Orbit simulator

Prepared by: Joost Aan de Brugh

Document Change Record

Issue	Date	Changed Section	Description of Change
0.1	30 November 2016	All	Initial version.
0.2	21 June 2017	1.1, 3	Height input and implementation.

Table of Contents

Abbreviations and Acronyms

Item	Meaning
MIPrep	Multi-Instrument Preprocessor
UTC	Coordinated Universal Time

Reference Documents

[1] Paul Tol. Orbit simulation. Technical report, SRON, 2016.

[2] Wikipedia. Quaternions and spatial rotation. https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation.

[3] Jean Meeus. *Astronomical Algorithms*. Willmann-Bell, Richmond, Virginia, 1991.

This document describes an algorithm to simulate a granule or an orbit by calculating the geometry. It is based on the orbit simulator described in [1], but is extended to support all possible inclination angles, avoids one numeric interpolation step, and some mathematical derivations are explained more extensively. This orbit simulator is incorporated in the software package MIPrep, so MIPrep can directly be used to collocate auxiliary data.

1 Input

One input file is a recipe for one granule or one full revolution. However, the properties are set up such that a large part is constant for the same satellite and only a fraction changes when simulating different revolutions by the same satellite.

1.1 The satellite

The satellite is assumed to orbit the Earth in a circle, not an ellipse. Moreover, the Earth is approximated as a sphere, not a ellipsoid.

The orbit is a circular motion in the reference frame of the Earth's center point. That means that satellite is considered to rotate around the sun as part of the Earth, though it rotates around the Earth's center on its own, ignoring the Earth's rotation around her own axis.

The satellite as an orbiting object is specified with three parameters:

- The time for one revolution (t_r) or the satellite height (Z).
- The local solar time at the northward equator pass (T_{eq}).
- The orbit inclination angle (γ).

The time of one revolution and the height of the satellite orbit are bound to each other with the law of gravity. One should be given and the other will be calculated. If the height is given, it will be converted to the distance from the Earth center: $z = Z + r_E$ with $r_E = 6371$ km. The relationship between the revolution time and the satellite height is

$$z = \sqrt[3]{\frac{Gm_E \cdot t_r^2}{(2\pi)^2}} \tag{1}$$

with $Gm_E = 398600.4418\ \mathrm{km^3 s^{-2}}$, the universal gravitational constant multiplied by the mass of the Earth, in other words, the Earth's gravitational parameter.

The orbit inclination is the angle between the orbital plane and the equator plane. The conventions is to define this angle between 0° and 180°, where 0° is an eastward orbit and 180° is a westward orbit. With 90° inclination, the orbit covers the poles.

Because the Earth's rotation is ignored by the satellite, the local solar time at the sub-satellite point is not influenced by the Earth's rotation. In other words, the local solar time only depends on the longitude of the sub-satellite point if the Earth would not rotate. For example, if the orbital inclination is 90°, the local solar time is constant until it passes a pole, in which case the local solar time shifts twelve hours. For all orbital inclinations, there is a periodic local solar time that is the same for each revolution. Together with the orbital inclination, the local solar time at one orbital phase, e.g. the northward equator pass, determines the local solar time everywhere in the orbit.

One revolution is defined from equator pass to equator pass. If the northward equator pass is during the day (between 6:00 and 18:00 local solar time), one revolution will be defined from southward equator pass to southward equator pass. Otherwise, it will be defined from northward equator pass to northward equator pass. So, the daytime part of the revolution is always in the middle.

The northward equator pass time is indeed northward because the inclination angle is defined between 0° and 180°. When setting an inclination angle between −180° and 0°, the northward equator pass will be southward and that is confusing. Technically, the orbit stays the same if the inclination angle is taken negative and the northward equator pass time is shifted twelve hours. But then, the northward equator pass is southward, so it is adviced to stick to the conventions and define the inclination angle between 0° and 180°. For an orbit with southward flying direction during the day, a local solar time at northward equator pass between 18:00 and 6:00 should be defined. Note that the algorithm will ensure that the day part of the revolution is in the middle.

1.2 A revolution

One revolution is defined with two more parameters:
- The date: year month and day.
- The longitude at daytime equator pass (λ_{eq}).

The date is only used for the sun model, most importantly for the seasonal cycle. The daytime equator pass can be the northward equator pass. That is if the local solar time at northward equator pass is between 6:00 and 18:00. Otherwise, this is the southward equator pass. The longitude of the daytime equator pass is always $\frac{t_r}{\mathrm{day}} \cdot 2\pi$ west of the previous revolution.

1.3 Scanlines

The revolution is sampled in scanlines. These scanlines are assumed to be pictures taken at one time, so each scanline corresponds to one sub-satellite point in the orbit. These scanlines are defined by the following additional parameters.
- Orbit phase at the start of scanning (p_s).
- Orbit phase at the end of scanning (p_e).
- Time per scan (t_s).

The orbit phase is defined as a number between 0 and 1, where 0 corresponds to the night-time equator pass at the beginning of the revolution, $\frac{1}{2}$ corresponds to the daytime equator pass and 1 corresponds to the night-time equator pass at the end of the revolution.

The time per scan is the time in UTC between two scanlines. A total of $\frac{t_r}{t_s}$ scanlines fit in one revolution where t_s is the time per scan. The scans are sampled such that the first scan is always exactly at the starting phase defined, so the last scan is generally not exactly at the defined ending orbit phase.

1.4 The swath

Each scanline observes a number of ground scenes at defined observation angles. One angle, β, is in the flight direction (positive is looking forward) and the other angle, α is perpendicular to the flight direction (positive is looking to the right). Under normal circumstances, the swath is approximately perpendicular to the flying direction, so α generally reaches much larger values than β.

To define α and β, two projections are made of the viewing line. One projection is on the plane constructed by the satellite, the sub-satellite point and the flying direciton. The other on the plane through the satellite perpendicular to the flying direction (down and to the side). The angles are defined as those between the actual viewing lines and the projections, not between the zenith (satellite to sub-satellite point) and the projections. This implies that the angles can never add up to more than 90°. Becuase the program allows any input of α and β, this law is checked in the code. See Fig. 1 for an illustration.

The swath is defined with the following additional input parameters.
- Number of detector rows (N_x).
- Polynomial parameters for α.
- Polynomial parameters for β.

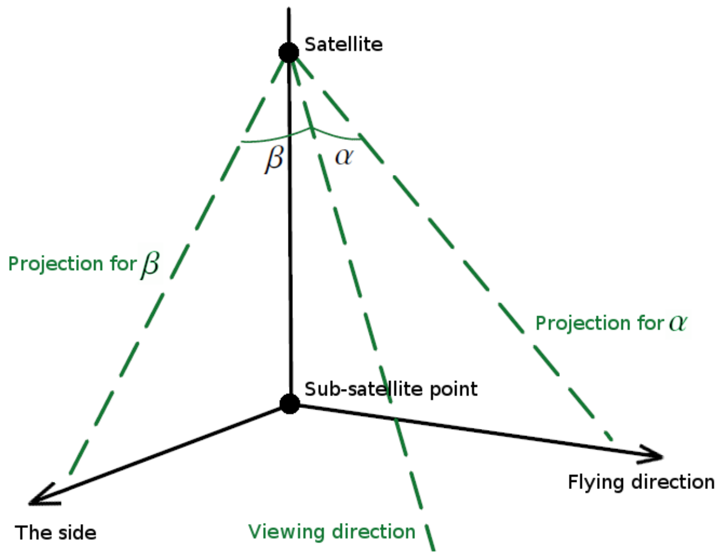

Figure 1: Illustration on how angles α and β are defined for a viewing direction.

The number of rows is the number of observations per scanline, each with a different set of observation angles α and β. Both angles are expressed as polynomial function of the row, where the row number is first converted to a number from -1 (first row) to $+1$ (last row).

$$q = \frac{2x}{N_x - 1} - 1 \tag{2}$$

Here, x is the zero-based row index ($x \in \{0, \ldots, N_x - 1\}$) and N_x is the number of rows. For only one row, we will take $q = 0$.

The polynomial parameters are a list of numbers. The first number is the factor for q^0, followed by q^1, q^2, ..., continuing until the input list is finished. This works the same for α and β. For a theoretical perfect swath, the polynomial values for β are (0) and for α are $(0, \alpha_{\max})$.

2 Output

The output is a MIPrep orbit object that can be written in a NetCDF using the write routine from the base orbit object. The swaths are concatenated to a one-dimensional obervation dimension. The row index is the quickly varying dimension and the scanline index is the slowly varying dimension. The orbit contains the following information.

- The UTC time in seven integers (year, month, day, hour, minute, second, millisecond). Note that pixels of the same swath (same scanline index) have the same time.
- The latitude and longitude of the pixel center.
- The latitude and the longitude of the pixel corners. The first corner is towards the previous row and the previous scanline. The second corner is towards the next row and the previous scanline. The third corner is towards the next row and the next scanline. The fourth corner is towards the previous row and the next scanline.
- The solar zenith and azimuth angles.
- The viewing zenith and azimuth angles. Because of the spherical assumption and the similar scan angles each scanline, the viewing zenith angles of all pixels with the same row index are the same.

All angles and geo-locations are expressed in degrees.

3 Implementation

This chapter contains all the mathematical derivations implemented in the algorithm.

All geometry is calculated in units of an Earth radius. This means that the satellite altitude must be converted.

$$h = \frac{z}{r_E} \tag{3}$$

with z measured from the Earth center.

3.1 Calculation of the sub-satellite points

One revolution is defined from the night-time equator pass to the next night-time equator pass. The middle of the revolution is the daytime equator pass. In the calculation, it is important to have a clearly defined point in the middle of the orbit. The user interface provides a local solar time for the northward equator pass, which may be the desired daytime point in the middle of the orbit, but may also be the night-time equator pass time (for a satellite that flies southwards during the day). To ensure a clearly defined daytime equator pass point while keeping the correct orbit, we redefine T_{eq} as the solar time of daytime equator pass, not necessarily northwards. For an orbit with northward equator pass in the day, nothing needs to be redefined. For a southward equator pass during the day, we shift T_{eq} twelve hours and take γ negative. In such a case, the daytime equator pass is always at angle γ with respect to the east at happens at T_{eq} local solar time.

The scanlines will be calculated using the orbit phase. The input phase is defined from 0 to 1, going from night-time equator pass to the next night-time equator pass. This phase will be converted to an angle phase defined from −180° to +180° (or $-\pi$ to $+\pi$).

$$\varphi = 2\pi\left(p_s + y\frac{t_s}{t_r}\right) - \pi \tag{4}$$

Here, y is the zero-based scanline index and φ is the angle phase as function of scanline index. The given ending phase determines the number of scanlines that fit with:

$$N_y = \left\lfloor (p_e - p_s)\frac{t_r}{t_s} \right\rfloor + 1 \tag{5}$$

The UTC time at equator pass is calculated by converting solar time to universal time using the equator pass longitude λ_{eq}.

$$t_{\mathrm{eq}} = T_{\mathrm{eq}} - \frac{\lambda_{\mathrm{eq}}}{2\pi}\mathrm{day} \tag{6}$$

The UTC scan time is calculated with the equator time and the angle phase.

$$t = t_{\mathrm{eq}} + \frac{\varphi}{2\pi}t_r \tag{7}$$

Next to calculate is the geo-locations of the sub-satellite points. This requires some trigonometry. The idea is to start at the equator point and rotate this point along the orbit with angle φ. This is done using quaternions [2]. We first have to define a right-handed cartesian coordinate system. This is defined as.

- Positive X: From the center to the equator point.
- Positive Y: To the east from equator point.
- Positive Z: From the center to the north pole.

Then, the original point is at $[1,0,0]$ and the rotation axis is $[0,-\sin\gamma,\cos\gamma]$, where γ is the inclination angle. Note that this angle can be negative because of the transformation performed for a southward daytime equator pass.

For the quaternion arithmetic, we define:

$$\begin{aligned} s &= \sin\frac{\varphi}{2} \\ c &= \cos\frac{\varphi}{2} \\ S &= \sin\gamma \\ C &= \cos\gamma \end{aligned} \tag{8}$$

And the arithmetic is as follows.

$$\begin{aligned} Q &= c - sSj + sCk \\ Q^\dagger &= c + sSj - sCk \\ P &= i \\ pQ^\dagger &= ci + sCj + sSk \\ QpQ^\dagger &= c^2 i + scCj + scSk + scSk + s^2SC - s^2S^2 i + scCj - s^2C^2 i - s^2SC \\ QpQ^\dagger &= \left(c^2 - s^2\right) i + 2scCj + 2scSk \end{aligned} \tag{9}$$

This is converted back to a point using

$$\begin{aligned} \cos\varphi &= c^2 - s^2 \\ \sin\varphi &= 2sc \end{aligned} \tag{10}$$

Thus, the point is

$$P_s = \begin{bmatrix} \cos\varphi \\ \sin\varphi\cos\gamma \\ \sin\varphi\sin\gamma \end{bmatrix} \tag{11}$$

The Z coordinate should be the sine of the latitude.

$$\phi_s = \arcsin\left(\sin\varphi\sin\gamma\right) \tag{12}$$

Here, the ϕ_s means latitude at sub-satellite point.

For the longitude, we can use the two-argument arctangent function with the X and the Y. And we have to correct for the longitude at the equator point and for the rotation of the Earth in the time that it takes for the satellite to go from the equator point to this point.

$$\lambda_s = \mathrm{Arctan}\left(\frac{\sin\varphi\cos\gamma}{\cos\varphi}\right) + \lambda_{\mathrm{eq}} - \varphi\frac{t_r}{\mathrm{day}} \tag{13}$$

Here, Arctan is the two-argument arctangent function that puts the result in the right quadrant. The two arguments are represented in one fraction, where the numerator is the first argument and the denominator is the second argument.

3.2 Calculation of the swath

To create a swath, the orientation of the satellite is needed. The swath is defined relative to the flying direction of the satellite. So, for each sub-satellite point, the angle of the flying direction with respect to the local northward direction is needed. Subsequently, the viewing angles from the satellite are used to acquire the viewed position on the Earth.

3.2.1 The flying direction

The flying direction is the derivative of the sub-satellite point with respect to the angle phase, without taking into account the rotation of the Earth. For simplicity, we also ignore λ_{eq} and return to the same coordinate system as in Sect. 3.1. We repeat the definition of the sub-satellite point from Eq. (11) and differentiate it with respect to the angle phase φ.

$$P_s = \begin{bmatrix} \cos\varphi \\ \sin\varphi\cos\gamma \\ \sin\varphi\sin\gamma \end{bmatrix}$$
$$\frac{dP_s}{d\varphi} = \begin{bmatrix} -\sin\varphi \\ \cos\varphi\cos\gamma \\ \cos\varphi\sin\gamma \end{bmatrix} \tag{14}$$

This is a normalized vector.

The local northward direction is obtained by defining the sub-satellite point as function of latutide and longitude (forcing it to be on the Earth surface).

$$P_s = \begin{bmatrix} \cos\phi_s\cos\lambda_s \\ \cos\phi_s\sin\lambda_s \\ \sin\phi_s \end{bmatrix} \tag{15}$$

and differentiate it with respect to the latitude with constant longitude.

$$\frac{dP_s}{d\phi_s} = \begin{bmatrix} -\sin\phi_s\cos\lambda_s \\ -\sin\phi_s\sin\lambda_s \\ \cos\phi_s \end{bmatrix} \tag{16}$$

To make progress, we will divide and multiply the X and the Y of this north direction with the X and the Y of point P_s, using the definition in Eq. 11 for the multiplication and the definition in Eq. 15 for the division.

$$\frac{dP_s}{d\phi_s} = \begin{bmatrix} -\sin\phi_s\cos\lambda_s\frac{\cos\varphi}{\cos\phi_s\cos\lambda_s} \\ -\sin\phi_s\sin\lambda_s\frac{\sin\varphi\cos\gamma}{\cos\phi_s\sin\lambda_s} \\ \cos\phi_s \end{bmatrix}$$
$$= \frac{1}{\cos\phi_s}\begin{bmatrix} -\sin\phi_s\cos\varphi \\ -\sin\phi_s\sin\varphi\cos\gamma \\ \cos^2\phi_s \end{bmatrix} \tag{17}$$

Now, we continue transforming back to the φ and γ perspective by using Eq. (12) to transform $\sin\phi_s$ to $\sin\varphi\sin\gamma$, and also transform the squared cosine of the latitude via $cos^2\phi_s = 1-\sin^2\phi_s$. We will only leave the prefactor.

$$\frac{dP_s}{d\phi_s} = \frac{1}{\cos\phi_s}\begin{bmatrix} -\sin\varphi\sin\gamma\cos\varphi \\ -\sin^2\varphi\sin\gamma\cos\gamma \\ 1-\sin^2\varphi\sin^2\gamma \end{bmatrix} \tag{18}$$

We will redefine the single-letter abbreviations with the sines and cosines of the phase angle and the inclination angle.

$$
\begin{aligned}
s &= \sin\varphi \\
c &= \cos\varphi \\
S &= \sin\gamma \\
C &= \cos\gamma \\
\frac{dP_s}{d\varphi} &= \begin{bmatrix} -s \\ cC \\ cS \end{bmatrix} \\
\frac{dP_s}{d\phi_s} = \frac{1}{\cos\phi_s} &\begin{bmatrix} -scS \\ -s^2SC \\ 1-s^2S^2 \end{bmatrix}
\end{aligned}
\tag{19}
$$

We wish to calculate the angle between the flying direction and the local north direction in the correct quadrant. This requires both the sine and the cosine with the right sign. In magnitude, the sine is equal to the cross product, because both vectors are normalized. The direction will determine the sign, which is determined by what direction is desired to be positive. We choose a positive angle when flying to the east. Then, $\frac{dP_s}{d\varphi} \times \frac{dP_s}{d\phi_s}$ is upward along the zenith (along P_s, which is $[c, sC, sS]$). Thus:

$$
\begin{aligned}
\sin\theta = P_s \cdot \left(\frac{dP_s}{d\varphi} \times \frac{dP_s}{d\phi_s}\right) = \frac{1}{\cos\phi_s} &\begin{vmatrix} c & sC & sS \\ -s & cC & cS \\ -scS & -s^2SC & 1-s^2S^2 \end{vmatrix} \\
= \frac{c^2C - s^2c^2S^2C + s^2c^2S^2C - s^2c^2S^2C + s^2C - s^4S^2C + s^4S^2C + s^2c^2S^2C}{\cos\phi_s} & \\
&= \frac{C}{\cos\phi_s}
\end{aligned}
\tag{20}
$$

The cosine is just the dot product, automatically resulting in the correct sign.

$$
\cos\theta = \frac{dP_s}{d\varphi} \cdot \frac{dP_s}{d\phi_s} = \frac{1}{\cos\phi_s}\left(s^2cS - s^2cSC^2 + cS - s^2cS^3\right) = \frac{cS}{\cos\phi_s} \tag{21}
$$

With the sine and the cosine, the angle is known in the correct quadrant using the two-argument arctangent. We can divide out the cosine of the latitude, because that is a positive number.

$$
\theta = \mathrm{Arctan}\left(\frac{\cos\gamma}{\cos\varphi\sin\gamma}\right) \tag{22}
$$

This angle is zero when flying to the north, and becomes positive when going north-east.

3.2.2 The viewed locations

For each detector row, a recipe is given to get the viewing angles at the satellite for the directions perpendicular (α) and along (β) the flying direction. Note that these angles are defined as the angles between the actual viewing direction and the viewing direction projected in one direction, so α and β cannot add up to 90° or more. If such angles are given for a detector row, the row is considered not to see anything and will be skipped in the simulation.

First, we calculate the distances between some points in Fig. 1. This figure is repeated here in Fig. 2, with points added after following the following (3D) draw protocool.

1. Draw the satellite (S).
2. Draw the sub-satellite on the Earth surface (O).

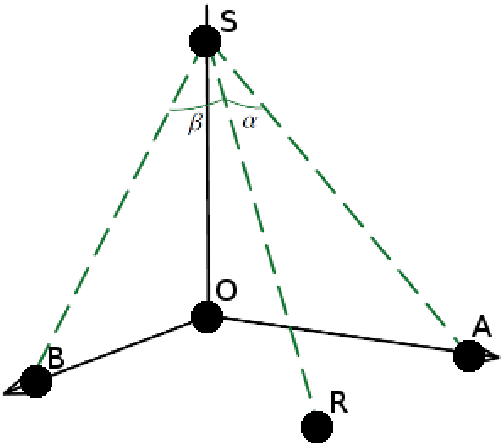

Figure 2: 3D drawing with points marked to calculate some distances using the input angles.

3. Draw the plane on O perpendicular to SO.
4. Draw a line from S at some 3D angle down until it hits the plane at representation point (R).
5. Draw a point (A) on the plane so that OA is parallel to the flying direction.
6. Draw a point (B) on the plane so that OB is perpendicular to the flying direction.

In the generic case, points A and B are not equal to R. That is the case if neither α nor β is zero. We will refresch the definitions of α and β.

- Angle α is the angle between SR and SA.
- Angle β is the angle between SR and SB.

Now, we want to calculate the distance OR, using Fig. 2.

$$\begin{aligned}
\sin\alpha &= \frac{RA}{SR} \\
\sin\beta &= \frac{RB}{SR} \\
OR = \sqrt{RA^2 + RB^2} &= SR\sqrt{\sin^2\alpha + \sin^2\beta} \\
SO = h - 1 = \sqrt{SR^2 - OR^2} &= SR\sqrt{1 - \sin^2\alpha - \sin^2\beta} \\
SR &= \frac{h-1}{\sqrt{1-\sin^2\alpha-\sin^2\beta}} \\
RA &= \frac{(h-1)\sin\alpha}{\sqrt{1-\sin^2\alpha-\sin^2\beta}} \\
RB &= \frac{(h-1)\sin\beta}{\sqrt{1-\sin^2\alpha-\sin^2\beta}} \\
OR &= \frac{(h-1)\sqrt{\sin^2\alpha+\sin^2\beta}}{\sqrt{1-\sin^2\alpha-\sin^2\beta}}
\end{aligned} \tag{23}$$

The root is negative if the angles add up to more than 90°, but that cannot be the case.

Now, we calculate the viewing point in the 2D plane SOR. For this, we make a new (2D) drawing (see Fig. 3).

1. Draw the satellite (S).
2. Draw the center of the Earth (C).
3. Draw the sub-satellite on the Earth surface (O).

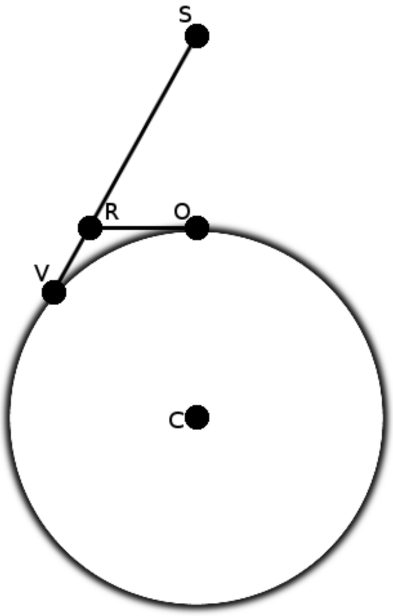

Figure 3: 2D draw protocol to clarify Eq. (24)

4. Draw a line perpendicular to SO to representation point (R).
5. Draw the Earth, a circle with center C through O.
6. Continue line SR until it hits the Earth surface at viewing point (V).

If line SR does not hit the Earth surface, the viewing direction misses the Earth and the detector row will be skipped. Whether this happens depends on the distance OR and the satellite altitude SO, which is $h-1$. The limit case is if angle CVS in 90°. In such a case:

$$\begin{aligned} \cos SCV &= \frac{1}{h} \\ \sin SCV &= \sqrt{1-\frac{1}{h^2}} \\ \tan SCV &= \sqrt{h^2-1} \\ ORS &= SCV \\ OR = \frac{SO}{\tan ORS} &= \frac{h-1}{\sqrt{h^2-1}} \\ OR &= \sqrt{\frac{h-1}{h+1}} \end{aligned} \tag{24}$$

If this OR distance is larger than this limit, it misses the Earth.

Asserting that angle CVS is obtuse, continue the draw protocol to get Fig. 4.

1. Draw lines CV and CS (the latter crosses O).
2. Draw a line from V perpendicular to CS (parallel to OR) until it crosses CS at (E).
3. Draw a line from V perpendicular to SV until it crosses CS at point (X).

With OR and SC known, all angles and lengths are fixed. These are resolved by calculating SX twice on a dif-

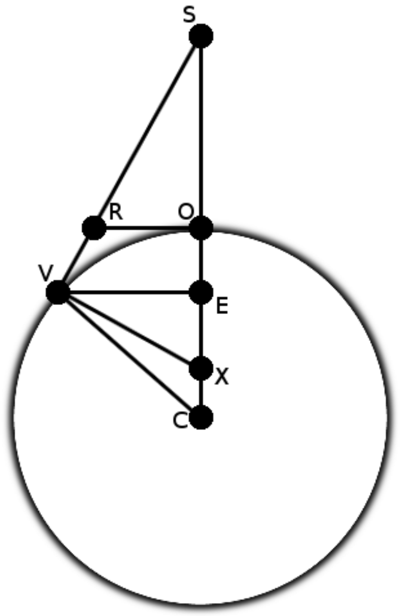

Figure 4: Continued 2D draw protocol to clarify equations from (26) onwards.

ferent way and equating their two definitions.

$$\begin{aligned}
\tan OSR &= \frac{OR}{h-1} \\
\sin OSR &= \frac{OR}{\sqrt{OR^2 + (h-1)^2}} \\
\cos OSR &= \frac{h-1}{\sqrt{OR^2 + (h-1)^2}} \\
\sin SCV &= EV \\
EVX = ESV &= OSR \\
SX = SE + EX &= EV\left(\tan OSR + \frac{1}{\tan OSR}\right) \\
SX &= \sin SCV\left(\frac{OR}{h-1} + \frac{h-1}{OR}\right)
\end{aligned} \tag{25}$$

The second way of calculating SX, we calculate CX first.

$$
\begin{aligned}
\frac{\sin CVS}{\sin CSV} &= h \\
\sin CSV &= \sin OSR \\
\sin CVS = h \sin OSR &= \frac{hOR}{\sqrt{OR^2 + (h-1)^2}} \\
\cos XVC = \sin CVS &= \frac{hOR}{\sqrt{OR^2 + (h-1)^2}} \\
\sin XVC &= \sqrt{1 - \cos^2 XVC} \\
\sin XVC &= \sqrt{1 - \frac{h^2 OR^2}{OR^2 + (h-1)^2}} \\
\sin XVC &= \sqrt{\frac{OR^2 + (h-1)^2 - h^2 OR^2}{OR^2 + (h-1)^2}} \\
\sin VXC = \cos OSR &= \frac{h-1}{\sqrt{OR^2 + (h-1)^2}} \\
CX = \frac{\sin XVC}{\sin VXC} &= \frac{\sqrt{OR^2 + (h-1)^2 - h^2 OR^2}}{h-1} \\
CX &= \sqrt{\frac{OR^2 + (h-1)^2 - h^2 OR^2}{(h-1)^2}} \\
CX &= \sqrt{1 - \frac{\left(h^2 - 1\right) OR^2}{(h-1)^2}} \\
CX &= \sqrt{1 - \frac{(h+1)\, OR^2}{h-1}}
\end{aligned}
\tag{26}
$$

Now, we combine the two definitions of SX, one using EV from the beginning and the one using CX at the end.

$$
\begin{aligned}
SX &= h - CX \\
\sin SCV \left(\frac{OR}{h-1} + \frac{h-1}{OR} \right) &= h - \sqrt{1 - \frac{(h+1)\, OR^2}{h-1}} \\
\sin SCV &= \frac{h - \sqrt{1 - \frac{(h+1)OR^2}{h-1}}}{\frac{OR}{h-1} + \frac{h-1}{OR}} \\
SCV &= \arcsin \left(\frac{h - \sqrt{1 - \frac{(h+1)OR^2}{h-1}}}{\frac{OR}{h-1} + \frac{h-1}{OR}} \right)
\end{aligned}
\tag{27}
$$

To continue calculating the geo-location of V, we need to return to the 3D representation. We have the distances OA (same as RB) and OB (same as RA). We now define x as the local eastward direction on the plane through O, A, B and R and y as the local northward direction in that plane. For this, we use the angle of the flying direction with the local north that we already have.

$$
\begin{aligned}
x &= OB \cos\theta + OA \sin\theta \\
y &= OA \cos\theta - OB \sin\theta
\end{aligned}
\tag{28}
$$

We will perform a quaternion rotation from point O to V using angle SCV. We will simplify the sub-satellite point to put it on zero longitude. We will define the uppercase and lowercase sines and cosines again.

$$\begin{aligned} S &= \sin\phi_s \\ C &= \cos\phi_s \\ s &= \sin\frac{SCV}{2} \\ c &= \cos\frac{SCV}{2} \end{aligned} \tag{29}$$

The rotation axis is $[-S, 0, C]$ for rotation in the x-direction and $[0, -1, 0]$ for rotation in the y-direction. Furthermore, we will define:

$$\begin{aligned} X &= \frac{x}{OR} \\ Y &= \frac{y}{OR} \end{aligned} \tag{30}$$

So, $X^2 + Y^2 = 1$. We add up the x and y contributions of the rotation to the quaternion using contribution factors X and Y. The magnitude of the rotation vector remains correct, because the x and y rotation vectors are perpendicular.

$$\begin{aligned} Q &= c - XsSi - Ysj + XsCk \\ Q^\dagger &= c + XsSi + Ysj - XsCk \\ P &= Ci + Sk \\ PQ^\dagger &= -XsSC + XsSC + cCi - YsSi + XsC^2j + XsS^2j + cSk + YsCk \\ PQ^\dagger &= cCi - YsSi + Xsj + cSk + YsCk \\ QPQ^\dagger &= XscSC - XYs^2S^2 + XYs^2 - XscSC - XYs^2C^2 + \\ &\left(c^2C - YscS - YscS - Y^2s^2C - X^2s^2C\right)i + \\ &\left(Xsc + XscC^2 - XYs^2SC + XscS^2 + XYs^2SC\right)j + \\ &\left(c^2S + YscC - X^2s^2S + YscC - Y^2s^2S\right)k \\ QPQ^\dagger &= \left(\left(c^2 - s^2\right)C - 2scYS\right)i + 2scXj + \left(\left(c^2 - s^2\right)S + 2scYC\right)k \end{aligned} \tag{31}$$

Now, substitute $\cos SCV = c^2 - s^2$ and $\sin SCV = 2sc$ and substitute the uppercase sines and consines back to sines and cosines of the latitude.

$$V = \begin{bmatrix} \cos SCV \cos\phi_s - Y \sin SCV \sin\phi_s \\ X \sin SCV \\ \cos SCV \sin\phi_s + Y \sin SCV \cos\phi_s \end{bmatrix} \tag{32}$$

The latitude and the longitude is acquired by the arcsine and the two-argument arctangent using the coordinates. And we give back the original longitude.

$$\begin{aligned} \phi &= \arcsin\left(\cos SCV \sin\phi_s + Y \sin SCV \cos\phi_s\right) \\ \lambda &= \lambda_s + \operatorname{Arctan}\left(\frac{X \sin SCV}{\cos SCV \cos\phi_s - Y \sin SCV \sin\phi_s}\right) \end{aligned} \tag{33}$$

3.2.3 Corners

Each pixel is surrounded by four corners, so that the entire orbit is a continuous mesh of quadrilaterals. This implies that orthogonally adjacent pixels share two corners and diagonally adjacent pixels share one corner.

These corners are calculated by calculating the viewed locations that correspond to the half-integer row and scanline indices using the same transformation formulas, Eqs. (2) and (4) using the indices:

$$\begin{aligned} x &\in \{-\frac{1}{2}, \frac{1}{2}, \ldots, N_x - \frac{1}{2}\} \\ y &\in \{-\frac{1}{2}, \frac{1}{2}, \ldots, N_y - \frac{1}{2}\} \end{aligned} \tag{34}$$

For each pixels, the corners correspoding to $x \pm \frac{1}{2}$ and $y \pm \frac{1}{2}$ are assigned. This automatically ensures that the desired corners are shared with the adjacent pixels and that the mesh is continuous.

3.3 Viewing geometry

The viewing zenith and azimuth angles can be calculated as well. The viewing zenith angle only depends on α, β and h and will therefore be the same for all scanlines. We almost already calculated the viewing zenith angle while calculating angle SCV.

$$\sin CVS = h \sin OSR = \frac{hOR}{\sqrt{OR^2 + (h-1)^2}} \tag{35}$$

Angle CVS is obtuse but has the same sine as the viewing zenith angle. Thus,

$$\theta_v = \arcsin\left(\frac{hOR}{\sqrt{OR^2 + (h-1)^2}}\right) \tag{36}$$

The azimuth depends on the geo-location, because that is defined with respect to the local north. We have to switch from the perspective of the sub-satellite point to the perspective of the viewed location, because the azimuth is defined with respect to the local north of the viewed location. Therefore, we cannot recycle the x and y used to get the latitude and longitude.

We put the longitude of the viewed position at zero. That leaves three angles, so uppercase and lowercase letters for the sines and cosines are not enough. Therefore, we define.

$$\begin{aligned} S_s &= \sin \phi_s \\ C_s &= \cos \phi_s \\ S_v &= \sin \phi \\ C_v &= \cos \phi \\ S_l &= \sin(\lambda_s - \lambda) \\ C_l &= \cos(\lambda_s - \lambda) \end{aligned} \tag{37}$$

The viewed location is at $[C_v, 0, S_v]$ and the sub-satellite point is at $[C_sC_l, C_sS_l, S_s]$. We subtract the viewed position from the sub-satellite point and project that on the local north (N) and on the local east (E) from the viewed position.

$$\begin{aligned} O - V = O = \begin{bmatrix} C_sC_l \\ C_sS_l \\ S_s \end{bmatrix} - \begin{bmatrix} C_v \\ 0 \\ S_v \end{bmatrix} &= \begin{bmatrix} C_sC_l - C_v \\ C_sS_l \\ S_s - S_v \end{bmatrix} \\ N &= \begin{bmatrix} -S_v \\ 0 \\ C_v \end{bmatrix} \\ E &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ (O-V) \cdot N = -C_sC_lS_v + S_vC_v + S_sC_v - S_vC_v &= S_sC_v - C_sC_lS_v \\ (O-V) \cdot E &= C_sS_l \end{aligned} \tag{38}$$

Thus, using the two-argument arctangent:

$$\varphi_v = \mathrm{Arctan}\left(\frac{\cos\phi_s \sin(\lambda_s - \lambda)}{\sin\phi_s \cos\phi - \cos\phi_s \sin\phi \cos(\lambda_s - \lambda)}\right) \tag{39}$$

And this is conform the conventions that zero azimuth is the satellite at the north and 90° azimuth is the satellite in the east.

3.4 The solar geometry

The model for the solar geometry is quite complicated, as it includes long-term effects such as the movement of the tropics of Cancer and Capricorn. With the approximations of a spherical Earth and a circular orbit, such complicated solar goemetry model seems overdone. The issue is that this solar model already existed [3] before the orbit simulator was constructed. So, it was chosen to adopt the entire model.

These formulas contain a lot of constant numbers, which are rotation phases, rotation periods and decay rates of several properties of the Earth's rotation around the sun.

For the date (y, m, d), the Julian day is calculated using a trick to cope with the unequal definitions of the months.

$$\begin{aligned} x &= \begin{matrix} 1 & \text{if} \quad m \leq 2 \\ 0 & \text{otherwise} \end{matrix} \\ J &= 1721089 + d + \left\lfloor -\frac{3}{4}\left\lfloor \frac{y-x}{100} \right\rfloor \right\rfloor + \left\lfloor 365\frac{1}{4}(y-x) \right\rfloor + \left\lfloor 367\left(x + \frac{(m-2)}{12}\right) \right\rfloor + \frac{t}{\mathrm{day}} - \frac{1}{2} \end{aligned} \tag{40}$$

The last two terms in J are the only non-integer parts and is the contribution of the UTC scan time of the relevant pixel. An integer J corresponds to UTC noon, declaring the $\frac{1}{2}$ at the end. Note that d represents the day part of the date and the text 'day' is the duration of one day to scale the scan time to a contribution in day units.

This Julian day is converted into a time in centuries using a convention for the time when it is zero. This time in century units is used to calculate long-term evolutions of properties of the Earth's orbit around the sun.

$$T = \frac{J - 2451545}{36525} \tag{41}$$

And the long-term properties that evolve are the latitudes of the tropics, also referred to as the mean obliquity of the ecliptic (ϵ), the Sun's mean anomaly (M) and the Sun's equation of center (C). Furhermore, the seasonal cycle (L) is slightly perturbed by long-term effects. These are all angles expressed in arcseconds or degrees.

$$\begin{aligned} \epsilon &= \left(84381.448 - 46.815T - 0.00059T^2 + 0.001813T^3\right) \cdot 1'' \\ L &= \left(280.46645 + 36000.7693T + 0.0003032T^2\right) \cdot 1^\circ \\ M &= \left(357.5291 + 35999.0503T - 0.0001559T^2 - 4.8 \cdot 10^{-7}T^3\right) \cdot 1^\circ \\ C &= (\left(1.9146 - 0.004817T - 0.000014T^2\right) \sin M + \\ &\quad (0.019993 - 0.000101T) \sin 2M + \\ &\quad 0.00029 \sin 3M) \cdot 1^\circ \end{aligned} \tag{42}$$

The Sun's true longitude is defined as a phase angle that is zero at the equinox in March.

$$\Theta = L + C \tag{43}$$

Now, we can compare the sun to our satellite, though we have to imagine the Sun orbiting the Earth instead of the Earth orbiting the Sun. The latitudes of the tropics of Cancer and Capricorn is equal to the inclination angle, and the true longitude Θ is the angle phase. As zero phase angle is the equinox in March, the sun is moving to the north, so we have to take the inclination angle positive. The normalized Sun's position is calculated by substituting γ and φ with ϵ and Θ in Eq. (11).

$$S = \begin{bmatrix} \cos\Theta \\ \sin\Theta\cos\epsilon \\ \sin\Theta\sin\epsilon \end{bmatrix} \tag{44}$$

Likewise, this can be transformed into a latitude and a longitude. The latitude is the Sun declination.

$$\delta = \arcsin(\sin\Theta\sin\epsilon) \tag{45}$$

And the longitude is similar, but that is compared to the March equinox point.

$$\alpha = \mathrm{Arctan}\left(\frac{\sin\Theta\cos\epsilon}{\cos\Theta}\right) \tag{46}$$

This longitude is converted to a longitude relative to the viewed location using the mean sidereal time at Greenwich (θ_m, an angle). This definition includes the diurnal cycle.

$$\begin{aligned} \theta_m &= \left(280.46061837 + 360.98564736629(J - 2451545.0) + 0.000387933T^2 - \frac{1}{38710000}T^3\right)\cdot 1^\circ \\ H &= \alpha - \theta_m - \lambda \end{aligned} \tag{47}$$

Now the sun has a latitude (δ) and a longitude (H) in the reference frame that the viewed location is at zero longitude, so the geometry is solved.

$$\begin{aligned} P_S &= \begin{bmatrix} \cos\delta\cos H \\ \cos\delta\sin H \\ \sin\delta \end{bmatrix} \\ P_V &= \begin{bmatrix} \cos\phi \\ 0 \\ \sin\phi \end{bmatrix} \\ \theta_s &= \arccos(P_S\cdot P_V) = \arccos(\cos\delta\cos H\cos\phi + \sin\delta\sin\phi) \end{aligned} \tag{48}$$

For the azimuth, we use the same formula, Eq. (39) as for the viewing azimuth angle, using the local north and east of the viewed location, substituting the $(\lambda_s - \lambda)$ with H, and ϕ_s with δ

$$\varphi_s = \mathrm{Arctan}\left(\frac{\cos\delta\sin H}{\sin\delta\cos\phi - \cos\delta\sin\phi\cos H}\right) \tag{49}$$