Development of the navigational system for a multi-spectral imaging nano-satellite

Ayush Mehta
Team Anant
Birla Institute of Technology and Science Pilani
ayushmehta2001108@gmail.com

Amay Sareen
Team Anant
Birla Institute of Technology and Science Pilani
amaysareen@gmail.com

Aaryav Mishra
Team Anant
Birla Institute of Technology and Science Pilani
mannaaryav@gmail.com

Abstract—**The navigational system of the satellite is required for determination of the position and velocity of a satellite. This is required for obtaining outputs from ephemeris models such as IGRF, WMM and Earth reflectivity for albedo. Given the high pointing requirements for an imaging payload, there exists a need for highly accurate position and velocity information as it affords greater room for inaccuracies to the entire attitude determination, estimation and control system. The navigation system comprises 2 major segments - orbit determination and orbit propagation.**

The orbit determination segment is responsible for calculating the current position and velocity vectors which is to be carried out using GNSS/GPS receivers. The data from the GNSS receivers provides measurements with errors within acceptable limits given the constraints by our payload. The data obtained from orbit determination is used for initialisation in orbit propagation.

The in-house orbit propagator has the responsibility of providing an estimated satellite position and velocity at any future point in time along with the reference frame vectors such as the magnetic field vector and Sun vector, which would be utilised in the attitude estimation assembly and power production. This information would in turn be useful for switching between the satellite's various modes of operation. The orbit propagator will be implemented in C due to faster run time and ease of implementation on microcontrollers. The accuracy of this system is verified against data from proven softwares such as NASA's General Mission Analysis Tool (GMAT).

We will also be calculating the magnetic field vector field and sun vector using our current position and power generated using the sun vector and the solar panel specifications.

There also arises a need to create a system of repeated propagation, given the increase in error as the satellite state is propagated further into the future. From our pointing requirements (obtained from payload) and the pointing error, we can calculate the tolerable error in position. Using this tolerable error and the modes of operations, we will then develop an algorithm which will optimise the usage of GPS receivers, as they consume significant energy. The exact specifications for such a system are also being investigated.

The Orbit propagator incorporates J2 perturbations, solar radiation pressure and atmospheric drag. Although various other perturbing forces act on the satellite as well such as gravitational effects from other celestial bodies, higher order perturbations from the Earth's gravitational field among others, the effect of these forces is considered negligible compared to the forces which are included. Various numerical schemes for propagation of satellite state were also investigated in order to find an optimal balance between accuracy and computational requirements.

TABLE OF CONTENTS

1. INTRODUCTION

Throughout the duration of any satellite mission, it is of vital importance to be able to determine the position and velocity of the satellite to within an acceptable margin of error, since this information is a necessity for various other computations such as the attitude determination and control algorithms as well as outputs from ephemeris models like IGRF, WMM and Earth reflectivity for albedo. They may also require knowledge of the position of the satellite relative to the sun. Thus the need for an accurate and validated satellite navigation for orbit-determination as well as a propagation system is evident.

The navigation system design has been made with the multispectral terrain imaging payload in mind. Considering that such payloads have high overall pointing requirements, the demand for accurate satellite position and velocity information is further highlighted. An increase in accuracy of a navigation system would allow greater tolerance of error from other, less accurate sources of information.

An error in position leads to attitude error through a direct pathway as well as an indirect pathway. The direct effect is observed as a pointing inaccuracy. A nadir pointing maneuver would be the clearest example of this as any error in position would directly correspond to an increase in the off-nadir tilt in the pointing direction. This leads to inaccurate pointing during the different modes of operation of our satellite, potentially causing issues such as inefficient down-linking and improper image processing.

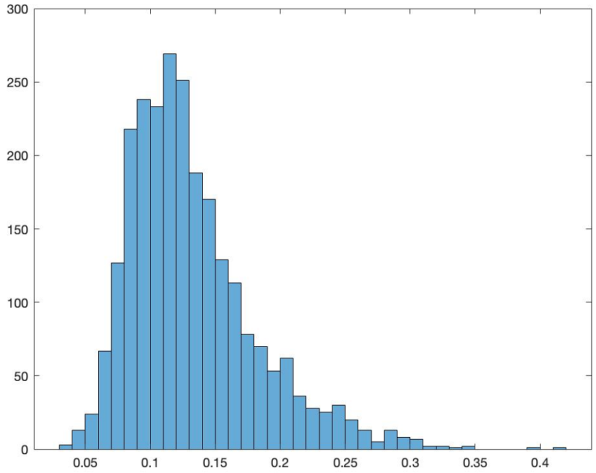


Figure 1: **Histogram of the error Euler angle between the true quaternion and the Kalman Filter output**;
with no position error

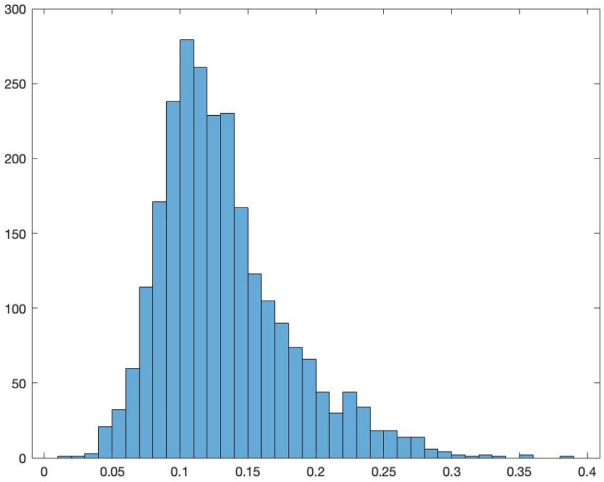


Figure 2: **Histogram of the error Euler angle between the true quaternion and the Kalman Filter output**;
with position error

The satellite's ADCS requires data on the earth's magnetic field vector with respect to the satellite, which is used both for creating a reference frame for attitude determination and estimation and for attitude control with the help of magnetorquers. Similarly, it also requires data on the sun vector and the orientation of the solar panels with respect to the sun vector to account for SRP as well as to enter the sun-pointing mode. An error in position would lead to an increase in the error present in each of these models. This constitutes the indirect influence of position error on attitude error.

There exist several methods for navigation in practice. These methods can broadly classify a satellite to be autonomous, semi-autonomous or non-autonomous.

Autonomous method of satellite navigation are still very much under development, being less accurate than other methods [17]. It however promises independence from any other artificial object for the creation of the satellite state vector[18]; making it suitable for satellites of strategic importance, since they should ideally continue functioning normally even in case of unforeseen failure of other satellites and ground stations. Since the mission is a multi-spectral imaging satellite of no strategic importance and the imaging payload requires higher accuracy for its pointing requirements, such a method is unsuitable given the present context.

Non-autonomous navigation systems on the other hand are completely dependent on one or more ground stations for the generation of the state vector, making them extremely inefficient.

A semi-autonomous navigation method constitutes the on-board processing of data from either a ground station (using TLE) or from satellite constellations (GNSS). This method not only allows for high precision measurements from systems that are utilised by several other users at any given time, but also provides a degree of autonomy in the absence of newer data, since the state vector can be propagated on board the satellite. Therefore, given the good balance of autonomy and accuracy that it offers, the navigation system we use is a semi-autonomous navigation system.

Our semi-autonomous system design calls for the use of antennas and/or a GNSS receiver on board the satellite. The two methods of semi-autonomous satellite navigation that were considered were through using a Two Line element (TLE)[21] or through a Global navigation satellite system (GNSS) receiver [22]. A TLE comprises of data sets published by NORAD (North American Aerospace Defence Command) that lists the orbital elements of objects in space above a certain size specification. This method uses publicly available propagation models (SGP4, SGP8) and has no on-board hardware requirements besides antenna for up linking the TLEs from ground station. This significantly reduces the navigation systems power requirements. However, it has drawbacks including a low update rate of at most 2 times a day for unclassified LEO satellites, no publicly available information by NORAD regarding error characteristics and a low accuracy of 1km at the time of measurement with long term propagation of TLEs, which leads to severe position error. This error necessitates the requirement of a computationally intensive Extended Kalman filter at all times for error reduction. Additionally, implementation of any such Kalman Filter would be hindered by the lack of any official information that may have otherwise been used to generate an error covariance matrix for the filter. These issues lead us to reject the TLEs and opt for a GNSS receiver instead. GNSS follows the concept of trilateration in 3D space. GNSS satellite constellations send signals radially to detect and locate receivers, when a minimum of 4 GNSS satellite signals are received, a position and time value output can be formulated by the receiver. An increase in the number of signals directly correlates to a more accurate position. Since in general more than 4 such GNSS satellites are present, this method provides us with a much higher degree of accuracy ($\sim$10m) and a high update rate of the order of 10 Hz [12], hence it doesn't necessitate the requirement of a Kalman filter for orbit determination. It does however, require a receiver and antenna on board and further imposes higher weight, power and space requirements on the satellite. As per the current design, these requirements can be are considered adequately feasible for our CubeSat.

While selecting a GNSS receiver and antenna, the following factors should be considered in order to ensure that the hardware can be accommodated within the limited nano-satellite constraints and still satisfy payload requirements: accuracy, average power consumption, dimensions, weight, communication interfaces provided (UART, I2C, CAN, etc.), and cost effectiveness. It is generally advisable to test the GNSS receiver's engineering model before moving on to buy its actual flight model given the potentially high cost of such components.

2. Force Model

The orbit propagator forms an essential part of the navigation system. It is required for estimation of the satellite position and velocity at some future time instance by propagating an initial update from the GPS (on current position and velocity of the satellite) until the desired time instance.

To carry out this propagation, it is important to know and account for the factors that affect our satellite orbit. For the acceleration due to these forces are used to derive the velocity and position of the satellite at a given instance from initial conditions, thus helping us gauge the satellite position and velocity in the orbit before the next update from the GPS.

The following forces constitute the Force model of the satellite:

Point Mass Gravitational Force

The most elementary force of our force is the two-body gravitational force between the earth and the cube-Sat. The equation for this force is given by:

$$\mathbf{F} = \frac{GMm}{r^2} \tag{1}$$

$$GM = \mu \tag{2}$$

$$\mu = 3.986 \times 10^{14} m^3/s^2$$

Where the above equation represents the Standard gravitational parameter of Earth and m is the mass of our cube-sat. Hence the two-body force is given by:

$$\mathbf{F} = \frac{\mu m}{r^2} \tag{3}$$

J2 Perturbations

While the standard first degree gravitational model works for spherical entities, it is important to note that the earth is not spherical, rather it is an oblate. J2 perturbations are the second degree orbital forces caused by the oblate nature of earth. The gravitational force, by taking into account these perturbations, is to be included in the force model.

The J2 perturbations cause a change in the longitude of ascending node (Ω) of the satellite orbit. (The phenomenon is called nodal regression) This change is, in Sun-synchronous orbits, a desirable one, as it helps maintain a constant Beta angle (Sun-orbital plane angle). In an SSO, by the time the earth completes one revolution around the sun, the satellite orbit would also have completed its one revolution.

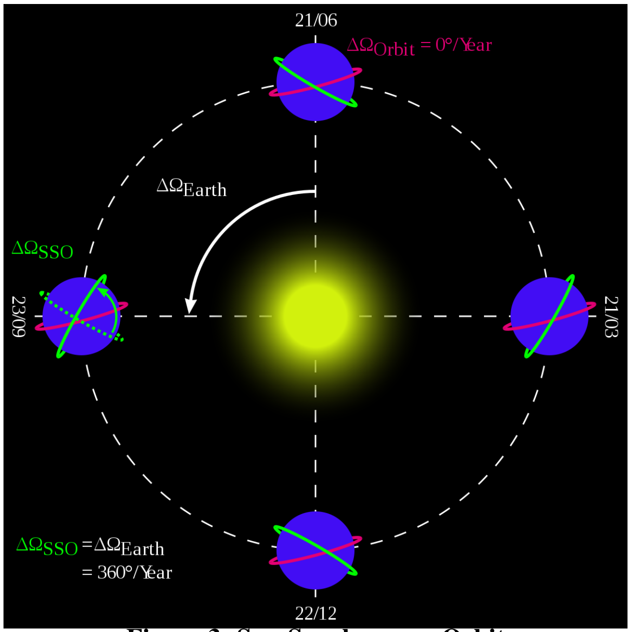


Figure 3: **Sun Synchronous Orbit**

For the derivation of the gravitational acceleration while involving the zonal harmonic J_2, we start with the simple representation of Gravitational potential for a point mass dm. [23]

$$U = -Gdm/\rho \tag{4}$$

Where all the terms have their usual meanings. Now, taking an arbitrary point P(r, ϕ, $lambda$) where r is the radial distance of the point, ϕ the geocentric lattitude and λ is the longitude; we integrate the above equation over the entire volume of the sphere, while including the zonal harmonics.[3]

$$U(r,\phi) = \frac{-\mu}{r}[1 + \sum_{k=2}^{\infty} J_k(\frac{R_e}{r})^k P_k cos(\phi)] \tag{5}$$

J_2 is the most dominant across all zonal harmonics. It has the value:

$$J_2 = 1.082629 \times 10^{-3}$$

Hence, we limit the equation to k=2. Here, P_k is the Legender polynomial function.

$$P_2(cos(\phi)) = \frac{2 - 3sin^2(\phi)}{2} \tag{6}$$

$$U(r,\phi) = \frac{-\mu J_2 R_e^2}{2r^3}(3sin^2(\phi) - 1) \tag{7}$$

In terms of Cartesian coordinates, $cos(\phi) = \frac{z}{r}$, so $sin^2(\phi) = 1 - \frac{z^2}{r^2}$ for which, we get:

$$U_{J_2} = \frac{\mu J_2 R_e^2}{2r^3}(1 - \frac{3z^2}{r^2}) \tag{8}$$

The required acceleration can be derived from the above potential using

$$a_{J_2} = -[\frac{\partial U_{J_2}}{\partial \hat{x}} + \frac{\partial U_{J_2}}{\partial \hat{y}} + \frac{\partial U_{J_2}}{\partial \hat{z}}] \tag{9}$$

Solving the above equation, we finally get the gravitational acceleration taking into account J_2 to be: [1]

$$a_x = \frac{\mu J_2 R_e^2}{2}(15\frac{xz^2}{r^7} - 3\frac{x}{r^5}) \tag{10}$$

$$a_y = \frac{\mu J_2 R_e^2}{2}(15\frac{yz^2}{r^7} - 3\frac{y}{r^5}) \quad (11)$$

$$a_z = \frac{\mu J_2 R_e^2}{2}(15\frac{z^3}{r^7} - 9\frac{z}{r^5}) \quad (12)$$

Here μ is the standard gravitational parameter of Earth and R_e is the average radius of the Earth. This Gravity model is JGM2.

The effect of J_2 as seen in Keplerian elements can be derived using the same above equations but for spherical coordinates instead (by considering the geocentric latitude and longitude and radius instead of Cartesian coordinates; [2],from which we get:

$$\dot{a} = \dot{i} = \dot{e} = 0 \quad (13)$$

$$\dot{\Omega} = -\frac{3}{2}nJ_2(\frac{R_e}{P})^2(5cos^2 i - 1) \quad (14)$$

$$\dot{\omega} = \frac{3}{4}nJ_2(\frac{R_e}{P})^2(5cos^2 i - 1) \quad (15)$$

$$\dot{M} = n[1 + \frac{3}{4}\sqrt{1-e^2}J_2(\frac{R_e}{P})^2(3cos^2 i - 1)] \quad (16)$$

Here the terms have their usual meanings, representing the 6 Keplerian elements.

While propagating in Keplerian elements may seem more convenient, considering the data derived from the GPS in this form, the gravity model is applied in the ECI (Earth Centred Inertial) frame. The reason for which is simply the fact that the ECI frame can be considered inertial. The use of an inertial frame is important to carry out integration in a fixed frame, such that we can propagate the orbit of our Cube-Sat before the next GPS update in an unchanging frame. Thus, the use of cartesian coordinates in the ECI frame proves to be much more convenient than the use of Keplerian parameters for orbit propagation. However, while the ECI frame may be effectively inertial, it is not a completely inertial frame; It is prone to a change in orientation over a longer course of time. A brief account of this phenomenon and the discussion of the different reference frames would be carried out in the next section.

Aerodynamic Drag Force

The aerodynamic drag force is the drag force caused due to the Earth's atmosphere. There are 3 types of satellite orbits. Namely Low Earth Orbits (LEO; below 2000km altitude), Middle Earth Orbits (MEO; 2000 to 35000km altitude), Geosynchronous orbits (35786 km) and finally High Earth Orbits (HEO; greater than 35786 km). Since the Earth's atmosphere stretches up to as far 10,000 km, the aerodynamic drag force cannot be neglected in LEO.

The drag force acting on our satellite is given by:

$$F_D = \frac{\rho v^2 C_D A}{2} \quad (17)$$

F_D = drag force
ρ = atmospheric density.
v = satellite velocity
A = The projected area along velocity vector. This projected area is the resultant of the area vectors of the 3 exposed sides in the direction of the velocity vector.
C_D is the drag coefficient, and its value can be experimentally determined in a wind tunnel [25]. However, since our orbit propagator's accuracy is to be compared with the outputs from GMAT. We take C_D to be equal to 2.2; the default value in GMAT's drag model.

The atmospheric density, ρ can be calculated using several methods, such as Jachia 1971, CIRA 1972, MSIS, etc.[4] But the method which GMAT uses is the Jachia Roberts method; a method which involves the use of exo-space temperature to calculate atmospheric density. This exo-space temperature further depends on the sun vector direction and albedo.

This method is primarily based on the analytical solution of the barometric and diffusion differential equations which are obtained by the integration of partial differential equation. However, this method has a high computational power requirement. Therefore, to fit our computational power budget, a much simpler model can be used. Which is,

$$\rho = \rho_o e^{(-\frac{h-h_o}{H})} \quad (18)$$

The ρ_o, h and h_o values can be taken from the appendix. It is important to note that this model assumes the density of the earth to be spherically symmetric in 3D space, this makes it inaccurate as compared to the other methods available. But it can still be adopted due to its simplicity and much lower computational requirement.

Solar Radiation Pressure (SRP)

The force caused due to solar radiation is based on the concept of momentum exchange between photons in light emitted by the sun and the satellite body.

This force, however, is not very prominent in LEO satellites. It is dominated by the drag force. While drag acts on the LEO satellite throughout its orbit, SRP only acts when the satellite is exposed to the sun and is not in the shadow of the earth or another celestial body. The general equation for this force is given by:

$$F_{SRP} = -P_\odot C_{SRP} e_{sat\odot} \quad (19)$$

$$P_\odot = SolarRadiationPressure = \frac{S_\odot}{cr^2_{sat\odot}} \quad (20)$$

$S_\odot$= Solar constant= 1362 Wm^-2
c = Speed of light = 299792458 m/s
r = Distance from the sun
C_{SRP}= Solar Drag coefficient.
S = Sun facing area.
$e_{sat\odot}$ = unit vector directed from the satellite to the center of the sun.

The value of $e_{sat\odot}$ can be calculated using fundamental linear algebra; we already know the vector from the Earth's centre to our Cube-Sat and the Sun-to-Earth vector, which is computed from the date. The resultant of these 2 vectors is all that's to be calculated.

To calculate the e_{sat} We first calculate the Julian centuries elapsed from epoch. [7]

$$T = \frac{JulianDate - 2451545}{36525} \quad (21)$$

Then we calculate the unit vector in the Earth-to-Sun direction:

$$e_{ES} = [cos\phi_{ecliptic} \quad cos\epsilon sin\phi_{ecliptic} \quad sin\epsilon sin\phi_{ecliptic}] \quad (22)$$

Where

$$\phi = 280.460 + 36000.771.T$$

is the mean longitude of the sun,

$$M = 357.5277233 + 35999.05034.T$$

is the mean anomaly of the sun,

$$\phi_{ecliptic} = \phi + 1.914666471.sinM + 0.019994643.sin2M$$

is the Longitude of the ecliptic, and

$$\epsilon = 23.439291 - 0.0130042.T$$

is Obliquity of the ecliptic.

The resultant of this unit vector and the unit vector from the earth's center to Cube-sat is then found out.

The solar constant is the flux density of energy radiated by sun at a distance of 1AU. It keeps fluctuating based on a 11 year cycle, these fluctuations are hard to predict.

The Solar drag coefficient (C_{SRP}) denotes how transparent or reflective the surface is. If C_{SRP} = 1 it is purely transparent and if C_{SRP}= 2, the surface is purely reflective. For satellites the C_{SRP} is generally taken to be 1.8.

While this model is a low fidelity and basic model, with comparatively lower accuracy, it is however practical enough to be used for a LEO satellite.

Since the aim is to strive for higher accuracy models in the navigation system, a higher fidelity model can be developed to account for SRP. [24]

$$F_{SRP} = -P_{\odot} \sum_{i=1}^{N} S_i cos\theta^i_{SRP}[2(\frac{R^i_{diff}}{3} + R^i_{spec} cos\theta^i_{SRP})n^i_I + (1 - R^i_{spec})e_{sat\odot}] \quad (23)$$

It can be observed that the summation is only carried out for plates with $cos\theta^i_{SRP} > 0$. This is an indicator that the force due to SRP is not acting continuously on the plates of the Cube-sat, while in orbit. The variables in the above equation are: θ^i_{SRP}= inclination of the i^{th} plate to the spacecraft-to-sun vector.
S_i= Area of the plate.
R^i_{diff}= Diffuse Reflectivity
R^i_{spec}= Specular Reflectivity
n^i_I= $A^T n^i_B$ where A is the attitude matrix (a 3x3 matrix representing the attitude of the Cube-Sat at a given instance) of the satellite and n^i_B is the outward unit normal vector of the plate in the satellite body frame (A 3-axes non-inertial frame with its origin on the satellite body; as shown in figure)

The Specular reflectivity is the ratio of incident power to the normal reflected power and the Diffuse reflectivity is the ratio of the incident power to the scattered reflected power. Values for the same are experimentally determined through a simple experiment.

Relative Specular Reflectance can be measured using the following setup:

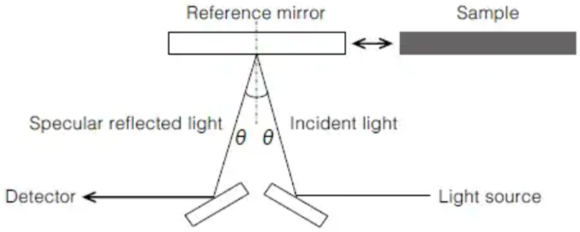


Figure 4: **Specular Reflectance Setup**

First we shine light on our sample, then we measure the specular light reflected at the same angle as the angle of incidence to get the relative specular reflectance measurement. (Using the Specular Reflectance measurement device).
The relative Specular reflectance is given by:

$$RelativeReflectance(R) = \frac{R_s}{R_r} \times 100\% \quad (24)$$

Here R_s is the Amount of light reflected from the sample and R_r is the Amount of light Reflected from the reference plate. Barium Sulphate or a mirror can be taken as the reference plate.

Now, **Relative Diffuse Reflectance** can be measured using the following setup:

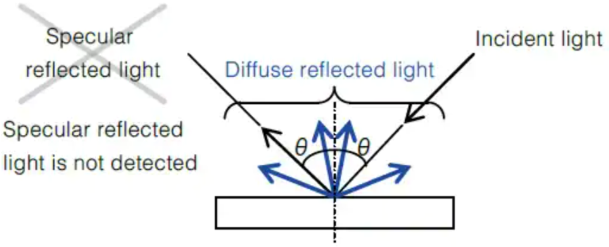


Figure 5: **Diffuse Refleactance Setup**

It is measured by using an integrating sphere. a device which can measure the diffuse reflected light excluding the Specular reflected light in the process. A material such as Barium Sulphate or specialised fluorine based polymer is used as the reference plate.

After having measured the Specular and Diffuse reflectance for the materials on board, the higher fidelity SRP model can be integrated into our Cube-Sat.

3. Numerical Methods

Orbit propagation is done through integrating the acceleration from our SGP model, taking the initial velocity and positions, over the required period of time. To carry the integration of this IVP (Initial Value Problem) out, there are several methods that can be used on board.

While choosing the right method the error of the output (when compared with more complex methods provided for orbit propagation on GMAT), and the complexity of the method are to be taken into account. An account on the methods that we tested and the results we got out of it are mentioned in the subsequent sub-sections.

Euler Explicit

The Euler explicit method is a primitive method used to solve Initial Value Problems (IVPs), which cannot be solved using the more traditional methods for solving differential equations. (The methods used to solve exact, linear or separable differential equations)

It is a first-order numerical procedure and is the most basic method for the integration of ODEs. It is also the simplest Runga-Kutta method.

$$
\begin{aligned}
y'(t) &= f(t, y(t)) && (25)\\
y(t_o) &= y_o && (26)\\
y_{n+1} &= y_n + hf(t_n, y_n) && (27)
\end{aligned}
$$

This method is the least robust of all the other methods used to solve ODEs, it is only first-order convergent. Therefore, the error in the computed solution is unacceptably poor and requires a very small step size to achieve a decent level of accuracy. Hence, we ruled this method out.

Adams Methods

There are 2 kinds of Adams methods, the Adam's Bashforth methods, which are explicit methods to solve IVPs and the Adam's Moulton methods, which are of the implicit type.

These are adaptive methods which are of (k+1)th order for a k degree polynomial The Adam's Bashforth equation is of the form:

$$y_1 \quad = \quad y_0 + h(q_0 + \frac{1}{2} \bigtriangledown q_0 + \frac{5}{12} \bigtriangledown^2 q_0 + ...) \quad (28)$$

Where h is the stepsize and $\bigtriangledown q_0 = q_0 - q_{-1}$;
$q_o, q_{-1}, ...$ being the derivatives computed at the points $t_0, t_{-1}, ...$ The solution at each step given by $y_0, y_1, ...$ (Reference[i])

Similarly, the Adams-Moulton equation is given by:

$$y_0 \quad = \quad y_{-1} + h(q_0 - \frac{1}{2} \bigtriangledown q_0 - \frac{1}{12} \bigtriangledown^2 q_0 + ...) \quad (29)$$

This is a comparatively complex numerical method which we implemented on GMAT, to compare with the outputs we got from the other, more simpler methods. Mainly the RK-4 method, which we finally decided to implement on board.

The Runge-Kutta Integrators

The Runge-Kutta scheme is a single step method used to solve differential for n coupled variables.

$$\frac{dr^i}{dt} \quad = \quad f(t, r) \quad (30)$$

Here f^i is the i^{th} value, and n is the total number of variables. The time increment of a given stage can be given as a multiple of a_i of the total time step desired. Thus for the i^{th} stage the interval used for calculation is $a_i\delta$. The integral at that stage is give by:

$$k_i^{(n)} \quad = \quad \delta t f(t + a_i\delta t, r^{(n)}(t) + \sum_{j=1}^{i-1} b_{ij} k_j^{(n)}) \quad (31)$$

Where b_{ij} is set of coefficients specific to the Runge-Kutta instance being calculated. It is different for differnt orders of the integrator. The total integration step is calculated using another constant c_j;

$$r^{(n)}(t + \delta t) = r^{(n)}(t) + \sum_{j=1}^{stages} c_j k_j^{(n)} \quad (32)$$

These are the equations that are used for non-adaptive Runge-Kutta methods. Such as the RK-4 method we finally decided on using. The RK-4 is a 4th order ODE solver. We use the RK-4 integrator in 4 stages, where $a_i = 0, \frac{1}{2}, \frac{1}{2}, 1$;
$b_{ij} = 0, \frac{h}{2}, \frac{h}{2}, h$ and $c_j = \frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}$.

The adaptive methods of GMAT however use 2 different orders of integration. The stage estimates of the equations in this other order of the integrator can be denoted by k_j^*. The 2 stage estimates cana be selected such that they are equal. Hence the error in the integration can be represented as a factor of $(c_j - c_j^*)$ and k_j. Once the estimated error is calculated, the step size of integration can be adapted such that we get the desired accuracy of integration.

Labeling the desired accuracy as α and the obtained accuracy as ϵ. The new step size is therefore given as:

$$\delta t_{new} \quad = \quad \sigma \delta t(\frac{\alpha}{\epsilon})^{1/m} \quad (33)$$

Here m is the order of truncation of the series, and σ is the safety factor, to avoid unnecessary iterations. It is usually taken to be 0.9. If the new δ value still doesn't give the desired accuracy, we increase the m value by one.
This procedure may be repeated until the best δ value is found.

The RK-89 method is an adaptive method as described above, using the 8th and 9th order integrators to find the most accurate time step.

Due to this we take the propagated coordinates using RK-89 to be the closest to the true value, hence using it as the reference for error calculation.[20]
Doing this, we noticed that the error by using RK-4 was within the acceptable range. Therefore, due to the comparatively lower complexity and acceptable errors, we use RK-4 as our integrator.

4. Frames of reference

There are several reference frames that can be used in practice. However, the ones that are majorly used are the Keplerian elements, ECI, ECEF [19], and body frames respectively. The inter conversion between these frames is a continuous process throughout the mission.

We receive our position data from the GNSS in the form of LLA (Latitude, Longitude, Altitude), which is converted to the ECEF frame and subsequently to the ECI 2020 and J2000 frames.[14] The conversion to the body frame is done from the ECI 2020 frame with the help of an attitude matrix; the discussion of which is not in the scope of this text, for it forms the basis of attitude control.

LLA coordinates

The LLA or the Latitude, Longitude and Altitude are the coordinates our GNSS returns the position and velocity of our satellite in. The LLA can be taken in 2 different reference frames, namely the geocentric and geodetic reference frames. Our GNSS returns data in geodetic LLA, from which it is converted to the ECEF coordinates.

In geodetic coordinates, the latitude (λ) is the angle between the equatorial plane and the normal to the reference ellipsoid. [9] The flattening of the aforementioned reference ellipsoid is given by

$$f = \frac{R_e - R_{pole}}{R_e} \tag{34}$$

Where R_e is the equatorial radius of the earth and R_{pole} is the distance of the center of the earth to a pole. The eccentricity of this ellipsoid is given by:

$$e = \sqrt{f(2-f)} \tag{35}$$

While there may exist many reference ellipsoid models in practice, $f \approx 1/298.257$ is a valid approximation, considering the fact that in all models the difference between the equatorial and polar radii is less than 22 km.

The eccentricity of this ellipsoid is hence given by e = 0.0818. With the semi-major and semi-minor axes being the equatorial and polar radius respectively.
The geodetic longitude is the angle in the equatorial plane between the line that connects the Earth's center with the prime meridian and another line that connects the center with the meridian on which the point lies.

A meridian is a direct path on the surface of the datum that is the shortest distance between the poles. [10]

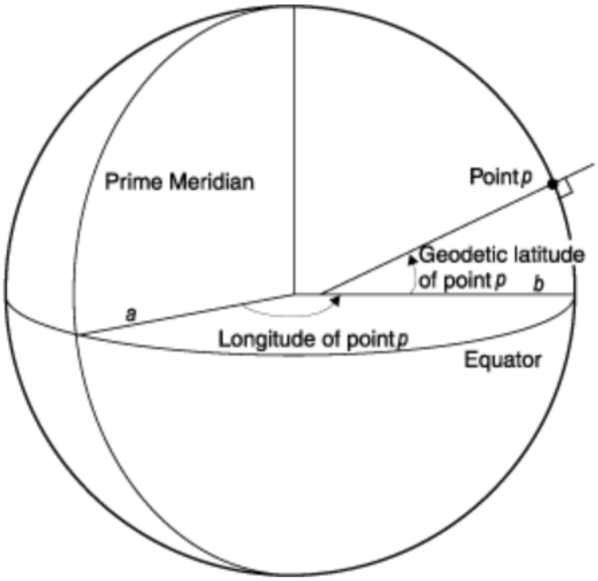


Figure 6: **LLA Reference Frame**

ECEF frame

The ECEF or the Earth Centered Earth Fixed frame is a non-inertial coordinate frame fixed to and rotating with the Earth. The equatorial plane is its fundamental frame (XY frame) The 3 axes that form this frame are the X axis, which is in the direction of a vector from the center of the earth towards the point of intersection of the equator and the prime meridian, the Z axis, which is in the direction of a vector from the Earth's center to the geographical north pole and the Y axis, which is in the direction of the cross product of the other 2 axes.

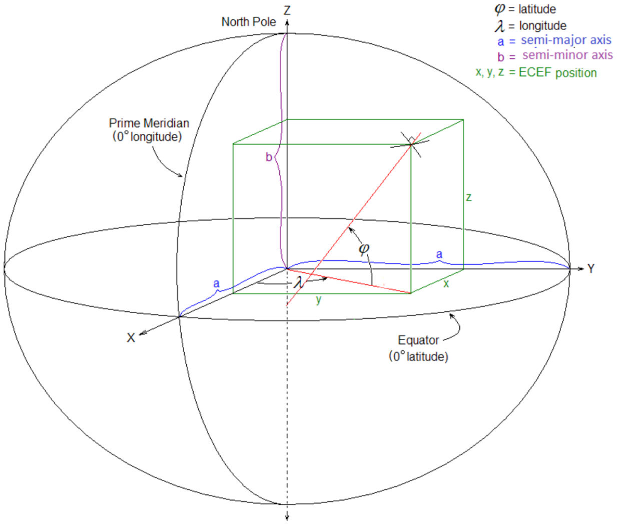


Figure 7: **ECEF Frame**

To convert the LLA coordinates into ECEF, we require the geodetic latitude (λ), longitude (ϕ) and altitude (h). Firstly, we compute the distance between the z axis and the normal to the ellipsoid. [9]

$$N = \frac{R_e q}{\sqrt{1 - e^2 sin^2(\lambda}} \tag{36}$$

The we proceed with the conversion to ECEF using:

$$x = (N + h)cos(\lambda)cos(\phi) \tag{37}$$

$$y = (N + h)cos(\lambda)sin(\phi) \tag{38}$$

$$z = [N(1 - e^2) + h]sin(\lambda) \tag{39}$$

ECI frame

The ECI frame or the Earth Centered Inertial frame is a frame that can be considered inertial within the time frame of the mission. The frame does undergo a change in orientation but that change is brought about over the course of decades (caused due to earth's axial precession, nutation and polar motion).

This coordinate frame is made up of a Z-axis pointing towards the pole, an X-axis pointing towards the ares star, or the vernal equinox and a Y-axis in the direction of the cross product of the X and Z axes.

This frame differs from the ECEF frame by just one rotation of the the XY plane by an angle of θ_{GMST}, where GMST stands for Greenwich Mean Sidereal Time.[8]
This transformation follows the simple principle of $\theta = \omega t$, since the ECEF frame rotates with the Earth, with the same angular velocity as that of the earth, while the ECI frame remains inertial.

To calculate the θ_{GST}, we first need to calculate the Julian date, which is given by:

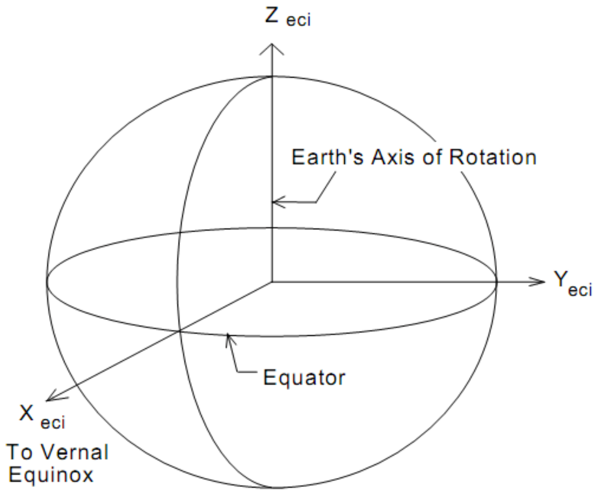


Figure 8: **ECI Frame**

$$JD(Y,M,D,h,m,s) = 1,721,013.5 + 367Y - INT(\frac{7}{4}(Y + INT(\frac{M+9}{12})) + INT(\frac{275M}{9}) + D + \frac{60h+m+s/60*}{1440} \quad (40)$$

Where Y, M, D denote year, month and day respectively. h, m and s denote the hour, minute and second. INT denotes the integer part and 60* translates to 61s for days with a leap second.

For the conversion to GMST, we use T_o, which is the number of Julian centuries elapsed from the epoch J2000 to 00:00 of the date in question. [7]

$$T_o = \frac{JD(Y,M,D,0,0,0) - 2,451,545}{36,525} \quad (41)$$

$$\theta_{GMST} = 24,110.54841 + 8,640,184.812866T_o + 0.093104T_o^2 - 6.2 \times 10^{-6}T_o^3 + 1.002737909350795 \times (3600h + 60m + s) \quad (42)$$

This θ is obtained in degrees by dividing it with 240° (1s = 1/240°) ([11]) Finally, the conversion of the ECEF coordinates to ECI is done using [16]:

$$\begin{bmatrix} X_{ECI} \\ Y_{ECI} \\ Z_{ECI} \end{bmatrix} = \begin{bmatrix} X_{ECEF} \\ Y_{ECEF} \\ Z_{ECEF} \end{bmatrix} \begin{bmatrix} cos(\theta_{GMST}) & -sin(\theta_{GMST}) & 0 \\ sin(\theta_{GMST}) & cos(\theta_{GMST}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (43)$$

J2000 Frame

The J2000 frame is the ECI frame defined by the Earth's mean equator and the direction of the equinox, with respect to the Earth's center at 12:00 Terrestrial time on 1st January 2000. [6] It is the latest global standard reference frame, the predecessor to which is the M50 frame; similarly defined, but for 1st January 1950 instead. GMAT propagates the orbit in this frame, so the comparison of our orbit propagator data from MATLAB with GMAT requires us to convert our output coordinates from ECI2020 to J2000 frame.

In the previous sections it was mentioned that the ECI frame is not inertial but can be assumed to be inertial over a comparatively short period of time. Due to the non-inertial nature of the ECI frame over the course of decades, the J2000 differs from ECI 2020.

Factors affecting the orientation of the ECI frame

The factors which effect the ECI frame causing it to change over a period of time are: Axial precession, nutation and polar motion.

Axial Precession is a gravity-induced, slow, and continuous change in the orientation of an Earth's rotational axis. It is mainly caused by the uneven gravitational forces of Sun and the Moon due to Earth's oblate nature. It causes gradual shift in the orientation of Earth's axis of rotation in a cycle of approximately 26,000 years, Regardless which it is the motion that affects the ECI frame the most.

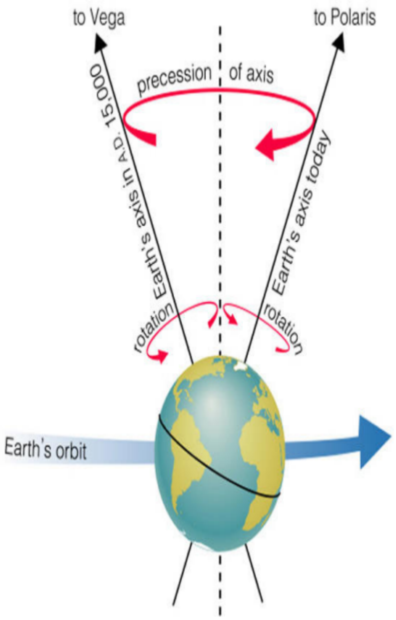


Figure 9: **Axial Precession**

Nutation is caused principally due to the gravitational forces of the moon and sun acting on the non-spherical figure of the earth; gravitational forces which vary with the planetary motion. This variation results in wobbling of the Earth, caused due to a change in inclination of the Earth's rotational axis. [13]

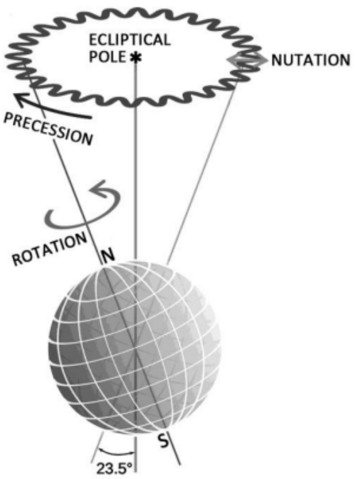


Figure 10: **Earth Nutation**

Polar motion of the Earth is the motion of the Earth's rotational axis relative to its crust. It is caused partly due to motions in the Earth's core and mantle, and partly to the redistribution of water mass as the Greenland ice sheet melts, and to isostatic rebound, which is the slow rise of land that was formerly burdened with ice sheets or glaciers.

While there are force models to account for the above 2 factors of precession and nutation; equations that help inter convert between an ECI frame and J2000, there is no such model to account for polar motion yet. However, the direction of polar motion prior to and after 2000 can be observed and a transformation matrix can be created with the difference, to minimise the error along the Z axis with respect to the GMAT output.

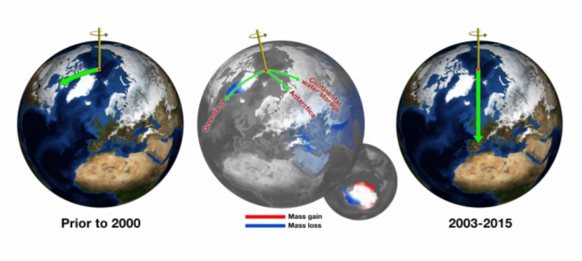


Figure 11: **Polar Motion**

5. Validation

To validate our force model and the numerical method used on board, we compare our output from the Orbit Propagator written in C and MATLAB, with the output from the Orbit Propagator of GMAT.

In this process we consider the GMAT outputs to be the most accurate and closest to the true value. Considering the fact that it uses more force models which are also more complex than the ones we are using for our OP.

The Orbit propagator of GMAT propagates in the J2000 frame. However, since our coordinates are propagates in the ECI2020 frame, we are required to transform our OP output to J2000 frame before comparing it with GMAT.

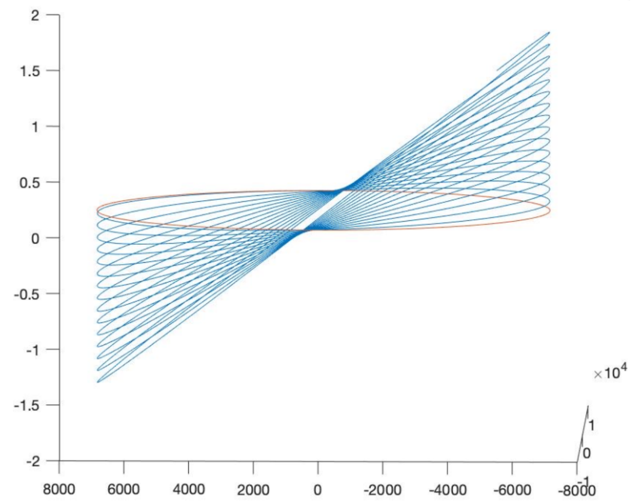


Figure 12: **GMAT Orbit (Blue) vs. MATLAB orbit (Orange) neglecting frame conversion to J2000.** (All the axes are in meters and the orbit which is being propagated is equatorial)

Coordinate transformation to J2000

The factors that cause a change in orientation in the ECI frames are Precession, Nutation and Polar motion respectively as discussed in the previous sections. This shift in the coordinate frame can be accounted for to carry out frame transformations using few equations.

Initially we had compared the coordinates from our OP for an equatorial orbit without converting the frame to J2000; this resulted in an error on the z axis, which was an unexpected outcome for a completely equatorial orbit, as seen in the figure.

Here the Blue lines depict the Orbit propagated by GMAT

Precession - Precession consists of 3 rotations through 3 angles:
Here, X1, Y1 and Z1 are the ECI coordinates of date.
- A positive Rotation about the Z1-axis, through an angle (90 - ζ)
-A positive rotation bout the Xi-axis through the angle (θ)
-A negative rotation about the Z2-axis through the angle (90+z)
The effective precession matrix formed by the product of the bove 3 rotations is given by equation (44)

$$D = \begin{bmatrix} A & B & -cosz.sin(\theta) \\ C & D & -sinz.sin(\theta) \\ sin(\theta).cos(\zeta) & -sin(\theta)sin(\zeta) & cos(\theta) \end{bmatrix} \tag{44}$$

Where,

$$A = cosz.cos(\theta)cos(\zeta) - sinz.sin(\zeta) \tag{45}$$
$$B = cosz.cos(\theta)sin(\zeta) - sinz.cos(\zeta) \tag{46}$$
$$C = sinz.cos(\theta)cos(\zeta) - cosz.sin(\zeta) \tag{47}$$
$$D = -sinz.cos(\theta)sin(\zeta) - cosz.cos(\zeta) \tag{48}$$

$$\zeta = 2306.2181T + 0.30188T^2 + 0.017988T^3 \tag{49}$$

$$\theta = 2004.3109T + 0.42665T^2 + 0.041833T^3 \tag{50}$$

$$z = 2306.2181T + 1.09468T^2 + 0.018203T^3 \tag{51}$$

T is the Julian centuries elapsed from epoch, that is 1st January 2000

$$T = \frac{JulianDate - 2451545}{36525} \tag{52}$$

Nutation - Nutation can be expressed as a sum of 3 rotations, which are as follows:
- A positive rotation along X2 axis, through the angle $\bar{\epsilon}$
- A negative rotation about Z2 axis, through the angle of $\triangle\psi$
- A negative rotation about the X3 axis through the angle of ϵ
The effective nutation matrix is given as a product of the above 3 rotations.

$$C = \begin{bmatrix} cos \triangle \psi & -sin \triangle \psi.cos\bar{\epsilon} & -sin \triangle \psi.sin\bar{\epsilon} \\ cos\epsilon sin \triangle \psi & A & B \\ sin\epsilon sin \triangle \psi & C & D \end{bmatrix} \tag{53}$$

Where,

$$\begin{aligned} A &= cos\epsilon cos \triangle \psi cos\bar{\epsilon} + sin\epsilon sin\bar{\epsilon} && (54) \\ B &= cos\epsilon cos \triangle \psi sin\bar{\epsilon} - sin\epsilon cos\bar{\epsilon} && (55) \\ C &= sin\epsilon cos \triangle \psi cos\bar{\epsilon} - cos\epsilon sin\bar{\epsilon} && (56) \\ D &= sin\epsilon cos \triangle \psi sin\bar{\epsilon} - cos\epsilon cos\bar{\epsilon} && (57) \end{aligned}$$

Here T is the Julian centuries from epoch. [7]

$\bar{\epsilon}$ = Mean Obliquity of the ecliptic = $\epsilon_o - 46.8150T - 0.00059T^2 + 0.001813T^3$ arc seconds

$$\triangle\psi = \sum_{i=1}^{1} 06 \triangle \psi_i \tag{58}$$

Which is the Nutation in longitude, where;

$\triangle\psi_i = (A_i + B_i.T)sin(a_{1i}.l + a_{2i}.l^{'} + a_{3i}.F + a_{4i}.D + a_{5i}.\Omega)$

l = Mean Anomaly of Moon = $485866.733 + (1325^{\gamma} + 715922.633)T + 31.310T^2 + 0.064T^3$ arc seconds

l' = Mean Anomaly of Sun = $1287099.804 + (99^{\gamma} + 1292581.244)T - 0.577T^2 - 0.012T^3$ arc seconds

F = Mean Longitude of Moon - Ω = $335778.877 + (1342^{\gamma} + 295263.137)T - 13.257T^2 + 0.011T^3$ arc seconds

D = Mean Elongation of Moon from Sun = $1072261.307 + (1236^{\gamma} + 1105601.328)T - 6.891T^2 + 0.019T^3$ arc seconds

D = Mean Elongation of Moon from Sun = $1072261.307 + (1236^{\gamma} + 1105601.328)T - 6.891T^2 + 0.019T^3$ arc seconds

Ω = Longitude of Ascending Node of Lunar Mean Orbit on Ecliptic Measured From Mean Equinox of Date = $450160.280 - (5^{\gamma} + 482890.539)T + 7.455T^2 + 0.008T^3$ arc seconds
$1^{\gamma} = 129600$ arc second

$$\triangle\epsilon = \sum_{i=1}^{1} 06 \triangle \epsilon_i \tag{59}$$

Which is the Nutation in Obliquity

Where $\triangle\epsilon_i = (C_i + D_i.T)cos(a_{1i}.l + a_{2i}.l^{'} + a_{3i}.F + a_{4i}.D + a_{5i}.\Omega)$
and
$\epsilon = \bar{\epsilon} + \triangle\epsilon$

Which is the true obliquity of Ecliptic.

The values of A_i, B_i, C_i, $a_{1i}, a_{2i}, a_{3i}, a_{4i}$ and a_{5i} can be obtained from the appendix.

Force Model validation

Having converted our output coordinates from the ECI 2020 frame to J2000 frame, we can now compare the implemented J2, SRP and atmospheric drag with the GMAT output.

We do this by individually taking the difference between the output for each individual force in our code and its corresponding GMAT output; subsequently plotting the error curve.

Since our 2 body gravitational force model had no errors when compared to GMAT, we first plotted the error plot solely taking into account J2 perturbations for an equatorial orbit. The max errors were seen to be around 14m for the x and y axes, and 33m for the z axis. These errors are very small and hence can be neglected.

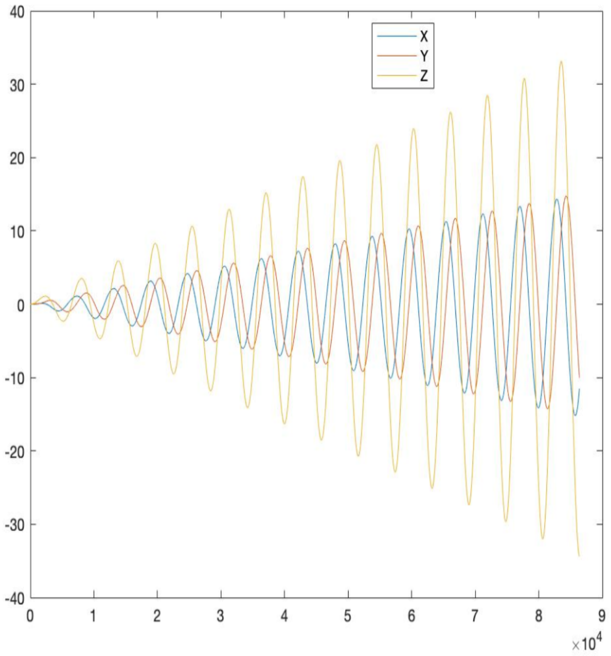


Figure 13: **Deviation in OP and GMAT(meters) vs time(sec)** (SSO orbit and have accounted for J2 Perturbation only)

Next we plotted the error plot for the Aerodynamic drag model. It was seen that the error in our simplistic exponentially decaying spherical atmospheric model was very less, when compared to the more complex Jachia-Roberts model used by GMAT; an aerodynamic drag model varying with temperature, and implemented using nested RK-4. Hence, leading us to rule this method out.

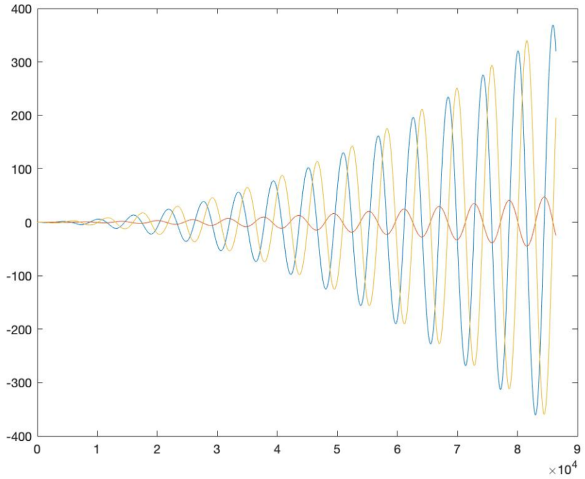


Figure 14: **Deviation in OP and GMAT(meters) and time(sec)** Forces considered: J2, Aerodrag for an SSO orbit

Next, we implemented only the SRP model, and plotted the error.

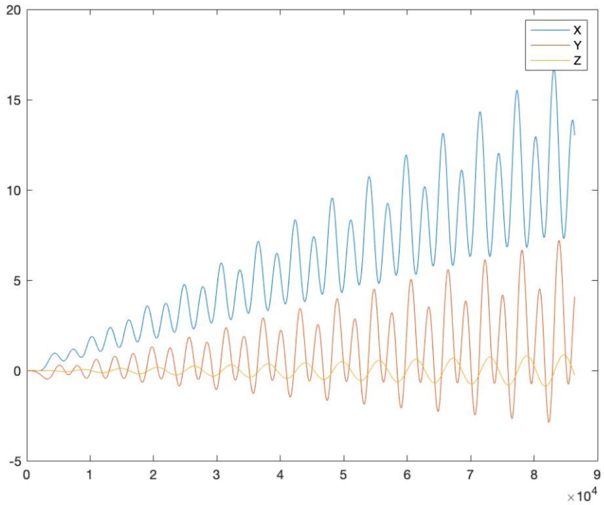


Figure 15: **Deviation in OP and GMAT(meters) and time(sec)** Forces considered: point mass 2 body and SRP for an SSO orbit

As the errors seemed to be in acceptable range for the individual forces, we moved on to comparing the entirety of our Force model in C and MATLAB with the GMAT model.

In our Force model we have taken 4 forces into consideration; The 2 body point force, Gravitational force due to J2, Aerodynamic drag force and Solar Radiation Pressure. Error is first calculated by comparing our code which accounts for these above 4 forces and The GMAT model for the same; taking a time step of 0.01 sec.

Since our orbit propagator would be implemented in the form of a C code on board, the initial MATLAB code is converted to C and is then compared with the GMAT code.

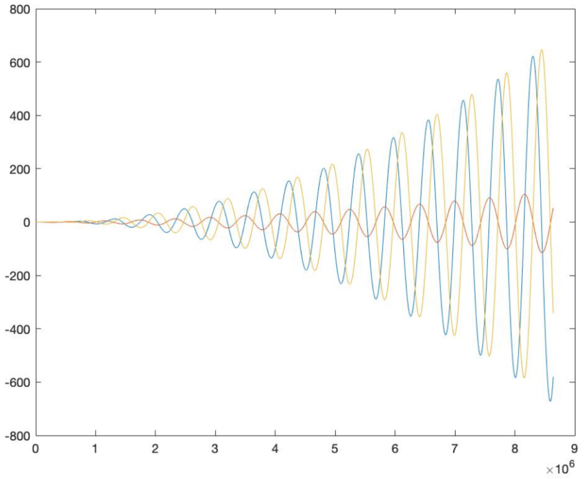


Figure 16: **Deviation in OP and GMAT(meters) and time(0.01 x sec)** Forces considered: J2, Aerodrag, SRP for an SSO orbit

It can be seen from the plot that the error attained the maximum value of 600m in the span of 1 day. Our payload can tolerate an error of around 0.3 degrees, which can roughly be translated to 3 km of position error, assuming no attitude errors outside of the direct error due to position shift. Considering the high update frequency of the GNSS, the error we got is not close to the maximum tolerable error and is hence, in acceptable range.

Though these were the results plotted by comparing the 4 forces we are considering on board, GMAT's force model accounts for more forces and is also more complex. It has numerous force models such as point mass and spherical harmonic gravity models, atmospheric drag, solar radiation pressure, tide models, and relativistic corrections.[5] Due to this, the complete GMAT force model can be considered to be the true force model. We compare this to the outputs we get having considered only the 4 major forces.

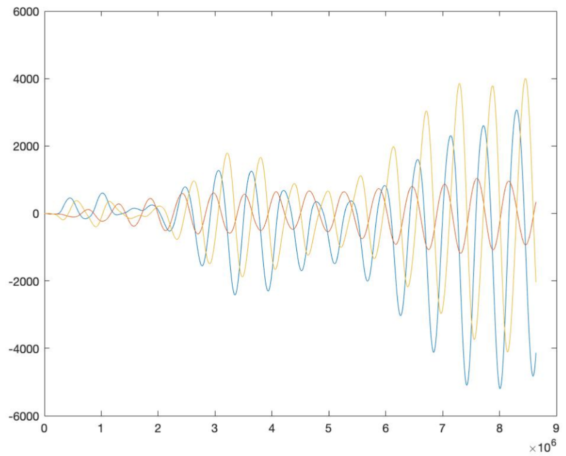


Figure 17: **Deviation in OP and GMAT(meters) and time(0.01 x sec)** GMAT's most realistic simulation has been used to compute for an SSO orbit against OP

From the above plot, it can be seen that the error is of about 4km, when propagated for a day. However, since our GNSS' update frequency is a few hours, the error within that time is well within the acceptable range.

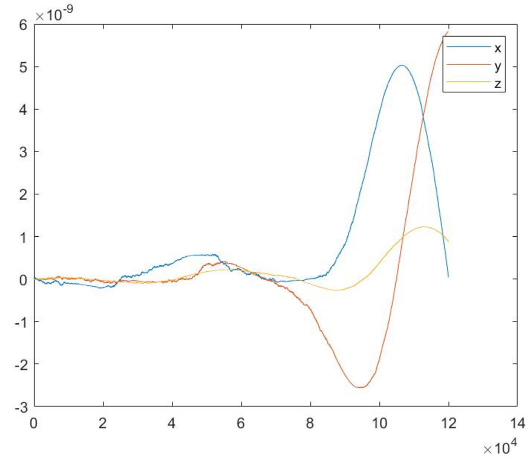


Figure 19: **Error plot of RK-4 vs RK89**

Numerical Methods

The error plots after using the RK-89. RK-4 and Adam's methods for orbit propagation considering only the gravity model (up to second degree perturbations) was plotted with the step size of 0.1 seconds for 120000 steps, before finally picking RK-4 as our integrator; discussed in the previous sections.

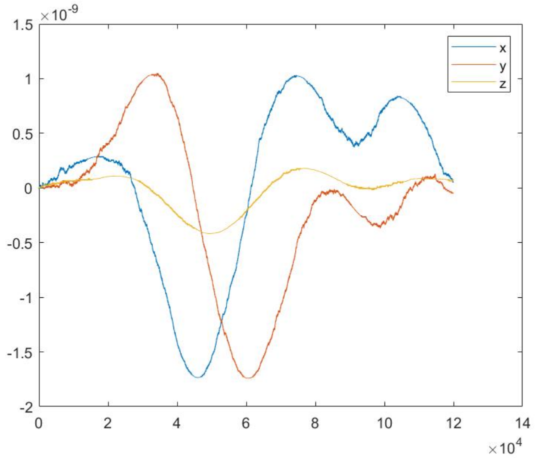


Figure 18: **Error plot of RK-4 vs Adam's Methods**

It can be seen from the above plots that the order of the error in RK-4, as compared to the more complex, higher order integrators; RK-89 and Adam's, is 10^{-9}. Which can be considered almost negligible for our numerical integration model.

6. Comparison of computational load

The computational load of our orbit propagator is an important factor to be considered for its integration into the navigational hardware of our nano-satellite. We seek to employ a comparatively lightweight, yet sufficiently accurate orbital propagation program. To attain this goal, we have written our orbit propagator in C. We came to this conclusion after having compared our C code to the myriad other programs/softwares used for orbital propagation. Namely, GMAT (General Mission Analysis Tool), STK (Systems Tool kit, by AGI), SGP4 and SGP8 (Simplified General Perturbations 4 and 8).

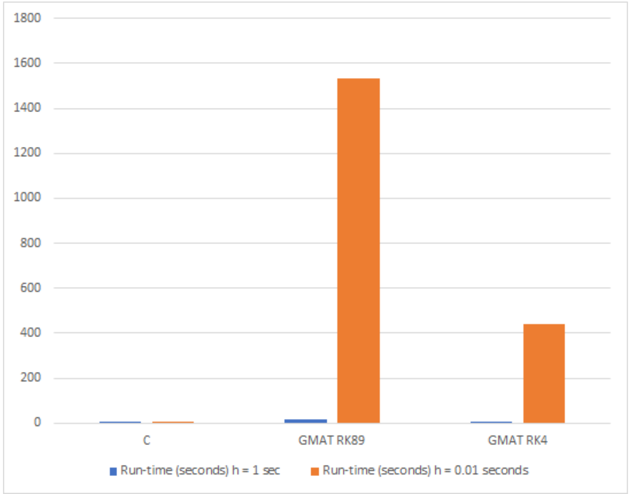


Figure 20: **GMAT vs. C program, run-time comparison**

The primary indicator of a software's computational load is the run-time of the code, a factor which also served as the primary basis for our comparison. In Table 1 and from the bar graph, it can be seen that STK and GMAT with a RK89 integrator (Using the highest accuracy models in both GMAT and STK) proved to be the slowest, hence having the highest computational cost, though their accuracy is the best when compared to the other propagators in the chart. GMAT with a RK4 integrator is the the propagation model akin to our propagator, since we use the same numerical integrator. However, its run-time is much higher than our orbital propagation code in C. This is because GMAT uses MATLAB and Python programs to carry out orbital propagation, which are much more computationally intensive than C programs. Hence, our orbit propagator code in C proved to be comparatively more lightweight than the GMAT and STK propagator models, while also being sufficiently accurate, as seen in the comparisons done in the previous section.

As evident in the Table 1, the C program is not the fastest propagator. It was observed that SGP4 and SGP8, both run as a MATLAB code, proved to be faster than our C code. However, they are not accurate enough. They are analytic and hence low fidelity propagators which consider secular and periodic variations due to Earth oblateness, solar and lunar gravitational effects, gravitational resonance effects and orbital decay using a simple drag model to approximate the motion of the object [26] , they do not integrate the force equations and rely on the aforementioned approximations alone for orbital propagation. This provides a very lightweight propagation method at the high cost of accuracy for a near-earth orbit like ours. The determination error for a near-earth orbit is a maximum of 40 km, when propagated for 3 days [27]. Which is much higher than the errors seen in our Orbital propagator which, as seen in the previous section, do not exceed 1 km, when the orbit is propagated for a day. Since our goal is mainly to attain the highest possible accuracy for a minimal computational load, our C orbit propagator proves to be most suitable for our mission.

7. Additional modules incorporated with orbit propagation

IGRF

The IGRF model that we use for the calculation of the magnetic field vectors is IGRF-13. Our program carries out the conversion of the FORTRAN subroutines into MATLAB,

Table 1: **Run-time comparison**

Propagator	Time-step (h)	Run-time
C Program	1 second	1.45 seconds
GMAT RK4	1 second	7.584 seconds
GMAT RK89	1 second	17.600 seconds
C Program	0.01 seconds	2.18 seconds
GMAT RK4	0.01 seconds	440.181 seconds
GMAT RK89	0.01 seconds	1534.180 seconds
STK	1 second	1361.16 seconds
SGP4	-	0.359 seconds
SGP8	-	0.636 seconds

such that the calculations can be carried out there.

The error while using this program is around 0.1nT in magnitude when compared with NGDC's (National Geophysical Data Centre) model on the same; a very minimal error which comes well within our desired range.

SSE

The Summarized Sun-Earth vector, abbreviated as SSE, is the vector we are using for generating the CSS (Coarse Sun Sensor) reference frame. For the calculation of this vector, we take into account the resultant of the Sun irradiance vector and the albedo irradiance vector; which is the vector in the direction of the light reflected off of the Earth's atmosphere.

The albedo irradiance is calculated by discretising the Earth surface into elements covering 1° latitude and 1.25° longitude. The Earth reflectivity data being used was derived from the TOMS mission data released by NASA. The following expression is used to calculate the effective reflectivity from Earth by combining the irradiance vectors from all elements lying in the intersection of the fields of view of the satellite and Sun:

$$\vec{l_g} = \sum_{Sat \bigcap Sun} E_c \vec{r_g} \tag{60}$$

where E_c is the irradiance from the Earth cell and $\vec{r_g}$ is the unit vector defining the cell to satellite direction.
The term E_c is calculated as follows:

$$E_c = \frac{\rho(\phi_g, \theta_g) E_{AM0} A_c(\phi_g) \vec{r}_{sun}^{T} \vec{n_c} \vec{r_g}^{T} \vec{n_c}}{\pi |\vec{r_{sat}}|^2} \tag{61}$$

where ρ is reflectivity, E_{AM0} is AM0 irradiance that falls from Sun, onto Earth. A_c is area of the cell. $\vec{r}_{sun}$ is Earth cell to Sun unit vector and $\vec{r_{sat}}$ is the non-normalised cell to satellite vector. n_c is the cell normal unit vector.

8. GNSS-Orbit Propagator integration

We must update the orbit propagator's initial coordinates as the error accumulates over time due to numerical propagation and other unaccounted forces such as gravitational influences of other celestial bodies, higher order perturbations due to Earth's gravity, and so on.

Given the low power usage for the GNSS receiver (100mW average), we have opted to leave it on at all times except if the emergency mode of operation is activated. This means we can update the parameters of orbit-propagator at any time, but it will place more computational strain on the on-board computer. Instead we utilise a trial and error approach if it's within acceptable bounds. The orbit propagator will only be used for flight planning since the GPS will be active at all times. Update rate will become more frequent before entering into imaging mode as it will require more precision during that period. The update rate is comparatively reduced before entering down-linking mode, as it demands significantly less accuracy.

There are three possible down-linking modes at an altitude of around 600 kilometres, each about 8 hours apart, assuming the ground station's field of vision (FOV) is 10 degrees from the horizon. Given that the control assembly takes around 10 minutes to converge to an attitude, data regarding the ground station or imaging location would need to be provided at least 10 minutes ahead of time. Thus, the orbit propagator will begin updating 15 minutes before the ground station or imaging location arrives. After 15 minutes, the inaccuracy is in the hundreds of metres, which is within acceptable limits. Consequently, in imaging mode, we'll update it every minute, whilst in down-linking mode, we'll update it every 5 minutes, since this will give us the needed accuracy for the flight plan. Each orbit propagator run will retain flight data for one orbit which lasts approximately 1.5 hours at a height of 600 kilometres.

9. Conclusion

For the purpose of verification and testing of other algorithms as well as for on-board functionality, an in-house, lightweight satellite orbit propagator was developed. Further additions were made to the propagator to calculate magnetic field and sun vectors. Albedo accounting for the sun vector was also done using a reflectivity model on a discretised Earth surface.

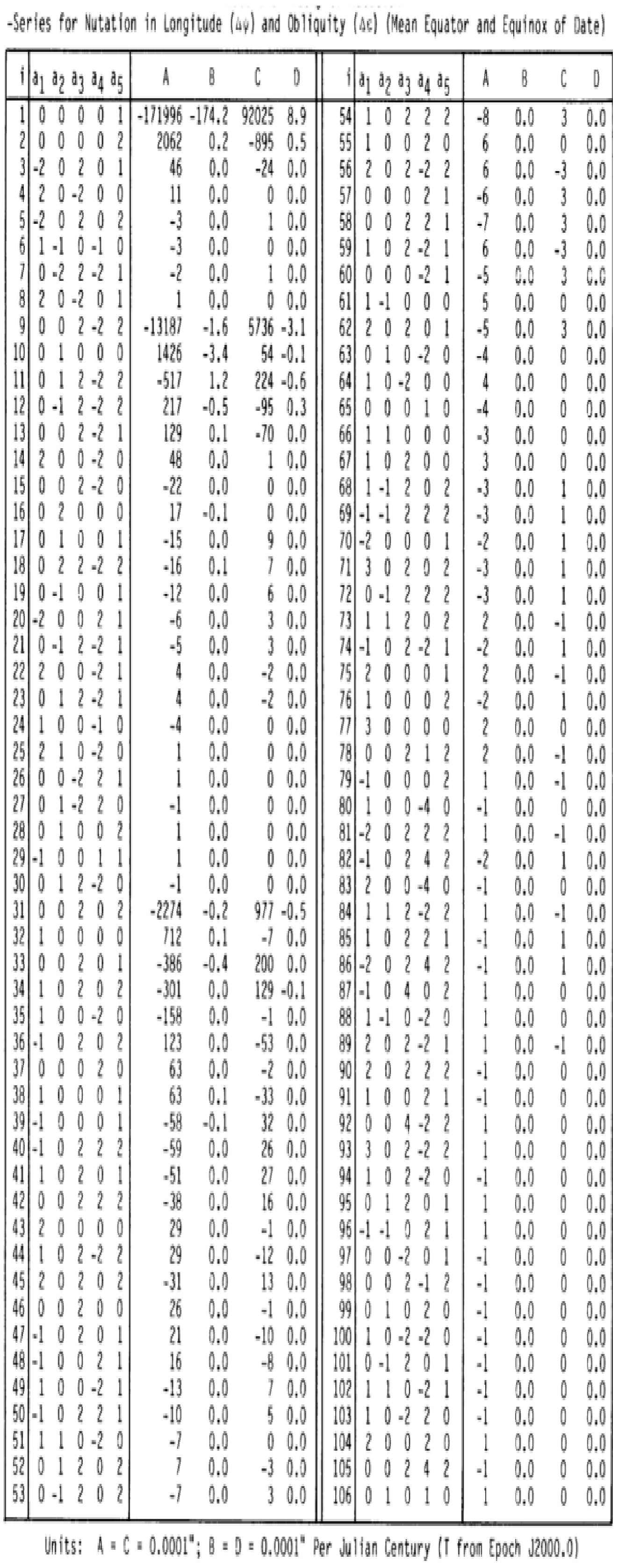
The satellite state is propagated with the assumption that Earth point mass, J2, atmospheric drag and SRP would be the only forces acting on the satellite that would have a significant effect on satellite state within the required time period of propagation.

Each of the forces was individually tested against the data generated using the same forces through NASA's GMAT software. The error for each of the forces was found to be within acceptable bound considering the direct and indirect effect of position error on attitude error.

Similarly each of the reference vectors are verified against existing and validated models and are again found to be within acceptable margins of error.

APPENDICES

A. SERIES FOR NUTATION IN LONGITUDE AND OBLIQUITY



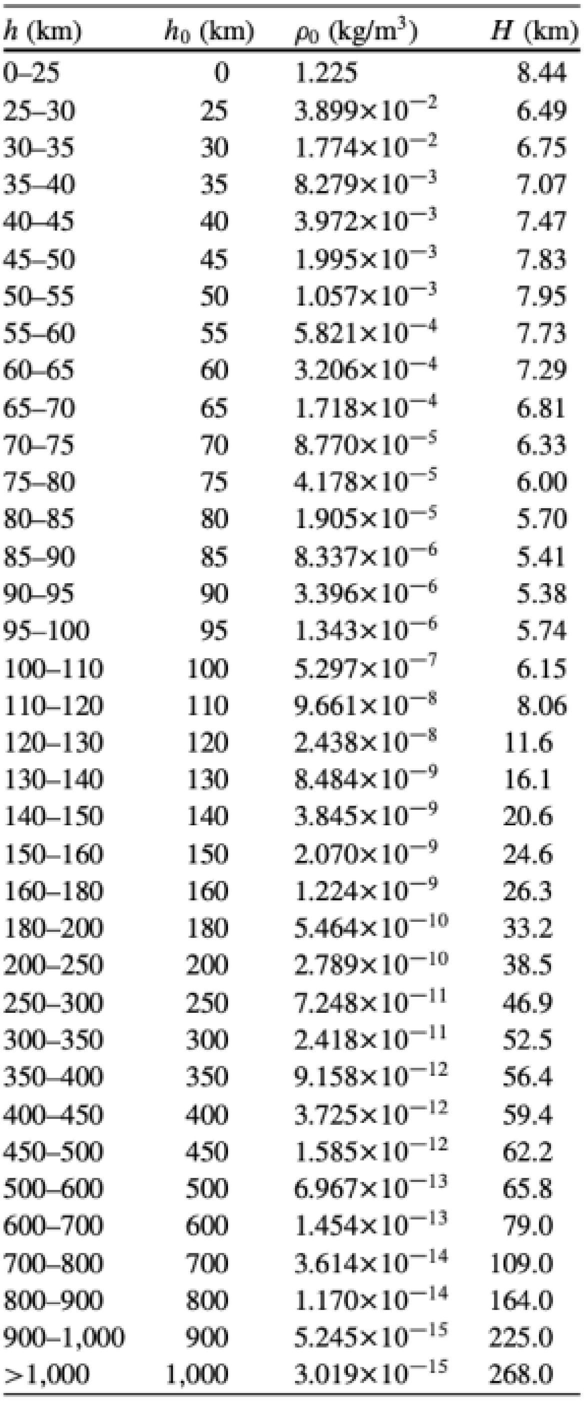
-Series for Nutation in Longitude (Δψ) and Obliquity (Δε) (Mean Equator and Equinox of Date)

i	a_1	a_2	a_3	a_4	a_5	A	B	C	D	i	a_1	a_2	a_3	a_4	a_5	A	B	C	D
1	0	0	0	0	1	-171996	-174.2	92025	8.9	54	1	0	2	2	2	-8	0.0	3	0.0
2	0	0	0	0	2	2062	0.2	-895	0.5	55	1	0	0	2	0	6	0.0	0	0.0
3	-2	0	2	0	1	46	0.0	-24	0.0	56	2	0	2	-2	2	6	0.0	-3	0.0
4	2	0	-2	0	0	11	0.0	0	0.0	57	0	0	0	2	1	-6	0.0	3	0.0
5	-2	0	2	0	2	-3	0.0	1	0.0	58	0	0	2	2	1	-7	0.0	3	0.0
6	1	-1	0	-1	0	-3	0.0	0	0.0	59	1	0	2	-2	1	6	0.0	-3	0.0
7	0	-2	2	-2	1	-2	0.0	1	0.0	60	0	0	0	-2	1	-5	0.0	3	C.C
8	2	0	-2	0	1	1	0.0	0	0.0	61	1	-1	0	0	0	5	0.0	0	0.0
9	0	0	2	-2	2	-13187	-1.6	5736	-3.1	62	2	0	2	0	1	-5	0.0	3	0.0
10	0	1	0	0	0	1426	-3.4	54	-0.1	63	0	1	0	-2	0	-4	0.0	0	0.0
11	0	1	2	-2	2	-517	1.2	224	-0.6	64	1	0	-2	0	0	4	0.0	0	0.0
12	0	-1	2	-2	2	217	-0.5	-95	0.3	65	0	0	0	1	0	-4	0.0	0	0.0
13	0	0	2	-2	1	129	0.1	-70	0.0	66	1	1	0	0	0	-3	0.0	0	0.0
14	2	0	0	-2	0	48	0.0	1	0.0	67	1	0	2	0	0	3	0.0	0	0.0
15	0	0	2	-2	0	-22	0.0	0	0.0	68	1	-1	2	0	2	-3	0.0	1	0.0
16	0	2	0	0	0	17	-0.1	0	0.0	69	-1	-1	2	2	2	-3	0.0	1	0.0
17	0	1	0	0	1	-15	0.0	9	0.0	70	-2	0	0	0	1	-2	0.0	1	0.0
18	0	2	2	-2	2	-16	0.1	7	0.0	71	3	0	2	0	2	-3	0.0	1	0.0
19	0	-1	0	0	1	-12	0.0	6	0.0	72	0	-1	2	2	2	-3	0.0	1	0.0
20	-2	0	0	2	1	-6	0.0	3	0.0	73	1	1	2	0	2	2	0.0	-1	0.0
21	0	-1	2	-2	1	-5	0.0	3	0.0	74	-1	0	2	-2	1	-2	0.0	1	0.0
22	2	0	0	-2	1	4	0.0	-2	0.0	75	2	0	0	0	1	2	0.0	-1	0.0
23	0	1	2	-2	1	4	0.0	-2	0.0	76	1	0	0	0	2	-2	0.0	1	0.0
24	1	0	0	-1	0	-4	0.0	0	0.0	77	3	0	0	0	0	2	0.0	0	0.0
25	2	1	0	-2	0	1	0.0	0	0.0	78	0	0	2	1	2	2	0.0	-1	0.0
26	0	0	-2	2	1	1	0.0	0	0.0	79	-1	0	0	0	2	1	0.0	-1	0.0
27	0	1	-2	2	0	-1	0.0	0	0.0	80	1	0	0	-4	0	-1	0.0	0	0.0
28	0	1	0	0	2	1	0.0	0	0.0	81	-2	0	2	2	2	1	0.0	-1	0.0
29	-1	0	0	1	1	1	0.0	0	0.0	82	-1	0	2	4	2	-2	0.0	1	0.0
30	0	1	2	-2	0	-1	0.0	0	0.0	83	2	0	0	-4	0	-1	0.0	0	0.0
31	0	0	2	0	2	-2274	-0.2	977	-0.5	84	1	1	2	-2	2	1	0.0	-1	0.0
32	1	0	0	0	0	712	0.1	-7	0.0	85	1	0	2	2	1	-1	0.0	1	0.0
33	0	0	2	0	1	-386	-0.4	200	0.0	86	-2	0	2	4	2	-1	0.0	1	0.0
34	1	0	2	0	2	-301	0.0	129	-0.1	87	-1	0	4	0	2	1	0.0	0	0.0
35	1	0	0	-2	0	-158	0.0	-1	0.0	88	1	-1	0	-2	0	1	0.0	0	0.0
36	-1	0	2	0	2	123	0.0	-53	0.0	89	2	0	2	-2	1	1	0.0	-1	0.0
37	0	0	0	2	0	63	0.0	-2	0.0	90	2	0	2	2	2	-1	0.0	0	0.0
38	1	0	0	0	1	63	0.1	-33	0.0	91	1	0	0	2	1	-1	0.0	0	0.0
39	-1	0	0	0	1	-58	-0.1	32	0.0	92	0	0	4	-2	2	1	0.0	0	0.0
40	-1	0	2	2	2	-59	0.0	26	0.0	93	3	0	2	-2	2	1	0.0	0	0.0
41	1	0	2	0	1	-51	0.0	27	0.0	94	1	0	2	-2	0	-1	0.0	0	0.0
42	0	0	2	2	2	-38	0.0	16	0.0	95	0	1	2	0	1	1	0.0	0	0.0
43	2	0	0	0	0	29	0.0	-1	0.0	96	-1	-1	0	2	1	1	0.0	0	0.0
44	1	0	2	-2	2	29	0.0	-12	0.0	97	0	0	-2	0	1	-1	0.0	0	0.0
45	2	0	2	0	2	-31	0.0	13	0.0	98	0	0	2	-1	2	-1	0.0	0	0.0
46	0	0	2	0	0	26	0.0	-1	0.0	99	0	1	0	2	0	-1	0.0	0	0.0
47	-1	0	2	0	1	21	0.0	-10	0.0	100	1	0	-2	-2	0	-1	0.0	0	0.0
48	-1	0	0	2	1	16	0.0	-8	0.0	101	0	-1	2	0	1	-1	0.0	0	0.0
49	1	0	0	-2	1	-13	0.0	7	0.0	102	1	1	0	-2	1	-1	0.0	0	0.0
50	-1	0	2	2	1	-10	0.0	5	0.0	103	1	0	-2	2	0	-1	0.0	0	0.0
51	1	1	0	-2	0	-7	0.0	0	0.0	104	2	0	0	2	0	1	0.0	0	0.0
52	0	1	2	0	2	7	0.0	-3	0.0	105	0	0	2	4	2	-1	0.0	0	0.0
53	0	-1	2	0	2	-7	0.0	3	0.0	106	0	1	0	1	0	1	0.0	0	0.0

Units: A = C = 0.0001"; B = D = 0.0001" Per Julian Century (T from Epoch J2000.0)

Figure 21: Source: IAU (1980) Theory of Nutation

B. SERIES FOR THE AERODRAG MODEL



h (km)	h_0 (km)	ρ_0 (kg/m^3)	H (km)
0–25	0	1.225	8.44
25–30	25	3.899×10^{-2}	6.49
30–35	30	1.774×10^{-2}	6.75
35–40	35	8.279×10^{-3}	7.07
40–45	40	3.972×10^{-3}	7.47
45–50	45	1.995×10^{-3}	7.83
50–55	50	1.057×10^{-3}	7.95
55–60	55	5.821×10^{-4}	7.73
60–65	60	3.206×10^{-4}	7.29
65–70	65	1.718×10^{-4}	6.81
70–75	70	8.770×10^{-5}	6.33
75–80	75	4.178×10^{-5}	6.00
80–85	80	1.905×10^{-5}	5.70
85–90	85	8.337×10^{-6}	5.41
90–95	90	3.396×10^{-6}	5.38
95–100	95	1.343×10^{-6}	5.74
100–110	100	5.297×10^{-7}	6.15
110–120	110	9.661×10^{-8}	8.06
120–130	120	2.438×10^{-8}	11.6
130–140	130	8.484×10^{-9}	16.1
140–150	140	3.845×10^{-9}	20.6
150–160	150	2.070×10^{-9}	24.6
160–180	160	1.224×10^{-9}	26.3
180–200	180	5.464×10^{-10}	33.2
200–250	200	2.789×10^{-10}	38.5
250–300	250	7.248×10^{-11}	46.9
300–350	300	2.418×10^{-11}	52.5
350–400	350	9.158×10^{-12}	56.4
400–450	400	3.725×10^{-12}	59.4
450–500	450	1.585×10^{-12}	62.2
500–600	500	6.967×10^{-13}	65.8
600–700	600	1.454×10^{-13}	79.0
700–800	700	3.614×10^{-14}	109.0
800–900	800	1.170×10^{-14}	164.0
900–1,000	900	5.245×10^{-15}	225.0
>1,000	1,000	3.019×10^{-15}	268.0

Figure 22: Source: [11]

REFERENCES

[1] Prasenjit Sengupta, Satellite Relative motion propagation and control in the presence of J2 perturbations/

[2] A. Conway and John E. Prussing; Orbital Mechanics

[3] Jerome R. Vetter; The evolution of Earth gravitational models used in astrodynamics

[4] Anoop Kumar Arudra; Atmospheric Desnity estimation using satellite precision orbit ephemerides

[5] NASA, GMAT Force model documentation;

[6] Earth-Centered inertial; Wikipedia

[7] Vallado, D.A.: Fundamentals of Astrodynamics and Applications, 3rd edn. Microcosm Press, Hawthorne and Springer, New York (2007)

[8] Bate, R.R., Mueller, D.D., White, J.E.: Fundamentals of Astrodynamics. Dover Publications, New York (1971)

[9] Farrell, J., Barth, M.: The Global Positioning System Inertial Navigation. McGraw-Hill, New York (1998)

[10] Geodetic Latitude and Longitude, IBM website

[11] Markley, F. L., Crassidis, J. L. (2014). Fundamentals of spacecraft attitude determination and control (pp. 361-364). New York, NY, USA:: Springer New York.

[12] Blewitt, G. (1997). Basics of the GPS technique: observation equations. Geodetic applications of GPS, 10-54.

[13] Smith, M. L. (1977). Wobble and nutation of the Earth. Geophysical Journal International, 50(1), 103-140.

[14] Attitude Reference Frames, aisolutions website: ai-solutions.com

[15] Solar Radiation Pressure, aisolutions website: ai-solutions.com

[16] Grewal, M. S., Weill, L. R., Andrews, A. P. (2007). Global positioning systems, inertial navigation, and integration. John Wiley Sons.

[17] Psiaki, M. L. (1999). Autonomous low-earth-orbit determination from magnetometer and sun sensor data. Journal of Guidance, Control, and Dynamics, 22(2), 296-304.

[18] Bhaskaran, S., Desai, S., Dumont, P., Kennedy, B., Null, G., Owen Jr, W., ... Werner, R. (1998). Orbit determination performance evaluation of the deep space 1 autonomous navigation system.

[19] Use of Coordinate system in orbit propagation; Stack Exchange website space.stackexchange.com

[20] NASA; General Mission Analysis Tool (GMAT) Mathematical Specifications DRAFT

[21] Kahr, E., Montenbruck, O., O'Keefe, K. P. (2013). Estimation and analysis of two-line elements for small satellites. Journal of Spacecraft and Rockets, 50(2), 433-439.

[22] Grenfell, P. W. (2020). GNSS-based relative navigation for LEO nanosatellite laser communications (Doctoral dissertation, Massachusetts Institute of Technology).

[23] Montenbruck, O., Gill, E. (2012). Satellite orbits: models, methods and applications. Springer Science Business Media.

[24] Ziebart, M. (2004). Generalized analytical solar radiation pressure modeling algorithm for spacecraft of complex shape. Journal of spacecraft and rockets, 41(5), 840-848.

[25] Drag force in wind tunnels: A new method; Science direct website: sciencedirect.com/science/article/pii/S0378437116306744

[26] Orbit propagators for satellites; AGI website: help.agi.com/stk

[27] Wei Dong, Zhao Chang-yin (2011). An Accuracy Analysis of the SGP4/SDP4 Model. Chinese Astronomy and Astrophysics.

BIOGRAPHY

Ayush Mehta *is currently pursuing his Bachelors in Chemical Engineering (Hons) from Birla Institute of Technology and Science, Pilani. He is currently Attitude Determination and Control Subsystem Team Lead for his university's student satellite program. He has mainly worked on developing the navigational system for the CubeSat mission.*

Amay Sareen *is currently pursuing his Bachelors in Chemical Engineering and Masters in Physics from Birla Institute of Technology Science, Pilani. He has extensive experience in working with CubeSat navigation systems, orbit propagators and algorithms sensors for attitude determination estimation. He is currently an Executive Committee member for his university's student satellite program and led the Attitude Determination Control subsystem till August 2021*

Aaryav Mishra *is currently a sophomore pursuing his Bachelors in Electrical and Electronics Engineering and Masters in Physics from Birla Institute of Technology Science, Pilani. He is currently a member of the Attitude Determination and Control Subsystem Team for his university's student satellite program.*