding algorithm to space them evenly over the sphere with a minimum spacing of $\pi/64$, or 2.8125 degrees.

Table 3 shows the total error in area, force magnitude and moment magnitude due to sampling roll attitude at 5, 10, and 20 degree intervals and then interpolating estimates for 2.5, 5, and 10 degree intervals. The total error is the average over all examined attitudes.

Table 3: Roll interpolation error

	Area	Force Magnitude	Moment Magnitude
interpolated from $5\,^\circ$	0.16%	0.16%	0.88%
interpolated from $10\,^\circ$	0.29%	0.28%	1.25%
interpolated from $20\,^\circ$	1.06%	0.98%	3.00%

Table 4 shows the total error in area, force magnitude and moment magnitude due to sampling pitch and yaw attitudes at $\pi/32$ radian (5.625 degree), $\pi/16$ radian (11.25 degree), and $\pi/8$ radian (22.5 degree) intervals and then interpolating estimates for $\pi/64$ radian (2.8125 degree), $\pi/32$ radian (5.625 degree), and $\pi/16$ radian (11.25 degree) intervals.

Table 4: Pitch-Yaw interpolation error

	Area	Force Magnitude	Moment Magnitude
interpolated from $5.625\,^\circ$	0.45%	3.66%	0.48%
interpolated from $11.25\,^\circ$	1.22%	8.08%	1.29%
interpolated from $22.5\,^\circ$	2.77%	18.3%	2.83%

These sweeps show that the calculations are more sensitive to pitch and yaw attitude than to roll attitude. An angular resolution of of $\pi/32$ radians (5.625 degrees) for pitch and yaw and 10 degrees for roll is sufficient to keep the overall area error under 1%, but the force magnitude error of 3.66% is still very significant.

9 CEV Solar Pressure Overview

The 2.5 degree attitude sweep used for finding the angular resolution trade-offs is also useful as an overview of all possible CEV solar pressure data. The sweep generated area, force, and torque results for 590112 unique combinations of pitch, yaw and roll (specified in view coordinates). With optimized solar array gimbal angles, the maximum observed difference between the solar flux direction and the total solar pressure force direction is 3.23 degrees.

Figure 7 is a histogram showing the distribution of illuminated area for all attitudes. The median area is about 73 m^2, but the distribution is highly skewed. A few attitudes result in areas lower than 70 m^2 going down to as low as 32 m^2. Many attitudes result in higher areas of up to 78 m^2, but none go as high as 80 m^2. The 79 m^2 result from the hand calculation is toward the end of the distribution's upper tail.

Figure 7: Histogram of illuminated area for all attitudes

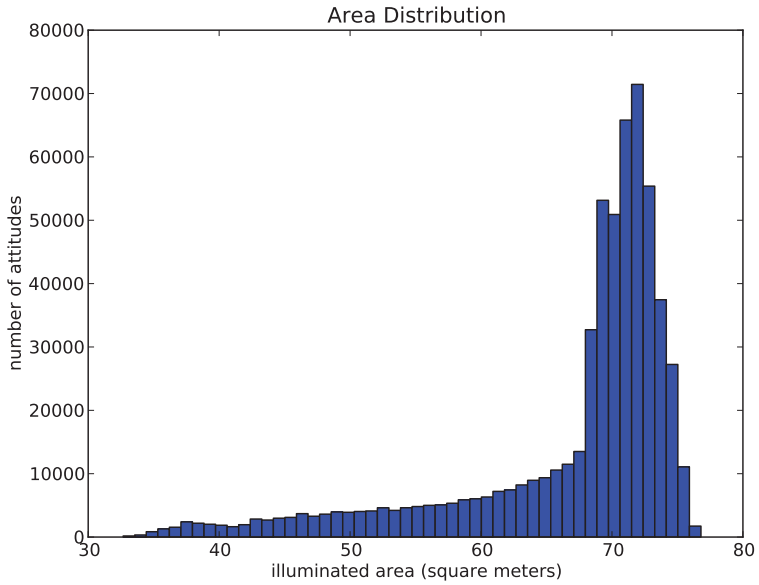


Figure 8 shows the magnitude of the solar pressure force (in view coordinates) as a function of roll attitude, with one line plotted for every combination of pitch and yaw. Force magnitude ranges between 200 and 400 μN. Some (pitch, yaw) combinations result in a fairly constant solar pressure force near the top of its range, while others tend to sweep between the range extremes as the craft rolls.

Figure 9 shows the magnitude of the solar pressure torque (in view coordinates, assuming a center of mass at the model origin) as a function of roll attitude, with one line plotted for every combination of pitch and yaw. Torque magnitude ranges between 0 and 1100 $\mu N \cdot m$. Many (pitch, yaw) combinations result in a fairly constant solar pressure torque as roll changes, evenly distributed in the 0 to 600 $\mu N \cdot m$ range. However, there are some (pitch, yaw) combinations that result in the torque sweeping between 400 $\mu N \cdot m$ and 1100 $\mu N \cdot m$ with roll changes.

Discontinuities due to the solar panel gimbal angle optimization are present in some of the torque magnitude lines that sweep with roll changes. In some attitudes, a small change in roll can cause a large change in the optimum solar panel gimbal angles. The discontinuity in the torque plot shows the effect of reconfiguring the solar arrays.

Figure 10 shows the solar pressure force magnitude in model coordinates, where roll does not change the results. The force is greatest when the spacecraft's nose is facing the Sun (yaw and pitch both zero). The force is smallest when the spacecraft is yawed 90 degrees so that one of the solar arrays is shielded from the Sun.

Figure 11 shows the solar pressure torque magnitude in model coordinates (assuming a center of

Figure 8: Solar pressure force magnitude vs. roll at all values of (pitch,yaw)

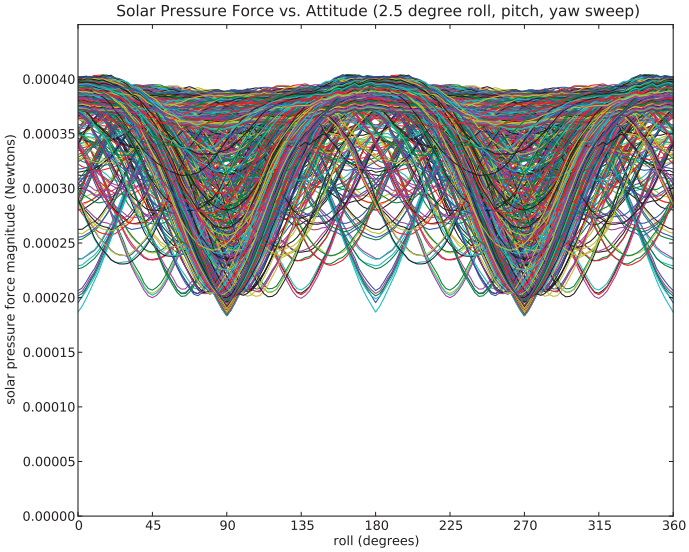


Figure 9: Solar pressure torque magnitude vs. roll at all values of (pitch,yaw)

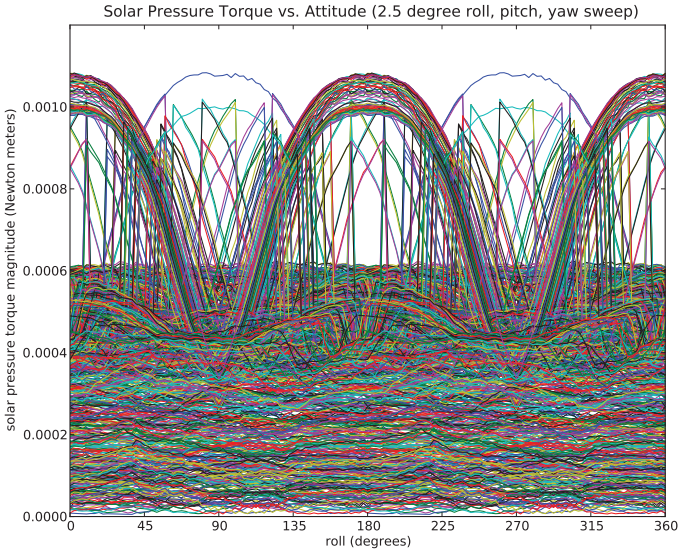


Figure 10: Solar pressure force magnitude in model coordinates

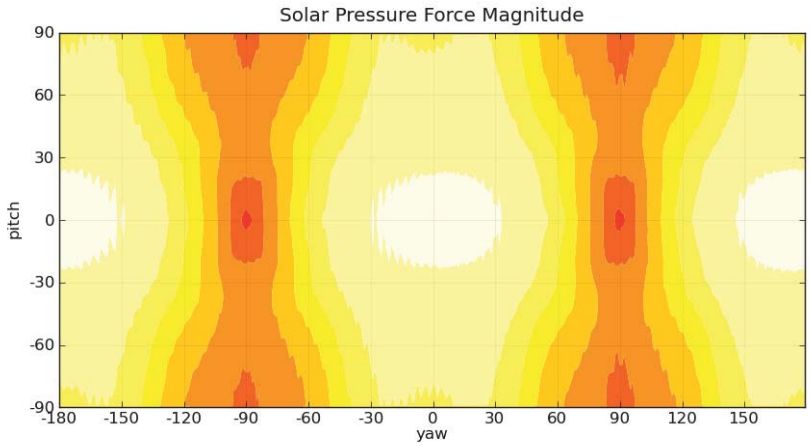
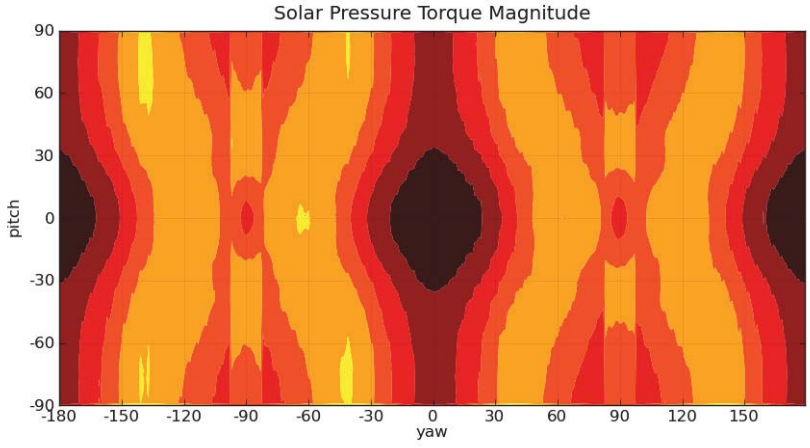


Figure 11: Solar pressure torque magnitude in model coordinates



mass at the model origin), where roll does not change the results. The torque is smallest (near zero) when the spacecraft's nose is facing the Sun (yaw and pitch both zero). The torque is greatest when the spacecraft is yawed about 60 degrees, presenting the least symmetrical aspect to the Sun.

10 Computing Other Attributes

10.1 Power Generation

Solar power generation of spacecraft components in the presence of self shading is usually difficult to compute, but the solar pressure rasterization process is easily extended to compute power for each pixel in the image and sum those powers to get totals for the whole vehicle and each model component.

An additional material component is added to specify the power generation potential of each material in $watts/m^2$, using zero for materials that do not generate solar power. The cross sectional area represented by each pixel exposed to the Sun is found during the solar pressure computation, and it is multiplied by the power generation potential of the component it represents to get the power generated by that area. Totals are found for the vehicle as a whole and for each of the separately modeled vehicle components.

The values currently reported assume a solar flux constant of $1367\ watts/m^2$ multiplied by a solar array conversion efficiency of 17%. Results can be scaled proportionally for other values of solar constant or efficiency.

10.2 Free Molecular Flow Drag

If the view direction is assumed to be the direction of wind rather than the direction of light, the raster algorithm can be used to compute free molecular flow drag. In free molecular flow drag conditions, the atmosphere is so thin that the air molecules do not react with each other, so fluid dynamics models are not required. The rasterization process will identify vehicle areas that are exposed to the wind. There are free molecular flow contribution from areas that are not exposed to wind, but in orbital conditions they are small enough to ignore.

After rasterization, each pixel is assumed to represent a small flat area. The normal and parallel components (Bird[4] *corrected*) of free molecular flow drag force due to a flat surface of area A are:

$$C_n = \left[\left(\frac{(1+\epsilon)}{\sqrt{\pi}} s \sin\alpha + \frac{(1-\epsilon)}{2} \sqrt{\frac{T_r}{T_\infty}} \right) e^{-s^2 \sin^2\alpha} + \left((1+\epsilon)(\frac{1}{2} + s^2 \sin^2\alpha) + \frac{(1-\epsilon)}{2} \sqrt{\frac{T_r}{T_\infty}} \sqrt{\pi} s \sin\alpha \right) (1 + \mathrm{erf}(s \sin\alpha)) \right] / s^2$$

$$C_p = \frac{(1-\epsilon)}{\sqrt{\pi}} s \cos\alpha \left[e^{-s^2 \sin^2\alpha} + \sqrt{\pi} s \sin\alpha (1 + \mathrm{erf}(s \sin\alpha)) \right] / s^2$$

The total force on a vehicle due to free molecular flow drag is these normal and parallel components multiplied by the area where they are in effect, summed over the entire vehicle surface, and multiplied by the dynamic pressure:

$$\bar{F} = \left(\frac{\rho V^2}{2} \right) \sum_i A_i (C_{n_i} \bar{n}_i + C_{p_i} \bar{p}_i)$$

The total moment about a moment reference center (*MRC*) due to free molecular flow drag is found by including a cross product with the vector from the MRC to the center of the area ($\bar{r}$):

$$\bar{M} = \left(\frac{\rho V^2}{2} \right) \sum_i \bar{r}_i \times A_i (C_{n_i} \bar{n}_i + C_{p_i} \bar{p}_i)$$

α is the angle of incidence of the relative wind. $\alpha = 0$ means the area is parallel to the wind, $\alpha = 90$ means the area is facing the wind, and $\alpha = -90$ means the area is facing away from the wind. Force components where $\alpha < 0$ are non-zero, but very small (when the other variables are typical of orbit conditions). Note that the angle of incidence use for drag calculations is not the same the angle of incidence used for solar pressure calculations ($\alpha = 90 - \theta$).

ϵ is the *accommodation coefficient* —the percent of molecules that are specularly reflected, and (1 - ϵ) is the percent that are absorbed and then diffusely re-emitted. ϵ can be different for each material in a model, but by default a value of 0 (fully diffuse) is used for all materials. Schaff[6] lists ϵ values for several materials ranging from 0.03 to 0.13.

s is the molecular speed ratio (stream speed over most probable molecular speed). It is inversely proportional to the square root of the gas temperature and zero for a stationary object. Bird says it is typically near 10 in orbital conditions. It is calculated with:

$$R_{universal} = 8.3144621$$

$$R_{specific} = R_{universal} / (0.001 \cdot mw)$$

$$T_r = (1-\epsilon) T_{wall} + \epsilon\, T_\infty$$

$$s = \frac{V}{\sqrt{2 R_{specific} T_\infty}}$$

$R_{universal}$ is the Universal Gas Constant. $R_{specific}$ is the specific gas constant for air with the molecular weight mw (which defaults to $24.97578\ kg/kg-mole$). T_∞ is the free stream temperature, which defaults to $746.22\ K$. T_{wall} is the spacecraft wall temperature, which defaults to a best practice value of $300\ K$. ρ is the density of air, which defaults to $4.4 \times 10^{-9}\ kg/m^2$. V is the vehicle speed relative to the wind, defaulting to $7000\ m/s$.

The default values of $mw, T_\infty, T_{wall}, \rho, V$, and ϵ were chosen because they are the values used for the analysis in the Orion Aero Data Book. The mw, T_∞ and ρ values are a function of altitude, and they come from the ***GRAM-07*** atmosphere model at an altitude of $480,000\ ft$. The V value was chosen to be representative of low Earth orbit, but a more detailed simulation might compute it from orbital velocity, the rotational speed of the Earth's atmosphere, and winds at a given altitude (also available from GRAM-07).

$\left(\frac{T_r}{T_\infty}\right)$ (aka, ***tRatio***) is the ratio of the vehicle surface temperature to the temperature of the gas molecules.

erf is the Gauss error function (a sigmoidal function related to the standard normal distribution).

The proof of concept program allows non-default values of s, ***tRatio***, $mw, T_\infty, T_{wall}, \rho, V, \epsilon$, and A_{ref} (the reference area) to be easily specified using "override" variables.

The total force $\bar{F}$ and moment $\bar{M}$ vectors are often described using various coefficients that are independent of the dynamic pressure ($\rho V^2/2$) and cross sectional area intercepting the wind (A_{ref}):

$$\bar{F} = \left(\frac{\rho V^2}{2}\right) A_{ref}(C_x\bar{x} + C_y\bar{y} + C_z\bar{z})$$

$$\bar{M} = \left(\frac{\rho V^2}{2}\right) A_{ref}L_{ref}(M_x\bar{x} + M_y\bar{y} + M_z\bar{z})$$

L_{ref} is calculated assuming it is the diameter of a circle with the given A_{ref} area: $L_{ref} = 2\sqrt{A_{ref}/\pi}$

In the program's model frame of reference, then wind is moving in the $-z$ direction, and the $-C_z$ coefficient is C_d, the *coefficient of drag*. The magnitude of the components perpendicular to the wind ($\sqrt{C_x^2 + C_y^2}$) is C_l, the *coefficient of lift*.

The program can optionally report results in the Orion CM structural frame of reference (described in Appendix B):

- positive X toward the back (engine bell) of the vehicle
- positive Y toward the starboard solar array side of the vehicle
- positive Z toward the top (pilot's head) of the vehicle
- the Moment Reference Center is moved to the point where the cone apex of the CEV back shell would be if it were extended to be a full cone (instead of the model origin)

In the Orion CM structural frame of reference:

- C_x is the *axial force coefficient*
- C_y is the *side force coefficient*
- C_z is the *normal force coefficient*
- M_x is the *rolling moment coefficient*
- M_y is the *pitching moment coefficient*
- M_z is the *yawing moment coefficient*

If the reference area A_{ref} is the cross sectional area of the vehicle intercepting the wind (which varies with attitude), then the coefficients will be independent of the vehicle scale. This is the default for the program, and the computed value of A_{ref} is one of the outputs.

The Orion Aero Data Book results are reported using a fixed (does *not* vary with attitude) value for A_{ref} of $19.86\ m^2$ (or $30791\ in^2$) that is based on the vehicle "nose facing the wind" cross sectional area (without solar arrays). The "override" variable for A_{ref} causes the program to report coefficients using a fixed reference area (and a fixed L_{ref} calculated from the fixed A_{ref}).

11 Summary

A raster based approach to calculating solar pressure for space vehicles is described. This approach takes into account the shading of vehicle components by other vehicle components. A two dimensional view of a three dimensional model is used to determine the parts of the craft exposed to solar pressure, and each pixel in the 2D image is treated as a flat plate when calculating solar pressure.

A proof-of-concept program was written to determine the accuracy and performance of the raster based approach.

The raster approach has the advantage of accounting for complicated geometry without a lot of modeling code. A 3D graphics library's most basic function is rendering a view of an object for display on a computer screen, and the only significant additional information needed from the model is the depth coordinate of each visible pixel.

The disadvantage of the raster approach is that the image is an approximation. Even with a relatively high resolution, overall accuracy was measured to be no better than 0.3% error.

On a 2010 vintage desktop computer, the typical run time of a single solar pressure calculation at a 512x512 resolution is 0.3 seconds. Pre-calculating the solar pressure for all possible attitudes and model configurations for fast table lookup during a simulation has been suggested, but it is not computationally feasible without some simplifying assumptions. One helpful assumption is that the solar array gimbals will always take on the optimal values to point the arrays toward the Sun, so only one set of gimbal angles needs to be computed for each potential spacecraft attitude. With

this assumption, a table of solar pressure as a function of spacecraft pitch and yaw[4] at a resolution of 1.0 degree can be created in a few hours. Attitude angles between the 1.0 degree increments can be interpolated.

Recent additions to the program allow it to be called directly from other software, providing an alternative to a table lookup interface.

Potential future work:

- Make it easier to add new models by loading an mesh exported from a CAD system instead of using OpenGL calls to build models
- Increase performance by taking better advantage of parallelism. Find and fix the performance problems observed when running on remote servers (allowing multiple networked machines to be used effectively)

[4]Roll can be omitted from the table if results are reported in model coordinates

12 Sources

[1] R.M. Georgevic, *Mathematical Model of the Solar Radiation Force and Torques Acting on the Components of a Spacecraft*
NASA Technical memorandum 33-494
October 1, 1971

[2] Peter C. Hughes, *Spacecraft Attitude Dynamics*
Dover Publications, Inc., Mineola N.Y
2004

[3] Michael Duckwitz, *Best Fit Plane Removal*
http://hkn.colorado.edu/resources/latex/plane-removal/plane-removal.pdf
January 24, 2007

[4] G.A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*
Oxford University Press
1994

[5] J. A. Storch, *Aerodynamic Disturbances on Spacecraft in Free-Molecular Flow*
Aerospace Corp report number TR-2003(3397)-1
2002

[6] S. A. Schaff and P. L. Chambre, *Flow of Rarefied Gasses*
Princeton University Press
1958

Figure 12: The 6 vertices of an octahedron locate the first attitudes

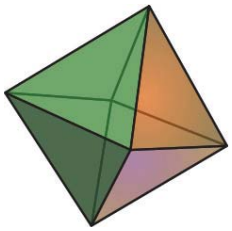
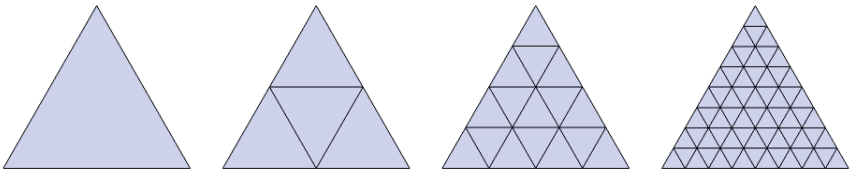


Figure 13: Triangular faces are subdivided to get additional evenly spaced vertices



Appendices

A Triangular Gridding Algorithm

A triangular gridding algorithm is used to nearly evenly distribute pitch and yaw attitudes over the surface of a sphere when examining all attitudes, avoiding redundant calculations near the "poles" as seen when using linearly spaced values of pitch and yaw.

The algorithm starts by approximating a sphere as an octahedron (Figure 12) - a platonic solid with 8 triangular faces and 6 vertices. Each of the 6 vertices is an evenly spaced (pitch,yaw) combination for spacecraft attitude.

To generate more vertices at closer intervals, the edges of each triangular face are split in the center to form new vertices, and lines are formed between the new points dividing each triangle into 4 new triangles (Figure 13).

Dividing the faces of the octahedron one time changes the minimum angle between the vertices from $\pi/2$ to $\pi/4$ radians, resulting in 18 even spaced vertices. Recursively applying this procedure generates $66, 258, 1026, 4098,$ ... $2^{2n} + 2$ evenly spaced vertices, reducing the minimum angle by a factor of 2 with each iteration.

The triangle gridding algorithm has been removed in recent versions of the program, because the increased complexity in interpreting results outweighs the smaller table sizes that it enables. The description of it remains to help document all the techniques that were investigated to reduce lookup table size.

B CEV Model To Orion CM Structural Coordinates

The proof of concept program optionally includes code that saves the output forces and torques in the structural coordinate frame $[x_s, y_s, z_s]$ used by Orion instead of the program's internal model coordinate frame $[x_m, y_m, z_m]$. This allows the outputs to be used for table lookup of solar pressure effects.

Locations in the two frames have their axes switched, and the origin is offset by 3.803 meters:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} 3.803 - z_m \\ -x_m \\ y_m \end{bmatrix}$$

Force directions in the two frames can be converted by simply switching axes:

$$F_{total_s} = \begin{bmatrix} F_{x_s} \\ F_{y_s} \\ F_{z_s} \end{bmatrix} = \begin{bmatrix} -F_{z_m} \\ -F_{x_m} \\ F_{y_m} \end{bmatrix}$$

The torque about the origin in the model frame is converted by moving the point of application to the structural frame origin:

$$\begin{aligned} M'_{total_m} &= M_{total_m} - (r_{origin_s} \times F_{total_m}) \\ \begin{bmatrix} M'_{x_m} \\ M'_{y_m} \\ M'_{z_m} \end{bmatrix} &= \begin{bmatrix} M_{x_m} \\ M_{y_m} \\ M_{z_m} \end{bmatrix} - \left(\begin{bmatrix} 0 \\ 0 \\ 3.803 \end{bmatrix} \times \begin{bmatrix} F_{x_m} \\ F_{y_m} \\ F_{z_m} \end{bmatrix} \right) \end{aligned}$$

and then switching axes to the structural frame:

$$M_{total_s} = \begin{bmatrix} M_{x_s} \\ M_{y_s} \\ M_{z_s} \end{bmatrix} = \begin{bmatrix} -M'_{z_m} \\ -M'_{x_m} \\ M'_{y_m} \end{bmatrix}$$

If the center of mass is supplied in structural coordinates, the torque can be calculated there:

$$M_{cm_s} = M_{total_s} - (r_{cm_s} \times F_{total_s})$$