General Mission Analysis Tool (GMAT) Mathematical Specifications DRAFT

May 7, 2013

NASA Goddard Space Flight Center
Greenbelt RD
Greenbelt, MD 20771

Contents

List of Figures

List of Tables

Chapter 1

Time

Time is the primary independent variable in GMAT. Time is used in integrating the equations of motion, and calculating planetary ephemerides, the orientations of planets and moons, and atmospheric density among others. GMAT uses three types of time systems depending on the type of calculations being performed: universal time systems based on the Earth's rotation with respect to the Sun; dynamic time systems that are based on the dynamic motion of the solar system and take into account relativistic effects; and atomic time systems based on the oscillation of the cesium atom. Each of these time systems has specific uses and is discussed below. In addition, universal, dynamic, and atomic time systems can be expressed in different time formats. The two time formats used in GMAT are the Modified Julian Date (MJD) format, and the Gregorian Date (GD) format. In the next section, we'll take a look at the time systems used in GMAT, and when GMAT uses each time system. Then we'll look at the different time formats.

1.1 Time Systems

GMAT uses several different time systems in physical models and spacecraft dynamics modelling. The choice of time system for a particular calculation is determined by which time system is most natural and convenient, as well as the accuracy required. In general, for determining Earth's orientation at a given epoch, we use one of several forms of Universal Time (UT), because universal time is based on the Earth's rotation with respect to the Sun. Planetary ephemerides are usually provided with time in a dynamic time system, because dynamic time is the independent variable in the dynamic theories and ephemerides. The independent variable in spacecraft equations of motion in GMAT is time expressed in an atomic time system. Let's look at each of these three systems, starting with atomic time.

1.1.1 Atomic Time: TAI and A.1

Atomic time (AT) is a highly accurate time system which is independent of the rotation of the Earth.[3] Therefore, AT is a natural system for integrating a spacecraft's equations of motion. AT is defined in terms of the oscillations of the cesium atom at mean sea level. The duration of the SI second is defined to be 9,192,631,770 oscillations of the cesium nuclide 133Ce. Two atomic times systems are are used in GMAT: A.1, and international atomic time (TAI). A.1 is in advance of TAI by 0.0343817 seconds.

$$A.1 = TAI + 0.0343817sec \tag{1.1}$$

where

$$TAI = UTC + \Delta AT \tag{1.2}$$

and ΔAT is the number of leap seconds, added since 1972, needed to keep $\mid UTC - UT1 \mid \leq 0.9$sec. GMAT reads ΔAT from the file named *tai-utc.dat*. For times that appear before the first epoch on the file, GMAT uses the first value found in the file. For times that appear after the last epoch, GMAT uses the last value contained in the file. Currently, GMAT uses A.1 time as the independent variable in the equations of motion. TAI is used as a time system for defining spacecraft state information.

Now let's look at the universal time system.

1.1.2 Universal Time: UTC and UT1

All of the universal time (UT) scales are based on the Earth's rotation with respect to a fixed point (sidereal time) or with respect to the Sun (solar time). The observed universal time (UTO) is determined from observations of stellar transits to determine mean local sidereal time. UT1 is UTO corrected for the Earth's polar motion and is used when the instantaneous orientation of the Earth is needed. UTC is the basis for all civil time standards. It is also known as Greenwich mean time (GMT) and Zulu time (Z). The UTC time unit is defined to be an SI second, but UTC is kept within 0.9 seconds of UT1 by occasional leap second adjustments. The equation relating UTC and UT1 is

$$UTC = UT1 - \Delta UT1 \tag{1.3}$$

In GMAT, $\Delta UT1$ is read from the file *eopc04.62-now* provided by the International Earth Rotation and Reference Systems Service (IERS). The file containing the latest measurements and predictions can be found at
http://www.iers.org/. For times past the last epoch contained in the file, GMAT uses the last value of $\Delta UT1$ contained in the file. GMAT uses UTC as a time system to define spacecraft state information. UT1 is used to determine the Greenwich hour angle and for the sidereal time portion of FK5 reduction.

1.1.3 Dynamic Time: TT, TDB and TCB

Dynamical time is the independent variable in the dynamical theories and ephemerides. This class of time scales contains terrestrial time (TT), Barycentric Dynamical time (TDB), and Barycentric Coordinate Time (TCB). TDB is the independent variable in the equations of motion referred to the solar system barycenter. It is also the coordinate time in the theory of general relativity. Despite the fact that the Jet Propulsion Laboratory (JPL) J2000.0 ephemerides are referred to in TDB, TT is frequently used. This is because TT and TDB always differ by less than 0.002 sec. As higher accuracy or more sensitive missions are planned, the difference may need to be distinguished. In this section we'll discuss how to calculate TT, TDB and TCB, and discuss where each is used in GMAT.

TT is the independent variable in the equations of motion referred to the Earth's center. It is also the proper time in the theory of general relativity. The unit of TT is a day of 86400 SI seconds at mean sea level. In GMAT, TT is used in FK5 reduction, and as an intermediate time system in the calculation of TDB and TCB. TT can be calculated from the following equation:

$$TT = TAI + 32.184 \text{ sec} \tag{1.4}$$

Calculating TDB exactly is a complicated process that involves iteratively solving a transcendental equation. For this reason, it is convenient to use the following approximation

$$TDB \approx TT + \underbrace{0.001658 \sin M_E + 0.00001385 \sin 2M_E}_{\text{units of seconds}} \tag{1.5}$$

Note that the term in the underbrace has units of seconds, and depending upon the units of TT, which is usually in days, a conversion of the term may be necessary before performing the addition with TT. M_E is the Earth's mean anomaly with respect to the sun and is given approximately as

$$M_E \approx 357.5277233 + 35,999.05034 T_{TT} \tag{1.6}$$

where T_{TT} is the time in TT expressed in the Julian Century format. T_{TT} can be calculated from

$$T_{TT} = \frac{JD_{TT} - 2,451,545.0}{36,525} \tag{1.7}$$

where JD_{TT} is the time in TT expressed in the Julian Date format. For a more complete discussion of the TDB time system, see Vallado[3] (pp. 195-198) and Seidelmann[4] (pp. 41-48). GMAT uses TDB as the default time system in the JPL ephemerides files. There is an option to use TT in the ephemerides using the `UseTTForEphemeris` flag.

The last dynamic time system GMAT uses is Barycentic Coordinate Time. In 1992, the IAU adopted this system and clarified the relationships between space-time coordinates.[4] In general, calculating TCB requires a four-dimensional space-time transformation that is well beyond the scope of this discussion. However, TCB can be approximated using the following equation:

$$TCB - TDB = L_B(JD - 2443144.5)86400 \tag{1.8}$$

The present estimate of the value of L_B is $1.550505x10^{-8}$ $(+/-1x10^{-14})$ (Fukushima et al., Celestial Mechanics, 38, 215, 1986). It is important to note that the main difference between TDB and TCB is a secular drift, and that as of the J2000 Epoch, the difference was approximately 11.25 seconds and growing.[5] GMAT uses time in the TCB system to evaluate the IAU data for the spin axes and prime meridian locations of all planets and moons except for Earth.[1] Note that Seidelmann[1] mistakenly says that time in TCB should be used in the equations given for the pole and meridian locations of the planets. The correct time to use is TDB, and GMAT uses this time system.

1.2 Time Formats

There are two time formats that GMAT uses to represent time in the systems discussed above. These formats are called the Gregorian Date (GD), and the Julian Date (JD). The difference between the GD and JD formats is how they represent the Year, Month, Day, Hours, Minutes, and Seconds of a given date. The GD format is well known, and the J2000 epoch is expressed as, 01 Jan 2000 12:00:00.000 TT. The reference epoch for the GD calendar is the beginning of the Christian Epoch. The JD format represents an epoch as a continuous number containing the day and the fraction of day.

The J2000 epoch is commonly used in astrodynamics as a reference epoch for planetary and other data. The J2000 epoch occurred at 01 Jan 2000 12:00:00.000 TT. The time system, TT, is important for precise applications! While the J2000 epoch is a specific instant in time, the numerical value changes depending upon which time system you express it in. We can make an analogy with vector algebra where we have an abstract quantity that is a vector, and can't write down a set of numbers representing the vector until we choose a coordinate system. Similarly, the J2000 epoch can be written in any of the different time systems and formats. All of the following are equivalent definitions of the J2000 Epoch:

2451545.0	TT
2451544.9996274998411	TAI
2451544.9992571294708	UTC
2451544.9999999990686	TDB

01 Jan 2000	12 : 00 : 00.000000000	TT
01 Jan 2000	11 : 59 : 27.815986276	TAI
01 Jan 2000	11 : 58 : 55.815986276	UTC
01 Jan 2000	11 : 59 : 59.999919534	TDB

In the next two sections we'll look at how to convert an epoch in a given time system from the GD format to the JD format, and vice versa.

1.2.1 Julian Date and Modified Julian Date

The Julian date is a time format in which we can express a time known in any of the Atomic, Universal or Dynamic time systems. The Julian Date is composed of the Julian day number and the decimal fraction of the current day. Seidelmann[4] (pp. 55-56) says "The Julian day number represents the number of days that has elapsed, at Greenwich noon on the day designated, since ...the epoch noon Jan 1 4713 B.C. in the Julian proleptic calendar. The Julian date (JD) corresponding to any instant is, by simple extension to this concept, the Julian day number followed by the fraction of the day elapsed since the preceding noon".

The fundamental epoch for most astrodynamic calculations is the J2000.0 epoch.[4] This epoch is GD 01 Jan 2000 12:00:00.000 in the TT time system and is expressed as JD 2451545.0 TT. To convert between Julian Date format and Gregorian Date format, GMAT uses Algorithm 14 from Vallado[3]

$$JD = 367Y - Int\left(\frac{7\left(Y + Int\left(\frac{M+9}{12}\right)\right)}{4}\right) + Int\left(\frac{275M}{9}\right) + D + 1,721,013.5 + \frac{\frac{\frac{s}{60}+m}{60}+H}{24} \tag{1.9}$$

where Y is the four digit year, Int signifies real truncation, and M, D, H, m and s are month, day, hour, minutes, and seconds respectively. This equation is valid for the time period 01 Mar. 1900 to 28 Feb. 2100.

For numerical reasons it is often convenient to work in a Modified Julian Date (MJD) format to ensure we can capture enough significant figures using double precision computers. In GMAT the MJD system is defined as

$$MJD = JD - 2,430,000.0 \tag{1.10}$$

where the reference epoch expressed in the GD format is 05 Jan 1941 12:00:00.000. However, we must be careful in calculating the Modified Julian Date, or we will lose the precision we are trying to gain. GMAT calculates the MJD as follows:

$$\text{JDay} = 367Y - Int\left(\frac{7\left(Y + Int\left(\frac{M+9}{12}\right)\right)}{4}\right) + Int\left(\frac{275M}{9}\right) + D + 1,721,013.5 \tag{1.11}$$

$$\text{PartofDay} = \frac{\frac{\frac{s}{60}+m}{60}+H}{24} \tag{1.12}$$

$$\text{MJD} = (\text{JDay} - 2,430,000.0) + \text{PartofDay} \tag{1.13}$$

The important subtlety is that we must subtract the MJD reference from the JD, before we add the fraction of day, to avoid losing precision in the MJD.

1.2.2 Gregorian Date

The Gregorian Date format is primarily used as a time system in which to enter state information in GMAT. GD is not a convenient time format for most mathematical calculations. Hence, GMAT often takes input in the GD format and converts it to a MJD format for use internally. The algorithm for converting from GD to JD is taken from Vallado[3] and reproduced here verbatim.

$$T_{1900} = \frac{JD - 2,415,019.5}{365.25}$$

$$Year = 1900 + \text{TRUNC}(T_{1900})$$

$$LeapYrs = \text{TRUNC}((Year - 1900 - 1)(0.25))$$

$$Days = (JD - 2,415,019.5) - ((Year - 1900)(365.0) + LeapYrs)$$

IF $Days < 1.0$ THEN

$$Year = Year - 1$$

$$LeapYrs = \text{TRUNC}((Year - 1900 - 1)(0.25))$$

$$Days = (JD - 2,415,019.5) - ((Year - 1900)(365.0) + LeapYrs)$$

If ($Year$ Mod 4) = 0 Then

$$LMonth[2] = 29$$

$$DayofYr = \text{TRUNC}(Days)$$

Sum days in each month until
$LMonth + 1$ summation $> DayofYr$

$Mon = \#$ of months in summation

$Day = DayofYr - LMonth$ summation

$$\tau = (Days - DayofYr)24$$

$$h = \text{TRUNC}(Temp)$$

$$min = \text{TRUNC}((Temp - h)60)$$

$$s = \left(Temp - h - \frac{min}{60}\right)3600$$

1.3 Conclusions

In this chapter we looked at three time systems that GMAT uses to perform internal calculations: atomic time, universal time, and dynamic time. Atomic time is used to integrate spacecraft equations of motion, while universal time is used to determine the sidereal time and greenwhich hour angle for use in FK5 reduction. Dynamic time systems are used in the JPL ephemerides and in the IAU planetary orientation data. Time, in any of these time systems, are represented in two formats: the Gregorian Date, and Julian Date. We looked at how to convert between different time systems, and between different time formats.

Chapter 2

Coordinate Systems

2.1 General Coordinate System Transformations

GMAT has the capability to take a position and velocity vector in one coordinate system, and convert them to another coordinate system that may be both translating and rotating with respect to the original system. In this section we derive the equations governing coordinate system transformations and describe the algorithm GMAT uses to transform position and velocity vectors.

We start by defining some notation. In Fig. 2.1, we see an illustration of a point "p" and two coordinate systems $\mathcal{F}_\mathcal{O}$ and $\mathcal{F}_\mathcal{F}$. Define the the position of p expressed in $\mathcal{F}_\mathcal{O}$ as $\mathbf{r}_p^\mathcal{O}$. Define the position of point p with respect to frame $\mathcal{F}_\mathcal{F}$ as $\mathbf{r}_p^\mathcal{F}$ *It is important to note that using this notation,* $\|\mathbf{r}_p^\mathcal{O}\| \neq \|\mathbf{r}_p^F\|$, *because the transformation defined in the notation contains a translation from the origin of* $\mathcal{F}_\mathcal{O}$ *to the origin of* $\mathcal{F}_\mathcal{F}$ *as well as a coordinate rotation.* $\mathbf{r}_{f/i}$ is the vector from the origin of $\mathcal{F}_i$ to origin of $\mathcal{F}_f$. Define the rotation matrix that rotates from $\mathcal{F}_\mathcal{I}$ to $\mathcal{F}_\mathcal{F}$ as $\mathbf{R}^{\mathcal{F}/\mathcal{I}}$. Finally, let's define the angular velocity $\boldsymbol{\omega}_{f/i}$ as the angular velocity of $\mathcal{F}_\mathcal{I}$ with respect to $\mathcal{F}_\mathcal{F}$. To simplify the notation, we assume that a vector is expressed in the frame denoted by the superscript. If need to define a point "p" with respect to $\mathcal{F}_\mathcal{O}$, but express this result in $\mathcal{F}_\mathcal{F}$ we use the notation $[\mathbf{r}_p^\mathcal{O}]^\mathcal{F}$ curly brackets. In summary, we have

$\mathbf{r}_p^\mathcal{O}$	Position of point p w/r/t frame $\mathcal{F}_\mathcal{O}$ expressed in $\mathcal{F}_\mathcal{O}$
$[\mathbf{r}_p^\mathcal{O}]^\mathcal{F}$	Position of point p w/r/t frame $\mathcal{F}_\mathcal{O}$ expressed in $\mathcal{F}_\mathcal{F}$
$\mathbf{r}_{f/o}^\mathcal{F}$	Position vector from origin of $\mathcal{F}_\mathcal{O}$ to origin of $\mathcal{F}_\mathcal{F}$ expressed in $\mathcal{F}_\mathcal{F}$
$\mathbf{R}^{\mathcal{F}/\mathcal{O}}$	Rotation matrix from frame $\mathcal{F}_\mathcal{O}$ to $\mathcal{F}_\mathcal{F}$
$\boldsymbol{\omega}_{f/o}^\mathcal{O}$	Angular velocity of frame $\mathcal{F}_\mathcal{O}$ w/r/t $\mathcal{F}_\mathcal{F}$, expressed in frame $\mathcal{F}_\mathcal{O}$
$\boldsymbol{\omega}_{f/o}^\mathcal{F}$	Angular velocity of frame $\mathcal{F}_\mathcal{O}$ w/r/t $\mathcal{F}_\mathcal{F}$, expressed in frame $\mathcal{F}_\mathcal{F}$

From inspection of Figure 2.1 we can write

$$\mathbf{r}_o^\mathcal{F} = \underbrace{\mathbf{R}^{\mathcal{F}/\mathcal{I}}\mathbf{r}_o^\mathcal{I}}_{Rot.} + \underbrace{\mathbf{r}_{i/f}^\mathcal{F}}_{Trans.} \tag{2.1}$$

Equation (2.1) is the equation used to convert a vector known in frame $\mathcal{F}_i$ to a vector in frame $\mathcal{F}_f$, where both a rotation and a translation are required. The first term in Eq. (2.1) is the term that performs the rotation portion of the transformation. Here, $\mathbf{r}_i$ is the position vector w/r/t to $\mathcal{F}_i$ and is expressed in $\mathcal{F}_i$. $\mathbf{R}_{fi}$ is the rotation matrix that rotates from $\mathcal{F}_i$ to $\mathcal{F}_f$. $\mathbf{r}_{if}$ is the vector that goes from the origin of $\mathcal{F}_f$ to the origin of $\mathcal{F}_i$, and is expressed in $\mathcal{F}_f$.

We also need to be able to determine the time rate of change of a vector in frame $\mathcal{F}_f$ if we know the time rate of change of the vector in $\mathcal{F}_i$. To determine the equation that describes the transformation, we must take the derivative of Eq. (2.1) with respect to time.

$$\frac{d\mathbf{r}_f}{dt} = \frac{d\mathbf{R}_{fi}\mathbf{r}_i}{dt} + \frac{d\mathbf{r}_{if}}{dt} \tag{2.2}$$

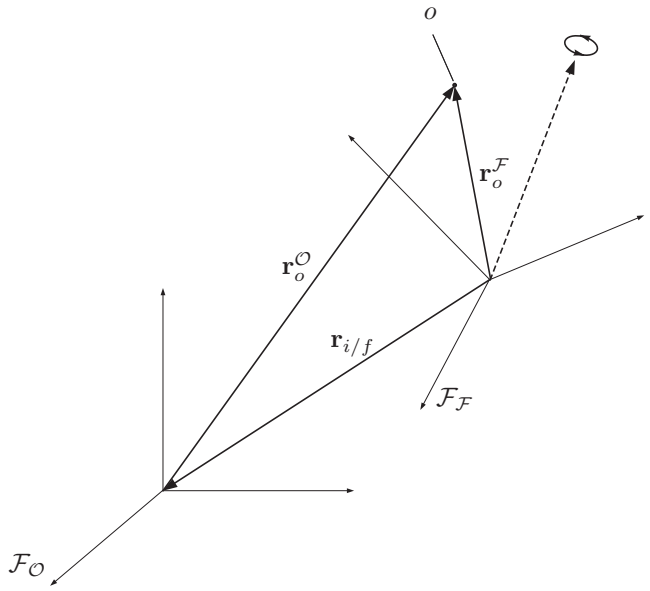


Figure 2.1: Illustration of a Translating and Rotating Coordinate System

Let's use a single dot above a variable to denote the first derivative of that variable with respect to time. Then, we can expand this to obtain

$$\dot{\mathbf{r}}_f = \dot{\mathbf{R}}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i + \dot{\mathbf{r}}_{if} \tag{2.3}$$

In Eq. (2.3) we see a term that contains the time derivative of the rotation matrix from $\mathcal{F}_i$ to $\mathcal{F}_f$. We can write the time derivative of $\mathbf{R}_{fi}$ as

$$\dot{\mathbf{R}}_{fi} = \mathbf{R}_{fi}\boldsymbol{\omega}_{fi}^{\mathrm{X}} = \{\boldsymbol{\omega}_{fi}^{\mathrm{X}}\}_f\mathbf{R}_{fi} \tag{2.4}$$

where $\boldsymbol{\omega}_{fi}$ is the angular velocity of $\mathcal{F}_i$ with respect to $\mathcal{F}_f$ expressed in $\mathcal{F}_i$. The skew symmetric matrix, $\boldsymbol{\omega}^x$, is defined as

$$\boldsymbol{\omega}^x = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \tag{2.5}$$

In summary, using Eq. (2.4) to transform a derivative vector from $\mathcal{F}_i$ to $\mathcal{F}_f$ we can use any of the following three equations:

$$\dot{\mathbf{r}}_f = \underbrace{\mathbf{R}_{fi}\boldsymbol{\omega}_{fi}^{\mathrm{X}}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{2.6}$$

$$\dot{\mathbf{r}}_f = \underbrace{\{\boldsymbol{\omega}_{fi}^{\mathrm{X}}\}_f\mathbf{R}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{2.7}$$

$$\dot{\mathbf{r}}_f = \underbrace{\dot{\mathbf{R}}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{2.8}$$

We choose between Eqs. (2.6), (2.7), or (2.8) depending on the type of information we have, and which frame is most convenient to express the angular velocity $\boldsymbol{\omega}_{fi}$ in. In general, we know $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$. To perform the transformation we need to determine $\mathbf{R}$, $\dot{\mathbf{R}}$, and $\dot{\mathbf{r}}_{if}$ and these quantities depend on $\mathcal{F}_i$ and $\mathcal{F}_f$.

One of the difficulties in implementing coordinate system transformations in GMAT is that we often can't calculate $\mathbf{R}_{fi}$ and $\dot{\mathbf{R}}_{fi}$ directly. For example, it is nontrivial to directly calculate the rotation matrix from the Earth fixed frame to the Moon fixed frame. Hence, we need to choose a convenient intermediate coordinate system. We choose the axis system defined by Earth's mean equinox and mean equator at the J2000 epoch, denoted $\mathcal{F}_{J_{2k}}$, as the intermediate reference frame for all transformations that require an

intermediate transformation. This choice is motivated by the fact that most of the data needed to calculate $\mathbf{R}$ and $\dot{\mathbf{R}}$ is given so that it is fast and convenient to calculate $\mathbf{R}_{J_{2k},i}$, and $\dot{\mathbf{R}}_{J_{2k},i}$.

The steps taken to perform a general coordinate transformation in GMAT are described below and illustrated in Fig. 2.2. We start with a vector and its first derivative known in frame $\mathcal{F}_i$, and wish to determine the vector and its first derivative with respect to frame $\mathcal{F}_f$. However, we assume that the transformation to go directly from $\mathcal{F}_i$ to $\mathcal{F}_f$ is not known.

The first step in the process is to perform a rotation from $\mathcal{F}_i$ to $\mathcal{F}_{J_{2k}}$. We define this intermediate system as $\mathcal{F}_1$. No translation is performed in step one. Using only the rotation portions of from Eqs. (2.1) and (2.8) we see that

$$\{\mathbf{r}_i\}_1 = \mathbf{R}_{J_{2k},i}\mathbf{r}_i \tag{2.9}$$

$$\{\dot{\mathbf{r}}_i\}_1 = \dot{\mathbf{R}}_{J_{2k},i}\mathbf{r}_i + \mathbf{R}_{J_{2k},i}\dot{\mathbf{r}}_i \tag{2.10}$$

The second step is to perform a translation from the origin of $\mathcal{F}_i$ to the origin of $\mathcal{F}_f$. We define this second intermediate system as $\mathcal{F}_2$. $\mathcal{F}_2$ has the same origin as $\mathcal{F}_f$ but has the same axes as $\mathcal{F}_{J_{2k}}$. From inspection of Fig.2.2 we can see that

$$\{\mathbf{r}_i\}_1 = \{\mathbf{r}_{Ri}\}_{J_{2k}} + \{\mathbf{r}_{fR}\}_{J_{2k}} + \{\mathbf{r}_f\}_2 \tag{2.11}$$

Solving for $\mathbf{r}_f$ we obtain

$$\{\mathbf{r}_f\}_2 = \{\mathbf{r}_i\}_1 - \{\mathbf{r}_{Ri}\}_{J_{2k}} - \{\mathbf{r}_{fR}\}_{J_{2k}} \tag{2.12}$$

where $\{\mathbf{r}_{Ri}\}_{J_{2k}}$ is the vector from the origin of $\mathcal{F}_i$ to the origin of $\mathcal{F}_R$ expressed in $\mathcal{F}_{J2k}$. Similarly $\{\mathbf{r}_{fR}\}_{J_{2k}}$ is the vector from the origin of $\mathcal{F}_R$ to the origin of $\mathcal{F}_f$ expressed in $\mathcal{F}_{J2k}$. Because the vector $\{\mathbf{r}_f\}_2$ is expressed in an inertial system we can we can take the derivative of Eq .(2.12) to obtain

$$\{\mathbf{v}_f\}_2 = \{\mathbf{v}_i\}_1 - \{\mathbf{v}_{Ri}\}_{J_{2k}} - \{\mathbf{v}_{fR}\}_{J_{2k}} \tag{2.13}$$

where $\{\mathbf{v}_{Ri}\}_{J_{2k}}$ is the velocity of the origin of $\mathcal{F}_R$ w/r/t the origin of $\mathcal{F}_i$. Similarly, $\{\mathbf{v}_{fR}\}_{J_{2k}}$ is the velocity of the origin of $\mathcal{F}_f$ w/r/t the origin of $\mathcal{F}_R$. Finally, we perform a rotation from $\mathcal{F}_{J_{2k}}$ to $\mathcal{F}_f$ about the origin of $\mathcal{F}_f$ to obtain the desired quantities.

$$\mathbf{r}_f = \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{2.14}$$

$$\dot{\mathbf{r}}_f = \dot{\mathbf{R}}_{f,J_{2k}}\{\mathbf{r}_f\}_2 + \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{2.15}$$

2.2 Pseudo-Rotating Coordinate Systems

In mission analysis, sometimes it is useful to consider a rotating coordinate system to be inertial at a given instant in time. In this case, we ignore the effects of rotation on the velocity. Let's call systems where we neglect the rotational effects on velocity pseudo-rotating coordinate systems.

To perform transformations to a pseudo-rotating coordinate system, the equations to convert a position vector do not change and are given by Eqs. (2.9) and (2.14). However, the velocity conversion equations change because we neglect the terms that contain $\dot{\mathbf{R}}$. For pseudo-rotating coordinate systems the velocity transformations shown in Eqs. (2.10) and (2.15) become

$$\left\{\frac{d\mathbf{r}_i}{dt}\right\}_1 = \mathbf{R}_{J_{2k},i}\frac{d\mathbf{r}_i}{dt} \tag{2.16}$$

and

$$\frac{d\mathbf{r}_f}{dt} = \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{2.17}$$

To perform the transformations describe in the last few sections, we need to be able to calculate the rotation matrix between any coordinate system and $\mathcal{F}_{J_{2k}}$, and the derivative of the rotation matrix. In the following sections we calculate these matrices for the systems used in GMAT. We assume that we want the

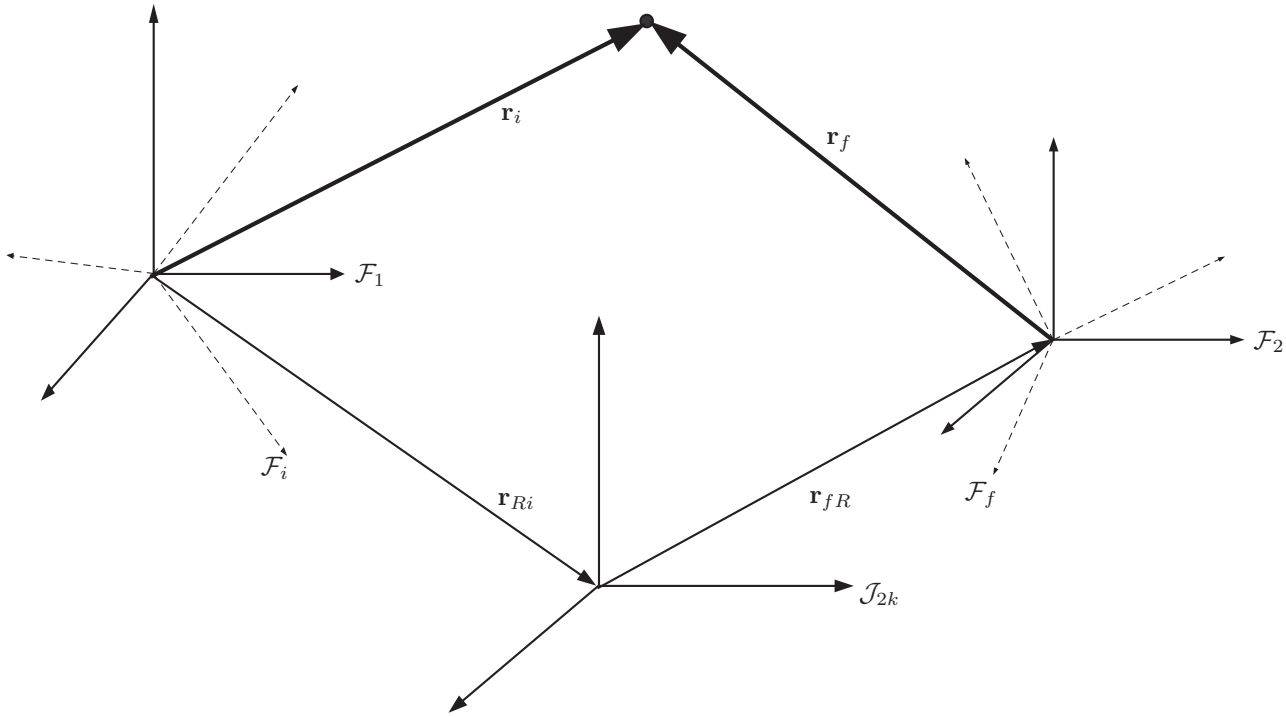


Figure 2.2: General Coordinate System Transformation Approach in GMAT

transformation from some generic frame $\mathcal{F}_i$ to $\mathcal{F}_{J_{2k}}$ which is the Earth Mean J2000 Equatorial (MJ2000Eq) system. The rotation matrix from $\mathcal{F}_{J_{2k}}$ to $\mathcal{F}_i$ can be found by the simple relationship.

$$\mathbf{R}_{i,J_{2k}} = \mathbf{R}^{-1}_{J_{2k},i} = \mathbf{R}^{T}_{J_{2k},i} \tag{2.18}$$

and

$$\dot{\mathbf{R}}_{if} = \dot{\mathbf{R}}^{-1}_{J_{2k},i} = \dot{\mathbf{R}}^{T}_{J_{2k},i} \tag{2.19}$$

2.3 ITRF and ICRF

The computation for the ICRF and ITRF tranformations in GMAT are from Kaplan[6] and Capolla[7] *et.al.* and employ the IAU 2000A nutation IAU 2006 precession models. The transformation is performed using three intermediate rotations as follows:

$$\mathbf{R} = \mathbf{C}^T\mathbf{R}_3(-\theta)\mathbf{W} \tag{2.20}$$

where $\mathbf{W}$ is the polar motion matrix, $\mathbf{R}_3$ is a 3 rotation through the Earth rotation angle θ (see Section 2.7.1), and $\mathbf{C}$ is a single matrix that captures precession, nutation, and frame bias. The time derivative of the rotation matrix assumes that the only significant time variation of the rotation matrix is due to the Earth's spin and is computed using

$$\mathbf{R} = \mathbf{C}^T\mathbf{R}_3(-\theta)\boldsymbol{\omega}_E^x\mathbf{W} \tag{2.21}$$

where

$$\boldsymbol{\omega}_E^x = \begin{pmatrix} 0 & -\omega_e & 0 \\ \omega_e & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.22}$$

and ω_e is computed using Eq. (2.98).

$\mathbf{W}$ is computed from Eq. 6.15 in Kaplan[6] as shown below.

$$\mathbf{W} = \mathbf{R}_3(-s')\mathbf{R}_2(x_p)\mathbf{R}_1(y_p) \tag{2.23}$$

The variable s' is computed from

$$s' = (-47\mu\text{as})T_{TT} \tag{2.24}$$

where T_{TT} is given by Eq. (1.7)and x_p and y_p are interpolated using third order lagrange interpolation from Earth Orientation Parameters (EOP) files provided by the IERS. The Earth Rotation angle is computed from

$$\theta = 2\pi(0.7790572732640 + 1.00273781191135448(JD_{UT1} - 2451545.0)) \tag{2.25}$$

where JD_{UT1} is the Julian date expressed in UT1 using Eq. (1.3).

$\mathbf{C}^T$ is computed using

$$\mathbf{C}^T = \begin{pmatrix} 1 - bX^2 & -bXY & X \\ -bXY & 1 - bY^2 & Y \\ -X & -Y & 1 - b(X^2 + Y^2) \end{pmatrix} \mathbf{R}_3(s) \tag{2.26}$$

where

$$b = \frac{1}{1 + \sqrt{1 - X^2 - Y^2}} \tag{2.27}$$

The variables X, Y, are respectively the x-component and y-component of the Celestial Intermediate Pole unit vector (CIP), and s is called the Celestial Intermediate Origin (CIO) locator. The computation of X, Y, and s requires evaluating series expansions with thousands of terms and many trigonometric function evaluations and is computationally expensive. Vallado *et. al.* show that it is possible to precompute X, Y, and s at one day intervals and interpolate the values using ninth order Lagrange interpolation and provide values that are accurate to within the error of the theory itself. The interpolation of X, Y, and s is over two orders of magnitude faster than the series evaluation. Interpolation of data at 1 day intervals is possible because all physical affects with a period of two days or less are not included in the theory for X, Y, s and are accounted for in the daily observations provided by the IERS in the EOP files.[6] GMAT interpolates tabulated values X, Y, s created using the IAU SOFA routines. s is computed using the routine iauS06a.c which uses the 2000A nutation model and 2006 precession model. X and Y are computed using the IAU SOFA routine called iauXy06 which also uses the the 2000A nutation model and 2006 precession model.

2.4 Transformation from ICRT to MJ2000Eq

With the inclusion of ICRF-based systems, GMAT supports two types of axis systems: (1) axis systems that are based on IAU-1976 theory (called MJ2000Eq in GMAT), and (2) axis systems based on IAU-2000 theory (called ICRF in GMAT). Rotation from ICRF to MJ2000Eq is performed using the Frame Bias matrix, **B**, given by Kaplan.[6]

$$\mathbf{B} = \begin{pmatrix} 1 - 0.5(d\alpha_0^2 + \xi_0^2) & d\alpha_0 & -\xi_0 \\ -d\alpha_0 - \eta_0\xi_0 & 1 - 0.5(d\alpha_0^2 + \eta_0^2) & -\eta_0 \\ \xi_0 - \eta_0 d\alpha_0 & \eta_0 + \zeta_0 d\alpha_0 & 1 - 0.5(\eta_0^2 + \xi_0^2) \end{pmatrix} \tag{2.28}$$

where $d\alpha_0 = -14.6$ mas, $\xi_0 = -16.6170$ mas, and $\eta_0 = -6.8192$ mas.

The rotation between any two axes, defind here as $\mathcal{A}_1$ and $\mathcal{A}_2$, are performed in three steps:

1. Convert from $\mathcal{A}_1$ to the base axes for $\mathcal{A}_1$.
2. If $\mathcal{A}_1$ and $\mathcal{A}_2$ have different base axis systems, use **B** to convert from $\mathcal{A}_1$ to the base axes of $\mathcal{A}_2$
3. Convert from the base axes for $\mathcal{A}_2$ to $\mathcal{A}_2$

2.5 The $\mathcal{F}_{J_{2k}}$ Inertial System and FK5 Reduction

It is well know that Newton's laws must be applied in an inertial system. The struggle to determine a truly inertial system has continued since Newton's time. In reality, the best we can do is approximate a truly inertial system in which to apply Newton's laws. In GMAT that system is the FK5 system, here called $\mathcal{F}_{J_{2k}}$. The $\mathcal{F}_{J_{2k}}$ system is referenced to the Earth's equator and the Earth's orbit about the sun. Because neither of these two planes are fixed in space, we must pick an epoch and define an inertial system based on the geometry at that epoch. This epoch is commonly chosen as the J2000 epoch. In this section, we present the definition of the $\mathcal{F}_{J_{2k}}$ system, and discuss the transformation from $\mathcal{F}_{J_{2k}}$ to the Earth Fixed system. This transformation is called FK5 reduction. We begin with a conceptual discussion of how the Earth's spin axis moves with respect to inertial space. We conclude this section with a presentation of the mathematical theory of FK5 reduction.

2.5.1 Overview of FK5 Reduction

The inertial system most commonly used in astrodynamics as of this writing is the FK5 system. We call this system $\mathcal{F}_{J_{2k}}$. The $\mathcal{F}_{J_{2k}}$ system is used for many calculations in GMAT. The two most important are for integrating equations of motion, and as an intermediate system for coordinate system transformation. $\mathcal{F}_{J_{2k}}$ is used throughout the astrodynamics community as the coordinate system to represent time varying data such as planetary ephemerides and planetary pole and prime meridian locations.

The rigorous mathematical definition of $\mathcal{F}_{J_{2k}}$ is complex. So, let's start with a simple qualitative explanation. The nominal z-axis of $\mathcal{F}_{J_{2k}}$ is normal to the Earth's equatorial plane. The nominal x-axis points along the line formed by the intersection of the Earth's equatorial plane and the ecliptic plane, in the direction of Aries. The nominal y-axis completes the right-handed system. Both the equatorial and ecliptic planes move slowly with respect to inertial space. The rigorous definition of FK5 uses the mean planes of the ecliptic and equator, at the J2000 epoch. We'll take a closer look at the mathematical definitions of the mean ecliptic and equator below.

FK5 reduction is the transformation that rotates a vector expressed in the $\mathcal{F}_{J_{2k}}$ system to the Earth Fixed coordinate system. To perform this transformation obviously requires an understanding of how the Earth's orientation changes with respect to $\mathcal{F}_{J_{2k}}$. The time varying orientation of the Earth is complex and is due to complicated interactions between the Earth and it's external environment and complicated internal dynamics. In fact, the dynamic orientation of the Earth is so complicated that we can't model it completely and FK5 reduction is a combination of dynamics models and empirical observations that are updated daily.

We describe the orientation of the Earth using three types of motion. The first type, including precession and nutation, describes how the Earth's principal moment of inertia moves with respect to inertial space.[4]

The motion is illustrated in Fig. 2.3. The principal moment of inertia is defined as the Celestial Ephemeris Pole, and due to the fact that Earth's mass distribution changes with time, the Celestial Ephemeris Pole is not constant with respect to the Earth's surface. Precession is the coning motion that the Celestial Ephemeris Pole makes around the ecliptic north pole. The other principal component of the motion of the Celestial Ephemeris Pole is commonly called nutation and is the oscillation in the angle between the Celestial Ephemeris Pole and the north ecliptic pole. The theory of Precession and Nutation come from dynamics models of the Earth's motion. The second type of motion is called sidereal time, and represents a rotation about the Celestial Ephemeris Pole. The sidereal time model is a combination of theory ad observation. The third motion is that of the Earth's instantaneous spin axis with respect to the Earth's surface. As, we'll see below, the Earth's spin axis is not constant with respect to the Earth's crust and it's motion is called Polar Motion. A portion of polar motion is due to complicated dynamics, and a portion is due to unmodelled errors in nutation. Polar motion is determined from observation. Now that we've had a brief introduction to precession, nutation, sidereal time, and polar motion, let's look at each in more detail.

Precession

As we mentioned above, precession is the coning motion of the Celestial Ephemeris Pole about the ecliptic north pole and is illustrated in Fig 2.3. The motion is caused by two primary effects. The first is the motion of the ecliptic plane due to the gravitational effects of the Sun, Moon, and planets on the Earth's orbit, and is called planetary precession. If the Earth's equator were fixed in inertial space, the effects of planetary precession would cause a precession of the equinox of about 12" per century and a decrease in the obliquity of the ecliptic of about 47" per century.[4] The second cause of precession is due to the gravitational attraction of the Sun and Moon on the irregular mass distribution of the Earth. This causes a change in the orientation of the Earth's equatorial plane with respect to inertial space with a smooth, long-period motion with a period of 26,000 years. The combined effects of planetary and lunisolar precession are called general precession and account for the secular and long period motion of the Celestial Ephemeris Pole (the short period motion is called nutation). The secular and long period motion is often used to define a mean equator and equinox, because it does not contain the short period motion that is modelled using nutation. Precession is modelled using three cubic equations that are shown in Section

Nutation

Nutation is the most complex motion in FK5 reduction. According to Seidelmann,[4] nutation is "the short period motion of the Earth's rotation axis with respect to a space-fixed coordinate system." Nutation is actually a superposition of motions with many different periods, the longest of which is 18.6 years and is associated with the regression of the node of the Moon's orbit. There are nutation effects due to the gravitational torque of the Sun, Moon, and planets on the Earth's irregular mass distribution. There are also Nutation effects due to the fact that Earth is not a rigid body. Nutation motion has an amplitude of about 9" and is usually represented as the sum of two components one in longitude and one in obliquity.

Nutation is modelled by separating the free and forced motion of the Earth. The forcing terms are due to torques from the Sun, Moon, and planets. The free terms are determined by observation because they are beyond our current modelling abilities. The resulting theory is a series expansion that contains coefficients and is a function of the location of the Sun, Moon, and planets. Nutation is intimately connected with polar motion, and in fact, as we'll see in a later section, errors in nutation modelling are captured in polar motion measurements. Nutation is modelled by separating the free and forced motion of the Earth. The forcing

Figure 2.3: Inertial Motion of Earth's Spin Axis

terms are due to torques from the Sun, Moon, and planets. The free terms are determined by observation because they are beyond our current modelling abilities. The resulting theory is a series expansion that contains coefficients and is a function of the location of the Sun, Moon, and planets. Nutation is intimately connected with polar motion, and in fact, as we'll see in a later section, errors in nutation modelling are captured in polar motion measurements.

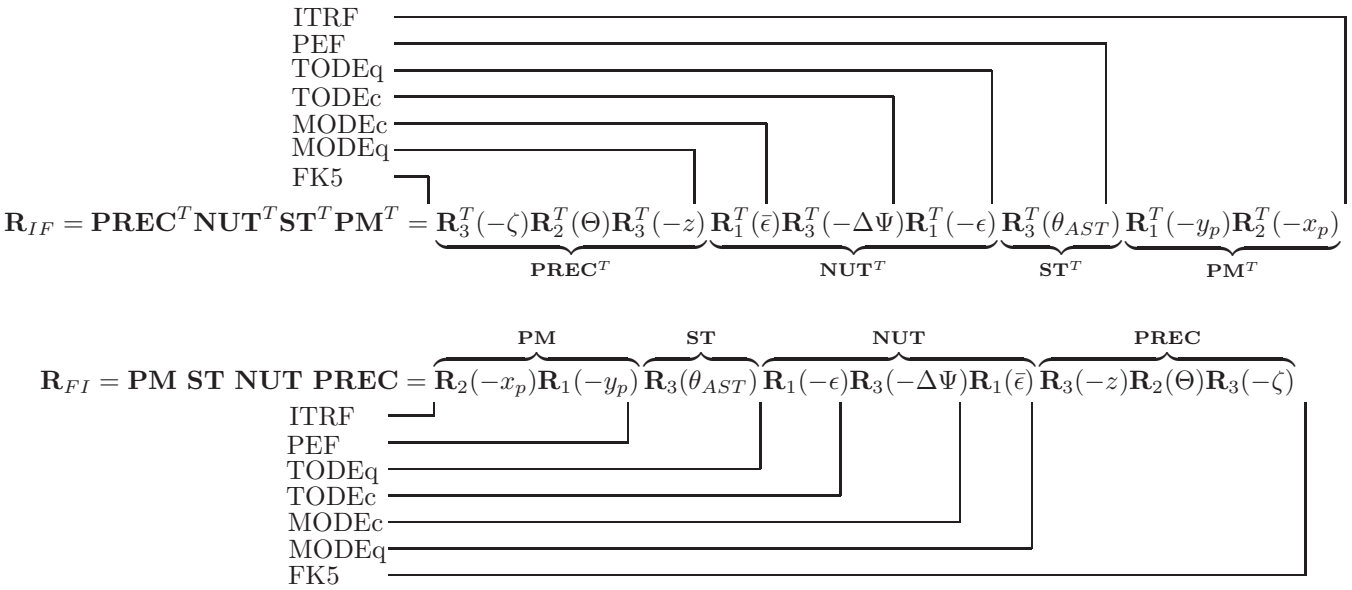


Figure 2.4: Intermediate Transformations and Coordinate Systems in FK5 Reduction

Sidereal Time

2.5.2 Precession Calculations

$$JD_{TDB} \approx JD_{TT} \tag{2.29}$$

$$T_{TDB} = \frac{JD_{TDB} - 2,451,545.0}{36525} \tag{2.30}$$

$$\zeta = 2306.2181^{"}T_{TDB} + 0.30188T_{TDB}^2 + 0.017998T_{TDB}^3 \tag{2.31}$$

$$\Theta = 2004.3109^{"}T_{TDB} - 0.42665T_{TDB}^2 - 0.041833T_{TDB}^3 \tag{2.32}$$

$$z = 2306.2181^{"}T_{TDB} + 1.09468T_{TDB}^2 + 0.018203T_{TDB}^3 \tag{2.33}$$

$$\mathbf{P} = \mathbf{R}_3(-z)\mathbf{R}_2(\Theta)\mathbf{R}_3(-\zeta) \tag{2.34}$$

2.5.3 Nutation Calculations

GMAT has the ability to use either the 1980 IAU Theory of Nutation, or the IERS 1996 Theory of Nutation. There are some calculations that are common to both, so let's look at them first. The mean obliquity of the ecliptic at the J2000 epoch, $\bar{\epsilon}$, is given by

$$\begin{aligned}\bar{\epsilon} =& 84381.448 - 46.8150T_{TDB} - 0.00059T_{TDB}^2 \\ &+ 0.001813T_{TDB}^3\end{aligned} \tag{2.35}$$

As we mentioned previously, Earth's nutation is caused by the combined gravitational effect of the Moon and Sun. So, we would expect to see the time dependent location of the Moon and Sun appear in the equations for Earth nutation. The theories of nutation described below take into account of the Moon and Sun position by modelling mean anomalies of the Moon and Sun, l and l' respectively, the mean argument of latitude of the Moon, F, the difference between the mean longitude of the Sun and Moon, D, and the

mean longitude of the ascending node of the Moon's orbit, Ω. The equations used to determine these values as a function of T_{TDB} are:

$$\begin{aligned} l =& 134.96340251^\circ + (1717915923.2178 T_{TDB} + \\ & 31.8792 T_{TDB}^2 + 0.051635 T_{TDB}^3 - 0.00024470 T_{TDB}^4)" \end{aligned} \tag{2.36}$$

$$\begin{aligned} l' =& 357.52910918^\circ + (129596581.0481 T_{TDB} - \\ & 0.5532 T_{TDB}^2 - 0.000136 T_{TDB}^3 - 0.00001149 T_{TDB}^4)" \end{aligned} \tag{2.37}$$

$$\begin{aligned} F =& 93.27209062^\circ + (1739527262.8478 T_{TDB} - \\ & 12.7512 T_{TDB}^2 + 0.001037 T_{TDB}^3 + 0.00000417 T_{TDB}^4)" \end{aligned} \tag{2.38}$$

$$\begin{aligned} D =& 297.85019547^\circ + (1602961601.2090 T_{TDB} - \\ & 6.3706 T_{TDB}^2 + 0.006593 T_{TDB}^3 - 0.00003169 T_{TDB}^4)" \end{aligned} \tag{2.39}$$

$$\begin{aligned} \Omega =& 125.04455501^\circ + (-6962890.2665 T_{TDB} + \\ & 7.4722 T_{TDB}^2 + 0.007702 T_{TDB}^3 - 0.00005939 T_{TDB}^4)" \end{aligned} \tag{2.40}$$

1980 Nutation Theory

$$\Delta\Psi_{1980} = \sum_{i=1}^{106} (A_i + B_i T_{TDB}) \sin a_p \tag{2.41}$$

$$\Delta\epsilon_{1980} = \sum_{i=1}^{106} (C_i + D_i T_{TDB}) \cos a_p \tag{2.42}$$

$$a_p = a_{1i} l + a_{2i} l' + a_{3i} F + a_{4i} D + a_{5i} \Omega \tag{2.43}$$

$$\mathbf{N} = \mathbf{R}_1(-\epsilon)\mathbf{R}_3(-\Delta\Psi)\mathbf{R}_1(\bar{\epsilon}) \tag{2.44}$$

1996 Nutation Theory

The 1996 theory of nutation published by the IERS is a higher fidelity model of Earth nutation. There are two primary differences between the 1908 IAU theory and the 1996 IERS theory. The first difference is the 1996 theory uses a 263 term series expansion for the effects of the Earth and Moon. The 1980 theory uses a 106 term series. The second difference is that the 1996 theory has a second series expansion to account for the effects of nutation caused by the more massive planets. The planetary series expansion is a 118 term series. Let's begin with the equations for the Earth and Moon's effect on Earth nutation, according to the 1996 IERS theory:

$$\Delta\Psi_{1996} = \sum_{i=1}^{263} (A_i + B_i T_{TDB}) \sin a_p + E_i \cos a_p \tag{2.45}$$

$$\Delta\epsilon_{1996} = \sum_{i=1}^{263} (C_i + D_i T_{TDB}) \cos a_p + F_i \sin a_p \tag{2.46}$$

$$a_p = a_{1i} M_\circ + a_{2i} M_\odot + a_{3i} u_{M\circ} + a_{4i} D_\odot + a_{5i} \Omega_\circ \tag{2.47}$$

To calculate the planetary effects on nutation, we begin by calculating the mean heliocentric longitude of the planets. Only the effects of Venus(V), Mars(M), JupiterJ), and Saturn(S) are included in the theory. We require the Earth's (E) mean longitude also. The mean longitudes are calculated using:

$$\begin{aligned} \lambda_V &= 181.979800853^\circ + 58,517.8156748 T_{TDB} \\ \lambda_E &= 100.466448494^\circ + 35,999.3728521 T_{TDB} \\ \lambda_M &= 355.433274605^\circ + 19,140.299314 T_{TDB} \\ \lambda_J &= 34.351483900^\circ + 3,034.90567464 T_{TDB} \\ \lambda_S &= 50.0774713998^\circ + 1,222.11379404 T_{TDB} \end{aligned}$$

The general precession in longitude, p_a, is calculated using

$$p_a = 1.39697137214^{\circ} T_{TDB} + 0.0003086 T_{TDB}^2$$

Finally, the planetary terms are calculated using:

$$\Delta\Psi_{pl} = \sum_{i=1}^{118} (A_i + B_i T_{TDB}) \sin a_{pl} \tag{2.48}$$

$$\Delta\epsilon_{pl} = \sum_{i=1}^{118} (C_i + D_i T_{TDB}) \cos a_{pl} \tag{2.49}$$

$$\begin{aligned} a_{pl} = \quad & a_{1i}\lambda_V + a_{2i}\lambda_E + a_{3i}\lambda_M + a_{4i}\lambda_J + a_{5i}\lambda_S + \\ & a_{6i}p_a + a_{7i}D + a_{8i}F + a_{9i}l + a_{10i}\Omega \end{aligned} \tag{2.50}$$

$$\Delta\Psi = \Delta\Psi_{1996} + \underbrace{\Delta\Psi_{pl}}_{optional} \tag{2.51}$$

$$\Delta\epsilon = \Delta\epsilon_{1996} + \underbrace{\Delta\epsilon_{pl}}_{optional} \tag{2.52}$$

$$\epsilon = \bar{\epsilon} + \Delta\epsilon \tag{2.53}$$

In GMAT, the planetary terms are optional. If the user has selected to include the planetary terms, the

$$\mathbf{N} = \mathbf{R}_1(-\epsilon)\mathbf{R}_3(-\Delta\Psi)\mathbf{R}_1(\bar{\epsilon}) \tag{2.54}$$

2.5.4 Sidereal Time Calculations

To calculate the sidereal time of the Earth, we need the current time which is then used to determine the Greenwich Mean Sidereal Time (GMST) and the equation of the equinoxes. GMST is calculated using:

$$\begin{aligned} \theta_{GMST} = & 1.00965822615e6 + 4.746600277219299e10 T_{UT1} \\ & + 1.396560 T_{UT1}^2 + 9.3e-5 T_{UT1}^3 \quad \text{(arcseconds)} \end{aligned} \tag{2.55}$$

The calculation of the equation of the equinoxes is dependent upon the time. If the Julian date falls after 2450449.5, then we use

$$EQ_{equinox} = \Delta\Psi \cos\epsilon + 0.00264^{"} \sin\Omega + 0.000063^{"} \sin 2\Omega \tag{2.56}$$

If the Julian date falls on or before 2450449.5 we use

$$EQ_{equinox} = \Delta\Psi \cos\epsilon \tag{2.57}$$

$$\theta_{AST} = \theta_{GMST} + EQ_{equinox} \tag{2.58}$$

$$\mathbf{ST} = \mathbf{R}_3(\theta_{AST}) \tag{2.59}$$

2.5.5 Polar Motion Calculations

$$\mathbf{PM} = \mathbf{R}_2(-x_p)\mathbf{R}_1(-y_p) \tag{2.60}$$

2.6 Deriving $\mathbf{R}_{J_{2k},i}$ and $\dot{\mathbf{R}}_{J_{2k},i}$ for Various Coordinate Systems

In GMAT, there are numerous coordinate systems that can be used to define variables, stopping conditions, spacecraft states, and other quantities. Some examples include the Earth centered mean ecliptic of J2000 system, the Earth-fixed system, the Mars equator system, and the Earth-Moon rotating system.

In the following subsections, we determine how $\mathbf{R}_{J_{2k},i}$ and $\dot{\mathbf{R}}_{J_{2k},i}$ are calculated in GMAT for all of the coordinate systems available in GMAT. Let's begin by looking at coordinate systems defined by the equator of a celestial body.

2.6.1 Equator System

The Equator axis system has the following nominal configuration:

- x-axis: Along the line formed by the intersection of the bodies equator and the ecliptic plane.
- y-axis: Completes the right-handed set.
- z-axis: Normal to the equatorial plane.

The Equator system in GMAT is a true equator of date axis system. The equatorial coordinate system is defined only for celestial bodies. For a particular body, the equatorial system is defined by the bodies equatorial plane and its intersection with the ecliptic plane, at the current epoch. The Earth and Moon have highly accurate models for their equatorial systems and and are treated at the end of this section. For the remaining bodies in the solar system, the equatorial coordinate system is calculated in GMAT using data published by the International Astronomical Union (IAU).[1] The IAU publishes data that gives the spin axis direction and prime meridian location of all the planets and major moons as a function of time. For the Earth, GMAT uses FK5 reduction for the Equator system. For the Moon, GMAT can use either the IAU data, or Euler angles provided in the JPL DE405 files.

Let's look more closely at the data provided by the IAU. Figure 2.5 contains an illustration of the three variables, α_o, δ_o, and W, that are used to define a body's spin axis and prime meridian location w/r/t MJ2000Eq. α_o and δ_o are used to define a body's spin axis direction. W is the body's sidereal time. The equations for α_o, δ_o, and W for the nine planets and the Earth's moon are found in Tables 2.1 and 2.2. From

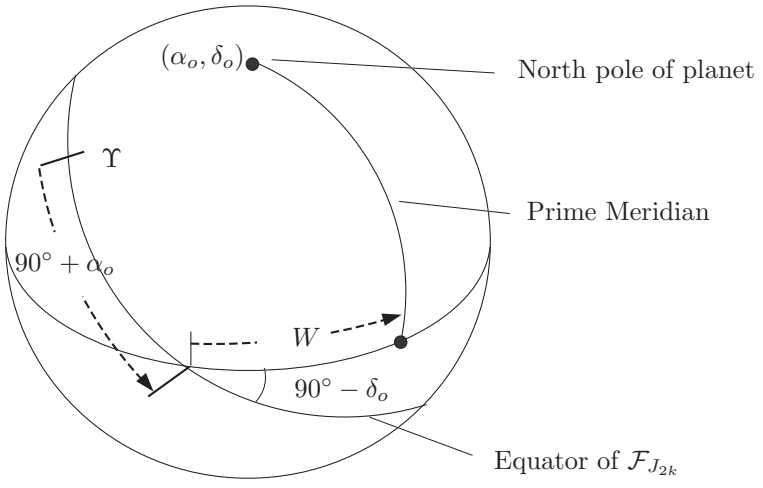


Figure 2.5: IAU Definition of Pole and Meridian Locations for Planets and Moons

inspection of Fig. 2.5 we see that

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3^T(90° + \alpha_o)\mathbf{R}_1^T(90° - \delta_o) \tag{2.61}$$

α_o and δ_o vary slowly with time, so we can assume the derivative of $\mathbf{R}_{Ii}$ for the Equator system is the zero matrix.

$$\dot{\mathbf{R}}_{J_{2k},i} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.62}$$

If the user chooses to use the DE405 files to determine the Moon's orientation, then GMAT gets a set of Euler angles and rates from the DE405 files. We then use the following equations to determine $\mathbf{R}_{J_{2k},i}$ and $\dot{\mathbf{R}}_{J_{2k},i}$.

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3(\theta_1)^T\mathbf{R}_1(\theta_2)^T \tag{2.63}$$

$$\dot{\mathbf{R}}_{J_{2k},i} = \mathbf{R}_3(\theta_1)^T\dot{\mathbf{R}}_1^T(\theta_2) + \dot{\mathbf{R}}_3^T(\theta_1)\mathbf{R}_1^T(\theta_2) \tag{2.64}$$

where

$$\dot{\mathbf{R}}_1(\theta_2) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & -\dot{\theta}_2\sin\theta_2 & \dot{\theta}_2\cos\theta_2 \\ 0.0 & -\dot{\theta}_2\cos\theta_2 & -\dot{\theta}_2\sin\theta_2 \end{pmatrix} \tag{2.65}$$

and

$$\dot{\mathbf{R}}_3(\theta_1) = \begin{pmatrix} -\dot{\theta}_1\sin\theta_1 & \dot{\theta}_1\cos\theta_1 & 0.0 \\ -\dot{\theta}_1\cos\theta_1 & -\dot{\theta}_1\sin\theta_1 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.66}$$

Finally, for the Earth, the Equator axis system a true of date equator system and is calculated using the algorithm described in Sec. 2.6.5.

2.6.2 MJ2000 Ecliptic (MJ2000Ec)

The MJ2000 Ecliptic axis system is defined as follows:

- x-axis: Along the line formed by the intersection of the Earth's mean equator and the mean ecliptic plane, at the J2000 epoch. The axis points in the direction of the first point of Aries.
- y-axis: Completes the right-handed set.
- z-axis: Normal to the Earth's mean equatorial plane at the J2000 Epoch.

The matrix to rotate from MJ2000 Ecliptic (MJ2000Ec) to MJ2000 Equatorial (MJ2000Eq) is a rotation about the x-axis through the obliquity of the ecliptic at the J2000 epoch which is 23.439291°:

$$\mathbf{R} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.91748206 & -0.397777156 \\ 0.0 & 0.39777716 & 0.9174820621 \end{pmatrix} \tag{2.67}$$

GMAT uses more significant digits than included here. The rotation matrix is constant by definition so its time derivative is identically the zero matrix.

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.68}$$

2.6.3 True of Epoch Equator (TOEEq)

The True of Epoch Equator axis system is defined as follows:

- x-axis: Along the true equinox at the chosen epoch. The axis points in the direction of the first point of Aries.
- y-axis: Completes the right-handed set.

- z-axis: Normal to the Earth's true equatorial plane at the chosen Epoch.

The TOEEq axis system is an intermediate system in FK5 reduction. $\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the TOEEq system are calculated using the following equations

$$\mathbf{R}_{Ii} = \mathbf{N}^T(t_o)\mathbf{P}^T(t_o) \tag{2.69}$$

where t_o is the epoch defined in the coordinate system description provided by the user in the epoch field. Hence, t_o is a constant value for the TOEEq system. For a given t_o, the matrices associated with the TOEEq system only need to be evaluated once and can be reused later when necessary. $\mathbf{P}(t_o)$ and $\mathbf{N}(t_o)$ are part of the FK5 reduction algorithm and are explained in detail in Sec. 2.5.3 and 2.5.2 . Because t_o is fixed for a TOEEq system, the derivative of $\mathbf{R}_{Ii}$ is identically equal to the zero matrix.

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.70}$$

2.6.4 Mean of Epoch Equator (MOEEq)

The Mean of Epoch Equator axis system is defined as follows:

- x-axis: Along the mean equinox at the chosen epoch. The axis points in the direction of the first point of Aries.
- y-axis: Completes the right-handed set.
- z-axis: Normal to the Earth's mean equatorial plane at the chosen Epoch.

The MOEEq is an intermediate system in FK5 reduction and $\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the MOEEq system can be calculated using the following equations

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o) \tag{2.71}$$

where t_o is the epoch defined in the coordinate system description provided by the user in the epoch field. Hence t_o is a constant value for the MOEEq system. For a given t_o, the matrices associated with the MOEEq system only need to be evaluated once and can be reused later when necessary. $\mathbf{P}(t_o)$ is described in Sec. 2.5.2. Because t_o is fixed for a MOEEq system, the derivative of $\mathbf{R}_{Ii}$ is the zero matrix.

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.72}$$

2.6.5 True of Date Equator (TODEq)

$\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the TODEq system can be calculated using the following equations Vallado[3] Fig. 3-29).

$$\mathbf{R}_{Ii} = \mathbf{N}^T(t_o)\mathbf{P}^T(t_o) \tag{2.73}$$

where t_o is the epoch. Unlike the TOEEq sytem, for the TODEq system t_o is a variable and usually comes from the epoch of the object whose state we wish to convert. $\mathbf{P}(t_o)$ and $\mathbf{N}(t_o)$ are part of the FK5 reduction algorithm and can be found in Vallado pgs. 214 - 219. Because t_o is not fixed for a TODEq system the derivative of $\mathbf{R}_{Ii}$ is non-zero. However, we will assume its derivative is negligibly small so that

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.74}$$

2.6.6 Mean of Date Equator (MODEq)

$\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the MODEq system can be calculated using the following equations

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o) \tag{2.75}$$

where t_o is the epoch. Unlike the MOEEq sytem, for the MODEq system t_o is a variable and usually comes from the epoch of the object whose state we wish to convert. $\mathbf{P}(t_o)$ and $\mathbf{N}(t_o)$ are part of the FK5 reduction algorithm and can be found in Vallado[3] pgs. 214 - 219. Because t_o is not fixed for a MODEq system, the derivative of $\mathbf{R}_{Ii}$ is non-zero. However, we will assume its derivative is negligibly small so that

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.76}$$

2.6.7 Mean of Date Ecliptic (MODEc)

$\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the MODEc system can be calculated using the following equations

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o)\mathbf{R}_1^T(\bar{\epsilon}) \tag{2.77}$$

where t_o is the epoch. For the MODEc system t_o is a variable and usually comes from the epoch of the object whose state we wish to convert. $\mathbf{P}(t_o)$ comes from the FK5 reduction algorithm and can be found in Vallado[3] pgs. 214 - 219. $\bar{\epsilon}$ is given by Vallado,[3] Eq. (3-52). For a more complete discussion, you can also refer to Seidelmann,[4] pgs. 114 - 115. Because t_o is not fixed for a MODEc system, the derivative of $\mathbf{R}_{Ii}$ is non-zero. However, we will assume its derivative is negligibly small so that

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.78}$$

2.6.8 True of Date Ecliptic (TODEc)

$\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$ for the TODEc system can be calculated using the following equations

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o)\mathbf{R}_1^T(\bar{\epsilon})\mathbf{R}_3^T(-\Delta\Psi) \tag{2.79}$$

where t_o is the epoch. Unlike the TOEEc sytem, for the TODEc system t_o is a variable and usually comes from the epoch of the object whose state we wish to convert. $\mathbf{P}(t_o)$is part of the FK5 reduction algorithm and can be found in Vallado pgs. 214 - 219. $\bar{\epsilon}$ is given by Vallado,[3] Eq. (3-52). $\Delta\Psi$ is given by Eq.(3-62) in Vallado.[3] For a more complete discussion, you can also refer to Seidelmann,[4] pgs. 114 - 115. Because t_o is not fixed for a MODEq system, the derivative of $\mathbf{R}_{Ii}$ is non-zero. However, we will assume its derivative is negligibly small so that

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.80}$$

2.6.9 Topocentric Horizon

The Topocentric coordinate system has its origin at a ground station with the following axes definitions

- x-axis: Completes the right handed set and points south in the local horizon system.
- y-axis: Points due East.
- z-axis: Normal to the local horizon. The local horizon is defined by the selection of the `HorizonReference` which is either `Sphere` or `Ellipsoid`.

The calculation of the Topocentric to MJ2000Eq systems is different for the different ground system state representations. Regardless of how the user inputs the state of a ground station, the Topocentric axes are always defined with respect to the reference shape of the central body. GMAT uses the appropriate transformation to convert the user input to the cartesian location of the ground station expressed in the local body-fixed frame. This cartesian representation is used to calculate the topocentric axes.

Define the axes of the topocentric coordinate system, expressed in the body fixed system as $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$. Define, x_F, y_F, and z_F as the location of the station in the local body-fixed coordinate system. Calculate the geocentric latitude to use as an initial guess to find the geodetic latitude

$$\phi_{gd} \approx \text{atan2}(z_F, r_{xy}); \tag{2.81}$$

The eccentricity of the body is calculated from the flatting f using

$$e^2 = 2f - f^2 \tag{2.82}$$

Set $\delta = 1.0$ to initialize the loop, then,
While ($\delta > 10^{-11}$)

$$\begin{aligned} \phi' &= \phi_{gd} && (2.83)\\ C &= \frac{R}{\sqrt{1 - e^2 \sin^2 \phi}} && (2.84)\\ \phi_{gd} &= \text{atan}\left(\frac{z_F + Ce^2 \sin\phi}{r_{xy}}\right) && (2.85)\\ \delta &= |\phi_{gd} - \phi'| && (2.86) \end{aligned}$$

EndWhile

The longitude of the station location, λ, is calculated using

$$\lambda = \text{atan2}(y_F, x_F) \tag{2.87}$$

Finally,

$$\hat{\mathbf{z}} = \begin{pmatrix} \cos\phi_{gd}\cos\lambda \\ \cos\phi_{gd}\sin\lambda \\ \sin\phi_{gd} \end{pmatrix} \tag{2.88}$$

$$\hat{\mathbf{y}} = \hat{\mathbf{k}} \times \hat{\mathbf{z}} \tag{2.89}$$

where

$$\hat{\mathbf{k}} = [0\ 0\ 1]^T \tag{2.90}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}} \tag{2.91}$$

$$\mathbf{R}_{FT} = [\hat{\mathbf{x}}\ \hat{\mathbf{y}}\ \hat{\mathbf{z}}] \tag{2.92}$$

The last step is to determine the rotation matrix from the topocenric system to the inertial system by using the rotation matrix from body fixed to inertial, $\mathbf{R}_{IF}$, at the desired epoch, t. We determine the body fixed rotation matrix using the algorithm in Sec. 3.1.9. Once we have evaluated $\mathbf{R}_{IF}$ and $\dot{\mathbf{R}}_{IF}$ we calculate $\mathbf{R}_{IT}$ and $\dot{\mathbf{R}}_{IT}$ using

$$\begin{aligned} \mathbf{R}_{IT} &= \mathbf{R}_{IF}\mathbf{R}_{FT} && (2.93)\\ \dot{\mathbf{R}}_{IT} &= \dot{\mathbf{R}}_{IF}\mathbf{R}_{FT} && (2.94) \end{aligned}$$

2.6.10 Celestial Body Fixed

The body fixed coordinate system is referenced to the body equator and the prime meridian of the body. The body fixed system for Earth is found by using FK5 reduction to the ITRF system as described by Vallado. The ITRF system is the earth fixed system.

Vallado denotes the four rotation sequences required to transform from the ITRF to the FK5 system as [PM], the polar motion, [ST], the sidereal time, [NUT], the nutation, and [PREC], the precession. GMAT calculates these four rotation matrices as described in Vallado. The rotation matrix from ITRF to FK5 can be written as follows.

$$\mathbf{R}_{Ii} = \mathbf{P}^T\mathbf{N}^T\mathbf{S}\mathbf{T}^T\mathbf{P}\mathbf{M}^T \tag{2.95}$$

GMAT assumes that the only intermediate rotation that has a significant time derivative is the sidereal time, [ST]. So, we can write

$$\dot{\mathbf{R}}_{Ii} = \mathbf{P}^T\mathbf{N}^T\dot{\mathbf{S}\mathbf{T}}^T\mathbf{P}\mathbf{M}^T \tag{2.96}$$

where $\dot{\mathbf{S}\mathbf{T}}$ is given by

$$\dot{\mathbf{S}\mathbf{T}} = \begin{pmatrix} -\omega_e \sin\theta_{AST} & \omega_e \cos\theta_{AST} & 0.0 \\ -\omega_e \cos\theta_{AST} & -\omega_e \sin\theta_{AST} & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.97}$$

and ω_e is given by

$$\omega_e = 7.29211514670698e^{-5}\left(1 - \frac{LOD}{86400}\right) \tag{2.98}$$

Note that the 2nd edition of Vallado[3] has inconsistencies in Eqs. (2.95) and (2.96) and they are discussed in the errata to the 2nd edition. We have modified equations Eqs. (2.95) and (2.96) according to the errata.

For bodies other than the earth, the IAU gives the spin axis direction as a function of time with respect to the MJ2000Eq system and rotation of the prime meridian in the MJ2000Eq system. This data for all of the planets and many moons can be found in "Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000" Seidelmann[1] *et.al.* Figure 1 in this document explains the three variables, α_o, δ_o, and W, that are used to define the body spin axis and prime meridian location w/r/t J2000. The values of α_o, δ_o, and W for the nine planets and the Earth's moon are found on pgs. 8 and 9.

Using the notation found in the reference we can write

$$\mathbf{R}_{Ii} = \mathbf{R}_3^T(90^\circ + \alpha_o)\mathbf{R}_1^T(90^\circ - \delta_o)\mathbf{R}_3^T(W) \tag{2.99}$$

For the derivative we assume that

$$\frac{d}{dt}\left(\mathbf{R}_3^T(90^\circ + \alpha_o)\right) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.100}$$

and

$$\frac{d}{dt}\left(\mathbf{R}_1^T(90^\circ - \delta_o)\right) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.101}$$

$$\frac{d}{dt}\left(\mathbf{R}_3^T(W)\right) = \begin{pmatrix} -\dot{W}\sin(W) & -\dot{W}\cos(W) & 0.0 \\ \dot{W}\cos(W) & -\dot{W}\sin(W) & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.102}$$

where $\dot{W}$ is the time derivative of W for the given body. Note, Seidelmann[1] does not provide the values for $\dot{W}$ so we include them in Table 2.1. In summary

$$\dot{\mathbf{R}} = \mathbf{R}_3^T(90^\circ + \alpha_o)\mathbf{R}_1^T(90^\circ - \delta_o)\frac{d}{dt}\mathbf{R}_3^T(W) \tag{2.103}$$

2.6.11 Body Inertial

The `BodyInertial` axis system is an inertial system based on the equator of the celestial body chosen as the origin of the system. The origin of a `BodyInertial` system must be a celestial body, and cannot be a spacecraft, libration point etc. The axes are defined as follows (except for Earth):

- x-axis: Along the line formed by the intersection of the bodies equator and the x-y plane of the FK5 system, at the J2000 epoch.
- y-axis: Completes the right-handed set.
- z-axis: Along the bodies instantaneous spin axis direction at the J2000 epoch.

For Earth, the `BodyInertial` axis system is the FK5 system. For all other bodies, the `BodyInertial` axis system is based upon the bodies equator and spin axis at the J2000 epoch. So, `BodyInertial` is essentially a true-of-epoch system referenced to the chosen central body. The body orientation at the J2000 epoch is calculated from the IAU Data in Seidelmann[1] for the Sun, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto. For the Moon, the orientation at the J2000 epoch comes from the DE405 files. Because the `BodyInertial` system is an inertial system, the derivative of the rotation matrix is always zero:

$$\dot{\mathbf{R}}_{Ii} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{2.104}$$

The rotation matrix, $\mathbf{R}_{Ii}$, is different for each celestial body. We begin by calculating the angles α and δ used to define the bodies orientation with respect to the FK5 system:

$$\alpha = \alpha_o(T_{J2000}) \tag{2.105}$$

$$\delta = \delta_o(T_{J2000}) \tag{2.106}$$

Where $T_{J2000} = 2451544.9999999990686$ TDB and the equations for α_o and δ_o are given by Seidelmann[1] and reproduced in Table 2.1. Finally, the rotation matrix is calculated using

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3^T(90^\circ + \alpha)\mathbf{R}_1^T(90^\circ - \delta) \tag{2.107}$$

The result is a rotation matrix, that is time invariant, for each celestial body.

2.6.12 Object Referenced

An object referenced system is a coordinate system whose axes are defined by the motion of one object with respect to another object. GMAT allows the user to define many different types of Object Referenced system. In Fig. 2.6 we see a diagram that defines the directions a user can choose from in creating an Object Referenced coordinate system. There are six directions. One is the relative position, denoted here by $\mathbf{r}$, of the secondary object with respect to the primary object, expressed in an inertial frame. The second is the the relative velocity, denoted here by $\mathbf{v}$, of the secondary object with respect to the primary, expressed in an inertial frame. The third direction is the vector normal to the direction of motion which is denoted by $\mathbf{n}$ and is calculated from $\mathbf{n} = \mathbf{r} \times \mathbf{v}$. The remaining three directions are the negative of of the first three.

In GMAT, a user can use the directions described above to define an Object Referenced coordinate system. In doing so, the user can choose two of the available directions, and define which of the three axes, x, y, and z, they desire the direction to be aligned with. Given this information, GMAT automatically constructs an orthogonal coordinate system. For example, if user chooses the x-axis to be in the direction of $\mathbf{r}$ and the z-axis to be in the direction of $\mathbf{n}$, the GMAT completes the right-handed set by setting the y-axis in the direction of $\mathbf{n} \times \mathbf{r}$. Obviously there are some choices that not yield an orthogonal system, or that yield a left handed system. GMAT does not allow the user to select these pairs of axes and throws an error message.

In general, given the unit vectors that define the axes system of the Object Referenced system, but expressed in the inertial frame, GMAT uses the following equations to determine $\mathbf{R}_{Ii}$ and $\dot{\mathbf{R}}_{Ii}$.

$$\mathbf{R}_{Ii} = [\ \hat{\mathbf{x}} \quad \hat{\mathbf{y}} \quad \hat{\mathbf{z}}\ ] \tag{2.108}$$

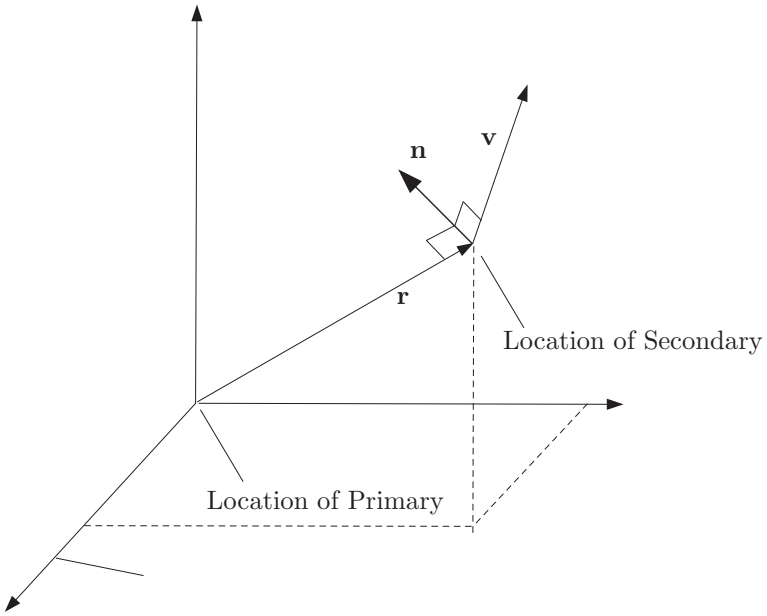


Figure 2.6: Diagram of an Object Referenced Coordinate System

$$\dot{\mathbf{R}}_{Ii} = \begin{bmatrix} \dot{\hat{\mathbf{x}}} & \dot{\hat{\mathbf{y}}} & \dot{\hat{\mathbf{z}}} \end{bmatrix} \tag{2.109}$$

Recall that the user chooses two axes to an Object Referenced system among the following choices: $\hat{\mathbf{r}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{n}}$, $-\hat{\mathbf{r}}$, $-\hat{\mathbf{v}}$, and $-\hat{\mathbf{n}}$. In general, one of the axes chosen by the user must be either $\hat{\mathbf{n}}$, or $-\hat{\mathbf{n}}$.

If the user defines the x-axis and y-axis then GMAT determines the z axis using

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \tag{2.110}$$

and

$$\dot{\hat{\mathbf{z}}} = \dot{\hat{\mathbf{x}}} \times \hat{\mathbf{y}} + \hat{\mathbf{x}} \times \dot{\hat{\mathbf{y}}} \tag{2.111}$$

If the user defines the y-axis and z-axis, then GMAT determines the x axis using

$$\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}} \tag{2.112}$$

and

$$\dot{\hat{\mathbf{x}}} = \dot{\hat{\mathbf{y}}} \times \hat{\mathbf{z}} + \hat{\mathbf{y}} \times \dot{\hat{\mathbf{z}}} \tag{2.113}$$

And finally, if the user defines the x-axis and z-axis then GMAT determines the y axis using

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{2.114}$$

and

$$\dot{\hat{\mathbf{y}}} = \dot{\hat{\mathbf{z}}} \times \hat{\mathbf{x}} + \hat{\mathbf{z}} \times \dot{\hat{\mathbf{x}}} \tag{2.115}$$

Depending on the users choice of axes for an Object Referenced coordinate system, GMAT will need to calculate $\hat{\mathbf{r}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{n}}$, $\dot{\hat{\mathbf{r}}}$, $\dot{\hat{\mathbf{v}}}$, and $\dot{\hat{\mathbf{n}}}$. These are given by:

$$\hat{\mathbf{r}} = \frac{\mathbf{r}}{\|\mathbf{r}\|} = \frac{\mathbf{r}}{r} \tag{2.116}$$

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{\mathbf{v}}{v} \tag{2.117}$$

$$\hat{\mathbf{n}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \tag{2.118}$$

$$\dot{\hat{\mathbf{r}}} = \frac{\mathbf{v}}{r} - \frac{\hat{\mathbf{r}}}{r}\left(\hat{\mathbf{r}} \cdot \mathbf{v}\right) \tag{2.119}$$

$$\dot{\hat{\mathbf{v}}} = \frac{\mathbf{a}}{v} - \frac{\hat{\mathbf{v}}}{v}\left(\hat{\mathbf{v}} \cdot \mathbf{a}\right) \tag{2.120}$$

$$\dot{\hat{\mathbf{n}}} = \frac{\mathbf{r} \times \mathbf{a}}{n} - \frac{\hat{\mathbf{n}}}{n}\left(\mathbf{r} \times \mathbf{a} \cdot \hat{\mathbf{n}}\right) \tag{2.121}$$

2.6.13 Geocentric Solar Ecliptic (GSE)

The Geocentric Solar Ecliptic system is a time varying axis system often used to describe and analyze the Earth's magnetic field. The coordinate system is defined such that

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_{sun}}{\|\mathbf{r}_{sun}\|} \tag{2.122}$$

where $\mathbf{r}_{sun}$ is the vector from the Earth to the Sun in the MJ2000Eq axis system. The z-axis is defined to be the ecliptic pole. To ensure we have an orthogonal system, we calculate $\hat{\mathbf{z}}$ using

$$\hat{\mathbf{z}} = \frac{\mathbf{r}_{sun} \times \mathbf{v}_{sun}}{\|\mathbf{r}_{sun} \times \mathbf{v}_{sun}\|} \tag{2.123}$$

Finally, the y-axis completes the right-handed set

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{2.124}$$

We can construct the rotation matrix that goes from the GSE axis system to the MJ2000Eq axis system as

$$\mathbf{R}_{Ii} = \left[\begin{array}{ccc} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{array}\right] \tag{2.125}$$

We also need to compute the derivative of the rotation matrix. We start by computing

$$\frac{d\hat{\mathbf{x}}}{dt} = \frac{\mathbf{v}_{sun}}{r_{sun}} - \hat{\mathbf{x}}\left(\hat{\mathbf{x}} \cdot \frac{\mathbf{v}_{sun}}{r_{sun}}\right) \tag{2.126}$$

where $\mathbf{v}_{sun}$ is the velocity of the Sun with respect to the Earth in the MJ2000Eq system. We can approximate the derivative of the z axis using

$$\frac{d\hat{\mathbf{z}}}{dt} \approx \mathbf{0} \tag{2.127}$$

$$\frac{d\hat{\mathbf{y}}}{dt} = \hat{\mathbf{z}} \times \frac{d\hat{\mathbf{x}}}{dt} \tag{2.128}$$

$$\dot{\mathbf{R}}_{Ii} = \left[\begin{array}{ccc} \dfrac{d\hat{\mathbf{x}}}{dt} & \dfrac{d\hat{\mathbf{y}}}{dt} & \dfrac{d\hat{\mathbf{z}}}{dt} \end{array}\right] \tag{2.129}$$

2.6.14 Geocentric Solar Magnetic (GSM)

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_{sun}}{\|\mathbf{r}_{sun}\|} \tag{2.130}$$

Let's define the spherical coordinates of the Earth's dipole in the Earth fixed frame to be λ_d and ϕ_d. The location of the dipole actually changes with time. Also, the dipole does not actually pass through the center of the Earth. However, GMAT currently assumes that the dipole direction is constant, and passes directly through the center of the Earth. If this approximation is not sufficient for future studies, the model will have to be updated.

$$\lambda_d = 70.1^\circ \text{ W} \tag{2.131}$$

$$\phi_d = 78.6^\circ \text{ N} \tag{2.132}$$

The dipole vector in the Earth Fixed system is simply

$$\{\mathbf{r}_d\}_F = \begin{pmatrix} \cos\phi_d \cos(-\lambda_d) \\ \cos\phi_d \sin(-\lambda_d) \\ \sin\phi_d \end{pmatrix} \tag{2.133}$$

If R_{IF} is the rotation matrix from the Earth Fixed frame to MJ2000Eq at the current epoch, then we can write the vector that describes the dipole direction in the inertial frame as

$$\{\mathbf{r}_d\}_I = R_{IF}\{\mathbf{r}_d\}_F \tag{2.134}$$

Then, the y-axis is defined as

$$\hat{\mathbf{y}} = \frac{\{\mathbf{r}_d\}_I \times \hat{\mathbf{x}}}{\|\{\mathbf{r}_d\}_I \times \hat{\mathbf{x}}\|} \tag{2.135}$$

the z-axis is defined as

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \tag{2.136}$$

and

$$\mathbf{R}_{Ii} = [\ \hat{\mathbf{x}} \quad \hat{\mathbf{y}} \quad \hat{\mathbf{z}}\ ] \tag{2.137}$$

To calculate the derivative of the rotation matrix, we know that

$$\frac{d\hat{\mathbf{x}}}{dt} = \frac{\mathbf{v}_{sun}}{r_{sun}} - \hat{\mathbf{x}}\left(\hat{\mathbf{x}} \cdot \frac{\mathbf{v}_{sun}}{r_{sun}}\right) \tag{2.138}$$

Let's define

$$\mathbf{y} = (R_{IF}\{\mathbf{r}_d\}_F) \times \hat{\mathbf{x}} \tag{2.139}$$

and

$$y = \|(R_{IF}\{\mathbf{r}_d\}_F) \times \hat{\mathbf{x}}\| \tag{2.140}$$

then

$$\frac{d\mathbf{y}}{dt} = \dot{\mathbf{y}} = \left(\dot{R}_{IF}\{\mathbf{r}_d\}_F\right) \times \hat{\mathbf{x}} + (R_{IF}\{\mathbf{r}_d\}_F) \times \frac{\hat{\mathbf{x}}}{dt} \tag{2.141}$$

Now we can write

$$\frac{d\hat{\mathbf{y}}}{dt} = \dot{\hat{\mathbf{y}}} = \frac{\dot{\mathbf{y}}}{y} - \hat{\mathbf{y}}\left(\hat{\mathbf{y}} \cdot \frac{\dot{\mathbf{y}}}{y}\right) \tag{2.142}$$

Finally,

$$\dot{\hat{\mathbf{z}}} = \dot{\hat{\mathbf{x}}} \times \hat{\mathbf{y}} + \hat{\mathbf{x}} \times \dot{\hat{\mathbf{y}}} \tag{2.143}$$

and

$$\dot{\mathbf{R}}_{Ii} = \left[\ \frac{d\hat{\mathbf{x}}}{dt} \quad \frac{d\hat{\mathbf{y}}}{dt} \quad \frac{d\hat{\mathbf{z}}}{dt}\ \right] \tag{2.144}$$

2.6.15 Body-Spin Sun Coordinates

The body-spin Sun coordinate system is a celestial-body-based coordinate system defined as follows:

- x points from the central body to the sun
- y completes the right-handed set
- z lies in the plane of the body's spin axis and the x axis

This system is similar to the GSM system with the following two differences: (1) The magnetic field vector is replaced with the body's spin axis, and (2) the system is based on the fixed frame of the central body and not always referenced to the Earth fixed system, and (3) x points in the opposite direction in the two systems.

Define the vector $\mathbf{r}_{sun}$ as the vector from the central body to the sun. Then,

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_{sun}}{\|\mathbf{r}_{sun}\|} \tag{2.145}$$

Define the body's spin axis in the body fixed system as

$$\mathbf{r}_s^I = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{2.146}$$

If $\mathbf{R}^{I/F}$ is the rotation matrix from the central body's fixed frame to MJ2000Eq at the current epoch, then we can write the vector that describes the spin axis in the inertial frame as

$$\mathbf{r}_s^I = \mathbf{R}^{I/F}\mathbf{r}_s^F \tag{2.147}$$

Then, the y-axis is defined as

$$\hat{\mathbf{y}} = \frac{\mathbf{r}_s^I \times \hat{\mathbf{x}}}{\|\mathbf{r}_s^I \times \hat{\mathbf{x}}\|} \tag{2.148}$$

the z-axis is defined as

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \tag{2.149}$$

and

$$\mathbf{R}_{Ii} = [\ \hat{\mathbf{x}} \quad \hat{\mathbf{y}} \quad \hat{\mathbf{z}}\ ] \tag{2.150}$$

To calculate the derivative of the rotation matrix, we know that

$$\frac{d\hat{\mathbf{x}}}{dt} = \frac{\mathbf{v}_{sun}}{r_{sun}} - \hat{\mathbf{x}}\left(\hat{\mathbf{x}} \cdot \frac{\mathbf{v}_{sun}}{r_{sun}}\right) \tag{2.151}$$

and

$$\frac{d\mathbf{y}}{dt} = \dot{\mathbf{y}} = \left(\dot{\mathbf{R}}^{I/F}\mathbf{r}_s^F\right) \times \hat{\mathbf{x}} + \mathbf{r}_s^I \times \frac{d\hat{\mathbf{x}}}{dt} \tag{2.152}$$

Now we can write

$$\frac{d\hat{\mathbf{y}}}{dt} = \dot{\hat{\mathbf{y}}} = \frac{\dot{\mathbf{y}}}{y} - \hat{\mathbf{y}}\left(\hat{\mathbf{y}} \cdot \frac{\dot{\mathbf{y}}}{y}\right) \tag{2.153}$$

Finally,

$$\dot{\hat{\mathbf{z}}} = \dot{\hat{\mathbf{x}}} \times \hat{\mathbf{y}} + \hat{\mathbf{x}} \times \dot{\hat{\mathbf{y}}} \tag{2.154}$$

and

$$\dot{\mathbf{R}}_{Ii} = \left[\ \frac{d\hat{\mathbf{x}}}{dt} \quad \frac{d\hat{\mathbf{y}}}{dt} \quad \frac{d\hat{\mathbf{z}}}{dt}\ \right] \tag{2.155}$$

2.7 Appendix 1: Derivatives of ObjectReferenced Unit Vectors

The derivations of the above quantities are shown below. We start by deriving two derivatives with respect to $\mathbf{n}$, where $\mathbf{n}$ is given by:

$$\mathbf{n} = \mathbf{r} \times \mathbf{v} \tag{2.156}$$

We need to determine two derivatives of $\mathbf{n}$. First

$$\frac{d\mathbf{n}}{dt} = \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) = \underbrace{\frac{d\mathbf{r}}{dt} \times \mathbf{v}}_{0} + \mathbf{r} \times \frac{d\mathbf{v}}{\partial t} \tag{2.157}$$

so we know that

$$\boxed{\frac{d\mathbf{n}}{dt} = \mathbf{r} \times \mathbf{a}} \tag{2.158}$$

The next useful derivative is

$$\frac{dn}{dt} = \frac{d\|\mathbf{n}\|}{dt} = \frac{d}{dt}\left(\mathbf{n}^T\mathbf{n}\right)^{1/2} = \frac{\mathbf{n}^T}{n}\frac{d\mathbf{n}}{dt} \tag{2.159}$$

So we can write

$$\boxed{\frac{dn}{dt} = \frac{\mathbf{n}}{n}\cdot(\mathbf{r}\times\mathbf{a})} \tag{2.160}$$

The following two derivatives are also useful

$$\frac{dr}{dt} = \frac{d\|\mathbf{r}\|}{dt} = \frac{d}{dt}(\mathbf{r}^T\mathbf{r})^{1/2} = \frac{\mathbf{v}\cdot\mathbf{r}}{r} \tag{2.161}$$

$$\boxed{\frac{dr}{dt} = \frac{\mathbf{v}\cdot\mathbf{r}}{r}} \tag{2.162}$$

$$\frac{dv}{dt} = \frac{d\|\mathbf{v}\|}{dt} = \frac{d}{dt}(\mathbf{v}^T\mathbf{v})^{1/2} = \frac{\mathbf{v}\cdot\mathbf{a}}{v} \tag{2.163}$$

so we can write

$$\boxed{\frac{dv}{dt} = \frac{\mathbf{v}\cdot\mathbf{a}}{v}} \tag{2.164}$$

$$\boxed{\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}} \tag{2.165}$$

$$\boxed{\hat{\mathbf{r}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}} \tag{2.166}$$

$$\boxed{\hat{\mathbf{n}} = \frac{\mathbf{r}\times\mathbf{v}}{\|\mathbf{r}\times\mathbf{v}\|}} \tag{2.167}$$

The time derivatives are derived as follows.

$$\dot{\hat{\mathbf{r}}} = \frac{\partial\hat{\mathbf{r}}}{\partial t} = \frac{\partial}{\partial t}\left(\mathbf{r}r^{-1}\right) = \frac{\mathbf{v}}{r} - \frac{\mathbf{r}}{r^2}\left(\frac{\mathbf{r}\cdot\mathbf{v}}{r}\right) \tag{2.168}$$

which can be rewritten as

$$\boxed{\frac{d\hat{\mathbf{r}}}{dt} = \frac{\mathbf{v}}{r} - \frac{\hat{\mathbf{r}}}{r}\left(\hat{\mathbf{r}}\cdot\mathbf{v}\right)} \tag{2.169}$$

$$\dot{\hat{\mathbf{v}}} = \frac{\partial\hat{\mathbf{v}}}{\partial t} = \frac{\partial}{\partial t}\left(\mathbf{v}v^{-1}\right) = \frac{\mathbf{a}}{v} - \frac{\mathbf{v}}{v^2}\left(\frac{\mathbf{v}\cdot\mathbf{a}}{v}\right) \tag{2.170}$$

which can be rewritten as

$$\boxed{\dot{\hat{\mathbf{v}}} = \frac{\mathbf{a}}{v} - \frac{\hat{\mathbf{v}}}{v}\left(\hat{\mathbf{v}}\cdot\mathbf{a}\right)} \tag{2.171}$$

Finally,

$$\dot{\hat{\mathbf{n}}} = \frac{d}{dt}\left(\mathbf{n}n^{-1}\right) = \frac{\mathbf{r}\times\mathbf{a}}{n} - \frac{\mathbf{n}}{n^3}\left(\mathbf{r}\times\mathbf{a}\cdot\mathbf{n}\right) \tag{2.172}$$

$$\boxed{\dot{\hat{\mathbf{n}}} = \frac{\mathbf{r}\times\mathbf{a}}{n} - \frac{\hat{\mathbf{n}}}{n}\left(\mathbf{r}\times\mathbf{a}\cdot\hat{\mathbf{n}}\right)} \tag{2.173}$$

2.7.1 Basic Rotation Matrices

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \tag{2.174}$$

$$\mathbf{R}_2 = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.175}$$

$$\mathbf{R}_3 = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.176}$$

Table 2.1: Recommended Values for Pole and Prime Meridian Locations of the Sun and Planets[1]

Name	Values
Sun	$\alpha_o = 286.13°$ (deg) $\delta_o = 63.87°$ (deg) $W = 84.10° + 14.1844000°d$ (deg) $\dot{W} = 14.1844000°$ (deg/s)
Mercury	$\alpha_o = 281.01 - 0.033T$ $\delta_o = 61.45 - 0.005T$ $W = 329.548 + 6.1385025d$ $\dot{W} = 6.1385025$
Venus	$\alpha_o = 272.76$ $\delta_o = 67.16$ $W = 160.20 - 1.4813688d$ $\dot{W} = -1.4813688$
Earth	$\alpha_o = 0.00 - 0.641T$ $\delta_o = 90.00 - 0.557T$ $W = 190.147 + 360.9856235d$ $\dot{W} = 360.9856235$ *Earth Data is included for completeness only, GMAT uses FK5 reduction for the Earth*
Mars	$\alpha_o = 317.68143 - 0.1061T$ $\delta_o = 52.88650 - 0.0609T$ $W = 176.630 + 350.89198226d$ $\dot{W} = 350.89198226$
Jupiter	$\alpha_o = 268.05 - 0.009T$ $\delta_o = 64.49 + 0.003T$ $W = 284.95 + 870.5366420d$ $\dot{W} = 870.5366420$
Saturn	$\alpha_o = 40.589 - 0.036T$ $\delta_o = 83.537 - 0.004T$ $W = 38.90 + 810.7939024d$ $\dot{W} = 810.7939024$
Uranus	$\alpha_o = 257.311$ $\delta_o = -15.175$ $W = 203.81 - 501.1600928d$ $\dot{W} = -501.1600928$
Neptune	$\alpha_o = 299.36 + 0.70 \sin N$ $\delta_o = 43.46 - 0.51 \cos N$ $W = 253.18 + 536.3128492d - 0.48 \sin N$ $\dot{W} = 536.3128492 - 0.48\dot{N} \cos N$ $N = 357.85 + 52.316T$ $\dot{N} = 6.0551 \times 10^{-4}$ (deg/day)
Pluto	$\alpha_o = 313.02$ $\delta_o = 9.09$ $W = 236.77 - 56.3623195d$ $\dot{W} = -56.3623195$

Table 2.2: Recommended Values for Pole and Prime Meridian Locations of Luna[1]

Name	Values
Luna	

$$\alpha_o = 269.9949 + 0.0031T - 3.8787\sin E1 - 0.1204\sin E2 + 0.0700\sin E3 - 0.0172\sin E4 + 0.0072\sin E6 - 0.0052\sin E10 - 0.0043\sin E13$$

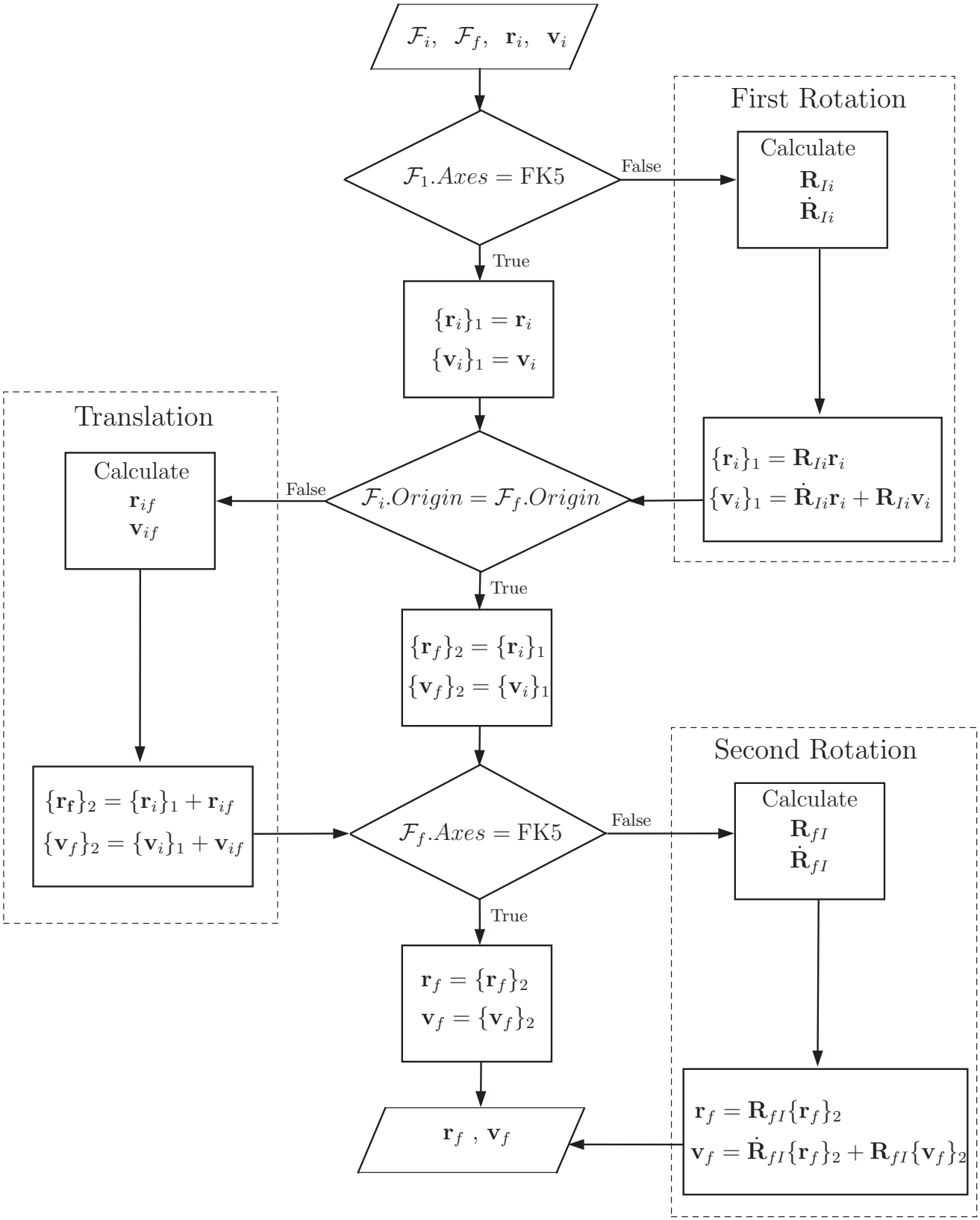
$$\delta_o = 66.5392 + 0.0130T + 1.5419\cos E1 + 0.0239\cos E2 - 0.0278\cos E3 + 0.0068\cos E4 - 0.00292\cos E6 + 0.0009\cos E7 + .0008\cos E10 - 0.0009\cos E13$$

$$W = 38.3213 + 13.17635815d - 1.4\times10^{-12}d^2 + 3.5610\sin E1 + 0.1208\sin E2 - 0.0642\sin E3 + 0.0158\sin E4 + 0.0252\sin E5 - 0.0066\sin E6 - 0.0047\sin E7 - 0.0046\sin E8 + 0.0028\sin E9 + 0.0052\sin E10 + 0.0040\sin E11 + 0.0019\sin E12 - 0.0044\sin E13$$

$$\dot{W} = +13.17635815 - 2.8\times10^{-12}d - .18870\cos E1 - .01280\cos E2 - .835\cos E3 + .211\cos E4 + .0248\cos E5 - .17\cos E6 - .061\cos E7 - .0015\cos E8 + .0049\cos E9 - .00083\cos E10 + .00001\cos E11 + .00031\cos E12 - .057\cos E13$$

where

$E1 = 125.045 - 0.0529921d$ $E2 = 250.089 - 0.1059842d$
$E3 = 260.008 + 13.0120009d$ $E4 = 176.625 + 13.3407154d$
$E5 = 357.529 + 0.9856003d$ $E6 = 311.589 + 26.4057084d$
$E7 = 134.963 + 13.0649930d$ $E8 = 276.617 + 0.3287146d$
$E9 = 34.226 + 1.7484877d$ $E10 = 15.134 - 0.1589763d$
$E11 = 119.743 + 0.0036096d$ $E12 = 239.961 + 0.1643573d$
$E13 = 25.053 + 12.9590088d$



* See Sec. 2.2 for Treatment of Pseudo-Rotating Coordinate Systems

Figure 2.7: Coordinate System Transformation Algorithm

Chapter 3

Calculation Objects

GMAT has the ability to calculate numerous quantities that are dependent upon the states of objects, coordinate systems, and the mission sequence. These calculation objects can range from the spacecraft state, to the local atmospheric density, to the positions of celestial bodies with respect to spacecraft, or other celestial bodies. In chapter, we present how GMAT performs these calculations by showing the mathematical algorithms.

The chapter begins by discussing different orbit state representations. Each of the orbit state representations available in GMAT are defined. Next we present the algorithms used to convert between different state representations. These include the Keplerian elements, modified Keplerian elements, Cartesian state, spherical state, and the equinoctial elements. In the second section we present how GMAT calculates all calculation parameters. Examples include the orbit period, percent shadow, and energy. The algorithms to calculate all parameters are included and described in detail. We conclude this chapter with a presentation of the algorithms used to calculate libration point and barycenter locations.

3.1 Spacecraft State Representations

There are several state representations that can be used in GMAT to define the state of a spacecraft object. These include the Keplerian elements, Cartesian state, equinoctial elements, spherical elements, and the modified Keplerian elements. In the following subsections, we discuss the definitions of these states types, and show how GMAT converts between the different state representations.

3.1.1 Definitions

The Keplerian elements are one of the most commonly used state representations. They provide a way to define the spacecraft state in way that provides an intuitive understanding of the motion of spacecraft in orbit. The Keplerian elements are denoted a, e, i, ω, Ω, and ν. They are defined in detail in Table 3.2 and illustrated in Fig. 3.1. Sections 3.1.2 and 3.1.3 show the algorithms that GMAT uses to convert between the Keplerian elements and the cartesian state.

The cartesian state is another common state representation and is often used in the numerical integration of the equations of motion. The cartesian state with respect to a given coordinate system is described in detail in Table 3.1.

The equinoctial elements are a set of non-singular elements that can be used to describe the state of a spacecraft. Because they are nonsingular, they are useful for expressing the equations of motion in Variation of Parameters (VOP) form. The elements can be unintuitive to use however. The equinoctial elements are described in detail in Table 3.4.

The modified Keplerian elements are similar to the Keplerian elements except a and e are replaced with the radius of periapsis r_p, and the radius of apoapsis r_a. r_p and r_a are often more convenient and intuitive for describing the dimensions of a Keplerian orbit than a and e. The modified Keplerian elements are defined in detail in Table 3.6. Note that both the Keplerian and modified Keplerian elements are undefined for parabolic orbits because the semimajor axis is infinite. Currently, GMAT does not support parabolic orbits

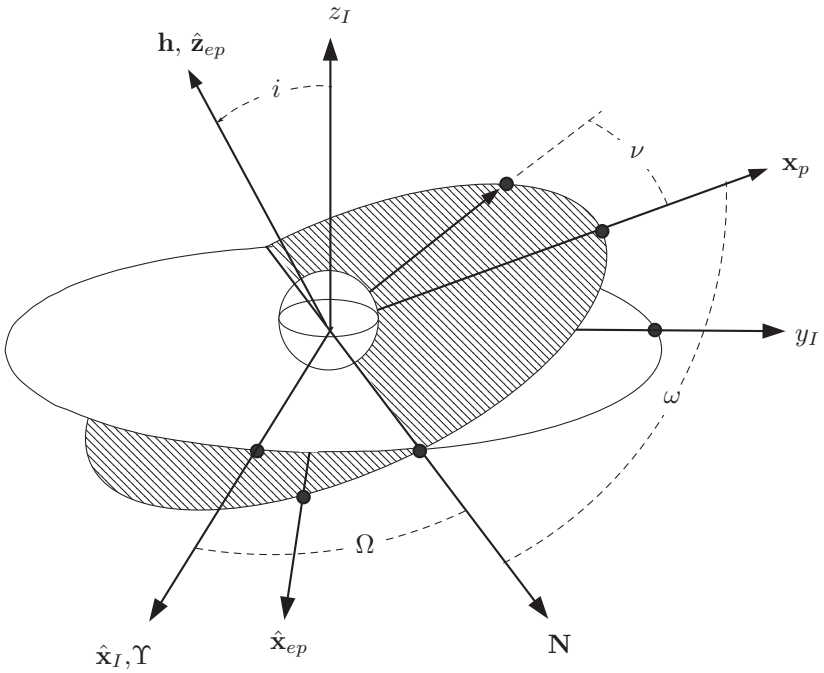


Figure 3.1: The Keplerian Elements

Table 3.1: The Cartesian State

Symbol	Description
x	x-component of position
y	y-component of position
z	z-component of position
$\dot{x}$	x-component of velocity
$\dot{y}$	y-component of velocity
$\dot{z}$	z-component of velocity

for this reason. Let's begin by looking at how GMAT converts from the Cartesian state to the Keplerian elements.

3.1.2 Cartesian State to Keplerian Elements

The conversion from the Cartesian state to the Keplerian elements has four special cases: elliptic inclined, circular inclined, elliptic equatorial, and circular equatorial. Certain orbital elements are undefined for some of the cases. For example, the right ascension of the ascending node, Ω, is undefined for equatorial orbits. However, computer systems don't handle undefined values gracefully. In this section, we'll see how the orbital elements are defined for each of the special cases, and how GMAT calculates the orbital elements for each case.[?]

Given: $\mathbf{r}$, $\mathbf{v}$, and μ

Find: a, e, i, ω, Ω, and ν

We begin by calculating the specific angular momentum and its magnitude.

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.1}$$

$$h = \|\mathbf{h}\| \tag{3.2}$$

Table 3.2: The Keplerian Elements (also see Fig. 3.1)
(See Table 3.3 for definitions of elements for near circular and near equatorial orbits.)

Symbol	Name	Description
a	semimajor axis	The semimajor contains information on the type and size of an orbit. If $a > 0$ the orbit is elliptic. If $a < 0$ the orbit is hyperbolic. $a = \infty$ for parabolic orbits.
e	eccentricity	The eccentricity contains information on the shape of an orbit. If $e = 0$, then the orbit is circular. If $0 < e < 1$ the orbit is elliptical. If $e = 1$ the orbit is parabolic. If $e > 1$ then the orbit is hyperbolic.
i	inclination	The inclination is the angle between the $\hat{\mathbf{z}}_I$ axis and the orbit normal direction $\mathbf{h}$. If $i \leq 90^\circ$ then the orbit is prograde. If $i > 90^\circ$ then the orbit is retrograde.
ω	argument of periapsis	The argument of periapsis is the angle between a vector pointing at periapsis and a vector pointing in the direction of the line of nodes. The argument of periapsis is undefined for circular orbits.
Ω	right ascension of the ascending node	Ω is defined as the angle between $\hat{\mathbf{x}}_I$ and $\mathbf{N}$ measured counterclockwise. $\mathbf{N}$ is defined as the vector pointing from the center of the central body to the spacecraft, when the spacecraft crosses the bodies equatorial plane from the southern to the northern hemisphere. Ω is undefined for equatorial orbits.
ν	true anomaly	The true anomaly is defined as the angle between a vector pointing at periapsis and a vector pointing at the spacecraft. The true anomaly is undefined for circular orbits.

Table 3.3: Keplerian Elements for Special Cases

Orbit Type	Numerical Threshold	Description
Elliptic Inclined	$e \geq 10^{-11}$, $i \geq 10^{-11}$	Ω is the angle between the x-axis and the line of nodes. ω is the angle between the line of nodes and the eccentricity vector, ν is the angle between the eccentricity vector and the spacecraft position vector.
Elliptic Equatorial	$e \geq 10^{-11}$, $i < 10^{-11}$	$\Omega = 0$, ω is the angle between the x-axis and the eccentricity vector, ν is the angle between the eccentricity vector and the spacecraft position vector.
Circular Inclined	$e < 10^{-11}$, $i \geq 10^{-11}$	Ω is the angle between the x-axis and the line of nodes, $\omega = 0$, ν is the angle between the line of nodes and the spacecraft position vector.
Circular Equatorial	$e < 10^{-11}$, $i < 10^{-11}$	$\Omega = 0$, $\omega = 0$, ν is the angle between the x-axis and the spacecraft position vector.

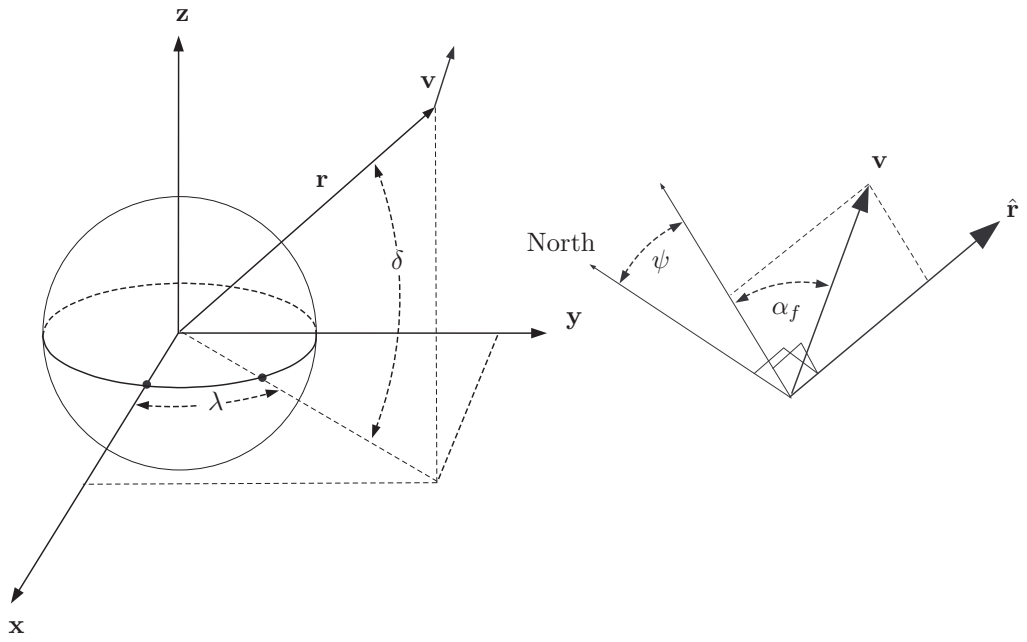


Figure 3.2: The Spherical Elements

A vector pointing in the direction of the line of nodes is

$$\mathbf{n} = [\ 0 \quad 0 \quad 1\ ]^T \times \mathbf{h} \tag{3.3}$$

$$n = \|n\| \tag{3.4}$$

The orbit eccentricity and energy are calculated using

$$r = \|\mathbf{r}\| \tag{3.5}$$

$$v = \|\mathbf{v}\| \tag{3.6}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r} \cdot \mathbf{v})\mathbf{v}}{\mu} \tag{3.7}$$

$$e = \|\mathbf{e}\| \tag{3.8}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.9}$$

For parabolic orbits, the semimajor axis is infinite and the energy is zero. Here we check to see if the orbit is near parabolic. If $|1 - e| < 10^{-7}$ an error message is reported and conversion is aborted.

The semimajor axis is computed using

$$a = -\frac{\mu}{2\xi} \tag{3.10}$$

Here we check to see if the conic section is nearly singular. If $|a(1 - e)| < .001(km)$ then an error message is thrown and conversion is aborted.

If the above tests pass, then we continue and calculate the inclination.

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{3.11}$$

There are four special cases for Ω, ω, and ν and each case is treated differently.

Special Case 1: Non-circular, Inclined Orbit

if $(e \geq 10^{-11})$ and $(i \geq 10^{-11})$, then

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{3.12}$$

Fix quadrant for Ω: if $n_y < 0$, then $\Omega = 2\pi - \Omega$

$$\omega = \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{e}}{ne}\right) \tag{3.13}$$

Fix quadrant for ω: if $e_z < 0$, then $\omega = 2\pi - \omega$

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) \tag{3.14}$$

Fix quadrant for ν: if $\mathbf{r}\cdot\mathbf{v} < 0$, then $\nu = 2\pi - \nu$
Special Case 2: Non-circular, Equatorial Orbit
if $(e \geq 10^{-11})$ and $(i < 10^{-11})$, then

$$\Omega = 0 \tag{3.15}$$

$$\omega = \cos^{-1}\frac{e_x}{e} \tag{3.16}$$

Fix quadrant for ω: if $e_y < 0$, then $\omega = 2\pi - \omega$

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) \tag{3.17}$$

Fix quadrant for ν: if $\mathbf{r}\cdot\mathbf{v} < 0$, then $\nu = 2\pi - \nu$
Special Case 3: Circular, Inclined Orbit
if $(e < 10^{-11})$ and $(i \geq 10^{-11})$, then

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{3.18}$$

Fix quadrant for Ω: if $n_y < 0$, then $\Omega = 2\pi - \Omega$

$$\omega = 0 \tag{3.19}$$

$$\nu = \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{r}}{nr}\right) \tag{3.20}$$

Fix quadrant for ν: if $r_z < 0$, then $\nu = 2\pi - \nu$
Special Case 4: Circular, Equatorial Orbit
if $(e < 10^{-11})$ and $(i < 10^{-11})$, then

$$\Omega = 0 \tag{3.21}$$

$$\omega = 0 \tag{3.22}$$

$$\nu = \cos^{-1}\left(\frac{r_x}{r}\right) \tag{3.23}$$

Fix quadrant for ν: if $r_y < 0$, then $\nu = 2\pi - \nu$

In the next section, we look at how to perform the inverse transformation and convert from Keplerian elements to the Cartesian state vector.

3.1.3 Keplerian Elements to Cartesian State

The transformation from the Keplerian elements to the Cartesian state is one of the most common state transformations in astrodynamics. We previously defined both state types and refer you to Tables 3.1 and 3.2 for their definitions. Below we show the algorithm that GMAT uses to convert from the Keplerian elements to the Cartesian state.[?]
Give: a, e, i, Ω, ω, ν, and μ
Find: $\mathbf{r}$ and $\mathbf{v}$

First check to ensure the keplerian elements are not singular. If $|a(1-e)| < .001(km)$ then the following error message is reported and conversion is aborted: "Warning: A nearly singular conic section was encountered while converting from the Keplerian elements to the Cartesian state. The radius of periapsis must be greater than 1 meter."

Next check that the magnitude of the position vector is not infinite. If $(1 + e\cos\nu <$ 1e-30) then the following error message is reported and conversion is aborted: "Warning: A near infinite radius was encountered while converting from the Keplerian elements to the Cartesian state. "

Check that the orbit is not parabolic, in which case, p is undefined. If $(|1-e| < 1e-7$ then the following error message is reported and conversion is aborted: "Warning: A nearly parabolic orbit was encountered while converting from the Keplerian elements to the Cartesian state. The Keplerian elements are undefined for a parabolic orbit.".

Finally, define ν_M as the true anomaly placed between $-\pi$ and π. If $e > 1$ and $\|\nu_M\| \geq \pi - acos(1/e)$ the following error message is reported and conversion is aborted:
"Error: The TA value is not physically possible for a hyperbolic orbit with the input values of SMA and ECC.
The allowed values are: $-limitTA < TA < limitTA$ (degrees). or equivalently
The allowed values are $TA < limitTA$ or $TA > 360 - limitTA$. " where the text "limitTA"' is replaced by the numeric value computed below in degrees:

$$limitTA = \pi - acos(1/e) \tag{3.24}$$

If the previous tests pass, we continue by calculating the semilatus rectum, and the radius.

$$p = a(1 - e^2); \tag{3.25}$$

If $1 + e\cos\nu < 1e - 10$, then the following warning is displayed but computation proceeds. "Warning: The orbital radius is large in the conversion from Keplerian to Cartesian state and the state may be near a singularity causing numerical errors in the conversion."

$$r = \frac{p}{1 + e\cos\nu} \tag{3.26}$$

The position components of the cartesian state vector are calculated using the following three equations.

$$x = r\left(\cos(\omega + \nu)\cos\Omega - \cos i\sin(\omega + \nu)\sin\Omega\right) \tag{3.27}$$

$$y = r\left(\cos(\omega + \nu)\sin\Omega + \cos i\sin(\omega + \nu)\cos\Omega\right) \tag{3.28}$$

$$z = r\left(\sin(\omega + \nu)\sin i\right) \tag{3.29}$$

Before calculating the velocity components we check to ensure the orbit is not parabolic. This avoids another possible division by zero.
if $(\|p\| < 1e - 30)$, then error and return: "Warning: GMAT does not support parabolic orbits in conversion from keplerian to cartesian elements".
If the orbit is not parabolic, we continue and calculate the velocity components using

$$\begin{aligned}\dot{x} = \sqrt{\frac{\mu}{p}}\left(\cos\nu + e\right)\left(-\sin\omega\cos\Omega - \cos i\sin\Omega\cos\omega\right) - \\ \sqrt{\frac{\mu}{p}}\sin\nu\left(\cos\omega\cos\Omega - \cos i\sin\Omega\sin\omega\right)\end{aligned} \tag{3.30}$$

$$\begin{aligned}\dot{y} = \sqrt{\frac{\mu}{p}}\left(\cos\nu + e\right)\left(-\sin\omega\sin\Omega + \cos i\cos\Omega\cos\omega\right) - \\ \sqrt{\frac{\mu}{p}}\sin\nu\left(\cos\omega\sin\Omega + \cos i\cos\Omega\sin\omega\right)\end{aligned} \tag{3.31}$$

$$\dot{z} = \sqrt{\frac{\mu}{p}}\left[(\cos\nu + e)\sin i\cos w - \sin\nu\sin i\sin\omega\right] \tag{3.32}$$

Now let's look at how to calculate the cartesian state given the equinoctial elements.

Table 3.4: The Equinoctial Elements

Symbol	Description
a	The semimajor contains information on the type and size of an orbit. If $a > 0$ the orbit is elliptic. If $a < 0$ the orbit is hyperbolic.
h	The projection of the eccentricity vector onto the $\hat{\mathbf{y}}_{ep}$ axis.
k	The projection of the eccentricity vector onto the $\hat{\mathbf{x}}_{ep}$ axis.
p	The projection of $\mathbf{N}$ onto the $\hat{\mathbf{y}}_{ep}$ axis.
q	The projection of $\mathbf{N}$ onto the $\hat{\mathbf{x}}_{ep}$ axis.
λ	The mean longitude.

3.1.4 Equinoctial Elements to Cartesian State

The equinoctial elements used in GMAT are defined in Table 3.4. The algorithm to convert from equinoctial elements to the cartesian state was taken from the GTDS Mathematical Theory.[8]
Given: a, h, k, p, q, λ, and μ
Find: $\mathbf{r}$ and $\mathbf{v}$

We begin by using the mean longitude, λ, to find the eccentric longitude F. The equation relating the two is transcendental:

$$\lambda = F + h\cos F - k\sin F \tag{3.33}$$

We use the Newton-Raphson method to solve for F, using λ as the initial guess. We iterate on the following equation until $|F(i+1) - F(i)| < 10^{-10}$.

$$F(i+1) = F(i) - \frac{f(F)}{f'(F)} \tag{3.34}$$

where

$$f(F) = F + h\cos(F) - k\sin(F) - \lambda \tag{3.35}$$
$$f'(F) = 1 - h\sin(F) - k\cos(F) \tag{3.36}$$

Once the eccentric longitude is calculated, we continue with

$$\beta = \frac{1}{1 + \sqrt{1 - h^2 - k^2}} \tag{3.37}$$

$$n = \sqrt{\frac{\mu}{a^3}} \tag{3.38}$$

$$r = a(1 - k\cos F - h\sin F) \tag{3.39}$$

The cartesian components expressed in the equinoctial coordinate system can be calculated using.

$$X_1 = a\left[(1 - h^2\beta)\cos F + hk\beta\sin F - k\right] \tag{3.40}$$

$$Y_1 = a\left[(1 - k^2\beta)\sin F + hk\beta\cos F - h\right] \tag{3.41}$$

$$\dot{X}_1 = \frac{na^2}{r}\left[hk\beta\cos F - (1 - h^2\beta)\sin F\right] \tag{3.42}$$

$$\dot{Y}_1 = \frac{na^2}{r}\left[(1 - k^2\beta)\cos F - hk\beta\sin F\right] \tag{3.43}$$

The transformation from the equinoctial system to the inertial Cartesian system is given by

$$\mathbf{r} = X_1\hat{\mathbf{f}} + Y_1\hat{\mathbf{g}} \tag{3.44}$$

$$\mathbf{v} = \dot{X}_1\hat{\mathbf{f}} + \dot{Y}_1\hat{\mathbf{g}} \tag{3.45}$$

where

$$\begin{bmatrix}\hat{\mathbf{f}} & \hat{\mathbf{g}} & \hat{\mathbf{w}}\end{bmatrix} = \frac{1}{1+p^2+q^2}\ \mathbf{Q} \tag{3.46}$$

and

$$\mathbf{Q} = \begin{pmatrix} 1-p^2+q^2 & 2pqj & 2p \\ 2pq & \left(1+p^2-q^2\right)j & -2q \\ -2pj & 2q & \left(1-p^2-q^2\right)j \end{pmatrix} \tag{3.47}$$

and j = 1 for direct orbits ($0 \le i \le 90^\circ$)
j = -1 for retrograde orbits ($90 < i \le 180^\circ$)

Currently GMAT sets $j = 1$ for all input states. Now let's look at how to calculate the cartesian state given the equinoctial elements.

3.1.5 Cartesian State to Equinoctial Elements

The equinoctial elements used in GMAT are defined in Table 3.4. The algorithm to convert from the cartesian state to the equinoctial elements was taken from the GTDS Mathematical Theory.[8]
Given: $\mathbf{r}$, $\mathbf{v}$, and μ
Find: a, h, k, p, q, λ, and μ

The orbit eccentricity and energy are calculated using

$$r = \|\mathbf{r}\| \tag{3.48}$$

$$v = \|\mathbf{v}\| \tag{3.49}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu} \tag{3.50}$$

$$e = \|\mathbf{e}\| \tag{3.51}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.52}$$

For parabolic orbits, the semimajor axis is infinite and the energy is zero. Here we check to see if the orbit is near parabolic. If $|1-e| < 10^{-7}$ an error message is reported and conversion is aborted.

The semimajor axis is computed using

$$a = -\frac{\mu}{2\xi} \tag{3.53}$$

Here we check to see if the conic section is nearly singular. If $|a(1-e)| < .001(km)$ then an error message is thrown and conversion is aborted.

The angular momentum unit vector is

$$\hat{\mathbf{h}} = \frac{\mathbf{r}\times\mathbf{v}}{\|\mathbf{r}\times\mathbf{v}\|} \tag{3.54}$$

The unit vectors that define the equinoctial coordinate system can be calculated using

$$f_x = 1 - \frac{\hat{h}_x^2}{1+\hat{h}_z j} \tag{3.55}$$

$$f_y = -\frac{\hat{h}_x\hat{h}_y}{1+\hat{h}_z j} \tag{3.56}$$

$$f_z = -\hat{h}_x j \tag{3.57}$$

where $j = 1$ if $\hat{h}_z > 0$ and $j = -1$ otherwise. Currently GMAT always assumes $j = 1$.

$$\hat{\mathbf{g}} = \hat{\mathbf{h}} \times \hat{\mathbf{f}} \tag{3.58}$$

We now have the necessary information to calculate the elements h, k, p, and q using the following relationships.

$$h = \mathbf{e} \cdot \hat{\mathbf{g}} \tag{3.59}$$

$$k = \mathbf{e} \cdot \hat{\mathbf{f}} \tag{3.60}$$

$$p = \frac{\hat{h}_x}{1 + \hat{h}_z j} \tag{3.61}$$

$$q = -\frac{\hat{h}_y}{1 + \hat{h}_z j} \tag{3.62}$$

The final element to calculate is the mean longitude, λ. We begin by computing the eccentric longitude, F, using

$$X_1 = \mathbf{r} \cdot \hat{\mathbf{f}} \tag{3.63}$$

$$Y_1 = \mathbf{r} \cdot \hat{\mathbf{g}} \tag{3.64}$$

and

$$\cos F = k + \frac{\left(1 - k^2\beta\right) X_1 - hk\beta Y_1}{a\sqrt{1 - h^2 - k^2}} \tag{3.65}$$

$$\sin F = h + \frac{\left(1 - h^2\beta\right) Y_1 - hk\beta X_1}{a\sqrt{1 - h^2 - k^2}} \tag{3.66}$$

$$F = \tan_2^{-1}\left(\frac{\sin F}{\cos F}\right) \tag{3.67}$$

where β is given by Eq. 3.37. The mean longitude is computed using the generalized Kepler equation

$$\lambda = F + h\cos F - k\sin F \tag{3.68}$$

Now let's look at transformations involving the spherical elements.

3.1.6 Cartesian State to SphericalAZFPA State

The spherical state, with azimuth, α_f, and flight path angle, ψ, is described in Table 3.5 and Fig. 3.2. The algorithm below shows how GMAT converts from the cartesian state to the spherical state with azimuth and flight path angle.

Given: $\mathbf{r}$ and $\mathbf{v}$

Find: r, λ, δ, v, ψ, and α_f

We begin by calculating the right ascension λ, and the declination δ.

$$r = \|\mathbf{r}\| \tag{3.69}$$

$$\lambda = \tan_2^{-1}(y, x) \tag{3.70}$$

$$\delta = \sin^{-1}(\frac{z}{r}) \tag{3.71}$$

The magnitude of the velocity vector is simply

$$v = \|\mathbf{v}\| \tag{3.72}$$

We calculate the vertical flight path angle, *psi*, using

$$\psi = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{v}}{rv}\right) \tag{3.73}$$

Table 3.5: The Spherical Elements

Symbol	Name	Description
r	r	Magnitude of the position vector, $\|\mathbf{r}\|$
λ	Right Ascension	The angle between the projection of $\mathbf{r}$ into the xy-plane and the x-axis measured counterclockwise.
δ	Declination	The angle between $\mathbf{r}$ and the xy-plane.
v	v	Magnitude of the velocity vector, $\|\mathbf{v}\|$.
ψ	Vertical flight path angle	The angle measured from a plane normal to $\mathbf{r}$ to the velocity vector $\mathbf{v}$, measured in the plane formed by $\mathbf{r}$ and $\mathbf{v}$
α_f	Flight path azimuth	The angle measured from vector perpendicular $\mathbf{r}$ and pointing north, to the projection of $\mathbf{v}$ into a plane normal to $\mathbf{r}$.
λ_v	Right ascension of velocity	The angle between the projection of $\mathbf{v}$ into the xy-plane and the x-axis measured counterclockwise.
δ_v	Declination of velocity	The angle between the velocity vector and the xy-plane.

To calculate the azimuth angle, α_z, we begin by calculating the rotation matrix from the frame in which the cartesian state is expressed in, $\mathcal{F}_i$, to a local frame, $\mathcal{F}_\ell$, where $\hat{\mathbf{z}}$ is a unit vector that points north. The basis vectors of $\mathcal{F}_\ell$ expressed in $\mathcal{F}_i$ can be calculated using

$$\hat{\mathbf{x}} = \begin{pmatrix} \cos(\delta)\cos(\lambda) \\ \cos(\delta)\sin(\lambda) \\ \sin(\delta) \end{pmatrix} \tag{3.74}$$

$$\hat{\mathbf{y}} = \begin{pmatrix} \cos(\lambda + \pi/2) \\ \sin(\lambda + \pi/2) \\ 0 \end{pmatrix} \tag{3.75}$$

$$\hat{\mathbf{z}} = \begin{pmatrix} -\sin(\delta)\cos(\lambda) \\ -\sin(\delta)\sin(\lambda) \\ \cos(\delta) \end{pmatrix} \tag{3.76}$$

We can write the tranformation matrix that goes from $\mathcal{F}_i$ to $\mathcal{F}_\ell$, $\mathbf{R}_{\ell i}$, as

$$\mathbf{R}_{\ell i} = [\ \hat{\mathbf{x}}\ \ \hat{\mathbf{y}}\ \ \hat{\mathbf{z}}\ ]^T \tag{3.77}$$

The velocity in the local frame, $\mathbf{v}'$, can be written as

$$\mathbf{v}' = \mathbf{R}_{\ell i}\mathbf{v} \tag{3.78}$$

Finally, we calculate the azimuth angle using

$$\alpha_f = \tan_2^{-1}(v'_y, v'_z) \tag{3.79}$$

Now that we have looked at how to convert from the Cartesian state to the spherical state, let's look at the inverse transformation that converts from the spherical state (with ψ and α_f) to the cartesian state.

3.1.7 SphericalAZFPA State to Cartesian State

In this section we present the algorithm used to convert from the spherical state (with ψ and α_f) to the cartesian state.
Given: r, λ, δ, v, ψ, and α_f
Find: $\mathbf{r}$ and $\mathbf{v}$

The components of the position vector are calculated using

$$x = r\cos\delta\cos\lambda \tag{3.80}$$
$$y = r\cos\delta\sin\lambda \tag{3.81}$$
$$z = r\sin\delta \tag{3.82}$$

We can write the velocity vector in terms of v, ψ, and α_f as,

$$\mathbf{v} = v\left[\cos(\psi)\hat{\mathbf{x}} + \sin(\psi)\sin(\alpha_f)\hat{\mathbf{y}} + \sin(\psi)\cos(\alpha_f)\hat{\mathbf{z}}\right] \tag{3.83}$$

where, $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{x}}$ are given in Eqs. (3.74), (3.75), and (3.76) respectively. Breaking down Eq. (3.83) into components gives us

$$v_x = v[\cos\psi\cos\delta\cos\lambda - \sin\psi(\sin\alpha_f\sin\lambda + \cos\alpha_f\sin\delta\cos\lambda)] \tag{3.84}$$

$$v_y = v[\cos\psi\cos\delta\sin\lambda + \sin\psi(\sin\alpha_f\cos\lambda - \cos\alpha_f\sin\delta\sin\lambda)] \tag{3.85}$$

$$v_z = v[\cos\psi\sin\delta + \sin\psi\cos\alpha_f cos\delta] \tag{3.86}$$

3.1.8 Cartesian State to SphericalRADEC State

The conversion form the Cartesian state to the spherical state with right ascension of velocity, λ_v, and declination of velocity, δ_v, is very similar to the transformation shown in Sec. 3.1.6. The algorithm to calculate λ_v and δ_v is shown below.
Given: **r** and **v**
Find: r, λ, δ, v, λ_v, and δ_v

To calculate r, λ, δ, and v we use Eqs. (3.69), (3.70), (3.71), and (3.72) respectively. The right ascension of velocity, λ_v, and declination of velocity, δ_v, are calculated using

$$\lambda_v = \tan_2^{-1}(v_y, v_x) \tag{3.87}$$

$$\delta_v = \sin^{-1}(\frac{v_z}{v}) \tag{3.88}$$

In the next section, we show the the transformation from the spherical state with right ascension of velocity, λ_v, and declination of velocity, δ_v, to the cartesian state.

3.1.9 SphericalRADEC State to Cartesian State

This transformation is similar to the conversion presented in Sec 3.1.7. The primary difference is how the velocity is represented.
Given: r, λ, δ, v, λ_v, and δ_v
Find: **r** and **v**

The position components are calculated using Eqs. (3.80), (3.81), and (3.82). The velocity components are calculated using

$$v_x = v\cos\lambda_v\cos\delta_v \tag{3.89}$$
$$v_y = v_x\tan\lambda_v \tag{3.90}$$
$$v_z = v\sin\delta_v \tag{3.91}$$

In the last few subsections, we have looked at transformations involving the spherical elements. Now let's look at transformations involving the modified Keplerian elements.

Table 3.6: The Modified Keplerian Elements

Symbol	Name	Description
r_p	radius of periapsis	The radius of periapsis is the radius at the spacecrafts closest approach to the central body. The radius of periapsis must be greater than zero, parabolic orbits are not currently supported.
r_a	radius of apoapsis	For an elliptic orbit r_a is the radius at the spacecrafts farthest distance from the central body and $r_a > r_p$. For hyperbolic orbits, $r_a < r_p$ and $r_a < 0$
i	inclination	The inclination is the angle between the $\hat{\mathbf{z}}_I$ axis and the orbit normal direction $\mathbf{h}$. If $i \leq 90°$ then the orbit is prograde. If $i > 90°$ then the orbit is retrograde.
ω	argument of periapsis	The argument of periapsis is the angle between a vector pointing at periapsis, $\mathbf{x}_p$, and a vector pointing at the spacecraft. The argument of periapsis is undefined for circular orbits.
Ω	right ascension of the ascending node	Ω is defined as the angle between $\hat{\mathbf{x}}_I$ and $\mathbf{N}$ measured counterclockwise. $\mathbf{N}$ is defined as the vector pointing from the center of the central body to the spacecraft, when the spacecraft crosses the bodies equatorial plane from the southern to the northern hemisphere. Ω is undefined for equatorial orbits.
ν	true anomaly	The true anomaly is defined as the angle between a vector pointing at periapsis, $\mathbf{x}_p$, and a vector pointing at the spacecraft. The true anomaly is undefined for circular orbits.

3.1.10 Keplerian or Cartesian, to Modified Keplerian Elements

The modified Keplerian elements, described in Table 3.6, are similar to the classical Keplerian elements. The modified Keplerian elements use the radius of apoapsis, r_a, and the radius of periapsis, r_p, to describe the size and shape of an orbit. The remaining elements, i, Ω, ω, and ν, are the same for both the Keplerian and modified Keplerian elements. The modified Keplerian elements, like the Keplerian elements, are undefined for parabolic orbits. Let's look at how GMAT calculates the modified
Given: a, e, i, ω, Ω, and ν, or $\mathbf{r}$, $\mathbf{v}$, and μ
Find: r_p and r_a

If we are given the Cartesian state, we first calculate the orbital elements using the algorithm in Sec. 3.1.3. Knowing the Keplerian elements, we calculate r_a and r_p using

$$r_a = a(1 + e) \tag{3.92}$$

$$r_p = a(1 - e) \tag{3.93}$$

Now let's look at the inverse transformation.

3.1.11 Modified Keplerian Elements to Keplerian Elements

The conversion from modified Keplerian elements to the Keplerian elements is discussed below. To perform the conversion, we use relationships that allow us to write the semimajor axis, a, and the eccentricity, e, in terms of r_a and r_p.
Given: r_p, r_a, i, ω, Ω, and ν
Find: a and e

We begin by calculating the eccentricity using

$$e = \frac{1 - \frac{r_p}{r_a}}{1 + \frac{r_p}{r_a}} \tag{3.94}$$

The semimajor axis is calculated using

$$a = \frac{r_p}{1 - e} \tag{3.95}$$

This concludes our discussion of state transformations. In the last few subsections we presented the algorithms used to convert between different orbit state representations used in GMAT. These include the Cartesian state, the Keplerian elements, the modified Keplerian elements, and two spherical state parameterizations. In the next section, we present the algorithms used to calculate properties such as orbit period, beta angle, and mean motion to name a few.

3.2 Simple Parameters

Simple parameters, which we will abbreviate as simply "parameters", are properties of spacecraft or other objects that are only dependent upon one of the following: CoordinateSystem, CentralBody, or None. An example of a simple parameter is the magnitude of a spacecrafts velocity vector. The spacecrafts velocity vector is dependent upon the coordinate system in which it is expressed. Once we have specified a coordinate system, it is trivial to calculate the velocity vector, and therefore its magnitude, in that coordinate system.

In GMAT, the syntax to specify a simple parameter is

`ObjectName.Dependency.ParameterName`

So, to calculate the magnitude of the velocity, of a spacecraft named `Sat`, in the Earth Fixed frame, we would use

`Sat.EarthFixed.VMAG`

GMAT has the ability to calculate many parameters in addition to `VMAG`. In the following subsections, we present the algorithms used to calculate all parameters in GMAT. We begin each subsection with a description of the parameter, and then give the type of dependency.

3.2.1 A1Gregorian

Description: `A1Gregorian` is the epoch of an object, in the A1 time system, given in the Gregorian date format.

Dependency: None.

The A1 date, in modified Julian date format is the current independent variable for time in GMAT. Therefore, it is not necessary to convert the date to another system for this parameter. The only calculation required for this parameter is to use the algorithm in Sec. **??** to convert from Modified Julian date format to Gregorian date format.

3.2.2 A1ModJulian

Description: `A1ModJulian` is the epoch of an object, in the A1 time system, given in the modified Julian date format.

Dependency: None.

The A1 date, in modified Julian date format is the current independent variable for time in GMAT. There are no calculations required for this parameter.

3.2.3 Altitude

Description: `Altitude` is the distance between a spacecraft and a plane tangent to the surface of the body at the sub-satellite point. GMAT assumes the body is an ellipsoid. The equatorial radius, and properties of the ellipsoid depend upon the particular body chosen by the user.

Dependency: Central Body.
Given: $\mathbf{r}$ in $\mathcal{F}_1$
Find: A
Definitions:

- $\mathcal{F}_1$ is the coordinate system in which GMAT originally knows $\mathbf{r}$
- $\mathcal{F}_F$ is body fixed system of the central body selected by the user.
- f is the bodies flattening coefficient
- R is the bodies mean equatorial radius
- ϕ_{gd} is the geodedic latitude of the spacecraft in the body fixed frame.
- h is the `Altitude` parameter

First we calculate ϕ_{gd} using the algorithm in Sec. 3.2.21. However, to calculate h, GMAT does not convert to degrees, or use the modulo function.

Then, with $\mathbf{r}$ expressed in $\mathcal{F}_F$, we perform

$$r_{xy} = \sqrt{x^2 + y^2} \tag{3.96}$$

$$e^2 = 2f - f^2 \tag{3.97}$$

$$h = \frac{r_{xy}}{\cos(\phi_{gd})} - \frac{R}{\sqrt{1 - e^2 \sin^2 \phi_{gd}}}; \tag{3.98}$$

3.2.4 AOP

Description: `AOP` is the argument of periapsis of a spacecraft. The argument of periapsis is the angle between the eccentricity vector and a vector in the direction of the right ascension of the ascending node. See below for treatment of circular and equatorial orbits. This algorithm is adopted from Vallado.[3]
Dependency: Coordinate System.
Given: $\mathbf{r}$ and $\mathbf{v}$
Find: ω

$$r = \|\mathbf{r}\|$$

$$v = \|\mathbf{v}\|$$

$$\mathbf{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu}$$

$$e = \|\mathbf{e}\|$$

Special Case: Circular Orbit
if $e < 10^{-11}$ then, $\omega = 0.0$ and return.
Otherwise continue,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}$$

$$h = \|\mathbf{h}\|$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right)$$

Special Case: Elliptic, Equatorial Orbit
if $i < 10^{-11}$ then,

$$\omega = \cos^{-1}\left(\frac{e_x}{e}\right) \tag{3.99}$$

where e_x is the first component of the eccentricity vector.
Fix quadrant for ω: if $e_y < 0$, then $\omega = 2\pi - \omega$

Otherwise continue
Special Case: Elliptic, Inclined Orbit

$$\mathbf{n} = [\ 0 \quad 0 \quad 1\ ]^T \times \mathbf{h}$$
$$\omega = \cos^{-1} \frac{\mathbf{n} \cdot \mathbf{e}}{\|\mathbf{n}\| \|\mathbf{e}\|}$$

Fix quadrant for ω: if $e_z < 0$, then $\omega = 2\pi - \omega$.
Finally, ω is converted to degrees.

3.2.5 Apoapsis

Description: `Apoapsis` is the parameter used in stopping conditions to allow the stopping condition algorithm to locate the time when a spacecraft is at apoapsis. Apoapsis is defined as a point, along an orbital path, when the component of velocity, in the spacecraft position vector direction, changes from positive to negative. The `Apoapsis` parameter is defined as the dot product of the position and velocity vectors.
Dependency: Central Body.
Given: $\mathbf{r}$, $\mathbf{v}$ in $\mathcal{F}_1$
Find: A
Definitions:

- $\mathcal{F}_1$ is the coordinate system in which GMAT originally knows $\mathbf{r}$ and $\mathbf{v}$
- $\mathcal{F}_2$ is a system with the MJ2000Eq axes, centered at the central body selected by the user.
- A is the `Apoapsis` parameter

if $(\mathcal{F}_1 \neq \mathcal{F}_2)$ convert $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}_2$. Then,

$$A = \mathbf{r} \cdot \mathbf{v} \tag{3.100}$$

3.2.6 AZI

Description: `AZI` is the azimuth angle of a spacecraft, as shown in Fig. 3.1 using the symbol α_f.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$
Find: α_f

AZI is calculated using the algorithm shown in Sec. 3.1.6. There is little benefit using a routine that calculates only α_f and not ψ.

3.2.7 BdotT and BdotR

Description: The "B" vector, $\mathbf{B}$, is only defined for hyperbolic orbits and is the vector from the center of mass of the central body, to the incoming hyperoblic asysmptote, such that the length of $\mathbf{B}$ is a minimum. Another way to say this is that $\mathbf{B}$ is perpendicular to the incoming asymptote. Let's define $\mathbf{S}$ as a unit vector in the direction of the incoming asymptote. Then, $\mathbf{T}$ is a unit vector perpendicular to $\mathbf{S}$, that lies in the xy-plane of the coordinate system, $\mathcal{F}_B$, chosen by the user. $\mathbf{R}$ is a unit vector perpendicular to both $\mathbf{S}$ and $\mathbf{T}$. Finally, `BdotT` is the dot product of $\mathbf{B}$ and $\mathbf{T}$, and `BdotR` is the dot product of $\mathbf{B}$ and $\mathbf{R}$. The method below was adopted from work by Kizner.[9]
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$, and definition of $\mathcal{F}_B$
Find: B_R and B_T
Definitions:

- $\mathcal{F}_1$ is the coordinate system in which GMAT originally knows $\mathbf{r}$ and $\mathbf{v}$
- $\mathcal{F}_B$ is the coordinate system in which to perform B-plane calculations. GMAT will place $\mathbf{T}$ in the xy-plane of $\mathcal{F}_B$. $\mathcal{F}_B$ must have a gravitational body at its origin.

- μ is the gravitational parameter of the central body at the origin of $\mathcal{F}_B$
- B_R is the dot product of $\mathbf{B}$ and $\mathbf{R}$
- B_T is the dot product of $\mathbf{B}$ and $\mathbf{T}$

if the selected coordinate system does not have a celestial body as its origin, then exit and throw an error message.

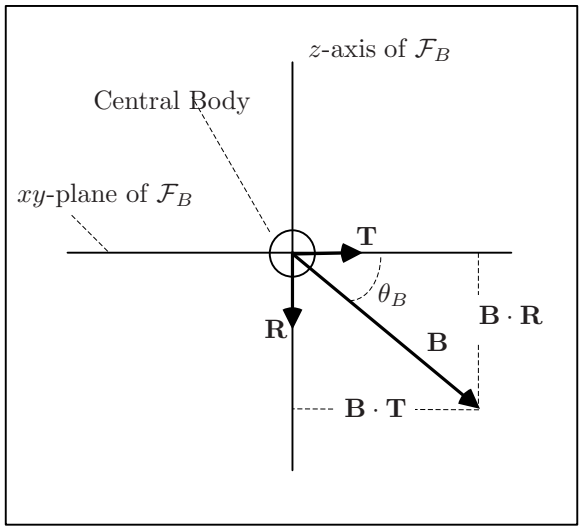


Figure 3.3: Geometry of the B-Plane as Seen From a Viewpoint Perpendicular to the B-Plane

if $\mathcal{F}_1 \neq \mathcal{F}_B$ convert $\mathbf{r}$ and $\mathbf{v}$ from $\mathcal{F}_1$ to $\mathcal{F}_B$

$$r = \|\mathbf{r}\|$$
$$v = \|\mathbf{v}\|$$

Calculate eccentricity related information

$$\mathbf{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu}$$
$$e = \|\mathbf{e}\|$$
$$\hat{\mathbf{e}} = \frac{\mathbf{e}}{e}$$

If $e \leq 1$, then the method fails and returns.

Now let's calculate the angular momentum and orbit normal vectors.

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}$$
$$h = \|\mathbf{r} \times \mathbf{v}\|$$
$$\hat{\mathbf{h}} = \frac{\mathbf{h}}{h}$$

A unit vector normal to both the eccentricity vector and the orbit normal vector is defined as:

$$\hat{\mathbf{n}} = \hat{\mathbf{h}} \times \hat{\mathbf{e}}$$

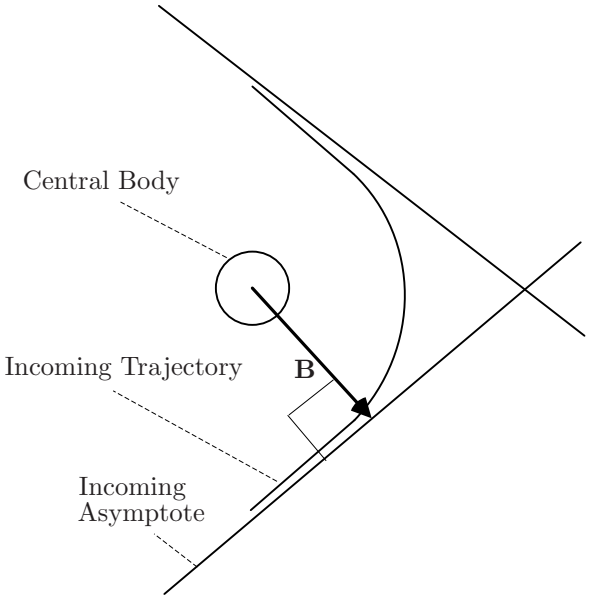


Figure 3.4: The B-Vector as Seen From a Viewpoint Perpendicular to Orbit Plane

The following relations are only true for hyperbolic orbits: The semiminor axis, b, can be calculated using

$$b = \frac{h^2}{\mu\sqrt{e^2-1}}$$

The incoming asymptote is defined using

$$\mathbf{S} = \frac{\hat{\mathbf{e}}}{e} + \sqrt{1-\left(\frac{1}{e}\right)^2}\,\hat{\mathbf{n}}$$

The B-vector, $\mathbf{B}$, is calculated using

$$\mathbf{B} = b\left(\sqrt{1-\left(\frac{1}{e}\right)^2}\,\hat{\mathbf{e}} - \frac{1}{e}\hat{\mathbf{n}}\right)$$

The remaining vectors, $\mathbf{T}$ and $\mathbf{R}$ are found using

$$\mathbf{T} = \frac{[\,S_y \quad -S_x \quad 0\,]^T}{\sqrt{S_x^2+S_y^2}}$$

$$\mathbf{R} = \mathbf{S}\times\mathbf{T}$$

Finally, the desired quantities are found using

$$B_T = \mathbf{B}\cdot\mathbf{T}$$

$$B_R = \mathbf{B}\cdot\mathbf{R}$$

if $\mathcal{F}_1 \neq \mathcal{F}_2$, convert $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}_2$

$$A = \mathbf{r}\cdot\mathbf{v} \tag{3.101}$$

3.2.8 BetaAngle

Definition: The Beta angle, β, is defined as the angle between the orbit normal vector, and the vector from the celestial body to the sun.

$$\hat{\mathbf{h}} = \frac{\mathbf{r}_{\oplus} \times \mathbf{v}_{\oplus}}{\|\mathbf{r}_{\oplus} \times \mathbf{v}_{\oplus}\|}$$

$$\hat{\mathbf{r}}_{s\oplus} = \frac{\mathbf{r}_{s\oplus}}{\|_{s\oplus}\|}$$

$$\beta = \sin^{-1}\left(\hat{\mathbf{h}} \cdot \hat{\mathbf{r}}_{s\oplus}\right) \tag{3.102}$$

- $\mathbf{r}_{\oplus}$: Position vector of spacecraft with respect to celestial body, in the EarthMJ2000Eq system.
- $\mathbf{v}_{\oplus}$: Velocity vector of spacecraft with respect to celestial body, in the EarthMJ2000Eq system.
- $\mathbf{r}_{s\oplus}$: Position vector from celestial body, to the sun.

3.2.9 BVectorAngle and BVectorMag

To avoid code reduplication, the magnitude and angle of the B vector, $\|\mathbf{B}\|$ and θ_B respectively, are calculated from the outputs of the B-Plane coordinates algorithm. The equations for $\|\mathbf{B}\|$ and θ_B are

$$\|\mathbf{B}\| = \sqrt{B_T^2 + B_R^2} \tag{3.103}$$

$$\theta_B = \tan^{-1}\frac{B_R}{B_T} \tag{3.104}$$

which is implemented using $atan2(B_R, B_T)$

3.2.10 C3Energy

Given: a, and μ
Find: C_3

$$C_3 = -\frac{\mu}{a} \tag{3.105}$$

Comment: a is calculated from the satellite cartesian state as shown in Section 3.1.2, and μ is associated with the specified central body.

3.2.11 DEC

Description: `DEC` is the declination of a spacecraft, as shown in Fig. 3.2 using the symbol δ.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$
Find: δ

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$r = \|\mathbf{r}\| \tag{3.106}$$

$$\delta = \sin^{-1}(\frac{z}{r}) \tag{3.107}$$

3.2.12 DECV

Description: `DECV` is the declination of velocity of a spacecraft.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$
Find: δ_v

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\delta_v = \sin^{-1}(\frac{v_z}{v}) \tag{3.108}$$

3.2.13 ECC

Description: `ECC` is the eccentricity of an orbit and must be greater than or equal to zero. The eccentricity contains information on the shape of an orbit. If ECC is zero then the orbit is circular. If ECC is greater than zero, but less than one, the orbit is elliptic. If ECC equals one, the orbit is parabolic. Finally, if ECC is greater than one, the orbit is hyperbolic. The algorithm used in GMAT to calculate SMA is adopted from Vallado.[3]
Dependency: Central Body.
Given: $\mathbf{r}$, $\mathbf{v}$, and μ (Central Body)
Find: e

$$r = \|\mathbf{r}\| \tag{3.109}$$

$$v = \|\mathbf{v}\| \tag{3.110}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu} \tag{3.111}$$

$$e = \|\mathbf{e}\| \tag{3.112}$$

3.2.14 FPA

Description: `FPA` is the orbit vertical Flight Path Angle as as shown in Fig. 3.2 using the symbol ψ.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$, and coordinate system $\mathcal{F}$.
Find: ψ

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\psi = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{v}}{rv}\right) \tag{3.113}$$

3.2.15 EA

Given: ν, e
Find: E

If e > (1 - $1e^{-11}$) then $E = 0$, return.

Otherwise,

$$\sin(E) = \frac{\sqrt{1-e^2}\sin(\nu)}{1+e\cos\nu} \tag{3.114}$$

$$\cos(E) = \frac{e+\cos\nu}{1+e\cos\nu} \tag{3.115}$$

$$E = \text{atan2}(\sin E, \cos E) \tag{3.116}$$

3.2.16 Energy

Description: `Energy` is the orbit energy.
Dependency: Central Body.
Given: **r**, **v**, and central body.
Find: ξ

Begin by converting **r** and **v** to a coordinate system with the origin equal to the central body defined by the user, and the MJ2000Eq axis system. Then,

$$r = \mathbf{r} \tag{3.117}$$

$$v = \mathbf{v} \tag{3.118}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.119}$$

3.2.17 HMAG

Description: `HMAG` is the magnitude of the orbit angular momentum.
Dependency: Central Body.
Given: **r**, **v**, and central body.
Find: h

Begin by converting **r** and **v** to a coordinate system with the origin equal to the central body defined by the user, and the MJ2000Eq axis system. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.120}$$

$$h = \|\mathbf{h}\| \tag{3.121}$$

3.2.18 HX,HY, and HZ

Description: `HX,HY,` and `HZ` are the components of the orbit angular momentum vector.
Dependency: Coordinate System.
Given: **r**, **v**, and coordinate system $\mathcal{F}$.
Find: h_x, h_y, and h_z

Begin by converting **r** and **v** to $\mathcal{F}$ if necessary. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = \begin{bmatrix} h_x & h_y & h_z \end{bmatrix}^T \tag{3.122}$$

3.2.19 HA

Description: `HA` is the orbit Hyperbolic Anomaly and is only defined for hyperbolic orbits. For non-hyperbolic orbits, `HA` returns a value of zero.
Dependency: Central Body. Given: ν, e
Find: H

If e $< (1 + 1e^{-11})$ then $H = 0$, return.

Otherwise,

$$\sinh(H) = \frac{\sin(\nu)\sqrt{e^2 - 1}}{1 + e\cos\nu} \tag{3.123}$$

$$H = \operatorname{asinh}(\sinh(H)) \tag{3.124}$$

3.2.20 INC

Description: `INC` is the inclination of an orbit in the chosen coordinate system.
Dependency: Coordinate System.
Given: **r**, **v**, and coordinate system *mathcalF*.
Find: *e*

Begin by converting **r** and **v** to *mathcalF* if necessary. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.125}$$

$$h = \mathbf{h} \tag{3.126}$$

$$i = \cos^{-1}(\frac{h_z}{h}) \tag{3.127}$$

3.2.21 Latitude

Description: `Latitude` is the geodetic latitude of a spacecraft. The geodedic latitude is defined as the the angle ϕ_{gc}, as shown in Fig. ( ), where the sub-satellite point is defined by the interscection of a line drawn from the spacecraft and perpendicular to a plane tangent to the surface of the body. GMAT assumes the body is an ellipsoid. The equatorial radius, and properties of the ellipsoid depend upon the particular body chosen by the user. The algorithm in GMAT is taken from Vallado.[3]

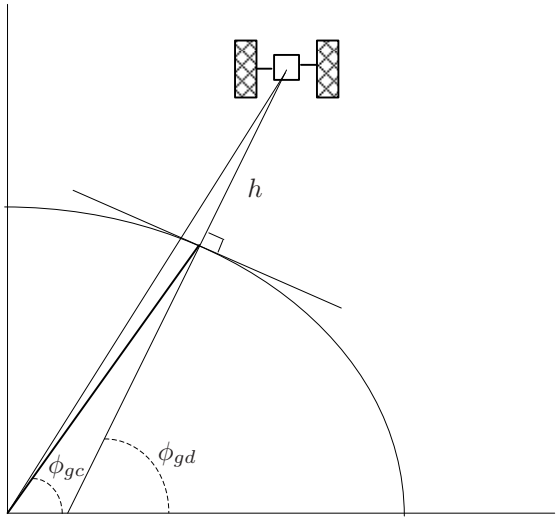


Figure 3.5: Geocentric and Geodetic Latitude

Dependency: Central Body.
Given: **r** in $\mathcal{F}_1$
Find: ϕ_{gc}
Definitions:

- $\mathcal{F}_1$ is the coordinate system in which GMAT originally knows **r**
- $\mathcal{F}_F$ is body fixed system of the central body selected by the user.
- f is the bodies flattening coefficient
- R is the bodies mean equatorial radius
- ϕ_{gd} is the geodedic latitude of the spacecraft in the body fixed frame.

if $\mathcal{F}_1 \neq \mathcal{F}_F$ convert $\mathbf{r}$ from $\mathcal{F}_1$ to $\mathcal{F}_F$. Then,

$$r_{xy} = \sqrt{x^2 + y^2} \tag{3.128}$$

Calculate the geocentric latitude to use as an initial guess to find the geodetic latitude

$$\phi_{gd} \approx \text{atan2}(z, r_{xy}); \tag{3.129}$$

$$e^2 = 2f - f^2 \tag{3.130}$$

Set $\delta = 1.0$ to initialize the loop, then,
While ($\delta > 10^{-7}$)

$$\phi' = \phi_{gd} \tag{3.131}$$

$$\phi_{gd} = \text{atan2}\left(z + \frac{Re^2 \sin^2 \phi_{gd}}{\sqrt{1 - e^2 \sin \phi_{gd}}}, r_{xy}\right) \tag{3.132}$$

$$\delta = |\phi_{gd} - \phi'| \tag{3.133}$$

EndWhile

After convergence, ϕ_{gd} is converted to degrees, and converted to fall between $-90°$ and $+90°$ degrees.

3.2.22 Longitude

Description: `Longitude` is the longitude of an object, in the body fixed frame of the central body chosen by the user.
Dependency: Central Body.
Given: $\mathbf{r}$, central body.
Find: ϕ

Begin by converting $\mathbf{r}$ to the body fixed system of the central body defined by the user. Then,

$$\phi = \tan_2^{-1}(y, x); \tag{3.134}$$

The calculation is completed by converting to degrees and setting the value to such that $-180 \leq \phi < 180$.

3.2.23 LST

Description: `LST` is the local sidereal time of an object, with respect to the selected central body. The local sidereal time is the sum of the longitude in the bodies fixed frame, and the mean sidereal time. This is illustrated in Fig. 3.6, where $\mathcal{F}_I$ is the body's equatorial inertial system (as described in Sec. 2.6.1), $\mathcal{F}_F$ is the body's fixed system (as described in Sec. 2.6.10). λ is the longitude of the object, in this case a spacecraft, and θ_{MST} is the mean sidereal time of the prime meridian.
Dependency: Central Body.
Given: $\mathbf{r}$, t_i (epoch of spacecraft in internal time system), and central body
Find: θ_{LST}
Definitions:

- $\mathcal{F}_I$ equatorial inertial system (as described in Sec. 2.6.1) of selected central body.
- $\mathcal{F}_F$ is the central body's fixed system (as described in Sec. 2.6.10)
- λ is the longitude of the object in $\mathcal{F}_F$
- θ_{MST} is the mean sidereal time of the central body's prime meridian.
- t_i (epoch of spacecraft in internal time system)

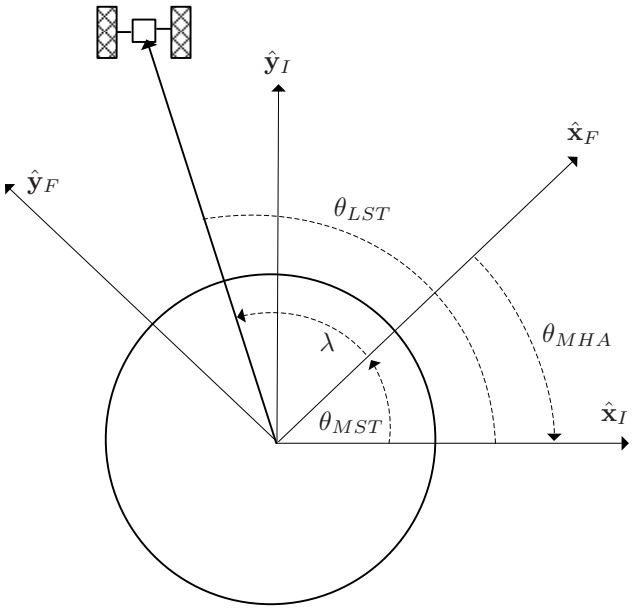


Figure 3.6: Local Sidereal Time Geometry

We begin by calculating λ using the algorithm described in Sec. 3.2.22. The mean sidereal time θ_{MST} is calculated differently for Earth than for other central bodies. If the central body is Earth, then we use the following equations to calculate θ_{MST}.

First, convert t_i, which is the spacecraft epoch in the interal time system (A1 Modified Julian Date), to T_{UT1}, which is the number elapsed Julian centuries from the J2000 epoch.

$$T_{UT1} = \frac{t_{ut1} - 21544.5}{36525} \tag{3.135}$$

$$\begin{aligned} \theta_{MST} =& 67310.54841^s + \\ & (876600^h \left(\frac{3600s}{1h}\right) + 8640184.812866) T_{UT1} + \\ & 0.093104 T_{UT1}^2 - 6.2 \times 10^{-6} T_{UT1}^3 \end{aligned} \tag{3.136}$$

3.2.24 MA

Given: ν, e
Find: M
If e < (1 - $1e^{-11}$) then calculate E using algorithm in Sec. 3.2.15. Then M is calculated using

$$M = E - e \sin E \tag{3.137}$$

Note: E must be expressed in radians in the above equation, and results in M in radians.
If e > (1 + $1e^{-11}$) then calculate H using algorithm in Sec. 3.2.19. Then M is calculated using

$$M = e \sinh H - H \tag{3.138}$$

Note: H must be expressed in radians in the above equation, and results in M in radians. GMAT outputs `MA` in degrees.
If neither of the above conditions are satisfied, $M = 0$, and output "Warning: Orbit is near parabolic in mean anomaly calculation. Setting MA = 0".

3.2.25 MHA

3.2.26 MM

Given: a, e, and μ
Find: n
The orbit is considered either circular or elliptic (both orbit types use the same equation to calculate n) if $e < 1 - 1e^{-11}$. In this case the mean motion, n, is calculated using

$$n = \sqrt{\frac{\mu}{a^3}} \tag{3.139}$$

The orbit is considered hyperbolic if $e > 1 + 1e^{-11}$. In this case the mean motion, n, is calculated using

$$n = \sqrt{-\frac{\mu}{a^3}} \tag{3.140}$$

If neither of the above two conditions are met, the mean motion is calculated using

$$n = 2\sqrt{\mu} \tag{3.141}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 3.1.2, and μ is associated with the specified central body.

3.2.27 OrbitPeriod

Given: a, and μ
Find: T
If a < 0, then $T = 0$, return.

Otherwise,

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{3.142}$$

Comment: a is calculated from the satellite cartesian state as shown in Section 3.1.2, and μ is associated with the specified central body.

3.2.28 PercentShadow

The `PercentShadow` parameter calculates the percentage of the apparent solar disk that is in view from the perspective of a spacecraft. The algorithm used in GMAT was adapted from Montenbruck[10] pgs. 80-83.

- $R_\odot$ = Radius of the Sun
- R_B = Radius of occulting body
- $R'_\odot$ = Apparent radius of the Sun
- R'_B = Apparent radius of occulting body
- $\mathbf{r}_\odot$ = Vector from central body to Sun
- $\mathbf{r}_B$ = Vector from central body to occulting body
- $\mathbf{r}$ = Vector from central body to s/c

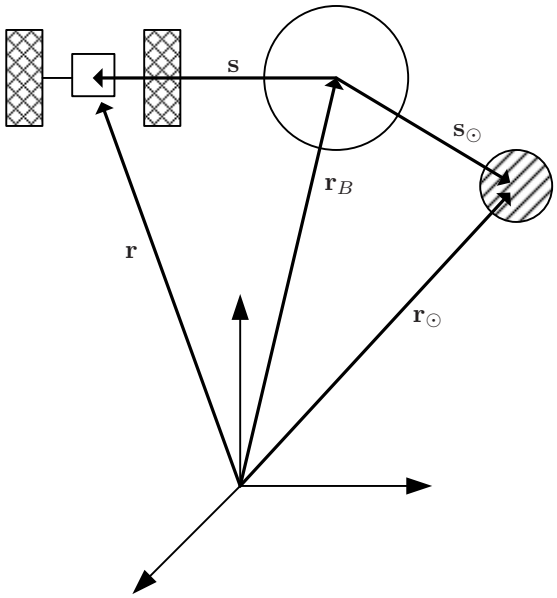


Figure 3.7: Shadow Geometry

We begin by calculating the vector from the occulting body to the spacecraft, $\mathbf{s}$, using

$$\mathbf{s} = \mathbf{r} - \mathbf{r}_B \tag{3.143}$$

and the vector from the occulting body to the sun, $\mathbf{s}_\odot$, using

$$\mathbf{s}_\odot = \mathbf{r}_\odot - \mathbf{r}_B \tag{3.144}$$

(Note that when the occulting body is the same as the central body, $\mathbf{s} = \mathbf{r}$, and $\mathbf{s}_\odot = \mathbf{r}_\odot$)

Next we calculate the apparent radius of the Sun and occulting body using

$$R'_\odot = \sin^{-1} \frac{R_\odot}{\|\mathbf{r}_\odot - \mathbf{r}\|} \tag{3.145}$$

$$R'_B = \sin^{-1} \frac{R_B}{\|\mathbf{r} - \mathbf{r}_B\|} \tag{3.146}$$

We can calculate the apparent separation of the two bodies, D', using

$$D' = \cos^{-1} \left(\frac{-\mathbf{s}^T (\mathbf{r}_\odot - \mathbf{r})}{s \|\mathbf{r}_\odot - \mathbf{r}\|} \right) \tag{3.147}$$

If $D' \geq R'_\odot + R'_B$, then the spacecraft is not in the body's shadow and

$$p = 0; \tag{3.148}$$

If $D' \leq R'_B - R'_\odot$, then the spacecraft is in full shadow and

$$p = 100; \tag{3.149}$$

If neither of the above conditions are met, the spacecraft is in partial shadow.

If $|R'_s - R'_B| < D' < R'_s + R'_B$, then we can calculate the percentage of shadow by calculating the area of overlap, A, of the two apparent disks as shown in Fig. 3.8.

$$A = R'^2_\odot \cos^{-1} \left(\frac{c_1}{R'_\odot} \right) + R'^2_B \cos^{-1} \left(\frac{D' - c_1}{R'_B} \right) - D' c_2 \tag{3.150}$$

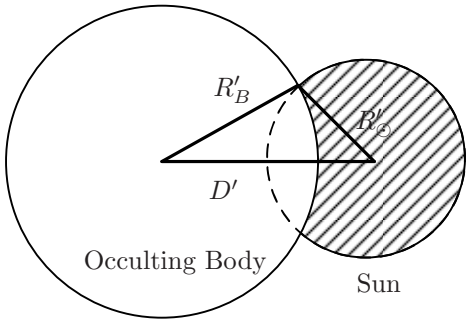


Figure 3.8: Occultation Geometry in Calculation of PercentShadow

where

$$c_1 = \frac{D'^2 + R'^2_\odot - R'^2_B}{2D'} \tag{3.151}$$

and

$$c_2 = \sqrt{R'^2_\odot - c_1^2} \tag{3.152}$$

The percent shadow can be calculated using

$$p = 100\frac{A}{\pi R'^2_\odot} \tag{3.153}$$

If the condition $|R'_\odot - R'_B| < D' < R'_\odot + R'_B$ is not satisfied, then the eclipse is annular and we use

$$p = 100\frac{R'^2_B}{R'^2_\odot} \tag{3.154}$$

3.2.29 RA

Description: `RA` is the right ascension of a spacecraft, as shown in Fig. 3.2 using the symbol λ.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$
Find: λ

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\lambda = \tan_2^{-1}(y, x) \tag{3.155}$$

3.2.30 RAV

Description: `RAV` is the right ascension of velocity of a spacecraft, as shown in Fig. 3.2 using the symbol λ_v.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$
Find: λ_v

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\lambda_v = \tan_2^{-1}(v_y, v_x) \tag{3.156}$$

3.2.31 RAAN

Description: `RAAN` is the right ascension of the ascending node as shown in Fig. 3.1 using the symbol Ω.
Dependency: Coordinate System.
Given: $\mathbf{r}$, $\mathbf{v}$, and coordinate system $\mathcal{F}$.
Find: e

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.157}$$

$$h = \|\mathbf{h}\| \tag{3.158}$$

$$\mathbf{n} = [\ 0 \quad 0 \quad 1\ ]^T \times \mathbf{h} \tag{3.159}$$

$$n = \|n\| \tag{3.160}$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{3.161}$$

if $(i \geq 10^{-11})$, then

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{3.162}$$

Fix quadrant for Ω: if $n_y < 0$, then $\Omega = 2\pi - \Omega$
if $(i < 10^{-11})$, then

$$\Omega = 0 \tag{3.163}$$

3.2.32 RadApo

Given: a, and e
Find: r_a
if $1 - e < 10^{-12}$ then $r_a = 0$. Note, this means that for parabolica, and hyperbolic orbits, GMAT outputs a value of zero for `RadApo`. Otherwise,

$$r_a = a(1 + e) \tag{3.164}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 3.1.2.

3.2.33 RadPer

Given: a, and e
Find: r_p

$$r_p = a(1 - e) \tag{3.165}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 3.1.2.

3.2.34 RLA and DLA

Description: `RLA` (λ_s) is the right ascenstion of the outgoing asypmptote of a hyperbolic trajectory. `DLA` (δ_s)is the declination of the outgoing asypmptote of a hyperbolic trajectory.
Dependency: Coordinate System.
Given: $\mathbf{r}$ and $\mathbf{v}$ and desired coordinate system.
Find: λ_s and δ_s

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to the desired coordinate system (an excpeption is thrown if the origin of the requested coordinate system is not a celestial body). The eccentricity vector $\mathbf{e}$ is computed using Eq. (3.111). If $\|\mathbf{e}\| < 1 + tol$ then RLA = DLA = NaN and return. Otherwise:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.166}$$

$$h = \|\mathbf{h}\| \tag{3.167}$$

$$C_3 = v^2 - \frac{2\mu}{r} \tag{3.168}$$

where μ is gravitational parameter of the central body at the origin of the given coordinate system. The outgoing asmympote unit vector, $\hat{\mathbf{s}}$, is computed using

$$\hat{\mathbf{s}} = \frac{1}{1 + C_3 \left(\frac{h}{\mu}\right)^2} \left(\frac{\sqrt{C_3}}{\mu} (\mathbf{h} \times \mathbf{e}) - \mathbf{e} \right) \tag{3.169}$$

RLA and DLA are computed from

$$\lambda_s = \tan^{-1}(s_y, s_x) \tag{3.170}$$

$$\delta_s = \sin^{-1}(s_z) \tag{3.171}$$

3.2.35 RMAG

Description: `RMAG` is the magnitude of the spacecraft's position vector.
Dependency: Central Body.
Given: $\mathbf{r}$ and central body.
Find: r

Begin by converting $\mathbf{r}$ to a coordinate system with the origin equal to the central body defined by the user, and the MJ2000Eq axis system. Then,

$$r = \|\mathbf{r}\| \tag{3.172}$$

3.2.36 SemilatusRectum

Description: `SemilatusRectum` is the orbit semilatus rectum, which is the magnitude of the position vector, when at true anomaly of 90°.
Dependency: Central Body.
Given: $\mathbf{r}$, $\mathbf{v}$, and μ (central body).
Find: p

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to a coordinate system with the origin equal to the central body defined by the user, and the MJ2000Eq axis system. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.173}$$

$$h = \|\mathbf{h}\| \tag{3.174}$$

$$p = \frac{h^2}{\mu} \tag{3.175}$$

3.2.37 SMA

Description: `SMA` is the semimajor axis of an orbit. The SMA contains information on the size and type of an orbit. If the SMA is positive, the orbit is elliptic. If the SMA is negative the orbit is hyperbolic. The SMA is undefined for parabolic orbits. The algorithm used in GMAT to calculate SMA is adopted from Vallado.[3]
Dependency: Central Body.
Given: $\mathbf{r}$, $\mathbf{v}$, and μ (Central Body)
Find: a

$$r = \|\mathbf{r}\| \tag{3.176}$$

$$v = \|\mathbf{v}\| \tag{3.177}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.178}$$

if $|1-e| > 10^{-30}$, then

$$a = -\frac{\mu}{2\xi} \tag{3.179}$$

otherwise, report error and return. Error: "Warning: Orbit is near parabolic and SMA is undefined".

3.2.38 TA

Description: `TA` is the orbit true anomaly as shown in Fig. 3.1 using the symbol ν.
Dependency: Central Body.
Given: $\mathbf{r}$, $\mathbf{v}$, and coordinate system $\mathcal{F}$.
Find: ν

Begin by converting $\mathbf{r}$ and $\mathbf{v}$ to $\mathcal{F}$ if necessary. Then,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{3.180}$$

$$h = \|\mathbf{h}\| \tag{3.181}$$

$$\mathbf{n} = [\ 0 \quad 0 \quad 1\ ]^T \times \mathbf{h} \tag{3.182}$$

$$n = \|n\| \tag{3.183}$$

$$r = \|\mathbf{r}\| \tag{3.184}$$

$$v = \|\mathbf{v}\| \tag{3.185}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu} \tag{3.186}$$

$$e = \|\mathbf{e}\| \tag{3.187}$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{3.188}$$

There are three special cases, and they are treated differently.
Special Case 1: Elliptic Orbit
if $(e \geq 10^{-11})$, then

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) \tag{3.189}$$

Fix quadrant for ν: if $\mathbf{r}\cdot\mathbf{v} < 0$, then $\nu = 2\pi - \nu$
Special Case 2: Circular, Inclined Orbit
if $(e < 10^{-11})$ and $(i \geq 10^{-11})$, then

$$\nu = \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{r}}{nr}\right) \tag{3.190}$$

Fix quadrant for ν: if $r_z < 0$, then $\nu = 2\pi - \nu$
Special Case 3: Circular, Equatorial Orbit
if $(e < 10^{-11})$ and $(i < 10^{-11})$, then

$$\nu = \cos^{-1}\left(\frac{r_x}{r}\right) \tag{3.191}$$

Fix quadrant for ν: if $r_y < 0$, then $\nu = 2\pi - \nu$

3.2.39 TAIModJulian

Description: `TAIModJulian` is the epoch in the TAI time system, expressed in the modified Julian date format. See Sec. 1.1.1 and 1.2.1 for more details.
Dependency: None.
Given: $A1$ (epoch in the internal, A1 time system).
Find: TAI

To convert from A1 to TAI we use the following equation

$$TAI = A1 - 0.0343817\text{sec} \tag{3.192}$$

3.2.40 TTModJulian

Description: `TTModJulian` is the epoch in the TT time system, expressed in the modified Julian date format. See Sec. 1.1.3 and 1.2.1 for more details.
Dependency: None.
Given: $A1$ (epoch in the internal, A1 time system).
Find: TT

To convert from A1 to TT we use the following equation

$$TT = A1 - 0.0343817\text{sec} + 32.184\text{sec} \tag{3.193}$$

3.2.41 TTGregorian

Description: `TTGregorian` is the epoch in the TT time system, expressed in the Gregorian date format. See Sec. 1.1.3 and 1.2.2 for more details.
Dependency: None.
Given: $A1$ (epoch in the internal, A1 time system).
Find: TT

To convert from A1 to TT we use Eq. (3.193). Then, knowing the epoch in the TT time system in the modified Julian date format, we use the algorithm in Sec. 1.2.1 to obtain the Gregorian date.

3.2.42 Umbra and Penumbra

The Umbra and Penumbra parameters are used to determine if a spacecraft is in the shadow of a celestial body. The algorithm used in GMAT is adapted from Montenbruck[10] pgs. 80-81. For both functions, if the value is less than 1, then the body is in shadow, if the function is greater than 1, then the body is not in shadow.

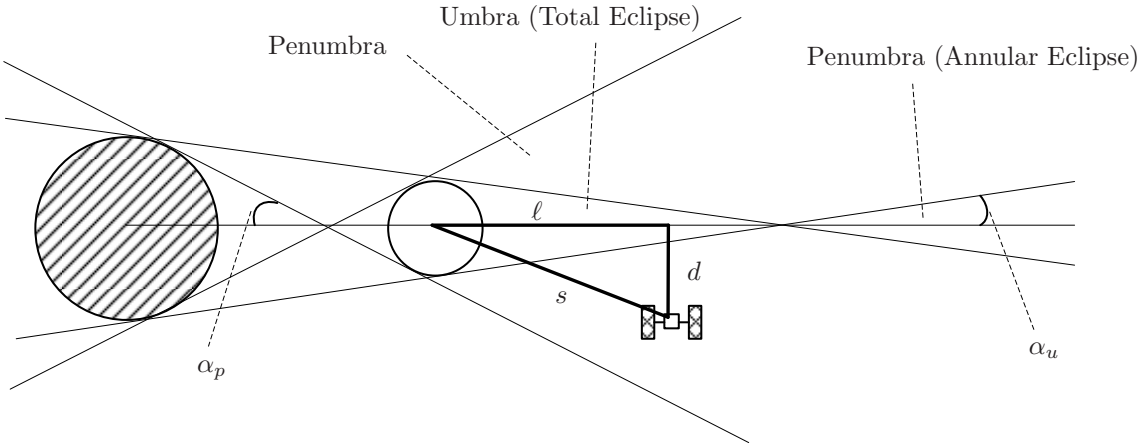


Figure 3.9: Geometry of Umbra and Penumbra Regions

For definitions of see Sec. 3.2.28.

$$\ell = \frac{-\mathbf{s}^T \mathbf{s}_\odot}{s_\odot} \tag{3.194}$$

$$d = \sqrt{s^2 - l^2} \tag{3.195}$$

$$\sin\alpha_p = \frac{R_\odot + R_B}{s_\odot} \tag{3.196}$$

$$\sin\alpha_u = \frac{R_\odot - R_B}{s_\odot} \tag{3.197}$$

The radii of the umbra and penumbra cones, r_p and r_u, at distance ℓ, are respectively

$$r_p = \tan\alpha_p \left(\ell + \frac{R_B}{\sin\alpha_p} \right) \tag{3.198}$$

$$r_u = \tan\alpha_u \left(\ell - \frac{R_B}{\sin\alpha_u} \right) \tag{3.199}$$

Finally, if $\ell \geq 0$

$$d_p = d - r_p \tag{3.200}$$

$$d_u = d - |r_u| \tag{3.201}$$

If $\ell > 0$ $d_u < 0$ and $r_u < 0$, then the object is in the total umbral eclipse region.
If $\ell > 0$ $d_u < 0$ and $r_u \geq 0$, then the object is in the annular umbral eclipse region.
If $\ell < 0$, then the object is on the day side of the occulting body and is not in shadow and

$$d_p = |d - r_p| \tag{3.202}$$

$$d_u = |d - |r_u|| \tag{3.203}$$

3.2.43 UTCModJulian

Description: `UTCModJulian` is the epoch in the UTC time system, expressed in the modified Julian date format. See Sec. 1.1.2 and 1.2.1 for more details.
Dependency: None.
Given: $A1$ (epoch in the internal, A1 time system).
Find: UTC

To convert from A1 to UTC we use the following equation

$$UTC = A1 - 0.0343817\text{sec} - \Delta AT \tag{3.204}$$

The default is to read ΔAT from the file named *tai-utc.dat*. ΔAT is the accumulated leap seconds since Jan. 1961.

3.2.44 VelApoapsis

Given: a, e, and μ
Find: v_a
If e > (1 - $1e^{-12}$) then $v_a = 0$.
Otherwise,

$$v_a = \sqrt{\frac{\mu}{a}\left(\frac{1-e}{1+e}\right)} \tag{3.205}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 3.1.2, and μ is associated with the specified central body.

3.2.45 VelPeriapsis

Given: a, e, and μ
Find: v_p

$$v_p = \sqrt{\frac{\mu}{a}\left(\frac{1+e}{1-e}\right)} \tag{3.206}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 3.1.2, and μ is associated with the specified central body.

3.2.46 VMAG

Description: `VMAG` is the magnitude of the spacecraft's velocity vector, when the velocity is expressed in the chosen coordinate system.
Dependency: Coordinate System.
Given: $\mathbf{v}$ and coordinate system $\mathcal{F}$.
Find: v

Begin by converting $\mathbf{v}$ to coordinate system $\mathcal{F}$ if necessary. Then,

$$v = \|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{3.207}$$

3.3 Other Calculations

3.3.1 MA to TA

Description: This algorithm shows how to calculate ν given M and e and is taken from Vallado.[3]
Given: M and e.
Find: ν

The algorithm is different for elliptic and hyperbolic orbits. Let's first look at what happens for elliptic orbits.
Elliptic Orbit Case
If $e <= 1$ then use the following algorithm:
Determine initial guess for the Eccentric anomaly
If (- $\pi < M < 0$) or $M > \pi$
$E = M - e$
Else
$E = M + e$
End
Iterate to determine the eccentric anomaly:

Iterate On: $E_{n+1} = E_n + \dfrac{M - E_n + e\sin E_n}{1 - e\cos E_n}$
Until: $|E_{n+1} - E_n| < 1e^{-8}$
Finally we convert the eccentric anomaly to the true anomaly using the algorithm given in sec. 3.3.2
Hyperbolic Orbit Case
If $e > 1$ then use the following algorithm:

We begin by choosing the initial guess for the hyperbolic anomaly. The initial guess depends on the value of the mean anomaly and the eccentricity:
If $e < 1.6$
If (- $\pi < M < 0$) or $M > \pi$
$H = M - e$
Else
$H = M + e$
End
Else
If ($e < 3.6$ & $|M| > \pi$)
$H = M - \text{sign}(M)e$
Else
$H = \frac{M}{e-1}$
End
End
Iterate to determine the Hyperbolic Anomaly:
Iterate On: $H_{n+1} = H_n + \dfrac{M + H_n - e\sinh H_n}{e\cosh H_n - 1}$
Until: $|H_{n+1} - H_n| < 1e^{-8}$
Convert the hyperolic anomaly to the true anomaly using the algorithm given in sec. 3.3.3

3.3.2 EA to TA

Description: This algorithm shows how to calculate ν given E and e and is taken from Vallado.[3]
Given: E and e.
Find: ν

$$\sin\nu = \frac{\sqrt{1-e^2}\sin(E)}{1-e\cos E} \tag{3.208}$$

$$\cos\nu = \frac{\cos E - e}{1-e\cos E} \tag{3.209}$$

$$\nu = \text{atan2}(\sin\nu, \cos\nu) \tag{3.210}$$

3.3.3 HA to TA

Description: This algorithm shows how to calculate ν given H and e and is taken from Vallado.[3]
Given: H and e.
Find: ν

$$\sin\nu = -\frac{\sqrt{e^2-1}\sinh(H)}{1-e\cosh H} \tag{3.211}$$

$$\cos\nu = \frac{\cosh H - e}{1-e\cosh H} \tag{3.212}$$

$$\nu = \text{atan2}(\sin\nu, \cos\nu) \tag{3.213}$$

3.4 Libration Points

We begin by assuming that the planets move in circular orbits about the sun, and the mass of a spacecraft is negligible compared to the mass of the planets. For illustrative purposes, lets consider the Earth and its orbit about the Sun. In this case, the libration points are locations in space where a spacecraft will stay fixed with respect to the Earth and Sun. Figure 3.10 shows a simple illustration. We see the Sun, the Earth's position with respect to the Sun, and the Libration points L_1 and L_2 at two different epochs. Notice that at t_1, the points L_1 and L_2 are on the Earth-Sun line. At a later time, t_2, although the Earth has moved with respect to the sun, L_1 and L_2 still lie on the Earth-Sun line.

The preceding example gives a brief qualitative description of two of the Earth-Sun libration points. In general, there are five libration points for a given three body system. To determine the locations of the libration points, it is convenient to work in a rotating coordinate system rather than the inertial system shown in Fig. 3.10. The system we use is constructed as follows:

- Define the primary as the heavier of the two bodies, the secondary as the lighter.
- Define the coordinate system x-axis as the axis pointing from the primary to the secondary.
- Define the y-axis to be orthogonal to the x-axis in the plane of the secondary's motion about the primary, pointing in the direction the secondary moves about the primary.
- Define the z-axis orthogonal to the x and y axes to form a right-handed system.
- Place the origin at center-of-mass of the system.

This coordinates system is illustrated in Fig. 3.11. The locations of the libration points in the rotating coordinate system can be found by calculating the values of γ that solve the following equations:

$$\begin{aligned}\gamma_1^5 - (3-\mu^*)\gamma_1^4 + (3-2\mu^*)\gamma_1^3 - \mu^*\gamma_1^2 \\ + 2\mu^*\gamma_1 - \mu^* = 0 \quad \text{(For L1)}\end{aligned} \tag{3.214}$$

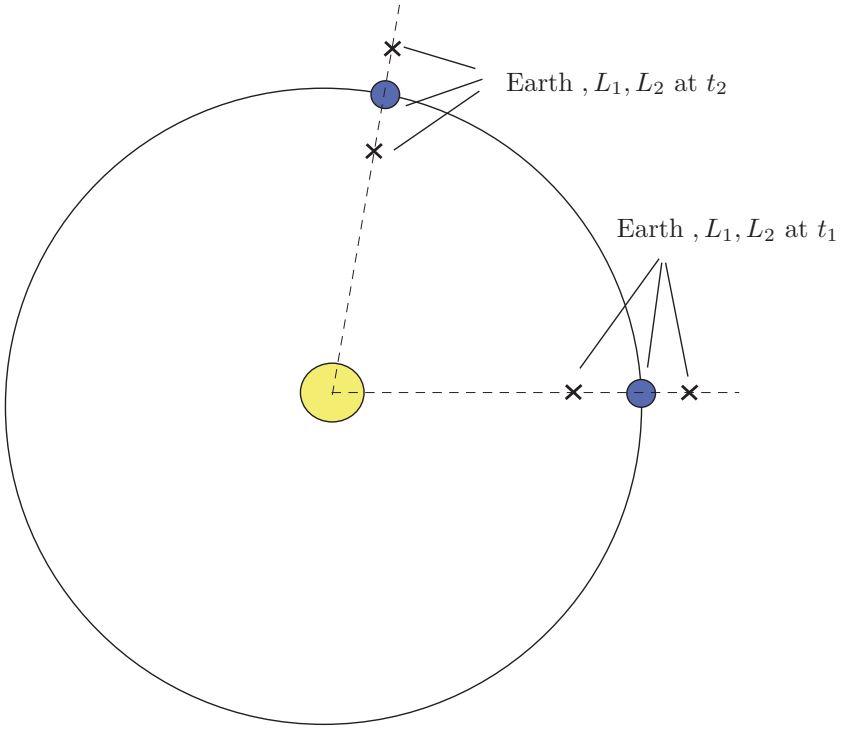


Figure 3.10: Geometry of Libration Points

$$\gamma_2^5 + (3-\mu^*)\,\gamma_2^4 + (3-2\mu^*)\,\gamma_2^3 - \mu^*\gamma_2^2 - 2\mu^*\gamma_2 - \mu^* = 0 \quad \text{(For L2)} \tag{3.215}$$

$$\gamma_3^5 + (2+\mu^*)\,\gamma_3^4 + (1+2\mu^*)\,\gamma_3^3 - (1-\mu^*)\,\gamma_3^2 - 2\,(1-\mu^*)\,\gamma_3 - (1-\mu^*) = 0 \quad \text{(For L3)} \tag{3.216}$$

where

$$\mu^* = \frac{m_2}{m_1 + m_2} \tag{3.217}$$

Equations (3.214)-(3.216) do not have exact analytic solutions.Szebehely[11] notes that they are most easily solved using an iterative method with the following as the initial guesses:

$$\gamma_1 = \gamma_2 = \left(\frac{\mu^*}{3\,(1-\mu^*)}\right)^{1/3} \tag{3.218}$$

$$\gamma_3 = 1 \tag{3.219}$$

GMAT uses the Newton-Raphson method to solve for the roots of the equations by iterating on

$$\gamma(i+1) = \gamma(i) - \frac{F(\gamma(i))}{F'(\gamma(i))} \tag{3.220}$$

until the the difference $|\gamma(i+1) - \gamma(i)| < 10^{-8}$. The derivative $F'(\gamma)$ for each libration point is shown below.

$$F'(\gamma) = 5\gamma_1^4 - 4\,(3-\mu^*)\gamma_1^3 + 3\,(3-2\mu^*)\,\gamma_1^2 - 2\mu^*\gamma_1 + 2\mu^* \quad \text{(For L1)} \tag{3.221}$$

$$F'(\gamma) = 5\gamma_2^4 + 4\,(3-\mu^*)\gamma_2^3 + 3\,(3-2\mu^*)\,\gamma_2^2 - 2\mu^*\gamma_2 - 2\mu^* \quad \text{(For L2)} \tag{3.222}$$

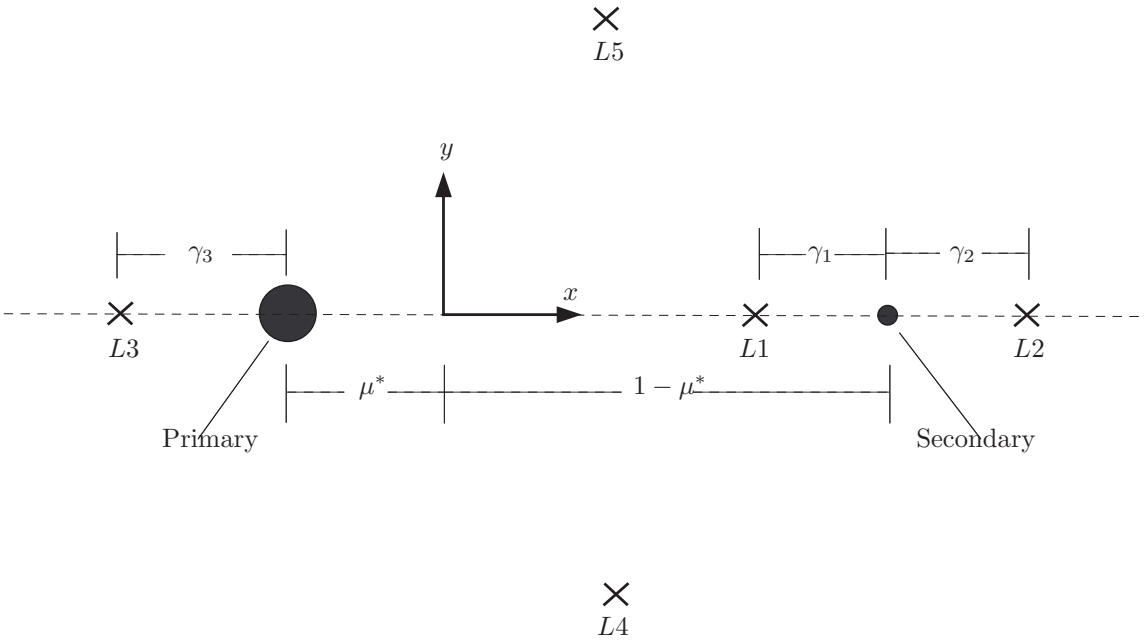


Figure 3.11: Location of Libration Points

$$F'(\gamma) = 5\gamma_3^4 + 4\left(2+\mu^*\right)\gamma_3^3 + 3\left(1+2\mu^*\right)\gamma_3^2 - 2\left(1-\mu^*\right)\gamma_3 - 2\left(1-\mu^*\right) \quad \text{(For L3)} \tag{3.223}$$

Table 3.7: Location of Libration Points in RLP Frame, with the Origin at the Primary Body

Point	x-Position	y-Position
L1	$1-\gamma_1$	0
L2	$1+\gamma_2$	0
L3	$-\gamma_3$	0
L4	1/2	$\sqrt{3}/2$
L5	1/2	$-\sqrt{3}/2$

We now need to redimensionalize the results found in the rotating system, and perform the necessary transformations to obtain the results in the MJ2000 system. Let's assume that $\mathbf{r}_s$, $\mathbf{v}_s$, and $\mathbf{a}_s$ are the position, velocity, and acceleration vectors respectively of the secondary body, with respect to the primary body, expressed in the FK5 system. Then, the position of the i^{th} libration point can be expressed in the rotating system with the origin centered on the primary body as

$$\mathbf{r}^i = r_s \begin{bmatrix} x_i & y_i & 0 \end{bmatrix}^T \tag{3.224}$$

where

$$r_s = \|\mathbf{r}_s\| \tag{3.225}$$

The velocity of the i^{th} libration point can be expressed in the rotating system with the origin centered on the primary body as

$$\mathbf{v}^i = \frac{\mathbf{v}_s \cdot \mathbf{r}_s}{r_s} \begin{bmatrix} x_i & y_i & 0 \end{bmatrix}^T \tag{3.226}$$

Now we have the redimensionalized position and velocity vectors of the libration point in the rotating coordinate system defined by the motion of the secondary body with respect to the primary body. To determine the position and velocity vectors in the FK5 system, with the origin located at the primary body,

we need to determine the rotation matrix and its derivative as follows:

$$\mathbf{R}^{Ii} = \begin{pmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 \\ \hat{x}_2 & \hat{y}_2 & \hat{z}_2 \\ \hat{x}_3 & \hat{y}_3 & \hat{z}_3 \end{pmatrix} \tag{3.227}$$

and

$$\dot{\mathbf{R}}^{Ii} = \begin{pmatrix} \dot{\hat{x}}_1 & \dot{\hat{y}}_1 & \dot{\hat{z}}_1 \\ \dot{\hat{x}}_2 & \dot{\hat{y}}_2 & \dot{\hat{z}}_2 \\ \dot{\hat{x}}_3 & \dot{\hat{y}}_3 & \dot{\hat{z}}_3 \end{pmatrix} \tag{3.228}$$

where

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_s}{r_s} \tag{3.229}$$

$$\hat{\mathbf{z}} = \frac{\mathbf{r}_s \times \mathbf{v}_s}{\|\mathbf{r}_s \times \mathbf{v}_s\|} \tag{3.230}$$

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{3.231}$$

and

$$\dot{\hat{\mathbf{x}}} = \dot{\hat{\mathbf{r}}}_s = \frac{\mathbf{v}_s}{r_s} - \frac{\hat{\mathbf{r}}_s}{r_s} \left(\hat{\mathbf{r}}_s \cdot \mathbf{v}_s\right) \tag{3.232}$$

$$\dot{\hat{\mathbf{z}}} = \frac{\mathbf{r}_s \times \mathbf{a}_s}{\|\mathbf{r}_s \times \mathbf{v}_s\|} - \frac{\hat{\mathbf{z}}}{\|\mathbf{r}_s \times \mathbf{v}_s\|} \left(\mathbf{r}_s \times \mathbf{a}_s \cdot \hat{\mathbf{z}}\right) \tag{3.233}$$

$$\dot{\hat{\mathbf{y}}} = \dot{\hat{\mathbf{z}}} \times \hat{\mathbf{x}} + \hat{\mathbf{z}} \times \dot{\hat{\mathbf{x}}} \tag{3.234}$$

GMAT currently assumes that the terms $\mathbf{r}_s \times \mathbf{a}_s$ are zero.

We finally arrive at the position of the Libration Point in the FK5 system with the origin at the primary by performing the calculations:

$$\mathbf{r} = \mathbf{R}^{Ii}\mathbf{r}^i \tag{3.235}$$

$$\mathbf{v} = \dot{\mathbf{R}}^{Ii}\mathbf{r}^i + \mathbf{R}^{Ii}\mathbf{v}^i \tag{3.236}$$

3.5 Barycenter

The barycenter of a system of point masses, $\mathbf{r}_b$, is also called the center of mass. If we have a system of n bodies, and we know the position of the i^{th} body with respect to a common reference system, then we can calculate the barycenter of the system using

$$\mathbf{r}_b = \frac{\sum_{i=1}^{n} m_i \mathbf{r}_i}{\sum_{i=1}^{n} m_i} \tag{3.237}$$

Similarly, we can calculate the velocity of the barycenter using the following equation

$$\mathbf{v}_b = \frac{\sum_{i=1}^{n} m_i \mathbf{v}_i}{\sum_{i=1}^{n} m_i} \tag{3.238}$$

3.6 Ground Station Model

This section contains algorithms for converting the station location parameters to body fixed and inertial cartesian coordinates. Ground stations are defined in the body fixed frame of the `CentralBody`. The user can define the ground system location using several methods. GMAT internally uses the cartesian location of the ground station with respect to the body fixed coordinate system. If the user provides the ground station location in cartesian elements, no transformation is required to obtain the body fixed representation. For spherical elements with respect to the Geocentric reference, GMAT uses the following transformation to calculate the cartesian station location in the body fixed frame,

$$\mathbf{r}_F = \begin{pmatrix} \cos\phi\cos\lambda \\ \cos\phi\sin\lambda \\ \sin\phi \end{pmatrix} \tag{3.239}$$

If the station location is defined in terms of the Geodetic reference, then the cartesian location is calculated as follows:[12]

$$C = \frac{R_b}{\sqrt{1 - e_b^2 \sin^2\phi}} \tag{3.240}$$

$$S = C\left(1 - e_b^2\right) \tag{3.241}$$

$$r_{xy} = (C + h_{e\ell\ell})\cos\phi \tag{3.242}$$

$$r_z = (S + h_{e\ell\ell})\sin\phi \tag{3.243}$$

$$\mathbf{r}_F = \begin{pmatrix} r_{xy}\cos\lambda \\ r_{xy}\sin\lambda \\ r_z \end{pmatrix} \tag{3.244}$$

To calculate the body-centered MJ2000Eq representation, we calculate the rotation matrix from fixed to inertial, $\mathbf{R}_{IF}$, using the algorithm in Sec 2.6.10 of the GMAT Mathematical Specification. Then,

$$\mathbf{r}_I = \mathbf{R}_{IF}\mathbf{r}_F \tag{3.245}$$

Chapter 4

Dynamics Modelling

In this chapter, we present the mathematical forumulation of the orbit and attitude dynamcis models in GMAT. One of the fundamental capabilities of GMAT is to model the motion of spacecraft in different flight regimes. The flight regime, such as low Earth, libration point, or lunar, are determined by the forces and perturbations that dominate the dynamics. The chapter begins with an overview of the orbital equations of motion and their variational equations. Next, we discuss the formulation for orbital perturbations and the equations used to model spacecraft thrust. The second half of the chapter is devoted to attitude modeling and we present the specifications for attitude conversions and kinematic attitude models.

4.1 Orbit Dynamics

4.1.1 Orbital Equations of Motion

The orbital equations of motion come from an application of Newton's laws of motion to a spacecraft in orbit. From Newton's Second Law we know that

$$\frac{d(m\mathbf{v})}{dt} = \sum F_{ext} \tag{4.1}$$

where m is the total mass of the spacecraft, $\mathbf{r}$ is the position vector of the spacecraft, t is time, and the right hand side of the equation represents the total sum of external forces. Solving for the acceleration gives us the second order differential equation

$$\frac{d^2\mathbf{r}}{d^2t} = \sum \frac{F_{ext}}{m} - \frac{\dot{m}}{m}\frac{\partial \mathbf{r}}{\partial t} \tag{4.2}$$

The terms included in the RHS of the equations of motion can be selected by the user, and the form of several terms (and whether they appear at all) are dependent upon the coordinate system of integration. If we include all of the possible forces GMAT can model in the summation on the RHS of Eq. (4.2), and we assume the origin of integration is a celestial body (the barycentric form is slightly different), then the orbital equations of motion are

$$\begin{aligned}\frac{d^2\mathbf{r}}{dt^2} = &-\frac{\mu}{r^3}\mathbf{r} + \nabla\phi^o_{sj} + G\sum_{\substack{k=1\\k\neq j}}^{n_b} m_k\left(\frac{\mathbf{r}_{ks}}{r^3_{ks}} - \frac{\mathbf{r}_{kj}}{r^3_{kj}}\right) + \frac{\dot{m}_s}{m}\frac{d\mathbf{r}}{dt} - \frac{1}{2}\rho v^2_{rel}\frac{C_d A}{m_s}\hat{\mathbf{v}}_{rel} + \frac{P_{SR}C_R A_\odot}{m_s}\hat{\mathbf{r}}_{s\odot} + \\ &\frac{\mu}{c^2r^3}\left(\left(4\frac{\mu}{r} - v^2\right)\mathbf{r} + 4(\mathbf{r}\cdot\mathbf{v})\mathbf{v}\right) + 2(\boldsymbol{\Omega}\times\mathbf{v}) + 2\frac{\mu}{c^2r^3}\left(\frac{3}{r^2}(\mathbf{r}\times\mathbf{v})(\mathbf{r}\cdot\mathbf{J}) + (\mathbf{v}\times\mathbf{J})\right)\end{aligned} \tag{4.3}$$

Table 4.1 below describes each force in the equation above.

Table 4.1: Force Models Available in GMAT

Description	Term
Central Body Point Mass	$-\frac{\mu}{r^3}\mathbf{r}$
Central Body Direct Nonspherical	$\nabla\phi^o_{sj}$
Direct Third Body Point Mass	$G\sum_{\substack{k=1\\k\neq j}}^{n_b} m_k\left(\frac{\mathbf{r}_{ks}}{r^3_{ks}}\right)$
Indirect Third Body Point Mass	$G\sum_{\substack{k=1\\k\neq j}}^{n_b} m_k\left(-\frac{\mathbf{r}_{kj}}{r^3_{kj}}\right)$
Spacecraft Thrust	$\frac{\dot{m}_s}{m}\frac{d\mathbf{r}}{dt}$
Atmospheric Drag	$-\frac{1}{2}\rho v^2_{rel}\frac{C_dA}{m_s}\hat{\mathbf{v}}_{rel}$
Solar Radiation Pressure	$\frac{P_{SR}C_RA_\odot}{m_s}\hat{\mathbf{r}}_{s\odot}$
Schwarzschild solution	$\frac{\mu}{c^2r^3}\left(\left(4\frac{\mu}{r}-v^2\right)\mathbf{r}+4(\mathbf{r}\cdot\mathbf{v})\mathbf{v}\right)+2(\boldsymbol{\Omega}\times\mathbf{v})$
Geodesic Precession	$2(\boldsymbol{\Omega}\times\mathbf{v})$
Lense-Thirring Precession	$2\frac{\mu}{c^2r^3}\left(\frac{3}{r^2}(\mathbf{r}\times\mathbf{v})(\mathbf{r}\cdot\mathbf{J})+(\mathbf{v}\times\mathbf{J})\right)$

4.1.2 Coordinate Systems for Integration of the Equations of Motion

4.1.3 Orbit Variational Equations and the State Transition Matrix

Estimation and optimization problems require first derivatives of the solution to the orbit final value problem with respect to orbital initial conditions. Those derivatives are provided by the orbit variational equations. The variatial equations are obtained by expanding the orbital equations of motion in a Taylor series and retaining only the linear terms. Assume the nonlinear dynamics have the following form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{4.4}$$

where

$$\mathbf{x} = \begin{bmatrix}\mathbf{r}^T & \mathbf{v}^T\end{bmatrix}^T = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{4.5}$$

and

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{r}}^T & \dot{\mathbf{v}}^T\end{bmatrix}^T = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \ddot{x} \quad \ddot{y} \quad \ddot{z}]^T \tag{4.6}$$

Expanding the dynamics equations and retaining only the first order terms yields

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}|_{ref} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\left(\mathbf{x} - \mathbf{x}|_{ref}\right) \tag{4.7}$$

We can rewrite Eq. (4.7) by defining $\delta\mathbf{x} = \dot{\mathbf{x}} - \dot{\mathbf{x}}|_{ref}$ as follows.

$$\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} \tag{4.8}$$

where the Jacobian of the state equations is defined as the "A" matrix as follows:

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{4.9}$$

From linear systems theory, the solution to Eq. (4.8) has the form

$$\delta\mathbf{x}(t_f) = \mathbf{\Phi}(t, t_o)\delta\mathbf{x}(t_o) \tag{4.10}$$

where $\dot{\mathbf{\Phi}}$ is governed by the following set of 36 differential equations:

$$\dot{\mathbf{\Phi}} = \mathbf{A}\mathbf{\Phi} \tag{4.11}$$

GMAT simultaneously integrates the variational equations along with the non-linear state equations from t_o to t_f to obtain the State Transition Matrix (STM):

$$\mathbf{\Phi}(t_o, t_f) = \int_{t_o}^{t_f} \mathbf{A}\mathbf{\Phi} dt \tag{4.12}$$

subject to the initial conditions

$$\mathbf{\Phi}(t_o, t_o) = \mathbf{I}_{6\times 6} \tag{4.13}$$

To perform the integration in Eq. (4.12) we require the partial derivatives contained in the matrix $\mathbf{A}$:

$$\mathbf{A} = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} = \begin{pmatrix} \dfrac{\partial \mathbf{v}}{\partial \mathbf{r}} & \dfrac{\partial \mathbf{v}}{\partial \mathbf{v}} \\ \dfrac{\partial \mathbf{a}}{\partial \mathbf{r}} & \dfrac{\partial \mathbf{a}}{\partial \mathbf{v}} \end{pmatrix} \tag{4.14}$$

Two of the derivatives are trivial:

$$\frac{\partial \mathbf{v}}{\partial \mathbf{r}} = \mathbf{0}_{3x3} \tag{4.15}$$

$$\frac{\partial \mathbf{v}}{\partial \mathbf{v}} = \mathbf{I}_{3x3} \tag{4.16}$$

The remaining two terms, $\partial\mathbf{a}/\partial\mathbf{r}$ and $\partial\mathbf{a}/\partial\mathbf{v}$, are dependent upon the specific perturbations included in the force model. The partial derivatives of perturbations are provided in the sections containing the specific formulation of the perturbing force.

4.1.4 Multiple Spacecraft Propgation and Coupled Propagation of the Equations of Motion

4.2 Force Modelling

4.2.1 n-Body Point Mass Gravity

The gravitational perturbation due to n point masses is well know. However, we will derive the governing differential equation here, as well as the componenents of the sensitivity matrix. Let's begin by defining some notation referring to Fig.4.1. Assume the j^{th} body is the central body of the integration.

- $\tilde{\mathbf{r}}_s$ is the position of the spacecraft with respect a hypothesized inertial frame.
- $\tilde{\mathbf{r}}_j$ is the position of the central body with respect a hypothesized inertial frame.
- $\tilde{\mathbf{r}}_k$ is the position of the k^{th} gravitational body with respect a hypothesized inertial frame.
- $\mathbf{r}$ is the position of the spacecraft with respect to the central body of integration (j^{th} body).
- $\mathbf{r}_k$ is the position of the k^{th} gravitational body with respect to the central body.

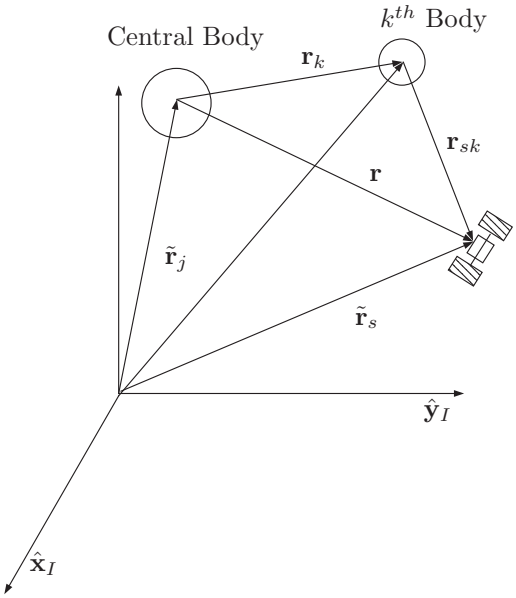


Figure 4.1: N-Body Illustration

We need the governing differential equation that describes the motion of the spacecraft with respect to the central body. However, we know that we must apply Newton's 2nd Law in an inertial frame. So, we begin by defining the relative position of the spacecraft with respect to the central body. From inspection of Fig.4.1 we see that

$$\tilde{\mathbf{r}}_j + \mathbf{r} = \tilde{\mathbf{r}}_s \tag{4.17}$$

By reordering and taking the second derivative with respect to time we obtain

$$\ddot{\mathbf{r}} = \ddot{\tilde{\mathbf{r}}}_s - \ddot{\tilde{\mathbf{r}}}_j \tag{4.18}$$

We can apply Newton's 2nd Law to the spacecraft and obtain

$$m_s \ddot{\tilde{\mathbf{r}}}_s = \sum_{k=1}^{n} F_k = G \sum_{k=1}^{n} \frac{m_s m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} (\mathbf{r}_k - \mathbf{r}) \tag{4.19}$$

where $(\mathbf{r}_k - \mathbf{r})$ is a vector from the spacecraft to the k^{th} body, m_s is the mass of the spacecraft, and m_k is the mass of the k^{th} body. We can write $\ddot{\tilde{\mathbf{r}}}_s$ as simply

$$\ddot{\tilde{\mathbf{r}}}_s = G \sum_{k=1}^{n} \frac{m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} (\mathbf{r}_k - \mathbf{r}) \tag{4.20}$$

We can apply Newton's 2nd Law to the j^{th} body and obtain

$$m_j \ddot{\tilde{\mathbf{r}}}_j = \frac{G m_s m_j}{r^3} \mathbf{r} + G \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{m_j m_k}{\|\mathbf{r}_k\|^3} \mathbf{r}_k \tag{4.21}$$

where the first term is the influence of the spacecraft on the central body, and the second term is the influence of the k point mass gravitational bodies. We can write $\ddot{\tilde{\mathbf{r}}}_j$ as simply

$$\ddot{\tilde{\mathbf{r}}}_j = \frac{G m_s}{r^3} \mathbf{r} + G \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{m_k}{\|\mathbf{r}_k\|^3} \mathbf{r}_k \tag{4.22}$$

Substituting Eq. (4.20) and (4.22) into (4.18) we get

$$\ddot{\mathbf{r}} = G\sum_{k=1}^{n} \frac{m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3}(\mathbf{r}_k - \mathbf{r}) - \frac{Gm_s}{r^3}\mathbf{r} - G\sum_{\substack{k=1\\k\neq j}}^{n} \frac{m_k}{\|\mathbf{r}_k\|^3}\mathbf{r}_k \tag{4.23}$$

Finally, collecting terms yields

$$\mathbf{a}_{pm} = \ddot{\mathbf{r}} = \underbrace{-\frac{\mu_j}{r^3}\mathbf{r}}_{1} + G\sum_{\substack{k=1\\k\neq j}}^{n} m_k \left(\underbrace{\frac{\mathbf{r}_k - \mathbf{r}}{\|\mathbf{r}_k - \mathbf{r}\|^3}}_{2} - \underbrace{\frac{\mathbf{r}_k}{\|\mathbf{r}_k\|^3}}_{3} \right) \tag{4.24}$$

We can break down the acceleration in the equation above into three physical categories. The first term is the acceleration on the spacecraft due to a point mass central body. The second type of terms are called direct terms. They account for the force of the k^{th} body on the spacecraft. The third type of terms are called indirect. They account for the force of the k^{th} body on the central body.

Let's look at the contributions to the sensitivity matrix due to point mass perturbations. We notice that $\mathbf{a}_{pm}$ is not a function of velocity. So,

$$\mathbf{A}_{pm} = \mathbf{D}_{pm} = \mathbf{0}_{3\times 3} \tag{4.25}$$

We also know that

$$\mathbf{B}_{pm} = \mathbf{I}_{3\times 3} \tag{4.26}$$

This leaves $\mathbf{C}_{pm}$ as the only non-trivial term for point mass gravitational effects. Let's look first at the derivatives of the point mass term. We can use the vector identity in Eq. (12.4) to arrive at

$$\frac{\partial}{\partial \mathbf{r}}\left(-\frac{\mu_j}{r^3}\mathbf{r}\right) = -\frac{\mu_j}{r^3}\mathbf{I}_3 + 3\mu_j\frac{\mathbf{r}\mathbf{r}^T}{r^5} \tag{4.27}$$

Similarly, applying Eq. (12.4) to the direct terms we see that

$$\frac{\partial}{\partial \mathbf{r}}\left(\sum_{\substack{k=1\\k\neq j}}^{n} \mu_k \frac{\mathbf{r}_k - \mathbf{r}}{\|\mathbf{r}_k - \mathbf{r}\|^3}\right) = -\sum_{\substack{k=1\\k\neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3}\mathbf{I}_3 + 3\sum_{\substack{k=1\\k\neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5}\right) \tag{4.28}$$

Finally, the derivative of the indirect terms are zero and we have

$$\mathbf{C}_{pm} = \underbrace{-\frac{\mu_j}{r^3}\mathbf{I}_3 + 3\mu_j\frac{\mathbf{r}\mathbf{r}^T}{r^5}}_{1} \underbrace{-\sum_{\substack{k=1\\k\neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3}\mathbf{I}_3 + 3\sum_{\substack{k=1\\k\neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5}\right)}_{2} \tag{4.29}$$

Combining similar terms we can express the result as

$$\mathbf{C}_{pm} = -\left(\frac{\mu_j}{r^3} + \sum_{\substack{k=1\\k\neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3}\right)\mathbf{I}_3 + 3\left(\mu_j\frac{\mathbf{r}\mathbf{r}^T}{r^5} + \sum_{\substack{k=1\\k\neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5}\right)\right) \tag{4.30}$$

4.2.2 Non-Spherical Gravity

GMAT integrates all spacecraft equations of motion using the Earth's Mean J2000 axis system. However, the user can choose central bodies other than the Earth as the origin of the coordinate system of integration. Gravitational forces are conservative and only a function of position. To calculate the gravitational force due to a non-spherical body, we need to determine the position of the spacecraft in the body fixed frame $\mathcal{F}_F$. However, the equations of motion are expressed in terms of the position of the spacecraft in the inertial frame.

We know from dynamics that the acceleration in an inertial frame can be calculated using

$$\mathbf{a}_{cb} = \nabla U \tag{4.31}$$

where U is the gravitational potential. The potential for a nonspherical body comes from the solution to Laplace's equation:

$$\nabla^2 U = 0 \tag{4.32}$$

The solution to this equation is most easily expressed in spherical, body-fixed coordinates because it allows for a convenient separation of variables.

In spherical coordinates the gradient of the gravitational potential is

$$\nabla U = \frac{\partial U}{\partial r}\mathbf{u}_r + \frac{1}{r}\frac{\partial U}{\partial \phi}\mathbf{u}_\phi + \frac{1}{r\cos\phi}\frac{\partial U}{\partial \lambda}\mathbf{u}_\lambda \tag{4.33}$$

We see that there are two singularities in Eq. (4.33). The first is when $r = 0$, which is a nonphysical case and we will not discuss it further. The second singularity occurs when $\phi = \pm 90°$. Pines[13] developed a uniform expression of the gravitational potential that avoids the singularity at the poles:

$$\begin{aligned} U = &\frac{\mu}{r}\left[1 + \sum_{n=1}^{\infty}\left(\frac{R_\otimes}{r}\right)^n \sum_{m=0}^{n} A_{nm}(u)[C_{nm} cos(m\lambda)\cos^m\phi \right. \\ &\left. + S_{nm} sin(m\lambda)\cos^m\phi]\right] \end{aligned} \tag{4.34}$$

Examining this form of the potential it is easy to see that there is not a singularity at the poles when taking the gradient in spherical coordinates. Pines rewrites Eq. (4.34) as

$$\begin{aligned} U = &\frac{\mu}{r}\left[1 + \sum_{n=1}^{\infty}\left(\frac{R_\otimes}{r}\right)^n \sum_{m=0}^{n} A_{nm}(u)[C_{nm} r_m(s,t) \right. \\ &\left. + S_{nm} i_m(s,t)]\right] \end{aligned} \tag{4.35}$$

where C_{nm} and S_{nm} are the gravitational coefficients, s, t, and u are given by

$$s = x/r, \qquad t = y/r, \qquad u = z/r = \sin\phi$$

and $r_m(s,t)$ and $i_m(s,t)$ are calculated using the recursive relationships

$$\begin{aligned} r_0 = 1, \qquad r_1 = s, \qquad i_0 = 0, \qquad i_1 = t \\ r_m = sr_{m-1} - ti_{m-1}, \qquad i_m = si_{m-1} + tr_{m-1} \end{aligned}$$

The coefficients $A_{nm}(u)$ are called "derived" Legendre functions and are given by

$$A_{nm}(u) = \frac{d^m}{du^m}(P_n(u)) \tag{4.36}$$

where we know from Rodrigues'[14] formula that

$$P_{n0}(u) = P_n(u) = \frac{1}{2^n n!}\frac{d^n}{du^n}(u^2 - 1)^n \tag{4.37}$$

and

$$P_{nm}(u) = (1 - u^2)^{m/2}\frac{d^m}{du^m}P_n(u) \tag{4.38}$$

For numerical reasons it is useful to normalize some of the terms in the potential function, U. By normalizing the spherical coefficients and the derived Legendre polynomials we can improve the stability of recursive algorithms used to calculate the Legendre polynomials and improved numerical problems. We

use the nondimensionalization approach and described by Lundberg.[14] Lundberg chooses the normalization factor so that the normalized spherical harmonics $\bar{C}_{nm}$ and $\bar{S}_{nm}$ will have a mean square value of one on the unit sphere. The normalized Legendre functions, $\bar{P}_{nm}$, are defined so that the product of the spherical harmonic coefficients and the corresponding Legendre functions remain constant, or

$$\bar{P}_{nm}\bar{C}_{nm} = P_{nm}C_{nm} \qquad \bar{P}_{nm}\bar{S}_{nm} = P_{nm}S_{nm} \tag{4.39}$$

GMAT uses the normalization factor N_{nm} given by

$$N_{nm} = \left[\frac{(n-m)!(2n+1)!}{(n+m)!}\right]^{1/2} \tag{4.40}$$

The non-dimensional spherical harmonic coefficients and Legendre functions are

$$\bar{P}_{nm} = N_{nm}P_{nm} \qquad \bar{C}_{nm} = \frac{C_{nm}}{N_{nm}} \qquad \bar{S}_{nm} = \frac{S_{nm}}{N_{nm}} \tag{4.41}$$

The derived Legendre polynomials are normalized using

$$\bar{A}_{nm} = N_{nm}A_{nm} \tag{4.42}$$

where $\bar{A}_{nm}$ are the normalized Legendre polynomials. Lundberg[14] showed that there are several recursive algorithms to compute $\bar{A}_{nm}$ but that only two are stable. GMAT uses the following algorithm to recursively calculate the derived Legendre polynomicals

$$\begin{aligned}\bar{A}_{nm} =& u\left[\frac{(2n+1)(2n-1)}{(n-m)(n+m)}\right]^{1/2}\bar{A}_{n-1,m} \\ &- \left[\frac{(2n+1)(n-m-1)(n+m-1)}{(2n-3)(n+m)(n-m)}\right]^{1/2}\bar{A}_{n-2,m}\end{aligned} \tag{4.43}$$

The recursive algorithm is started using

$$\bar{A}_{11} = \sqrt{3}\cos\phi \tag{4.44}$$

$$\bar{A}_{nn} = \cos\phi\sqrt{\frac{2n+1}{2n}}\bar{A}_{n-1,n-1} \tag{4.45}$$

The above equations are normalized using Eq. (4.42) and used in

The acceleration due to nonspherical gravity can be written as

$$\begin{aligned}\mathbf{a}_g =& \left(\frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u}\right)\hat{\mathbf{r}} \\ &+ \left(\frac{1}{r}\frac{\partial U}{\partial s} \quad \frac{1}{r}\frac{\partial U}{\partial t} \quad \frac{1}{r}\frac{\partial U}{\partial u}\right)^T\end{aligned} \tag{4.46}$$

To simplify the partial derivatives in Eq. (4.46), Pines defines some intermediate variables as follows

$$\rho = a/r \tag{4.47}$$

$$\begin{aligned}\rho_0 &= \mu/r \\ \rho_1 &= \rho\rho_0 \\ \rho_n &= \rho\rho_{n-1} \qquad \text{for } n > 1\end{aligned} \tag{4.48}$$

Using Lundberg's nondimensionalization approach, we can write

$$\begin{aligned}\bar{D}_{nm}(s,t) &= \bar{C}_{nm}r_m(s,t) + \bar{S}_{nm}i_m(s,t) \\ \bar{E}_{nm}(s,t) &= \bar{C}_{nm}r_{m-1}(s,t) + \bar{S}_{nm}i_{m-1}(s,t) \\ \bar{F}_{nm}(s,t) &= \bar{S}_{nm}r_{m-1}(s,t) - \bar{C}_{nm}i_{m-1}(s,t) \\ \bar{G}_{nm}(s,t) &= \bar{C}_{nm}r_{m-2}(s,t) + \bar{S}_{nm}i_{m-2}(s,t) \\ \bar{H}_{nm}(s,t) &= \bar{S}_{nm}r_{m-2}(s,t) - \bar{C}_{nm}i_{m-2}(s,t)\end{aligned} \tag{4.49}$$

The partial derivatives in Eq. (4.46) can be written as

$$\begin{aligned}\frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u} = \\ -\sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n} c_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{D}_{nm}\end{aligned} \tag{4.50}$$

$$\frac{1}{r}\frac{\partial U}{\partial s} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n}\bar{A}_{nm}(u)m\bar{E}_{nm} \tag{4.51}$$

$$\frac{1}{r}\frac{\partial U}{\partial t} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n}\bar{A}_{nm}(u)m\bar{F}_{nm} \tag{4.52}$$

$$\frac{1}{r}\frac{\partial U}{\partial s} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n} c_{n,m+1}\bar{A}_{n,m+1}(u)\bar{D}_{nm} \tag{4.53}$$

where

$$\begin{aligned} c_{n,m+1} &= [(n-m)(n+m+1)]^{1/2} \\ c_{n+1,m+1} &= \left[\frac{(n+m+2)(n+m+1)}{(2n+3)(2n+2)}\right]^{1/2}\end{aligned}$$

To calculate the nonzero portion of the sensitivity matrix, we begin by calcluting the following 9 terms:

$$a_{11} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} m(m-1)\bar{A}_{nm}\bar{G}_{nm} \tag{4.54}$$

$$a_{12} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} m(m-1)\bar{A}_{nm}\bar{H}_{nm} \tag{4.55}$$

$$a_{13} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} mc_{n,m+1}\bar{A}_{n,m+1}\bar{E}_{nm} \tag{4.56}$$

$$a_{14} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} mc_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{E}_{nm} \tag{4.57}$$

$$a_{23} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} mc_{n,m+1}\bar{A}_{n,m+1}\bar{F}_{nm} \tag{4.58}$$

$$a_{24} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} mc_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{F}_{nm} \tag{4.59}$$

$$a_{33} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} c_{n,m+2}\bar{A}_{n,m+2}\bar{D}_{nm} \tag{4.60}$$

$$a_{34} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} c_{n+1,m+2}\bar{A}_{n+1,m+2}\bar{D}_{nm} \tag{4.61}$$

$$a_{44} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n} c_{n+2,m+2}\bar{A}_{n+2,m+2}\bar{D}_{nm} \tag{4.62}$$

where

$$\begin{aligned} c_{n+1,m+2} &= c_{n+1,m+1}\left[(n-m)(n+m+3)\right]^{1/2} \\ c_{n,m+2} &= c_{n,m+1}\left[(n-m-1)(n+m+2)\right]^{1/2} \\ c_{n+2,m+2} &= c_{n+1,m+1}\left[\frac{(n+m+4)(n+m+3)}{(2n+5)(2n+4)}\right]^{1/2}\end{aligned}$$

Finally,

$$\mathbf{C}_g = \frac{\partial \mathbf{a}_g}{\partial \mathbf{r}} \tag{4.63}$$

where $\mathbf{C}_g$ is a symmetric matrix with components given by

$$\begin{aligned}
c_{11} &= a_{11} + s^2 a_{44} + a_4/r + 2sa_{14} && (4.64)\\
c_{12} &= c_{21} = a_{12} + sta_{44} + sa_{24} + ta_{14} && (4.65)\\
c_{13} &= c_{31} = a_{13} + sua_{44} + sa_{34} + ua_{14} && (4.66)\\
c_{22} &= -a_{11} + t^2 a_{44} + a_4/r + 2ta_{24} && (4.67)\\
c_{23} &= c_{32} = a_{23} + tua_{44} + ua_{24} + ta_{34} && (4.68)\\
c_{33} &= a_{33} + u^2 a_{44} + a_4/r + 2 * u * a_{34} && (4.69)
\end{aligned}$$

Note that

$$a_4 = \frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u} \tag{4.70}$$

and is given in Eq. (4.50).

4.2.3 Atmospheric Drag

The acceleration due to drag is given by

$$\mathbf{a}_d = -\frac{1}{2}\rho v_{rel}^2 \frac{C_d A}{m_s}\hat{\mathbf{v}}_{rel} \tag{4.71}$$

where

$$\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega}_{\otimes} \times \mathbf{r} + \mathbf{v}_w \tag{4.72}$$

and $\boldsymbol{\omega}_{\otimes}$ is the central bodies angular velocity vector, $\mathbf{v}_w$ is the local wind velocity, C_d is the drag coefficient, A is the cross sectional area normal to $\mathbf{v}_{rel}$, ρ is the atmospheric density, and m_s is the spacecraft mass.

The partial derivatives of the drag force with respect to position and velocity are:

$$\frac{\partial \mathbf{a}_d}{\partial \mathbf{r}} = -\frac{1}{2}\frac{C_d A}{m_s}\left(v_{rel}\mathbf{v}_{rel}\frac{\partial \rho}{\partial \mathbf{r}} + \rho \mathbf{v}_{rel}\hat{\mathbf{v}}_{rel}^T\left(\frac{\partial \mathbf{v}_w}{\partial \mathbf{r}} - \boldsymbol{\omega}_B^x\right) + \rho v_{rel}\left(\frac{\partial \mathbf{v}_w}{\partial \mathbf{r}} - \boldsymbol{\omega}_B^x\right)\right) \tag{4.73}$$

$$\frac{\partial \mathbf{a}_d}{\partial \mathbf{v}} = -\frac{1}{2}\frac{\rho C_d A}{m_s}\left(\mathbf{v}_{rel}\hat{\mathbf{v}}_{rel}^T + v_{rel}\mathbf{I}_3\right) \tag{4.74}$$

4.2.4 Solar Radiation Pressure

$$\mathbf{a}_s = -P_{SR}\frac{C_R A}{m_s}\hat{\mathbf{s}} \tag{4.75}$$

where $\hat{\mathbf{s}}$ is a unitized vector pointing from the spacecraft to the sun

$$\mathbf{s} = \mathbf{r}_s - \mathbf{r} \tag{4.76}$$

where $\mathbf{r}_s$ is the Sun's position vector and $\mathbf{r}$ is the spacecrafts position vector.

$$\mathbf{A}_s = \mathbf{D}_s = \mathbf{0}_{3\times 3} \tag{4.77}$$

$$\mathbf{B}_s = \mathbf{I}_{3\times 3} \tag{4.78}$$

$$\mathbf{C}_s = P_{SR}\frac{C_R A}{m_s}\left(\frac{1}{s^3}\mathbf{I}_3 - 3\frac{\mathbf{s}\mathbf{s}^T}{s^5}\right) \tag{4.79}$$

where

$$s = \|\mathbf{s}\| \tag{4.80}$$

4.2.5 Relativistic Corrections

The form of the relativitic correction to the Newtonian equations of motion depends upon the coordinate system in which they are expressed. The treatment in GMAT follows Huang[15] *et. al.*. For celestial-body-centered motion (not sun or solarsystem barycenter) expressed in the local J2000 axis system, the relativistic correction takes the following form:

$$\mathbf{a}_r = \frac{\mu}{c^2 r^3}\left(\left(4\frac{\mu}{r} - v^2\right)\mathbf{r} + 4(\mathbf{r}\cdot\mathbf{v})\mathbf{v}\right) + 2(\mathbf{\Omega}\times\mathbf{v}) + 2\frac{\mu}{c^2 r^3}\left(\frac{3}{r^2}(\mathbf{r}\times\mathbf{v})(\mathbf{r}\cdot\mathbf{J}) + (\mathbf{v}\times\mathbf{J})\right) \tag{4.81}$$

where

$$\mathbf{\Omega} = \frac{3}{2}\mathbf{v}_{B/S}\times\left(\frac{-\mu\mathbf{r}_{B/S}}{c^2 r_{B/S}^3}\right) \tag{4.82}$$

and

- μ is the gravitational parameter of the central body expressed in the local celestial body J2000 frame
- c is the speed of light
- $\mathbf{J}$ is the central body's angular momentum per unit mass.
- $\mathbf{r}$ is the vehicle's position in the local J000 system.
- $\mathbf{v}$ is the vehicle's velocity in the local J000 system.
- $\mathbf{r}_{B/S}$ is the central body's position with respect to the Sun.
- $\mathbf{v}_{B/S}$ is the central body's velocity with respect to the Sun.

The first term in Eq. 4.81 is the Schwarzschild solution, the second term is geodesic precesion, and the third term is Lense-Thirring precession. The vector J is computed using

$$\mathbf{J} = \mathbf{R}_B^{I/F}\begin{bmatrix}0 & 0 & \frac{2}{5}R_B^2\omega_B\end{bmatrix}^T \tag{4.83}$$

where R_B is the central body's mean equatorial radius,ω_B is the spin rate, and $\mathbf{R}_B^{I/F}$ is the body's fixed to inertial rotation matrix.

Eq. 4.81 is used for all bodies except the sun. In the case that the sun is the central body, the geodesic precesion term is omitted from the correction.

4.2.6 Spacecraft Thrust

$$\begin{aligned} F_T(T,P) &= C_1 + C_2P + (C_3 + C_4P + C_5P^2 + C_6P^{C_7} \\ &+C_8P^{C_9} + C_{10}P^{C_{11}} + C_{12}(C_{13})^{C_{14}P})\left(\frac{T}{T_{ref}}\right)^{1+C_{15}+C_{16}P} \end{aligned} \tag{4.84}$$

$$\begin{aligned} I_{sp}(T,P) &= K_1 + K_2P + (K_3 + K_4P + K_5P^2 + K_6P^{K_7} \\ &+K_8P^{K_9} + K_{10}P^{K_{11}} + K_{12}(K_{13})^{K_{14}P})\left(\frac{T}{T_{ref}}\right)^{1+K_{15}+K_{16}P} \end{aligned} \tag{4.85}$$

$$\dot{m} = f_d\frac{F_T(T,P)}{I_{sp}(T,P)g} \tag{4.86}$$

$$\mathbf{T} = f_s f_d F_T(T,P)\mathbf{R}_{iT}\hat{\mathbf{T}}_d \tag{4.87}$$

where $F_T(T,P)$ is given in equation 4.84, f_s and f_d are the thrust scale factor and duty cycle respectively, $\mathbf{R}_{iT}$ is the rotation matrix from the thruster coordinate system to the EarthMJ2000 equatorial system, and $\hat{\mathbf{T}}_d$ is the unitized thrust direction in the thruster coordinate system.

Table 4.2: Thrust and Isp Coefficient Units

Coeff.	Unit		
C_1	N	K_1	s
C_2	N/kPa	K_2	s/kPa
C_3	N	K_3	s
C_4	N/kPa	K_4	s/kPa
C_5	N/kPa^2	K_5	s/kPa^2
C_6	N/kPa^{C_7}	K_6	s/kPa^{C_7}
C_7	None	K_7	None
C_8	N/kPa^{C_9}	K_8	s/kPa^{C_9}
C_9	None	K_9	None
C_{10}	N/kPa$^{C_{11}}$	K_{10}	s/kPa$^{C_{11}}$
C_{11}	None	K_{11}	None
C_{12}	N	K_{12}	s
C_{13}	None	K_{13}	None
C_{14}	1/kPa	K_{14}	1/kPa
C_{15}	None	K_{15}	None
C_{16}	1/kPa	K_{16}	1/kPa

VNB Thruster System

It is possible to specify thrust with respect to rotating coordinates systems. Then, during integration of the equations of motion, GMAT uses the coordinate system definition to determine the thrust in the inertial system being used for numerical integration of the equations of motion. One of the coordinate systems useful in mission analysis is the Velocity-Normal-Binormal (VNB) system based on the motion of a spacecraft with respect to a reference origin. One way to configure a thruster to use a local VNB system is to set `CoordinateSystem` to local. Internally, GMAT creates a coordinate system based on the `Axes` and `Origin` specified chosen for the thruster. We illustrate this below by example. The thruster named `Thruster1` is configured to use a local VNB coordinate system based on the motion of the owner-spacecraft and the Earth.

```
Create Thruster Thruster1
Thruster1.CoordinateSystem = Local;
Thruster1.Origin = Earth;
Thruster1.Axes = VNB;
```

To convert the thrust from the requested local VNB system to the inertial system, GMAT creates a coordinate system configured as shown below. The `Origin` field on the thruster is used in two places on the coordinate system: as both the `Origin` and the `Primary`. The axes are set to ObjectReferenced, and the x-axis and y-axis are respectively set to "V" and "N".

```
 Create CoordinateSystem SATVNB;
GMAT SATVNB.Origin = DefaultSC;
GMAT SATVNB.Axes = ObjectReferenced;
GMAT SATVNB.Primary = Earth;
GMAT SATVNB.Secondary = DefaultSC;
GMAT SATVNB.XAxis = V;
GMAT SATVNB.YAxis = N;
```

Using this system, the x-axis is in the velocity direction, the n-axis is in the velocity direction, and the z-axis completes the right-handed set. Note, the secondary body is not set until the thruster is assigned to a spacecraft, and then, the secondary is set to be the owner- spacecraft. The script snippet above shows the configuration after the thruster has been attached to spacecraft "DefaultSC", using, for example, the script line `DefaultSC.Thrusters = {Thruster1};`

LVLH Thruster System

The LVLH system, similarly to the VNB system, is a local system that is constructed based on the motion of the owner spacecraft with respect to an origin and axes system chosen by the user. As an example, below the thruster named `Thruster1` is configured to use a local LVLH coordinate system based on the motion of the owner spacecraft "MySat" and the moon.

```
Create Thruster Thruster1
Thruster1.CoordinateSystem = Local;
Thruster1.Origin = Luna;
Thruster1.Axes = LVLH;
```

Internally, GMAT creates a coordinate system configured as shown below. The `Origin` field on the thruster is used in two places on the coordinate system, as both the `Origin` and the `Primary`. The axes are set to `ObjectReferenced`.

```
 Create CoordinateSystem SATLVLH;
GMAT SATLVLH.Origin = MySat;
GMAT SATLVLH.Axes = ObjectReferenced;
GMAT SATLVLH.Primary = Luna;
GMAT SATLVLH.Secondary = MySat;
GMAT SATLVLH.XAxis = -R;
GMAT SATLVLH.YAxis = -N;
```

Using this system, the y-axis is opposite of orbit normal, the z-axis points towards the origin, and the x-axis completes the right-handed set.

4.3 Attitude

The attitude of a spacecraft can be defined qualitatively as how the spacecraft is oriented in inertial space, and how that orientation changes in time. GMAT has the ability to model the orientation and rate of rotation of a spacecraft using several different mathematical models. Currently, GMAT assumes that a spacecraft is a rigid body.

There are many ways to quantitatively describe the orientation and rate of rotation of a spacecraft, just like there are many ways we can quantitatively describe an orbit state. Let's define any set of numbers that can uniquely define the spacecraft attitude as an *attitude parameterization*. GMAT allows its users to employ several common attitude parameterizations including quaternions, Euler angles, the Direction Cosine Matrix (DCM) or Attitude Matrix **A**, Euler angle rates, and the angular velocity vector. Given an initial attitude state, GMAT can propagate spacecraft attitude using one of several kinematic or dynamic attitude propagation models.

Our notation will be based on a compromise that spans multiple disciplinary fields, from astrodynamics and aircraft flight simulation to computer graphics, and is informed by the AIAA and CCSDS standards. We want a consistent definition in the inner core algorithms of GMAT combined with the capability of translating internal variables into interface variables that are easy for users to understand and work with. For example, if a user wishes to visualize the attitude of a specific mission spacecraft using quaternions extracted from real world telemetry, GMAT should be flexible enough for that user to define all four quaternion components, their locations with respect to each other, and their numerical signs. The user should also be able to specify the two reference frames the attitude parameterization is relating, and the direction the attitude parameters take a transformation. In other words, following the CCSDS standard, the user should be able to specify which is the "from" frame and which is the "to" frame.

We should also point out that the parameters presented in this document can describe not only the "transformation" of a vector between two reference frames, but also the "rotation" of a vector from one orientation to another in the same reference frame. The literature uses the definition of "transformation" and "rotation" quite liberally, and a detailed search has found little agreement on terminology and notation. Almost all authors agree that the components of the 3×3 matrix relating vectors to each other are "Direction Cosines". They are the dot products of the basis axes unit vectors of one reference frame with the basis axes unit vectors of the other reference frame. They are the cosines of the angles between each of the basis axes.

For now it is suggested that the matrix we use to transform or rotate vectors be labeled **R** to remain consistent with the mathematical development earlier in this specification in Chapter 2. The 3×3 matrix **R** will relate various reference frames to each other and can be used to rotate vectors between frames such as in the FK5 reduction process. Meanwhile, if we wish to explicitly model the attitude or orientation of a spacecraft, we can represent this with the 3×3 matrix **A**. The letter brings the word "Attitude" to mind, and is used by several well known attitude dynamics authors including Landis Markley who wrote the "Parameterization of the Attitude" section of "Spacecraft Attitude Determination and Control" by James R. Wertz, a very popular text in the field. To be consistent with Chapter 2, we should specify the direction of the transformation with additional notation.

Meanwhile, if we intentionally keep our notation as clutter free as possible, the definition of **A** will be as follows.

$$\mathbf{b} = \mathbf{A}\mathbf{r} \tag{4.88}$$

In this equation **b** is a vector in the spacecraft body frame, and **r** is a vector in the reference frame. The attitude matrix **A** transforms vectors from the reference frame to the body frame.

The human mind has some difficulty visualizing the orientation of a spacecraft when presented with the nine component attitude matrix. Likewise, it is difficult to know if an aircraft or spacecraft is "upside-down" given the four components of a quaternion vector. For human interface purposes, the attitude descriptor of choice is the set of three Euler Angles, such as the classic "roll", "pitch", and "yaw". The CCSDS specification avoids mentioning the DC matrix altogether. Meanwhile, GMAT will use it as an intermediate form of attitude parameterization when transforming from internal attitude propagation equations using quaternions, and human interpretable Euler Angles. GMAT will also use it for both Coordinate System Fixed and Spinning Spacecraft modes of attitude propagation.

In this chapter, we discuss the attitude parameterizations supported in GMAT, and how to convert between the different types. We discuss the internal state parameterization that GMAT uses. Next we investigate the types of attitude modes in GMAT and discuss in detail how GMAT propagates the spacecraft attitude in all of the Kinematic and Dynamic attitude modes. We conclude the chapter with a discussion of how GMAT converts between different attitude parameterizations.

4.3.1 Attitude Propagation

Given a set of initial conditions that define the attitude, GMAT can propagate the attitude using several methods. Currently, GMAT supports both Kinematic attitude and Dynamic attitude propagation. In Kinematic mode, the attitude is defined by describing the desired orientation with respect to other objects such as spacecraft or celestial bodies. With this information, GMAT can calculate the required attitude to satisfy the desired geometrical configuration. In Dynamic mode, the user must also supply spacecraft body inertial mass properties, and tell GMAT how to apply torques dynamically to the spacecraft. With these conditions specified, GMAT will numerically integrate quaternion attitude equations of motion combined with Euler's moment equations. The resulting attitude motion will display basic physics-based phenomena such as precession and nutation. Dynamic mode will allow a user to model the implementation of attitude control laws, as well as the effects of asymmetric propellant mass depletion and dynamic spacecraft component articulation. It will also eventually support the modeling of attitude perturbations such as Gravity Gradient, Aerodynamic Drag and Solar Photon Pressure Torques, and spacecraft magnetic interaction with planetary magnetic fields. This section presents the different Kinematic and Dynamic attitude modes, and how GMAT calculates the attitude state in each mode. Let's begin by looking at the internal attitude state representations and how the user can define initial conditions.

Internal State Representation and Attitude Initial Conditions

As mentioned in the introduction, certain attitude parameterizations are more useful for attitude propagation, while other attitude parameterizations are more intuitive for providing attitude initial conditions or output. GMAT uses different internal parameterizations of the attitude depending on the attitude mode. The type of parameterization is chosen to make the attitude propagation algorithms natural and convenient. For the Kinematic modes, GMAT uses the Direction Cosine Matrix (DCM) or "Attitude" matrix that represents the rotation from the inertial reference frame to the body frame. The notation for the Attitude matrix, including the direction of the rotation it parameterizes is $\mathbf{A}_{BI}$. The subscripts denote the direction of the rotation from Inertial to Body frames. For Dynamic mode, six degree of freedom (6DOF) attitude propagation, GMAT uses the quaternion that represents the rotation from the inertial reference frame to the body frame. The notation for the quaternion, including the direction of the rotation it parameterizes, is $\boldsymbol{q}_{BI}$. Once again, the subscripts denote the direction of the rotation from Inertial to Body frames. GMAT uses the angular velocity of the body with respect to the inertial frame, expressed in the body frame, $\{\boldsymbol{\omega}_{IB}\}_B$, as the rate portion of the state vector.

For convenience, the user can choose a coordinate system in which to define the initial attitude state. Let's call this system $\mathcal{F}_i$. The user can define the initial attitude with respect to $\mathcal{F}_i$ using Euler angles, the DCM or attitude matrix $\mathbf{A}$, or quaternions. The user can define the body rate with respect to $\mathcal{F}_i$ by defining the angular velocity in $\mathcal{F}_i$, $\{\boldsymbol{\omega}_{IB}\}_i$, by defining the angular velocity in the inertial frame as measured in body coordinates $\{\boldsymbol{\omega}_{IB}\}_B$ or by defining the Euler angle rates. Note that not all attitude modes require these three pieces of information. The specific inputs for each attitude mode are discussed below, along with details about how attitude propagation is performed in each mode.

Kinematic Attitude Propagation

The Kinematic attitude mode allows a user to define a geometrical configuration based on the relative position of a spacecraft with respect to other spacecraft or celestial bodies, and with respect to different coordinate systems. In Kinematic mode, GMAT does not integrate the attitude equations of motion, but rather calculates the attitude based on the geometrical definition provided by the user. There are two Kinematic modes to choose from. These are Coordinate System Fixed and Spinning Spacecraft. The different modes allow the user to conveniently define the spacecraft attitude depending on the type of

attitude profile needed for a specific mission. To begin, let's look at how GMAT calculates the attitude state in the Coordinate System Fixed attitude mode (CSFixed).

Coordinate System Fixed Mode

In the CSFixed attitude mode, the user specifies a coordinate system $\mathcal{F}_i$ in which to fix the attitude. $\mathcal{F}_i$ can be any of the default coordinate systems or any user defined coordinate system. GMAT calculates the attitude matrix of $\mathcal{F}_i$ using the following equation.

$$\mathbf{A}_{BI} = \mathbf{R}_{iI} \tag{4.89}$$

$\mathbf{R}_{iI}$ is the DCM matrix relating $\mathcal{F}_I$ to $\mathcal{F}_i$ and GMAT knows how to calculate this matrix for all allowable $\mathcal{F}_i$. For details on the calculation of this matrix for all coordinate systems in GMAT see Ch. 2. To calculate $\{\boldsymbol{\omega}_{IB}\}_B$, we start from Euler's equation:

$$\dot{\mathbf{A}}_{BI} = -\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B \mathbf{A}_{BI} \tag{4.90}$$

where

$$\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B \equiv \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \equiv \boldsymbol{\Omega} \tag{4.91}$$

and $\{\boldsymbol{\omega}_{IB}\}_B$ is the skew symmetric angular velocity "cross product matrix" of $\mathcal{F}_\mathcal{B}$ with respect to $\mathcal{F}_I$, expressed in $\mathcal{F}_B$. Solving Equation 4.90 for $\{\boldsymbol{\omega}^{\times}_{IB}\}_B$ we obtain

$$\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B = -\dot{\mathbf{A}}_{BI}\mathbf{A}_{BI}^T \tag{4.92}$$

where

$$\dot{\mathbf{A}}_{BI} = \dot{\mathbf{R}}_{iI} \tag{4.93}$$

Spinning Spacecraft Mode

In spinning spacecraft mode, GMAT propagates the attitude by assuming the spin axis direction is fixed in inertial space. The spacecraft attitude at some time, t, is kinematically propagated from the attitude initial conditions. The user defines the initial attitude with respect to $\mathcal{F}_i$ by providing $\mathbf{A}_{BI}(t_o)$ or an equivalent parameterization that is then converted to the Attitude matrix. The user also provides the angular velocity of the body axes with respect to the inertial axes expressed in $\mathcal{F}_i$, $\{\boldsymbol{\omega}_{IB}\}_i$.

To calculate $\mathbf{A}_{BI}(t)$ where t is an arbitrary epoch, we begin by calculating $\mathbf{A}_{B_oI}$, understanding that $\mathbf{A}_{B_oI} = \mathbf{A}_{BI}(t_o)$. We calculate $\mathbf{A}_{B_oI}$ using

$$\mathbf{A}_{B_oI} = \mathbf{A}_{Bi}\mathbf{R}_{iI}(t_o) \tag{4.94}$$

where $\mathbf{A}_{Bi}$ comes from user provided data, and $\mathbf{R}_{iI}(t_o)$ is calculated by GMAT and is dependent upon $\mathcal{F}_i$. See Ch. 2 for details on how GMAT calculates $\mathbf{R}_{iI}$ for all allowable coordinate systems in GMAT.

Before calculating $\mathbf{A}_{BI}(t)$ we must determine the spin axis in the body frame, $\{\boldsymbol{\omega}_{IB}\}_B$. If the user has chosen to provide the angular velocity vector in body frame coordinates, then we already have what we need. Otherwise, the user provides $\{\boldsymbol{\omega}_{IB}\}_i$. In spinning mode we assume the rotation rate magnitude is constant and also that the spin axis direction is constant in inertial space. This assumption is only realistic if the angular velocity is aligned with either the maximum or minimum primary axis of moment of inertia in the spacecraft body reference frame. If angular velocity is aligned with the axis of minimum moment of inertia, then we must also assume there is an active attitude control system holding it there. If there is no control system, and the spacecraft dissipates rotational kinetic energy (all real bodies will) then the direction of the angular velocity vector will migrate slowly until it is aligned with the axis of maximum moment of inertia.

Understanding these assumptions we can state $\{\boldsymbol{\omega}_{IB}\}_B(t) = \{\boldsymbol{\omega}_{IB}\}_B(t_o) = \{\boldsymbol{\omega}_{IB}\}_B$. We can find the spin axis in the body frame (if we need to) using $\mathbf{A}_{Bi}$ as follows

$$\{\boldsymbol{\omega}_{IB}\}_B = \mathbf{A}_{Bi}\{\boldsymbol{\omega}_{IB}\}_i \tag{4.95}$$

Once calculated, GMAT saves the parameters $\mathbf{A}_{B_oI}$ and $\{\boldsymbol{\omega}_{IB}\}_B$ for use in calculating the attitude and rate at other epochs.

GMAT calculates $\mathbf{A}_{BI}(t)$ using the Euler axis/angle to Attitude matrix conversion algorithm in Sec.4.3.2. The Euler axis is simply the unitized angular velocity vector or,

$$\mathbf{a} = \frac{\{\boldsymbol{\omega}_{IB}\}_B}{\omega_{IB}} \tag{4.96}$$

where

$$\omega_{IB} = \|\{\boldsymbol{\omega}_{IB}\}_B\| \tag{4.97}$$

The Euler angle ϕ is calculated using

$$\phi(t) = \omega_{IB}(t - t_o) \tag{4.98}$$

where t is the current epoch, and t_o is the spacecraft's initial epoch. Let's define the Attitude matrix that results from the Euler axis/angle to Attitude matrix conversion algorithm using $\mathbf{a}$ and $\phi(t)$, as $\mathbf{A}_{BB_o}(t)$. We can calculate $\mathbf{A}_{BI}(t)$ using

$$\mathbf{A}_{BI}(t) = \mathbf{A}_{BB_o}(t)\mathbf{A}_{B_oI} \tag{4.99}$$

To summarize, in spinning mode the user provides $\mathbf{A}_{Bi}$ and $\{\boldsymbol{\omega}_{IB}\}_i$ or $\{\boldsymbol{\omega}_{IB}\}_B$. GMAT assumes that that the spin axis direction is constant in both inertial and body space, and uses the Euler axis/angle method to propagate the attitude to find $\mathbf{A}_{BI}$. Now let's look at dynamic attitude propagation in GMAT.

Dynamic Attitude Propagation

In Dynamic mode, GMAT propagates quaternion attitude equations of motion along with Euler's moment equations. This allows the user to provide detailed mass moment of inertia properties, and to model spacecraft attitude control and environmental perturbing torques. The resulting numerically integrated attitude will exhibit basic dynamics properties such as precession and nutation. Detailed tracking of the dynamic inertia tensor will also allow the user to properly model the effect on the attitude trajectory of propellant mass depletion, articulation of various components, and implementation of attitude control laws. For modes where control laws cause the spacecraft to slew to a targeted pointing direction, GMAT will model line of sight angular acceleration, deceleration, overshoot, settling, and drift behavior. There are several Dynamic modes to choose from. The different modes allow the user to conveniently define the spacecraft attitude and controlled pointing depending on the type of attitude profile needed for a specific mission. To begin, let's look at how GMAT calculates the attitude state in the simplest possible dynamic mode, Rigid Torque Free attitude mode (RigidTorqueFreeQuat).

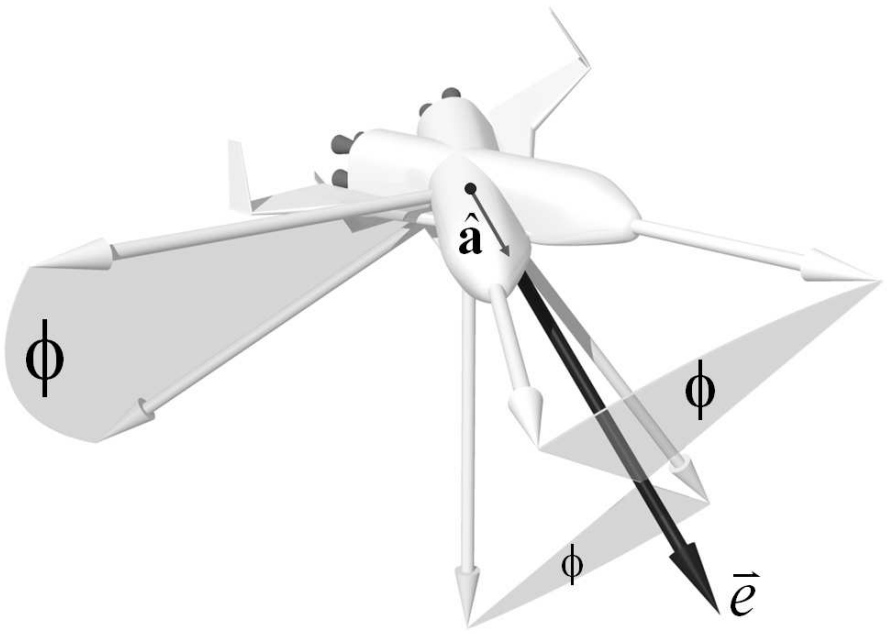


Figure 4.2: Euler Axis and Angle

Rigid Torque Free (Quaternion) Mode

The fundamental equations of attitude motion run most efficiently using the four component quaternion. There are equations of motion for Euler Angles, but these have singularities that cause problems in implementation. There are also equations of motion for the nine components of the Attitude matrix. Numerically integrating nine equations for the Attitude matrix is not as efficient as four equations for the quaternion. The quaternion is the most popular attitude parameterization in use today for propagating attitude using numerical integration schemes. Let's begin by defining the quaternion.

As shown above in Figure 4.2, any spacecraft attitude can be described as the rotation about a single axis via a fixed angle from an inertial frame orientation to the current body orientation. The single axis is called the Euler Axis, and the angle is the Euler Angle. In the figure, the Euler Axis is given the vector label $\vec{e}$. This is not the same as the three Euler Angles used as another of our attitude parameterizations. We will use the CCSDS Attitude Data Message standard to define the GMAT quaternion. The definition is based on the Euler Axis/Angle parameterization and is

$$\boldsymbol{q} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_c \end{bmatrix} \equiv \begin{bmatrix} \mathbf{q} \\ q_c \end{bmatrix} \equiv \begin{bmatrix} \mathbf{a}_1 \sin\frac{\phi}{2} \\ \mathbf{a}_2 \sin\frac{\phi}{2} \\ \mathbf{a}_3 \sin\frac{\phi}{2} \\ \cos\frac{\phi}{2} \end{bmatrix} \tag{4.100}$$

where $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ are components of the unit vector aligned with the Euler Axis or $\hat{\mathbf{a}}$. This unit vector is shown near the spacecraft center of mass in Figure 4.2. In Eq. 4.100 the Euler Angle is denoted using the symbol ϕ. There are actually two possible values of ϕ, since the rotation shown in Figure 4.2 can be the long way around or the short way around. The two possible values will add to 360 °. Our definition of quaternions assumes the smaller angle.

The subscript "c" on the fourth component of the quaternion comes from the CCSDS standard and most likely stands for "cosine". The CCSDS has cleverly avoided specifying whether the cosine "scalar" component should be placed before or after the vector component. This standard has thus eliminated the confusion of which component of the quaternion should be defined as q_1 and which as q_4. The CCSDS standard does place its initial definition for q_c after the definitions for q_1, q_2, and q_3. This lets us stack the four components in the vector shown in the second part of the definition in Eq. 4.100. If q_4 were substituted for q_c, we would have Landis Markley's definition from the "Parameterization of the Attitude" section in "Spacecraft Attitude Determination and Control" by Wertz. The third part of the definition above shows the 3×1 vector component of the quaternion as a single symbol $\mathbf{q}$, placed above q_c to form a second 4×1 matrix or vector. This also parallels Markley's definition and will conveniently fit into the GMAT equations relating quaternions to the Attitude matrix shown later in this section. Note that for the final part of the definition in Eq. 4.100, the CCSDS standard uses $\mathbf{e}$ to denote the unit vector aligned with the Euler Axis instead of $\mathbf{a}$. We have chosen $\mathbf{a}$ to represent "axis", and because that is how GMAT is using it later in the attitude parameterization conversion Section 4.3.2.

In torque free quaternion attitude propagation mode, the user provides four pieces of information. They first choose a coordinate system, $\mathcal{F}_i$, in which to define the initial conditions. Secondly, they define the initial attitude with respect to $\mathcal{F}_i$ by providing $\mathbf{A}_{Bi}$ or an equivalent parameterization that is then converted to the Attitude matrix. GMAT will then use $\mathbf{A}_{Bi}$ to calculate $\mathbf{A}_{BI}$ using Eq. 4.89. From this Attitude matrix the parameterization will next be converted by GMAT to the quaternion $\boldsymbol{q}_{Bi}$. Thirdly, the user provides the angular velocity of the body axes with respect to the inertial axes expressed in $\mathcal{F}_i$, or $\{\boldsymbol{\omega}_{IB}\}_i$. If it is more convenient for the user to provide angular velocity expressed in the spacecraft body frame, GMAT will accept this input as well. If the user provides angular velocity in $\mathcal{F}_i$, GMAT will use Eq. 4.95 to convert it to $\{\boldsymbol{\omega}_{IB}\}_B$. Fourthly, the user provides the mass moment of inertia properties. These can be in the form of the three principal moments of inertia I_{xx}, I_{yy}, and I_{zz}, or a full inertia tensor with the off-diagonal products of inertia I_{xy}, I_{xz}, and I_{yz} included. In a future release of GMAT the user will be able to use the GUI or scripting language to assemble a complex spacecraft model from components. Locating and orienting these components and describing how those that articulate can move, the user will provide enough information for GMAT to calculate a dynamic inertia tensor. This future capability will also include dynamically adjusting the components of the inertia tensor as propellant is consumed from tanks.

GMAT will assemble the data provided by the user into a state vector that can be propagated numerically. In this case, GMAT will need to create an initial state, and then provide its numerical integrators with the state vector derivative equations.

We will define an attitude state variable $\mathbf{x}$ such that

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{q}^T & \boldsymbol{\omega}^T \end{bmatrix}^T = \begin{bmatrix} q_1 & q_2 & q_3 & q_c & \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \tag{4.101}$$

then taking the derivative we arrive at

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{q}}^T & \dot{\boldsymbol{\omega}}^T \end{bmatrix}^T = \begin{bmatrix} \dot{q_1} & \dot{q_2} & \dot{q_3} & \dot{q_c} & \dot{\omega_1} & \dot{\omega_2} & \dot{\omega_3} \end{bmatrix}^T \tag{4.102}$$

where

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q} \tag{4.103}$$

and where

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix} \tag{4.104}$$

In Eq. 4.104, $[\omega\times]$ is the skew symmetric angular velocity cross product matrix originally introduced in Eq. 4.91. As in Eq. 4.91 the angular velocity used here represents the body with respect to the inertial frame. If we substitute Eq. 4.91 into Eq. 4.104 we get

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{4.105}$$

Substituting Eq. 4.105 into Eq. 4.103 and multiplying the terms together we get

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q} = \begin{bmatrix} \frac{1}{2}(\quad \omega_3 q_2 - \omega_2 q_3 + \omega_1 q_c) \\ \frac{1}{2}(-\omega_3 q_1 + \omega_1 q_3 + \omega_2 q_c) \\ \frac{1}{2}(\quad \omega_2 q_1 - \omega_1 q_2 + \omega_3 q_c) \\ \frac{1}{2}(-\omega_1 q_1 - \omega_2 q_2 - \omega_3 q_3) \end{bmatrix} \tag{4.106}$$

Next let's evaluate the time derivative of the body angular velocity terms. These are the last three components of the state vector derivative in Equation 4.132. Following the Markley/Wertz convention, if we let $\mathbf{L}$ represent angular momentum we can start with a simple equation for angular momentum measured in the Inertial reference frame.

$$\mathbf{L} = \mathbf{I} \cdot \boldsymbol{\omega} \tag{4.107}$$

Euler's second law says that, in an inertial reference frame, the time derivative of $\mathbf{L}$ is the applied torque, $\mathbf{T}$. We write this as

$$\dot{\mathbf{L}} = \frac{d}{dt}(\mathbf{I} \cdot \boldsymbol{\omega}) = \mathbf{T} \tag{4.108}$$

where $\mathbf{I}$ is the body oriented Inertia Tensor. This is a 3×3 matrix with the principal moments of inertia on the diagonal and the products of inertia on the off-diagonal. It is written as follows.

$$\mathbf{I} = \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{bmatrix} \tag{4.109}$$

where the moments and products of inertia are defined as follows.

$$I_{11} \equiv \sum_{i=1}^{n} m_i(\rho_{i2}^2 + \rho_{i3}^2) \tag{4.110}$$

$$I_{22} \equiv \sum_{i=1}^{n} m_i(\rho_{i3}^2 + \rho_{i1}^2) \tag{4.111}$$

$$I_{33} \equiv \sum_{i=1}^{n} m_i(\rho_{i1}^2 + \rho_{i2}^2) \tag{4.112}$$

$$I_{12} = I_{21} \equiv -\sum_{i=1}^{n} m_i\rho_{i1}\rho_{i2} \tag{4.113}$$

$$I_{23} = I_{32} \equiv -\sum_{i=1}^{n} m_i\rho_{i2}\rho_{i3} \tag{4.114}$$

$$I_{31} = I_{13} \equiv -\sum_{i=1}^{n} m_i\rho_{i3}\rho_{i1} \tag{4.115}$$

The signs for the products of inertia depend on how products of inertia are themselves defined. They can be positive or negative depending on individual authors. Meanwhile, each regular geometric shape, if constructed of a uniform solid material, will have an analytic formula for each of its moments and products of inertia. A spacecraft model composed of multiple elements can sum moments and products of inertia for all components into a single Inertia Tensor. For the current RigidTorqueFreeQuat mode we will assume ours is diagonal.

$$\mathbf{I} = \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix} \tag{4.116}$$

In this initial development, we will assume our spacecraft body frame is aligned with its principal axes of inertia, and that it is a completely rigid body. This will mean our inertia tensor is both diagonal and has a time derivative of zero. Let's apply these facts to Eq. 4.108.

$$\frac{d}{dt}(\mathbf{I}\cdot\boldsymbol{\omega}) = (\dot{\mathbf{I}}\cdot\boldsymbol{\omega}) + (\mathbf{I}\cdot\dot{\boldsymbol{\omega}}) \tag{4.117}$$

Since we already stated that the time derivative of I is zero, the first term on the RHS of Eq. 4.117 will drop out. We can now write Eq. 4.108 as follows.

$$\dot{\mathbf{L}} = \mathbf{T} = (\mathbf{I}\cdot\dot{\boldsymbol{\omega}}) \tag{4.118}$$

In Eq. 4.118 the angular momentum term is measured in the inertial frame. Let's express it now in the spacecraft body frame. Superscripts to the left will indicate the reference frame.

$${}^{I}\dot{\mathbf{L}} = {}^{B}\dot{\mathbf{L}} + {}^{B}\boldsymbol{\omega} \times {}^{B}\mathbf{L} \tag{4.119}$$

Combining Equations 4.108 and 4.119 gives us

$${}^{B}\mathbf{T} = {}^{B}\dot{\mathbf{L}} + {}^{B}\boldsymbol{\omega} \times {}^{B}\mathbf{L} \tag{4.120}$$

Substituting 4.118 for $\dot{\mathbf{L}}$ on the RHS of 4.120, and 4.107 for $\mathbf{L}$ on the RHS, we get the following.

$${}^{B}\mathbf{T} = {}^{B}(\mathbf{I}\cdot\dot{\boldsymbol{\omega}}) + {}^{B}\boldsymbol{\omega} \times {}^{B}\mathbf{I}\cdot\boldsymbol{\omega} \tag{4.121}$$

Assuming our Inertia Tensor is diagonal, if we multiply out the terms on the RHS of Eq. 4.121 we will get three equations for torque along the principal axes of the spacecraft body.

$$T_1 = I_{11}\dot{\omega_1} + (I_{22} - I_{33})\omega_2\omega_3 \tag{4.122}$$

$$T_2 = I_{22}\dot{\omega_2} + (I_{33} - I_{11})\omega_3\omega_1 \tag{4.123}$$

$$T_3 = I_{33}\dot{\omega_3} + (I_{11} - I_{22})\omega_1\omega_2 \tag{4.124}$$

These equations are equivalent to Equations 16-50a through 16-50c on Page 522 of Wertz.

Solving equations 4.122 through 4.124 for angular acceleration, we get

$$\dot{\omega_1} = \frac{T_1 - (I_{22} - I_{33})\omega_2\omega_3}{I_{11}} \tag{4.125}$$

$$\dot{\omega_2} = \frac{T_2 - (I_{33} - I_{11})\omega_3\omega_1}{I_{22}} \tag{4.126}$$

$$\dot{\omega_3} = \frac{T_3 - (I_{11} - I_{22})\omega_1\omega_2}{I_{33}} \tag{4.127}$$

These are the three angular velocity time derivative terms we needed for the state derivative vector in Equation 4.132. They are also known as Euler's Moment Equations. As long as we are rotating torque free, we will set the torque **T** terms in Equations 4.125 through 4.127 to zero. We now have everything GMAT needs to integrate our quaternion attitude equations of motion for a torque free rigid tumbling body.

These equations will model several types of rotational motion. They are "motionless hang", "flat spin", "spin with precession", and "spin with precession and nutation" which is the same as a full 3-axis tumble. Next, let's add a level of complexity and look at Articulated Torque Free attitude mode.

Articulated Torque Free (Quaternion) Mode

Let's assume our spacecraft is still rotating without the influence of control or environmental torques, but now it is moving articulated appendages. We can no longer assume our inertia tensor **I** is constant. It is also unlikely, now that objects are moving, that **I** will remain diagonal. Let's develop new equations to handle our dynamic inertia tensor. The definition of angular momentum is

$$\mathbf{L} \equiv \mathbf{I}\boldsymbol{\omega} \tag{4.128}$$

which applies to an inertia tensor with off-diagonal products of inertia. The initial conditions for both parameters on the RHS of Eq. 4.128 will be provided by the User. Multiplying both sides by the inverse of **I**, and then solving for angular velocity we get

$$\boldsymbol{\omega} = \mathbf{I^{-1}L} \tag{4.129}$$

If GMAT calculates changes in **I** based on articulation, and for later modes includes propellant mass consumption, and it integrates $\dot{\mathbf{L}}$ to get **L**, during each integration step we can use Eq. 4.129 to calculate current angular velocity ω. Solving Eq. 4.120 for $\dot{\mathbf{L}}$, we get

$${}^B\dot{\mathbf{L}} = {}^B\mathbf{T} - {}^B\boldsymbol{\omega} \times {}^B\mathbf{L} \tag{4.130}$$

If the user inputs all the same initial conditions as they did in RigidTorqueFreeQuat mode, with the addition of a full inertia tensor with products, and GMAT tracks articulation of components and adjusts the inertia tensor dynamically, the state vector to calculate will now be

$$\mathbf{x} = \begin{bmatrix}\boldsymbol{q}^T & \mathbf{L}^T\end{bmatrix}^T = \begin{bmatrix}q_1 & q_2 & q_3 & q_c & L_1 & L_2 & L_3\end{bmatrix}^T \tag{4.131}$$

where the initial condition of **L** is calculated from Eq. 4.128. To get angular velocity, rather than integrating Euler's Moment Equations, we will use Eq. 4.129 which means we need to invert a 3×3 matrix. Since a full inertia tensor is still symmetric, we can use Cramer's rule to invert it with no numerical pathologies. The time derivative of **x**, which we need to send to the GMAT numerical integrators will be

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\boldsymbol{q}}^T & \dot{\mathbf{L}}^T\end{bmatrix}^T = \begin{bmatrix}\dot{q_1} & \dot{q_2} & \dot{q_3} & \dot{q_c} & \dot{L_1} & \dot{L_2} & \dot{L_3}\end{bmatrix}^T \tag{4.132}$$

We get $\dot{\mathbf{q}}$ from Eq. 4.106 and $\dot{\mathbf{L}}$ from Eq. 4.130. For this torque free mode we can choose to set the torque vector in Eq. 4.130 to zero. We now have everything GMAT needs to integrate our quaternion attitude equations of motion for a torque free articulated tumbling body.

Attitude Hold Mode

Under Construction

Seek Coordinate System Fixed Mode

Under Construction

Target Pointing Slew Mode

Under Construction

Torque Perturbations Mode

Torque Perturbations are higher order terms that model environmental torques or torques applied as the result of human operator control inputs. These can be attitude maneuvers selected by a ground controller from a mouse activated GUI menu, or input through joysticks by an astronaut pilot. Full 6DOF maneuvering involved in manual or automated rendezvous, proximity operations, and docking will involve translational forces and equations of motion and is covered in Ch. TBD. Simple environmental torques include Gravity Gradient, Aerodynamic Drag and Solar Photon Pressure Torques, and spacecraft magnetic interaction with planetary magnetic fields. These torques can be added to the basic Euler Moment Equations presented above in the Torque Free Mode section.

4.3.2 Attitude Parameterizations and Conversions

This section details how GMAT converts between different attitude parameterizations. For each conversion type, any singularities that may occur are addressed. The orientation parameterizations in GMAT include the DCM or A Matrix, Euler Angles, quaternions, and Euler axis/angle. The body rate parameterizations include Euler angle rates and angular velocity. We begin with the algorithm to transform from the quaternions to the Attitude matrix.

Conversion: Quaternions to Attitude Matrix

Given: $\mathbf{q}$, q_c
Find: $\mathbf{A}$
Name: *QuatsToAMat*

$$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \tag{4.133}$$

$$\mathbf{q}^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{4.134}$$

$$c = \frac{1}{q_1^2 + q_2^2 + q_3^2 + q_c^2} \tag{4.135}$$

$$\mathbf{A} = c\left[(q_c^2 - \mathbf{q}^T\mathbf{q})\mathbf{I}_3 + 2\mathbf{q}\mathbf{q}^T - 2q_c\mathbf{q}^\times\right] \tag{4.136}$$

where $\mathbf{I_3}$ is a 3×3 identity matrix. Multiplying out Eq. 4.136 we get

$$\mathbf{A} = c\left(\begin{bmatrix} (q_c^2 - q_1^2 - q_2^2 - q_3^2) & 0 & 0 \\ 0 & (q_c^2 - q_1^2 - q_2^2 - q_3^2) & 0 \\ 0 & 0 & (q_c^2 - q_1^2 - q_2^2 - q_3^2) \end{bmatrix} + \begin{bmatrix} q_1^2 & q_1q_2 & q_1q_3 \\ q_2q_1 & q_2^2 & q_2q_3 \\ q_3q_1 & q_3q_2 & q_3^2 \end{bmatrix} + \begin{bmatrix} 0 & -q_cq_3 & q_cq_2 \\ q_cq_3 & 0 & -q_cq_1 \\ -q_cq_2 & q_cq_1 & 0 \end{bmatrix}\right) \tag{4.137}$$

Adding terms in Eq. 4.137 we get

$$\mathbf{A} = c \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_c^2 & 2(q_1 q_2 + q_3 q_c) & 2(q_1 q_3 + q_2 q_c) \\ 2(q_1 q_2 - q_3 q_c) & -q_1^2 + q_2^2 - q_3^2 + q_c^2 & 2(q_2 q_3 + q_1 q_c) \\ 2(q_1 q_3 + q_2 q_c) & 2(q_2 q_3 - q_1 q_c) & -q_1^2 - q_2^2 + q_3^2 + q_c^2 \end{bmatrix} \tag{4.138}$$

If we substitute the traditional symbol q_4 in for the CCSDS symbol q_c, we will get the exact identity shown in Equation 12-13a on Page 414 of Wertz, except for the constant term c. That is there to normalize the Attitude matrix if the quaternion magnitude is not exactly equal to 1.

It may seem a waste of space to multiply out all these terms when the math was so compact in the form shown above in Eq. 4.136. However, when coding this algorithm, several multiply by zero and add to zero operations will be avoided if Eq. 4.138 is used instead. So it may be a waste of space, but it is a savings in time when running GMAT.

Conversion: Attitude Matrix to Quaternions

Given: $\mathbf{A}$
Find: $\mathbf{q}$, q_c
Name: *AMatToQuats*

Define following vector

$$\mathbf{v} = [\, A_{11} \quad A_{22} \quad A_{33} \quad \text{trace}(\mathbf{A}) \,] \tag{4.139}$$

where the trace of $\mathbf{A}$ is

$$\text{trace}(\mathbf{A}) = A_{11} + A_{22} + A_{33} \tag{4.140}$$

Define i_m as the index of the maximum component of $\mathbf{v}$. Then use the following logic if $i_m = 1$

$$\mathbf{q}'' = \begin{pmatrix} 2v_{i_m} + 1 - \text{trace}(\mathbf{A}) \\ A_{12} + A_{21} \\ A_{13} + A_{31} \\ A_{23} - A_{32} \end{pmatrix} \tag{4.141}$$

if $i_m = 2$

$$\mathbf{q}'' = \begin{pmatrix} A_{21} + A_{12} \\ 2v_{i_m} + 1 - \text{trace}(\mathbf{A}) \\ A_{23} + A_{32} \\ A_{31} - A_{13} \end{pmatrix} \tag{4.142}$$

if $i_m = 3$

$$\mathbf{q}'' = \begin{pmatrix} A_{31} + A_{13} \\ A_{32} + A_{23} \\ 2v_{i_m} + 1 - \text{trace}(\mathbf{A}) \\ A_{12} - A_{21} \end{pmatrix} \tag{4.143}$$

if $i_m = 4$

$$\mathbf{q}'' = \begin{pmatrix} A_{23} - A_{32} \\ A_{31} - A_{13} \\ A_{12} - A_{21} \\ 1 + \text{trace}(\mathbf{A}) \end{pmatrix} \tag{4.144}$$

We normalize $\mathbf{q}''$ using

$$\mathbf{q}' = \frac{\mathbf{q}''}{\|\mathbf{q}''\|} \tag{4.145}$$

Finally,

$$\mathbf{q} = [\; q_1' \quad q_2' \quad q_3' \;]^T \tag{4.146}$$

and

$$q_c = q_c' \tag{4.147}$$

Conversion: Attitude Matrix to Euler Axis/Angle

Given: **A**
Find: **a**, ϕ
Name: *AMatToEulAxisAngle*

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \tag{4.148}$$

$$\phi = \cos^{-1}\left(\frac{1}{2}\left(\text{trace}(\mathbf{A}) - 1\right)\right) \tag{4.149}$$

$$\mathbf{a} = \frac{1}{2\sin\phi}\begin{pmatrix} A_{23} - A_{32} \\ A_{31} - A_{13} \\ A_{12} - A_{21} \end{pmatrix} \tag{4.150}$$

If $\|\sin\phi\| < 10^{-14}$, then we assume

$$\mathbf{a} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \tag{4.151}$$

Note that if $\|\sin\phi\| < 10^{-14}$ then $\cos\phi \approx 1$ and we arrive at an Attitude matrix of $\mathbf{I}_3$.

Conversion: Euler Axis/Angle to Attitude Matrix

Given: **a**, ϕ
Find: **A**
Name: *EulAxisAngleToAMat*

$$\mathbf{a}^{\times} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{4.152}$$

$$\mathbf{A} = \cos\phi\mathbf{I}_3 + (1 - \cos\phi)\mathbf{a}\mathbf{a}^T - \sin\phi\mathbf{a}^{\times} \tag{4.153}$$

Multiplying out the terms in Eq. 4.153 we get

$$\mathbf{A} = \begin{pmatrix} \cos\phi & 0 & 0 \\ 0 & \cos\phi & 0 \\ 0 & 0 & \cos\phi \end{pmatrix} + (1 - \cos\phi)\begin{pmatrix} a_1^2 & a_1a_2 & a_1a_3 \\ a_2a_1 & a_2^2 & a_2a_3 \\ a_3a_1 & a_3a_2 & a_3^2 \end{pmatrix} - \sin\phi\begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{4.154}$$

Continuing to multiply and sum terms, Eq. 4.154 becomes nine separate equations, one for each of the elements of the Attitude Matrix.

$$\begin{aligned}
\mathbf{A}_{11} &= \cos\phi + a_1^2 - a_1^2\cos\phi \\
\mathbf{A}_{12} &= a_1a_2 - a_1a_2\cos\phi + a_3\sin\phi \\
\mathbf{A}_{13} &= a_1a_3 - a_1a_3\cos\phi - a_2\sin\phi \\
\mathbf{A}_{21} &= a_2a_1 - a_2a_1\cos\phi - a_3\sin\phi \\
\mathbf{A}_{22} &= \cos\phi + a_2^2 - a_2^2\cos\phi \\
\mathbf{A}_{23} &= a_2a_3 - a_2a_3\cos\phi + a_1\sin\phi \\
\mathbf{A}_{31} &= a_3a_1 - a_3a_1\cos\phi + a_2\sin\phi \\
\mathbf{A}_{32} &= a_3a_2 - a_3a_2\cos\phi - a_1\sin\phi \\
\mathbf{A}_{33} &= \cos\phi + a_3^2 - a_3^2\cos\phi
\end{aligned} \tag{4.155}$$

It may seem a waste of space to multiply out all these terms when the math was so compact in the form shown above in Eq. 4.153. However, when coding this algorithm, several multiply by zero operations will be avoided if Equations 4.155 are used instead. So it may be a waste of space, but it is a savings in time when running GMAT.

Conversion: Euler Angles to Attitude Matrix

Given: Sequence order (i.e. 123, 121, ...**321**,...313), θ_1, θ_2, θ_3
Find: **A**
Name: *EulerAnglesToAMat*

We'll give an example for a 321 rotation, and then present results for the remaining 11 Euler angle sequences. First, let's define $\mathbf{A}_3(\theta_1)$, $\mathbf{A}_2(\theta_2)$, and $\mathbf{A}_1(\theta_3)$.

$$\mathbf{A}_3(\theta_1) = \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.156}$$

$$\mathbf{A}_2(\theta_2) = \begin{pmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix} \tag{4.157}$$

$$\mathbf{A}_1(\theta_3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{pmatrix} \tag{4.158}$$

Now we can write

$$\mathbf{A}_{321} = \mathbf{A}_1(\theta_3)\mathbf{A}_2(\theta_2)\mathbf{A}_3(\theta_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_3 & s_3 \\ 0 & -s_3 & c_3 \end{pmatrix} \begin{pmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{pmatrix} \begin{pmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.159}$$

where $c_1 = \cos\theta_1$, $s_1 = \sin\theta_1$ etc. We can rewrite $\mathbf{A}_{321}$ as

$$\mathbf{A}_{321} = \begin{pmatrix} c_2c_1 & c_2s_1 & -s_2 \\ -c_3s_1 + s_3s_2c_1 & c_3c_1 + s_3s_2s_1 & s_3c_2 \\ s_3s_1 + c_3s_2c_1 & -s_3c_1 + c_3s_2s_1 & c_3c_2 \end{pmatrix} \tag{4.160}$$

Equation 4.160 is equivalent to the matrix at the bottom of the left hand column of Table E-1 on Page 764 of Wertz. The approach for obtaining the Attitude matrix is similar for the remaining 11 Euler angle sequences. Rather than derive the Attitude matrices for the remaining 11 sequences, we present them in Table 4.3.

Conversion: Attitude Matrix to Euler Angles

Given: Sequence order (i.e. 123, 121, ...**321**,...313), **A**
Find: θ_1, θ_2, θ_3
Name: *AMatToEulerAngles*

We'll give an example for a 321 rotation, and then present results for the remaining 11 Euler angle sequences. Examining, Eq. 4.160, we see that

$$\frac{A_{21}}{A_{11}} = \frac{\cos\theta_2 \sin\theta_1}{\cos\theta_2 \cos\theta_1} \tag{4.161}$$

From this we can see that

$$\theta_1 = \tan^{-1}\frac{A_{21}}{A_{11}} \tag{4.162}$$

Further inspection of Eq. 4.160 shows us that

$$\theta_2 = \sin^{-1} A_{13} \tag{4.163}$$

At first glance, we may choose to calculate θ_3 using $\theta_3 = \tan^{-1}(A_{23}/A_{33})$. However, in the case that $\theta_2 = 90°$, this would result in the indeterminate case, $\theta_3 = \tan^{-1}(A_{23}/A_{33}) = \tan^{-1}(0/0)$. An improved method, found in the ADEAS mathematical specifications document, is to determine θ_3 using

$$\theta_3 = \tan^{-1}\left(\frac{A_{31}\sin\theta_1 - A_{32}\cos\theta_1}{-A_{21}\sin\theta_1 + A_{22}\cos\theta_1}\right) \tag{4.164}$$

Substituting values from Eq. 4.160 into Eq. 4.164, and using abbreviated notation, we see that

$$\theta_3 = \tan^{-1}\left(\frac{s_1(s_3s_1 + c_3s_2c_1) - c_1(-s_3c_1 + c_3s_2s_1)}{s_1(c_3s_1 - s_3s_2c_1) + c_1(c_3c_1 + s_3s_2s_1)}\right) \tag{4.165}$$

Now, if $\theta_2 = 90°$, and we substitute $c_2 = 0$ and $s_2 = 1$ into the above equation, we see we get a determinate form. Results for all twelve Euler Sequences are shown in Table 4.5.
Note: For all $\tan^{-1}$ we need to use a quadrant check (equivalent to atan2) to make sure the the correct quadrant is chosen.

Conversion: Angular Velocity to Euler Angles Rates

Given: Sequence (i.e. 123, 121, ... 313), θ_2, θ_3 $\boldsymbol{\omega}$
Find: $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$
Name: *AngVelToEulerAngles*

$$\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{pmatrix} = \mathbf{S}^{-1}(\theta_2,\theta_3)\boldsymbol{\omega} \tag{4.166}$$

$\mathbf{S}^{-1}(\theta_2,\theta_3)$ is dependent upon the Euler sequence. Table 4.4 contains the different expressions for $\mathbf{S}^{-1}(\theta_2,\theta_3)$ for each of the 12 unique Euler sequences.

Note: Each of the forms of $\mathbf{S}^{-1}$ have a possible singularity due to the appearance of either $\sin\theta_2$ or $\cos\theta_2$ in the denominator. If GMAT encounters a singularity, an error message is thrown, and the zero vector is returned.

Conversion: Euler Angles Rates to Angular Velocity

Given: Sequence (i.e. 123, 121, ... 313), θ_2, θ_3, $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$
Find: $\boldsymbol{\omega}$
Name: *EulerAnglesToAngVel*

$$\boldsymbol{\omega} = \mathbf{S}(\theta_2,\theta_3)\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{pmatrix} \tag{4.167}$$

$\mathbf{S}(\theta_2,\theta_3)$ is dependent upon the Euler sequence. Table 4.4 contains the different expressions for $\mathbf{S}^{-1}(\theta_2,\theta_3)$ for each of the 12 unique Euler sequences.

Conversion: Quaternions to Euler Angles

Given: $\mathbf{q}$, q_4, Euler Sequence
Find: θ_1, θ_2, and θ_3
Name: *QuatsToEulerAngles*

There is not a direct transformation to convert from the quaternions to the Euler Angles. GMAT first converts from the quaternion to the Attitude matrix using the algorithm presented above called "QuatsToAMat". The Attitude matrix is then used to calculate the Euler Angles for the given Euler angle sequence using the algorithm called "AMatToEulerAngles:.

Conversion: Euler Angles to Quaternions

Given: θ_1, θ_2, and θ_3, Euler Sequence
Find: $\mathbf{q}$, q_4
Name: *EulerAnglesToQuats*

There is not a direct transformation to convert from Euler Angles to quaternions. GMAT first converts from the Euler Angles to the Attitude matrix using the algorithm above called "EulerAnglesToAMat". The Attitude matrix is then used to calculate the quaternions using the algorithm called "AMatToQuats".

Table 4.3: Attitude Matrices for 12 Unique Euler Angle Rotation Sequences

$$\mathbf{A} = \mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_3c_2 & c_3s_2s_1 + s_3c_1 & -c_3s_2c_1 + s_1s_3 \\ -s_3c_2 & -s_3s_2s_1 + c_3c_1 & s_3s_2c_1 + c_3s_1 \\ s_2 & -c_2s_1 & c_2c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_3c_2 & c_3s_2c_1 + s_1s_3 & c_3s_2s_1 - s_3c_1 \\ -s_2 & c_2c_1 & c_2s_1 \\ s_3c_2 & s_3s_2c_1 - c_3s_1 & s_3s_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_2c_1 & s_2 & -c_2s_1 \\ -c_3s_2c_1 + s_3s_1 & c_3c_2 & c_3s_2s_1 + s_3c_1 \\ s_3s_2c_1 + c_3s_1 & -s_3c_2 & -s_3s_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_1 + s_3s_2s_1 & s_3c_2 & -c_3s_1 + s_3s_2c_1 \\ -s_3c_1 + c_3s_2s_1 & c_3c_2 & s_3s_1 + c_3s_2c_1 \\ c_2s_1 & -s_2 & c_2c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_1 - s_3s_2s_1 & c_3s_1 + s_3s_2c_1 & -s_3c_2 \\ -c_2s_1 & c_2c_1 & s_2 \\ s_3c_1 + c_3s_2s_1 & s_3s_1 - c_3s_2c_1 & c_3c_2 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_2c_1 & c_2s_1 & -s_2 \\ -c_3s_1 + s_3s_2c_1 & c_3c_1 + s_3s_2s_1 & s_3c_2 \\ s_3s_1 + c_3s_2c_1 & -s_3c_1 + c_3s_2s_1 & c_3c_2 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_2 & s_2s_1 & -s_2c_1 \\ s_3s_2 & c_3c_1 - s_3c_2s_1 & c_3s_1 + s_3c_2c_1 \\ c_3s_2 & -s_3c_1 - c_3c_2s_1 & -s_3s_1 + c_3c_2c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_2 & s_2c_1 & s_2s_1 \\ -c_3s_2 & c_3c_2c_1 - s_3s_1 & c_3c_2s_1 + s_3c_1 \\ s_3s_2 & -s_3c_2c_1 - c_3s_1 & -s_3c_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_1 - s_3c_2s_1 & s_3s_2 & -c_3s_1 - s_3c_2c_1 \\ s_2s_1 & c_2 & s_2c_1 \\ s_3c_1 + c_3c_2s_1 & -c_3s_2 & -s_3s_1 + c_3c_2c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_2c_1 - s_3s_1 & c_3s_2 & -c_3c_2s_1 - s_3c_1 \\ -s_2c_1 & c_2 & s_2s_1 \\ s_3c_2c_1 + c_3s_1 & s_3s_2 & -s_3c_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_1 - s_3c_2s_1 & c_3s_1 + s_3c_2c_1 & s_3s_2 \\ -s_3c_1 - c_3c_2s_1 & -s_3s_1 + c_3c_2c_1 & c_3s_2 \\ s_2s_1 & -s_2c_1 & c_2 \end{pmatrix}$$

$$\mathbf{A} = \mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_2c_1 - s_3s_1 & c_3c_2s_1 + s_3c_1 & -c_3s_2 \\ -s_3c_2c_1 - c_3s_1 & -s_3c_2s_1 + c_3c_1 & s_3s_2 \\ s_2c_1 & s_2s_1 & c_2 \end{pmatrix}$$

Table 4.4: Kinematics of Euler Angle Rotation Sequences

Euler Sequence	$\mathbf{S}(\theta_2, \theta_3)$	$\mathbf{S}^{-1}(\theta_2, \theta_3)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c_3c_2 & s_3 & 0 \\ -s_3c_2 & c_3 & 0 \\ s_2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c_3/c_2 & -s_3/c_2 & 0 \\ s_3 & c_3 & 0 \\ -s_2c_3/c_2 & s_3s_2/c_2 & 1 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c_3c_2 & -s_3 & 0 \\ -s_2 & 0 & 1 \\ s_3c_2 & c_3 & 0 \end{pmatrix}$	$\begin{pmatrix} c_3/c_2 & 0 & s_3/c_2 \\ -s_3 & 0 & c_3 \\ s_2c_3/c_2 & 1 & s_3s_2/c_2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s_2 & 0 & 1 \\ c_3c_2 & s_3 & 0 \\ -s_3c_2 & c_3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & c_3/c_2 & -s_3/c_2 \\ 0 & s_3 & c_3 \\ 1 & -s_2c_3/c_2 & s_3s_2/c_2 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s_3c2 & c_3 & 0 \\ c_3c_2 & -s_3 & 0 \\ -s_2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} s_3/c_2 & c_3/c_2 & 0 \\ c_3 & -s_3 & 0 \\ s_3s_2/c_2 & s_2c_3/c_2 & 1 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -s_3c_2 & c_3 & 0 \\ s_2 & 0 & 1 \\ c_3c_2 & s_3 & 0 \end{pmatrix}$	$\begin{pmatrix} -s_3/c_2 & 0 & c_3/c_2 \\ c_3 & 0 & s_3 \\ s_3s_2/c_2 & 1 & -s_2c_3/c_2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -s_2 & 0 & 1 \\ s_3c_2 & c_3 & 0 \\ c_3c_2 & -s_3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & s_3/c_2 & c_3/c_2 \\ 0 & c_3 & -s_3 \\ 1 & s_3s_2/c_2 & s_2c_3/c_2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c_2 & 0 & 1 \\ s_3s_2 & c_3 & 0 \\ c_3s_2 & -s_3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & s_3/s_2 & c_3/s_2 \\ 0 & c_3 & -s_3 \\ 1 & -s_3c_2/s_2 & -c_3c_2/s_2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c_2 & 0 & 1 \\ -c_3s_2 & s_3 & 0 \\ s_3s_2 & c_3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -c_3/s_2 & s_3/s_2 \\ 0 & s_3 & c_3 \\ 1 & c_3c_2/s_2 & -s_3c_2/s_2 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s_3s_2 & c_3 & 0 \\ c_2 & 0 & 1 \\ -c_3s_2 & s_3 & 0 \end{pmatrix}$	$\begin{pmatrix} s_3/s_2 & 0 & -c_3/s_2 \\ c_3 & 0 & s_3 \\ -s_3c_2/s_2 & 1 & c_3c_2/s_2 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} c_3s_2 & -s_3 & 0 \\ c_2 & 0 & 1 \\ s_3s_2 & c_3 & 0 \end{pmatrix}$	$\begin{pmatrix} c_3/s_2 & 0 & s_3/s_2 \\ -s_3 & 0 & c_3 \\ -c_3c_2/s_2 & 1 & -s_3c_2/s_2 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} s_3s_2 & c_3 & 0 \\ c_3s_2 & -s_3 & 0 \\ c_2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} s_3/s_2 & c_3/s_2 & 0 \\ c_3 & -s_3 & 0 \\ -s_3c_2/s_2 & -c_3c_2/s_2 & 1 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -c_3s_2 & s_3 & 0 \\ s_3s_2 & c_3 & 0 \\ c_2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -c_3/s_2 & s_3/s_2 & 0 \\ s_3 & c_3 & 0 \\ c_3c_2/s_2 & -s_3c_2/s_2 & 1 \end{pmatrix}$

Table 4.5: Computation of Euler Angles from Attitude Matrix

Euler Sequence	Euler Angle Computations		
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(-A_{32}/A_{33})$	$\theta_2 = \sin^{-1}(A_{31})$	$\theta_3 = \tan^{-1}\left(\frac{A_{13}\sin\theta_1 + A_{12}\cos\theta_1}{A_{23}\sin\theta_1 + A_{22}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(A_{23}/A_{22})$	$\theta_2 = \sin^{-1}(-A_{21})$	$\theta_3 = \tan^{-1}\left(\frac{A_{12}\sin\theta_1 - A_{13}\cos\theta_1}{-A_{32}\sin\theta_1 + A_{33}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(-A_{13}/A_{11})$	$\theta_2 = \sin^{-1}(A_{12})$	$\theta_3 = \tan^{-1}\left(\frac{A_{21}\sin\theta_1 + A_{23}\cos\theta_1}{A_{31}\sin\theta_1 + A_{33}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(A_{31}/A_{33})$	$\theta_2 = \sin^{-1}(-A_{32})$	$\theta_3 = \tan^{-1}\left(\frac{A_{23}\sin\theta_1 - A_{21}\cos\theta_1}{-A_{13}\sin\theta_1 + A_{11}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(-A_{21}/A_{22})$	$\theta_2 = \sin^{-1}(A_{23})$	$\theta_3 = \tan^{-1}\left(\frac{A_{32}\sin\theta_1 + A_{31}\cos\theta_1}{A_{12}\sin\theta_1 + A_{11}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(A_{12}/A_{11})$	$\theta_2 = \sin^{-1}(-A_{13})$	$\theta_3 = \tan^{-1}\left(\frac{A_{31}\sin\theta_1 - A_{32}\cos\theta_1}{-A_{21}\sin\theta_1 + A_{22}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(A_{12}/-A_{13})$	$\theta_2 = \cos^{-1}(A_{11})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{33}\sin\theta_1 - A_{32}\cos\theta_1}{A_{23}\sin\theta_1 + A_{22}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(A_{13}/A_{12})$	$\theta_2 = \cos^{-1}(A_{11})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{22}\sin\theta_1 + A_{23}\cos\theta_1}{-A_{32}\sin\theta_1 + A_{33}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(A_{21}/A_{23})$	$\theta_2 = \cos^{-1}(A_{22})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{33}\sin\theta_1 + A_{31}\cos\theta_1}{-A_{13}\sin\theta_1 + A_{11}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(A_{23}/-A_{21})$	$\theta_2 = \cos^{-1}(A_{22})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{11}\sin\theta_1 - A_{13}\cos\theta_1}{A_{31}\sin\theta_1 + A_{33}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(A_{31}/-A_{32})$	$\theta_2 = \cos^{-1}(A_{33})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{22}\sin\theta_1 - A_{21}\cos\theta_1}{A_{12}\sin\theta_1 + A_{11}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(A_{32}/A_{31})$	$\theta_2 = \cos^{-1}(A_{33})$	$\theta_3 = \tan^{-1}\left(\frac{-A_{11}\sin\theta_1 + A_{12}\cos\theta_1}{-A_{21}\sin\theta_1 + A_{22}\cos\theta_1}\right)$

4.4 Spacecraft Model

4.4.1 RF Hardware Models

The RF Hardware models in GMAT include transmitters, receiver, and transponders. For each type of RF Hardware there are models for signal transmission/reception feasibility, received frequency, and transmitted frequency to name a few. In the sections below we present these models for each type of RF Hardware.

Receiver

The receiver model supports a frequency model and a reception feasibility model. The frequency model allows the user to specify the frequency capabilities of the receiver using different methods. The feasibility model determines whether an RF signal sensed at the receiver can feasibly detected given the receivers capability as defined by the frequency model.

When the frequency model `CenterAndBandwidth` is selected, the feasibility test is as follows. Define the frequency transmitted by the originating transmitter as F_t, the receiver's center frequency as F_C, the bandwidth as ΔF, and the range rate between transmitter and receiver as $\dot{\rho}$ (which is positive when the distance ρ is increasing). The frequency at the receiver is calculated using

$$F_r = F_t(1 - \frac{\dot{\rho}}{c}) \tag{4.168}$$

where c is the speed of light. Define the upper limit of the receiver as $F_u = F_c + \Delta F/2$ and the lower limit as $F_l = F_c - \Delta F/2$. If

$$F_l \leq F_r \leq F_u \tag{4.169}$$

then the received frequency is received by the receiver model. If the above test fails, the receiver does not receive the signal.

Transmitter

Transponder

Antenna

4.4.2 Thruster Models

GMAT supports several thruster models. The thruster models employ physics and empirical data provided by the thruster manufacturer to model thrust and mass flow rate used in orbit and attitude equations of motion. The thrust magnitude and I_{sp} are assumed to be functions of thruster inlet flow conditions including pressure, temperature, and for bi-propellant thrusters, the oxidizer to fuel ratio.

In the following subsections we present models for thrust magnitude and mass flow rates for several thruster types. All thrusters have a location and orientation. The location is described in the spacecraft body system. The orientation can be described with respect to any coordinate system known to GMAT. Let's define the rotation matrix from the thruster frame $\mathcal{F}_T$ to Earth's MJ2000 Equator as $\mathbf{R}_T$. Then, the thrust used in the orbit equations of motion is

$$\mathbf{F}_T = F_T \mathbf{R}_T \hat{\mathbf{T}} \tag{4.170}$$

where F_T is the thrust magnitude and is thruster dependent, and $\hat{\mathbf{T}}$

Now let's look at how to calculate the thrust magnitude for a mono-propellant chemical thruster.

Mono-Propellant Chemical Thruster

temperature. The specific form of Eqs. (4.171) and (4.172) are determined by fitting test data to approximate thrust and I_{sp} as function T_i and P_i. The user can supply this relationship via a scripte

We assume thruster data is given as a function of thruster inlet properties (as opposed to thrust chamber properties), and thrust magnitude and I_{sp} are modelled using

$$F_T = f(P_i, T_i) \tag{4.171}$$

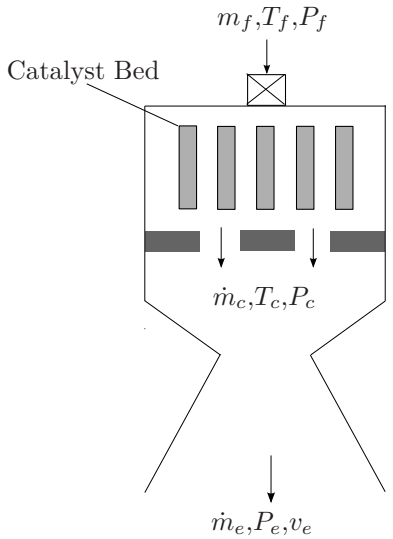


Figure 4.3: Mono-Prop Thruster Diagram

$$I_{sp} = f(P_i, T_i) \tag{4.172}$$

where P_i and T_i are the thruster inlet pressure and temperature. The specific form of Eqs. (4.171) and (4.172) are determined by fitting test data to approximate thrust and I_{sp} as function T_i and P_i. The user can supply this relationship via a scripted equation or by providing a function name. After calculating F_T and I_{sp}, we calculate the mass flow rate using

$$\dot{m}_e = \frac{F_T}{I_{sp}} \tag{4.173}$$

Bi-Propellant Chemical Thruster

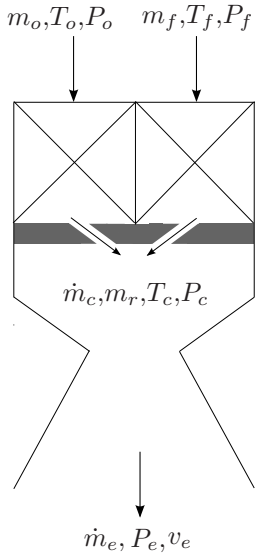


Figure 4.4: Bi-Prop Thruster Diagram

$$m_r = \frac{\dot{m}_o}{\dot{m}_f} \tag{4.174}$$

$$\dot{m}_c = \dot{m}_o + \dot{m}_f \tag{4.175}$$

$$T' = \frac{\dot{m}_o T_o + \dot{m}_f T_f}{\dot{m}_o + \dot{m}_f} \tag{4.176}$$

$$P' = \frac{\dot{m}_o P_o + \dot{m}_f P_f}{\dot{m}_o + \dot{m}_f} \tag{4.177}$$

$$F = f(P_c, T_c, of) \tag{4.178}$$

$$I_{sp} = f(P_c, T_c, of) \tag{4.179}$$

$$\dot{m}_e = \frac{\mathbf{F}_T}{I_{sp}} \tag{4.180}$$

Thruster Pulse Modelling

$$T' = \frac{T(t)}{T_{max}} = \begin{cases} \dfrac{t^2(t-t_{si})^2}{t_{si}^4} & t \le t_{si} \\ 1 & t_{si} < t < t_{sf} \\ \dfrac{(t-2t_{sf}+t_f)^2(t-t_f)^2}{(t_f-t_{sf})^4} & t_{sf} \le t \le t_f \end{cases} \tag{4.181}$$

The time to the thrust centroid, t_c, is calculated using

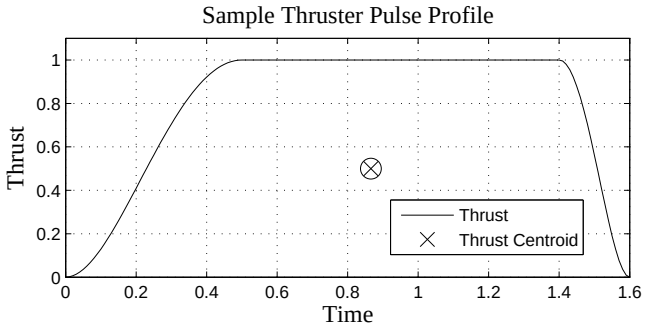


Figure 4.5: Sample Thrust Pulse Profile

$$t_c = \frac{\int_0^{t_f} t\ T'(t)dt}{\int_0^{t_f} T'(t)dt} \tag{4.182}$$

performing the integral yields

$$t_c = \frac{-4t_{si}^2 + 4t_{sf}^2 + 6t_{sf}t_f + 5t_f^2}{-14t_{si} + 14t_{sf} + 16t_f} \tag{4.183}$$

Thruster Hot Fire Test Data and Thruster Models

Thruster hot fire test data is used to develop empirical models that describe thruster performance as a function of inlet conditions such as fuel pressure and temperature. In this section we'll discuss how the empirical models are developed and discuss how the empirical models are consistent with the physical models. First we present the physics model for a thruster test stand experiment. Next we present what is measured during a thrust stand test, and show how the measurements are used in combination with the physics model to generate a model of thruster performance over a given range of thruster inlet conditions.

In Fig. 4.6 we see an illustration of a simple thrust test setup. The thruster is mounted to a rigid surface. The force due to thrust, F_T, can be written as

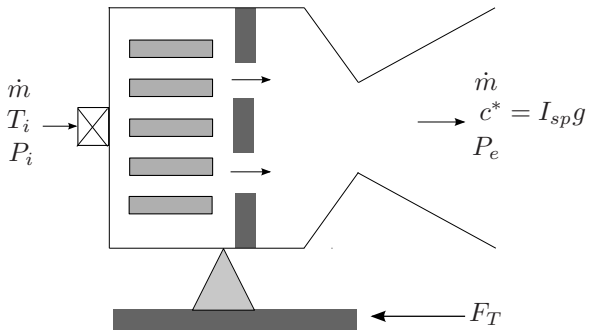


Figure 4.6: Thrust Stand Illustration

$$F_T = \dot{m}v_e - (P_e - P_a)A_e \tag{4.184}$$

where

$\dot{m}$	mass flow rate, kg/s
T_i	fuel inlet temperature, K°
P_i	fuel inlet pressure, Pa
c^*	characteristic velcocity, m/s
I_sp	specific impulse, s
g_o	9.801 m/s^2
P_e	nozzle exit pressure, Pa
A_e	nozzle exit area, m^2
F_T	force due to thrust, N

we can rewrite this as

$$F_T = \dot{m}\left(v_e - \frac{(P_e - P_a)A_e}{\dot{m}}\right) \tag{4.185}$$

From this equation we define the characteristic velocity, c^*, using

$$c^* = \left(v_e - \frac{(P_e - P_a)A_e}{\dot{m}}\right) \tag{4.186}$$

In practice, v_e or P_e are not measured. We'll assume the tests are performed in a vacuum so $P_a = 0$. To understand how we relate the measurements to physical model, let's define a new quantity I_{sp}, where

$$I_{sp} = \frac{F_T}{\dot{m}g_o} \tag{4.187}$$

we can rewrite this as

$$F_T = \dot{m}I_{sp}g_o \tag{4.188}$$

Comparing Eq.s (4.185) and (4.188) we see that

$$c^* = I_{sp}g_o = v_e - \frac{(P_e - P_a)A_e}{\dot{m}} \tag{4.189}$$

Equation (4.189) shows that I_{sp} is a measure of the effective (characteristic) exhaust velocity. I_{sp} contains information on the energy stored in the fuel and how that energy translates to exit velocity. When I_{sp} is calculated from measured thrust data, the I_{sp} contains a correction for exhaust velocity for force due to pressure $(P_e - P_a)A_e$.

The characteristic velocity, and hence I_{sp}, depend on the type of fuel and the inlet temperature and pressure of the fuel. Experimental data determines how

$$I_{sp}(T_i, P_i) = \left.\frac{F_T}{\dot{m} g_o}\right|_{(T_i, P_i)} \tag{4.190}$$

$\dot{m}$	Known
T_i	Known
P_i	Known
g_o	Known
F_T	Measured
I_sp	Calculated

4.4.3 Tank Models

Accurately modelling tank pressure changes is essential for accurate maneuver modelling and reconstruction. The following sections discuss three types of tank models: pressurant tank, regulated fuel tank, and blowdown fuel tank. For each tank there are three models: isothermal, heat transfer, and adiabatic.

The models used in GMAT are based on work by Estey[16] , Hearn[17] and Moran.[18] For each tank, we select a set of state variables that when defined, allow us to determine all remaining properties of the tank. For the state variables, we provide differential equations that describe how the state variables change with respect to time. The number of state variables varies between different tanks, and with the model type such as isothermal and heat transfer.

For each of the three tanks, we develop a heat transfer model, an adiabatic model, and an isothermal model. The heat transfer model is derived using the laws of conservation of energy and the conservation of mass. An adiabatic model is provided by setting the heat transfer rates to zero in the heat transfer model. The isothermal model for each tank is developed separately. Each ot these models is useful for certain classes of maneuvers. Isothermal models are useful for maneuvers with low mass flow rates, adiabatic models are useful for short maneuvers with high mass flow rates. Heat transfer models are useful for long maneuvers with large mass flow rates.

When developing heat transfer models, we'll assume that specific internal energy is given by

$$u = cT \tag{4.191}$$

specific enthalpy for a liquid is given by

$$h_\ell = c_\ell T_\ell \tag{4.192}$$

and specific enthalpy for a gas is given by

$$h_g = T_g(c_g + R_g) \tag{4.193}$$

The notation used in tank model development is shown below. After presenting notation, we present the dynamics model for a pressurant tank.

Nomenclature

A_g, A_ℓ, A_w	= Heat transfer area
c_v, c_g	= Specific heat at constant volume
D	= Tank diameter
d	= Liquid surface diameter
Gr	= Grashof number
h_ℓ, h_v	= Enthalpy
m_g, m_ℓ, m_w, m_v	= Mass
P_g, P_v, P_t	= Pressure
R_v, R_g	= Gas constant
$T_g, T_\ell, T_w, T_v, T_a$	= Temperature
u_g, u_ℓ, u_w, u_v	= Specific internal energy
V_g, V_ℓ, V_t	= Volume
$\dot{W}$	= Work rate
$\dot{Q}_g, \dot{Q}_v, \dot{Q}_l, \dot{Q}_w$	= Heat transfer rate
ν_l, ν_g, ν_v	= Specific volume

Subscripts

a	= Ambient
g	= Pressurant gas
ℓ	= Propellant liquid
t	= Total
v	= Propellant vapor
w	= Tank wall
e	= Exit-flow
i	= In-flow

Pressurant Tank

The pressurant tank model is the simplest of the tank models primarily due to the fact that there is only one substance, the pressurant gas, contained in the tank. In this section, we develop a state space model for pressurant tank dynamics. We choose the state variables to be pressurant gas mass and temperature, m_g and T_g respectively, and tank wall temperature T_w.

In Fig.4.11 we see an illustration of a pressurant tank. We divide the tank into two control volumes: the gas region and the tank wall region. The only mass flow in the system occurs where pressurant gas exits the tank. Heat transfer occurs between the gas and the wall, and the wall and the ambient surroundings.

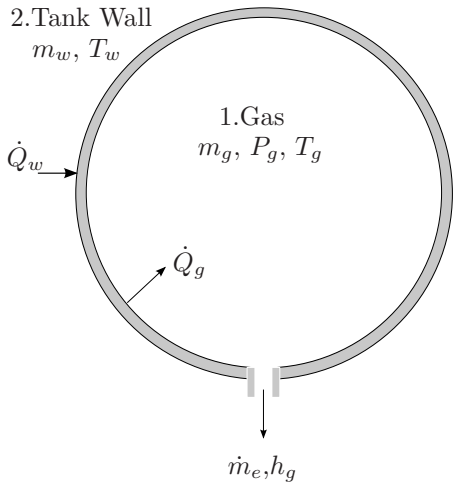


Figure 4.7: Pressurant Tank Diagram

Knowing the volume of the tank and the state variables m_g, T_g, and T_w, we calculate pressure from one of the following two equations of state:

$$P_g = \frac{m_g R_g T_g}{V_g} \tag{4.194}$$

or from the Beattie-Bridgeman Eq.

$$P_g = \frac{R_g T_g}{V_g} + \frac{a_g}{V_g^2} + \frac{b_g}{V_g^3} \tag{4.195}$$

The state variables m_g, T_g, and T_w are described by ordinary differential equations found by applying the first law of thermodynamics and the conservation of mass. The 1st Law applied to the gas control volume yields

$$\frac{d}{dt}\left(m_g u_g\right) = \dot{Q}_g - \dot{m}_e h_g \tag{4.196}$$

The 1st Law applied to the wall control volume yields

$$\frac{d}{dt}\left(m_w u_w\right) = \dot{Q}_w - \dot{Q}_g \tag{4.197}$$

and finally from conservation of mass we obtain

$$\dot{m}_g = -\dot{m}_e \tag{4.198}$$

For these equations to be useful for numerical integration, we need to expand the derivatives, and if necessary, decouple the equations (as we'll see, for the pressurant tank, the equations are not coupled).

Expanding the terms in Eq. (4.296) we have

$$\dot{m}_g c_g T_g + m_g c_g \dot{T}_g = \dot{Q}_g - \dot{m}_e T_g \left(c_g + R_g\right) \tag{4.199}$$

Similarly, expanding Eq. (4.297) we obtain

$$m_w c_w \dot{T}_w = \dot{Q}_w - \dot{Q}_g \tag{4.200}$$

Solving the system of equations yields the following differential equations of state for the pressurant tank heat transfer model.

$$\dot{m}_g = -\dot{m}_e \tag{4.201}$$

$$\dot{T}_g = \frac{1}{m_g c_g}\left(\dot{Q}_g - T_g R_g \dot{m}_e\right) \tag{4.202}$$

$$\dot{T}_w = \frac{1}{m_w c_w}\left(\dot{Q}_w - \dot{Q}_g\right) \tag{4.203}$$

The adiabatic model is obtained by setting the terms $\dot{Q}_g$ and $\dot{Q}_w$ to zero in the above equations. (Note for the adiabatic model there are only two state variables, m_g and T_g, as the wall temperature T_w is removed from the system of equations.) Similarly, the isothermal model is obtained by setting $\dot{T}_g$ and $\dot{T}_w$ to zero. So, for the isothermal model there is only one state variable m_g.

In summary, for the pressurant tank, all models calculate the tank pressure using

$$P_g = \frac{m_g R_g T_g}{V_g}$$

then the specific equations for the heat transfer, adiabatic, and isothermal models, are as follows
Pressurant Tank: Heat Transfer
State Variables: m_g, T_g, T_w

$$\begin{aligned}\dot{m}_g &= -\dot{m}_e \\ \dot{T}_g &= \frac{1}{m_g c_g}\left(\dot{Q}_g - T_g R_g \dot{m}_e\right) \\ \dot{T}_w &= \frac{1}{m_w c_w}\left(\dot{Q}_w - \dot{Q}_w\right)\end{aligned}$$

Pressurant Tank: Adiabtic
State Variables: m_g, T_g

$$\begin{aligned}\dot{m}_g &= -\dot{m}_e \\ \dot{T}_g &= \dot{m}_e \frac{T_g R_g}{m_g c_g}\end{aligned}$$

Pressurant Tank: Isothermal
State Variables: m_g

$$\dot{m}_g = -\dot{m}_e \tag{4.204}$$

Now let's look at a model for a fuel tank operating in blow down mode.

Blowdown Tank

The blowdown tank model is significantly more complex than the pressurant tank model due to the presence of liquid fuel and fuel vapor contained in the tank ullage. In this section, we develop a state space model for a blow down tank. We choose the state variables to be the liquid mass and temperature, m_ℓ and T_ℓ, the gas temperature T_g, and tank wall temperature T_w.

In Fig.4.12 we see an illustration of a blow down tank. We divide the tank into three control volumes: the gas region, the liquid region, and the tank wall region. Mass flow occurs where the pressurant gas exits the tank and at the boundary between the liquid and gas in the form of evaporation. Heat transfer occurs between all three control volumes as well as with the surroundings. In summary, the physical processes modelled for a blow down tank are

1. Vapor pressure is a function of liquid temperature.
2. Liquid density is a function of liquid temperature.
3. Heat transfer between the liquid and gas.
4. Heat transfer between the tank wall and gas.
5. Heat transfer between the tank wall and liquid.
6. Heat transfer between the surroundings and tank. wall.

The assumptions made in the tank model are

1. Pressurant does not dissolve in liquid ($m_g = C$).
2. Vapor and gas temperatures are equal.
3. Vapor and gas volumes are equal.

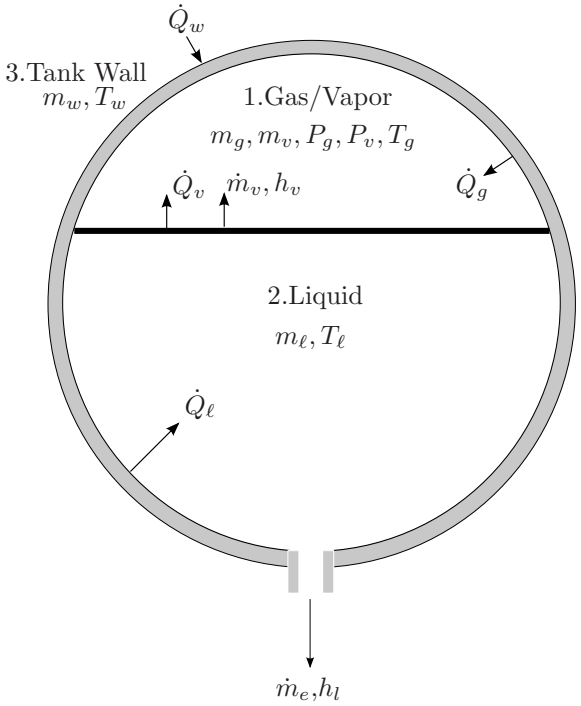


Figure 4.8: Blow Down Tank Diagram

Assume we are given m_g, the tank diameter D, and hence know the total tank volume V_t, and we know the physical constants associated with the liquid and gas (R_g, c_g, ν_g, c_ℓ, $\nu_\ell(T_\ell)$ and $P_v(T_\ell)$). We choose the state variables m_ℓ, T_ℓ, T_g, and T_w, all other tank properties can be calculated from these state variables using the following equations:

$$V_\ell = \nu_\ell(T_\ell) m_\ell \tag{4.205}$$
$$V_g = V_t - V_\ell \tag{4.206}$$
$$P_g = \frac{m_g R_g T_g}{V_g} \tag{4.207}$$
$$P_v = P_v(T_\ell) \tag{4.208}$$
$$m_v = \frac{P_v V_g}{R_v T_g} \tag{4.209}$$
$$P_t = P_v + P_g \tag{4.210}$$

To determine the state equations governing m_ℓ, T_ℓ, T_g, and T_w we apply the 1st law of thermodynamics and the law of conservation of mass. The 1st Law applied to the gas control volume is

$$\frac{d}{dt}\left(m_v u_v + m_g u_g\right) = \dot{Q}_v + \dot{Q}_g - P_t \dot{V}_g + \dot{m}_v h_v \tag{4.211}$$

The 1st Law applied to the liquid control volume is

$$\frac{d}{dt}\left(m_\ell u_\ell\right) = \dot{Q}_\ell - \dot{Q}_v + P_t\dot{V}_g - \dot{m}_v h_{lg} - \dot{m}_e h_\ell \tag{4.212}$$

The 1st Law applied to the wall control volume yields

$$\frac{d}{dt}\left(m_w u_w\right) = \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g \tag{4.213}$$

and finally from conservation of mass:

$$\dot{m}_\ell = -\dot{m}_e - \dot{m}_v \tag{4.214}$$

we also know that

$$\dot{m}_v = \frac{P_v\dot{V}_g}{R_v T_g} - \frac{P_v V_g \dot{T}_g}{R_v T_g^2} \tag{4.215}$$

where we assume that

$$\dot{P}_v \approx 0 \tag{4.216}$$

Equations (4.311) - (4.315) are five equations in five unknowns (m_v, m_ℓ, T_ℓ, T_g, and T_w). Our approach is to use Eq. (4.314) to eliminate $\dot{m}_v$ terms. The result is a system of four equations in four unknowns using Eqs. (4.311), (4.312), (4.313), and (4.315). The result we seek is four decoupled ordinary differential equations for m_ℓ, T_ℓ, T_g, and T_w.

Let's continue with Eq. (4.311). We need to rewrite the equation in terms of $\dot{m}_\ell$ and $\dot{T}_g$ ($\dot{T}_w$ and $\dot{T}_\ell$ don't appear explicitly). Expanding the derivatives assuming $\dot{m}_g = 0$ yields

$$\dot{m}_v c_v T_g + m_v c_v \dot{T}_g + m_g c_g \dot{T}_g = \dot{Q}_v + \dot{Q}_g - P_t\dot{V}_g + \dot{m}_v h_v \tag{4.217}$$

Now, substituting $\dot{m}_v = -\dot{m}_\ell - \dot{m}_e$ and noting that $\dot{V}_g = -\nu_l \dot{m}_\ell$ if we assume

$$\dot{\nu}_\ell = \frac{d\nu_\ell}{dT_\ell}\dot{T}_\ell \approx 0$$

we arrive at

$$\begin{aligned}\left(T_g R_v - P_t\nu_\ell\right)\dot{m}_\ell + \left(m_v c_v + m_g c_g\right)\dot{T}_g = \\ \dot{Q}_v + \dot{Q}_g - \dot{m}_e T_g R_v\end{aligned} \tag{4.218}$$

Now continuing with Eq. (4.312) expanding the derivatives and making similar substitutions as we made previously we obtain

$$\begin{aligned}\dot{m}_\ell c_\ell T_\ell + m_\ell c_\ell \dot{T}_\ell = \dot{Q}_\ell - \dot{Q}_v + P_t(-\nu_\ell\dot{m}_\ell) - \\ (-\dot{m}_\ell - \dot{m}_e)h_v - \dot{m}_e c_\ell T_\ell\end{aligned} \tag{4.219}$$

Grouping terms we obtain

$$\begin{aligned}(c_\ell T_\ell + P_t v_l - h_v)\dot{m}_\ell + (m_\ell c_\ell)\dot{T}_\ell = \\ \dot{Q}_\ell - \dot{Q}_v + \dot{m}_e(h_v - c_\ell T_\ell)\end{aligned} \tag{4.220}$$

For the wall region, described by Eq. (4.313), we arrive at

$$(m_w c_w)\dot{T}_w = \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g \tag{4.221}$$

Finally, by eliminating $\dot{m}_v$ in the Gas Law shown in Eq. (4.315) we obtain

$$-\dot{m}_\ell - \dot{m}_e = \frac{P_v(-\nu_\ell\dot{m}_\ell)}{R_v T_g} - \frac{P_v V_g \dot{T}_g}{R_v T_g^2} \tag{4.222}$$

Grouping terms yields the result

$$\left(1 - \frac{P_v \nu_\ell}{R_v T_g}\right) \dot{m}_\ell - \frac{P_v V_g}{R_v T_g^2} \dot{T}_g = -\dot{m}_e \tag{4.223}$$

Equations (4.318), (4.320), (4.321), and (4.323) are four coupled ordinary differential equations that can be decoupled by casting them in matrix form as follows:

$$\begin{pmatrix} A_{11} & 0 & A_{13} & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & 0 & A_{34} \\ A_{41} & & A_{43} & 0 \end{pmatrix} \begin{pmatrix} \dot{m}_\ell \\ \dot{T}_\ell \\ \dot{T}_g \\ \dot{T}_w \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \tag{4.224}$$

where

$$\begin{aligned} A_{11} &= T_g R_v - P_t \nu_\ell && (4.225)\\ A_{13} &= m_v c_v + m_g c_g && (4.226)\\ A_{21} &= c_\ell T_\ell + P_t v_l - h_v && (4.227)\\ A_{22} &= m_\ell c_\ell && (4.228)\\ A_{34} &= m_w c_w && (4.229)\\ A_{41} &= 1 - \nu_\ell / \nu_v && (4.230)\\ A_{43} &= -m_v / T_g && (4.231)\\ b_1 &= \dot{Q}_v + \dot{Q}_g - \dot{m}_e T_g R_v && (4.232)\\ b_2 &= \dot{Q}_\ell - \dot{Q}_v + \dot{m}_e (h_v - c_\ell T_\ell) && (4.233)\\ b_3 &= \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g && (4.234)\\ b_4 &= -\dot{m}_e && (4.235) \end{aligned}$$

The solution to the equations is

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43} b_1 - A_{13} b_4}{A_{11} A_{43} - A_{41} A_{13}} && (4.236)\\ \dot{T}_\ell &= \frac{1}{A_{22}} \left(b_2 - A_{21} \frac{A_{43} b_1 - A_{13} b_4}{A_{11} A_{43} - A_{41} A_{13}} \right) && (4.237)\\ \dot{T}_g &= \frac{A_{11} b_4 - A_{41} b_1}{A_{11} A_{43} - A_{41} A_{13}} && (4.238)\\ \dot{T}_w &= \frac{b_3}{A_{34}} && (4.239) \end{aligned}$$

For the adiabatic model we set all heat transfer rates, $\dot{Q}$, to zero in Eqs. (4.332)-(4.335) and so there are only two state variables as $\dot{T}_w = 0$ and so $T_w =$ constant.

Now let's develop equations for an isothermal model of a blow down tank. In the isothermal model, we assume $T_\ell = T_g = T_w = T$. The only state variable that requires a differential equation is m_ℓ. Because T_g, T_ℓ, and hence, P_v are constant, we know that

$$\dot{m}_v = \frac{P_v \dot{V}_g}{R_v T_g} \tag{4.240}$$

Subtituting this result into Eq.(4.314) and solving for $\dot{m}_\ell$ we obtain.

$$\dot{m}_\ell = -\frac{\dot{m}_e}{\left(1 - \frac{P_v \nu_\ell}{R_v T}\right)} \tag{4.241}$$

In summary for the heat transfer model for a blow down tank, we choose m_ℓ, T_ℓ, T_g, and T_w are state variables. Eqs. (4.305)-(4.310) are used to calculate the remaining tank properties, and Eqs. (4.336)-(4.339) are used to model the tank states as functions of time.

For the all three models, heat transfer, adiabatic, and isothermal, knowing the state variables m_ℓ, T_ℓ, T_g, and T_w we compute the remaining tank properties using

$$\begin{aligned} V_\ell &= \nu_\ell(T_\ell) m_\ell \\ V_g &= V_t - V_\ell \\ P_g &= \frac{m_g R_g T_g}{V_g} \\ P_v &= P_v(T_\ell) \\ m_v &= \frac{P_v V_g}{R_v T_g} \\ P_t &= P_v + P_g \end{aligned}$$

The models differ in the number of state variables and in the state rate equations. A summary is presented below.

Blow Down Tank: Heat Transfer
State Variables: m_ℓ, T_ℓ, T_g, T_w

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_\ell &= \frac{1}{A_{22}}\left(b_2 - A_{21}\frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}}\right) \\ \dot{T}_g &= \frac{A_{11}b_4 - A_{41}b_1}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_w &= \frac{b_3}{A_{34}} \end{aligned}$$

where A_{ij} and b_i are given by Eqs. (4.325)-(4.335).

Blow Down Tank: Adiabtic
State Variables: m_ℓ, T_ℓ, T_g

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_\ell &= \frac{1}{A_{22}}\left(b_2 - A_{21}\frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}}\right) \\ \dot{T}_g &= \frac{A_{11}b_4 - A_{41}b_1}{A_{11}A_{43} - A_{41}A_{13}} \end{aligned}$$

where A_{ij} and b_i are given by Eqs. (4.325)-(4.335). Note that all heat flow rates, $\dot{Q}$, are set to zero.

Blow Down Tank: Isothermal
State Variables: m_ℓ

$$\dot{m}_\ell = -\frac{\dot{m}_e}{\left(1 - \frac{P_v \nu_\ell}{R_v T}\right)}$$

Pressure Regulated Tank

The pressure regulated fuel tank model is the most complex tank model supported in GMAT. The model complexity is due to the presence of liquid fuel and fuel vapor contained in the tank ullage, and due to mass and energy transfer from the pressurant tank to the ullage of the regulated fuel tank. In this section, we develop a state space model for a pressure regulated tank. Note, to model a pressure regulated tank using a

heat transfer or adiabitic model, we must simultaneously solve the equations associated with the pressurant tank. For the regulated tank model, we choose the state variables to be the liquid mass and temperature, m_ℓ and T_ℓ, the gas temperature T_g, tank wall temperature T_w, and the incoming pressurant gas mass m_i.

In Fig.4.13 we see an illustration of a pressure regulated tank. Like the blow down tank model, we divide the tank into three control volumes: the gas region, the liquid region, and the tank wall region. Mass flow occurs where the pressurant gas exits the tank, at the boundary between the liquid and gas in the form of evaporation, and from the pressurant tank to the ullage of the regulated tank. Heat transfer occurs between all three control volumes as well as with the surroundings. Hence, the physical processes modelled for a blow down tank are the same as those listed for the blow down tank, with the added process of mass flow from the pressurant tank.

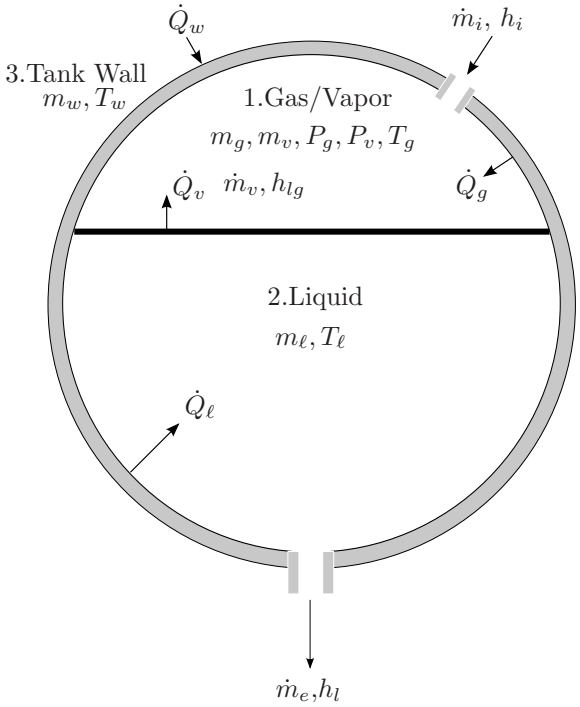


Figure 4.9: Pressure Regulated Tank Diagram

The derivation of the state equations for a pressure regulated tank follows naturally from the derivation of the blow down tank. The only control volume that differs between the two models is the gas/vapor control volume. Applying the 1st Law of thermodynamics to the gas/vapor control volume of the pressure regulated tank gives us

$$\frac{d}{dt}\left(m_v u_v + m_g u_g\right) = \dot{Q}_v + \dot{Q}_g - P_t \dot{V}_g + \dot{m}_v h_v + \dot{m}_p h_p \tag{4.242}$$

Taking the time derivative of the gas law for the gas contained in the tank ullage yields

$$\dot{m}_g = \frac{P_g \dot{V}_g}{R_g T} - \frac{P_g V_g \dot{T}_g}{R_g T_g^2} \tag{4.243}$$

Equations (4.342) and (4.343), together with equations (4.320), (4.321), and (4.323) are a system of 5 equations in 5 unknowns which can be decoupled using simple linear algebra. However, first we must expand Eqs. (4.342) and (4.343) and write them in terms of the state rate derivatives. Expanding Eq. (4.342) we arrive at

$$\begin{aligned}\left(R_v T_g - P_t \nu_\ell\right)\dot{m}_\ell + \left(m_v c_v + m_g c_g\right)\dot{T}_g + \\ \left(c_g T g - h_p\right)\dot{m}_g = \dot{Q}_v + \dot{Q}_g - \dot{m}_e R_v T_g\end{aligned} \tag{4.244}$$

Similarly, for Eq. (4.343) we obtain

$$\frac{\nu_\ell}{\nu_g}\dot{m}_\ell + \frac{m_g}{T_g}\dot{T}_g + \dot{m}_g = 0 \tag{4.245}$$

To integrate the state equations we must decouple the equations and this is easily done by casting the equations in matrix form and solving the system of equations. We can write the equations is state space for as follows

$$\begin{pmatrix} A_{11} & 0 & A_{13} & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{34} & 0 \\ A_{41} & 0 & A_{43} & 0 & 0 \\ A_{51} & 0 & A_{43} & 0 & A_{55} \end{pmatrix} \begin{pmatrix} \dot{m}_\ell \\ \dot{T}_\ell \\ \dot{T}_g \\ \dot{T}_w \\ \dot{m}_g \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ 0 \end{pmatrix} \tag{4.246}$$

where the coefficients A_{ij} and b_i are given by

$$\begin{aligned} A_{11} &= T_g R_v - P_t \nu_\ell && (4.247)\\ A_{13} &= m_v c_v + m_g c_g && (4.248)\\ A_{15} &= c_g T_g - h_p && (4.249)\\ A_{21} &= c_\ell T_\ell + P_t v_l - h_{lg} && (4.250)\\ A_{22} &= m_\ell c_\ell && (4.251)\\ A_{34} &= m_w c_w && (4.252)\\ A_{41} &= 1 - \nu_\ell/\nu_v && (4.253)\\ A_{43} &= -m_v/T_g && (4.254)\\ A_{51} &= \nu_\ell/\nu_g && (4.255)\\ A_{53} &= m_g/T_g && (4.256)\\ A_{55} &= 1 && (4.257)\\ b_1 &= \dot{Q}_v + \dot{Q}_g - \dot{m}_e R_v T_g && (4.258)\\ b_2 &= \dot{Q}_\ell - \dot{Q}_v + \dot{m}_e(h_{lg} - c_\ell T_\ell) && (4.259)\\ b_3 &= \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g && (4.260)\\ b_4 &= -\dot{m}_e && (4.261) \end{aligned}$$

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{55}A_{43}b_1 - b_4 A_{13}A_{55} + b_4 A_{15}A_{53}}{D} && (4.262)\\ \dot{T}_\ell &= \frac{b_2 - A_{21}\dot{m}_\ell}{A_{22}} && (4.263)\\ \dot{T}_g &= \frac{-A_{41}A_{55}b_1 + b_4 A_{11}A_{55} - b_4 A_{51}A_{15}}{D} && (4.264)\\ \dot{T}_w &= \frac{b_3}{A_{34}} && (4.265)\\ \dot{m}_g &= \frac{-b_1 A_{51}A_{43} + b_1 A_{41}A_{53} - b_4 A_{11}A_{53} + b_4 A_{51}A_{13}}{D} && (4.266) \end{aligned}$$

where

$$D = A_{55}A_{43}A_{11} - A_{43}A_{51}A_{15} + A_{41}A_{15}A_{53} - A_{41}A_{13}A_{55} \tag{4.267}$$

For the adiabatic model we set all heat transfer rates, $\dot{Q}$, to zero in Eqs. (4.358)-(4.361). For the adiabatic model there is only four state variables as $\dot{T}_w = 0$ and so $T_w =$ constant.

Now let's develop equations for an isothermal model of a pressure regulated tank. In the isothermal model, we assume $T_\ell = T_g = T_w = T$. The only state variables that require differential equations are m_ℓ and m_g. Because $T_g =$ constant, and hence, $P_v =$ constant, we know that

$$\dot{m}_\ell = -\frac{\dot{m}_e}{\left(1 - \dfrac{P_v \nu_\ell}{R_v T}\right)} \tag{4.268}$$

Similarly, for m_g we obtain

$$\dot{m}_g = \frac{\nu_\ell}{\nu_g} \frac{\dot{m}_e}{\left(1 - \frac{P_v \nu_\ell}{R_v T}\right)} \tag{4.269}$$

Heat Transfer

Heat transfer models are from Ring[19] and Incropera[2] and Pitts[20]

$$\dot{Q} = hA\Delta T \tag{4.270}$$

$$\frac{hL}{k} = Nu = c(Gr_L Pr)^a \tag{4.271}$$

so

$$h = \frac{kc}{L}(Gr_L Pr)^a \tag{4.272}$$

Table 4.6: Dimensionless Heat Transfer Terms[2]

Parameter	Definition	Interpretation
Grashof number (Gr_L)	$\frac{g\beta(T_s - T_\infty)L^3}{\nu^2}$	Measure of the ratio of buoyancy forces to viscous forces.
Prandtl number (Pr)	$\frac{c\mu}{k} = \frac{\mu}{\alpha}$	Ratio of the momentum and thermal diffusivities.
Nusselt number (Nu_L)	$\frac{hL}{k_f}$	Ratio of convection to pure conduction heat transfer.
Reynolds number (Re_L)	$\frac{VL}{\nu}$	Ratio of inertia and viscous forces.

Physiochemical Properties

Hydrazine Properties[21]

$$c = 3084 \text{J/kg/K}$$

$$\rho\ (\text{kg/m}^3) = 1025.6 - 0.87415\ T\ (^o\text{C}) - 0.45283e^{-3}\ T^2\ (^o\text{C})$$

$$\rho\ (\text{kg/m}^3) = 1230.6 - 0.62677\ T\ (^o\text{K}) - 0.45283e^{-3}\ T^2\ (^o\text{K})$$

Some models are from[22]

$$\rho_\ell(T_\ell) = K_1 + K_2 T_\ell + K_3 T_\ell^2\ (\text{kg/m}^3) \tag{4.273}$$

$$\frac{d\rho_\ell}{dT_\ell} = K_2 + 2K_3 T_\ell\ (\text{kg/m}^3) \tag{4.274}$$

$$P_v = C_1 e^{(C_2 + C_3 T_\ell + C_4 T_\ell^2)}\ (\text{N/m}^2) \tag{4.275}$$

$$\frac{dP_v}{dT_\ell} = C_1 ln(C_2 + C_3 T_\ell C_4 T_\ell^2)\,(C_3 + 2C_4 T_\ell) \tag{4.276}$$

Table 4.7: Constants for Density Equations

Const.	N_20_4	$CH_3N_2H_3$
K_1 (kg/m^3)	2066	1105.3
K_2 (kg/m^3/K)	-1.979	-0.9395
K_3 (kg/m^3/K^2)	-4.826e-4	0

Table 4.8: Constants for Vapor Pressure Equations

Const.	N_20_4	$CH_3N_2H_3$
C_1 (kg/m^3)	6895	6895
C_2 (kg/m^3/K)	18.6742	12.4293
C_3 (kg/m^3/K^2)	-5369.57	2543.37
C_4 (kg/m^3/K^2)	194721	0

$$m_d = P_g m_\ell^\alpha \left(\frac{T_\ell}{294}\right)^2 \tag{4.277}$$

$$\begin{aligned}\frac{dm_d}{dt} = m_\ell^\alpha \left(\frac{T_\ell}{294}\right)^2 \dot{P}_g + \alpha P_g m_\ell^{(\alpha-1)} \left(\frac{T_\ell}{294}\right)^2 \dot{m}_\ell \\ +2P_g m_\ell^\alpha \left(\frac{T_\ell}{294}\right) \dot{T}_\ell\end{aligned} \tag{4.278}$$

Table 4.9: Constants for Dissolved Pressurant Equations

Const.	N_20_4	$CH_3N_2H_3$
α	3.335e-11	2.059e-11

4.4.4 Mass Properties

Spherical Tank

($\hat{x}$ is out of the page)

Figure 4.10: Geometry For Mass Properties of Partially Filled Spherical Tank

$$V = \frac{1}{3}\pi\left(3r - h\right)h^2 \tag{4.279}$$

$$cg_z = \frac{-\frac{3}{4}h^2 + 3hr - 3r^2}{3r - h} \tag{4.280}$$

$$cg_x = cg_y = 0 \tag{4.281}$$

$$I_{zz} = \frac{\pi\rho}{2}\left(\frac{1}{5}\left(h - r\right)^5 - \frac{2}{3}r^2\left(h - r\right)^3 + r^4(h - r) + \frac{8}{15}r^5\right) \tag{4.282}$$

$$I_{xx} = \frac{\pi\rho}{2}\left(-\frac{3}{10}\left(h - r\right)^5 + \frac{1}{3}r^2\left(h - r\right)^3 + \frac{1}{2}r^4(h - r) + \frac{8}{15}r^5\right) \tag{4.283}$$

$$I_{yy} = I_{xx} \tag{4.284}$$

$$\mathbf{I}' = \mathbf{R}_{bt}^T \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \mathbf{R}_{bt} \tag{4.285}$$

Estey[16] gives appoximate equations for the area of different portions for a partially filled sphere. The area of the spherical shell in contact with the gaseous region is given by

$$A_g = 4.0675 \cdot V^{2/3}\left(\frac{V_g}{V}\right)^{0.62376} \tag{4.286}$$

The area of the boundary between the liquid and the gaseous region is given by

$$A_b = A_g - 3.4211 \cdot V^{2/3}\left(\frac{V_g}{V}\right)^{1.24752} \tag{4.287}$$

Finally, the area of the spherical shell in contact with the liquid is given by

$$A_l = \pi D^2 - A_g \tag{4.288}$$

4.5 Environment Models

4.5.1 Ephemerides

Analytic Ephemeris Model

- For a new body, the user must input the central body by choosing from the 9 Planets or the sun.
- The user must provide the epoch.
- The user must provide the keplerian elements, in the central body centered, MJ2000Eq axis system.
- The user can provide a μ value for use in the solution of the equations of motion.

The body should store the users original input for the state and epoch, and the state and epoch calculated at the last request for state information. Then, when the next request is made for state information, the epoch and state from the last request are used as the input state for next calculation.

Generation of Ephemeris from Minor Planet Center Data

GMAT allows the user to import initial state data for minor planets from the Minor Planet Center's (MPC) orbit database. The MPC datafile is a flat file available at here: http://www.cfa.harvard.edu/iau/MPCORB.html.

GMAT reads state information from the MPCORB file, and converts the data to the format used internally by GMAT's two-body ephemeris propagator. The two-body propagator is based on universal variables appraoch using f and g functions and requires an initial cartesian state with respect to the planet's cental body (in this case the sun), the central body μ value.

The file format for the MPCORB file is described on the MPC web site (http://www.cfa.harvard.edu/iau/info/MPOrbitFor and is shown below:

```
 Columns   F77    Use

    1 -   7  a7     Number or provisional designation
                      (in packed form)
    9 -  13  f5.2   Absolute magnitude, H
   15 -  19  f5.2   Slope parameter, G

   21 -  25  a5     Epoch (in packed form, .0 TT)
   27 -  35  f9.5   Mean anomaly at the epoch, in degrees

   38 -  46  f9.5   Argument of perihelion, J2000.0 (degrees)
   49 -  57  f9.5   Longitude of the ascending node, J2000.0
                      (degrees)
   60 -  68  f9.5   Inclination to the ecliptic, J2000.0 (degrees)

   71 -  79  f9.7   Orbital eccentricity
   81 -  91  f11.8  Mean daily motion (degrees per day)
   93 - 103  f11.7  Semimajor axis (AU)

  106        i1     Uncertainty parameter, U
                    If this column contains 'E' it indicates
                    that the orbital eccentricity was assumed.
                    For one-opposition orbits this column can
                    also contain 'D' if a double (or multiple)
                    designation is involved or 'F' if an e-assumed
                    double (or multiple) designation is involved.

  108 - 116  a10    Reference
```

```
 118 - 122  i5     Number of observations
 124 - 126  i3     Number of oppositions

    For multiple-opposition orbits:
    128 - 131  i4     Year of first observation
    132        a1     '-'
    133 - 136  i4     Year of last observation

    For single-opposition orbits:
    128 - 131  i4     Arc length (days)
    133 - 136  a4     'days'

 138 - 141  f4.2   r.m.s residual (")
 143 - 145  a3     Coarse indicator of perturbers
                   (blank if unperturbed one-opposition object)
 147 - 149  a3     Precise indicator of perturbers
                   (blank if unperturbed one-opposition object)
 151 - 160  a10    Computer name

There may sometimes be additional information beyond column 160
as follows:

 162 - 165  z4.4   4-hexdigit flags
                   The bottom 6 bits are used to encode a
                   value representing the orbit type (other
                   values are undefined):

                     Value
                       2  Aten
                       3  Apollo
                       4  Amor
                       5  Object with q < 1.381 AU
                       6  Object with q < 1.523 AU
                       7  Object with q < 1.665 AU
                       8  Hilda
                       9  Jupiter Trojan
                      10  Centaur
                      14  Plutino
                      15  Other resonant TNO
                      16  Cubewano
                      17  Scattered disk

                   Additional information is conveyed by
                   adding in the following bit values:

                      64  Unused
                     128  Unused
                     256  Unused
                     512  Unused
                    1024  Unused
                    2048  Unused
                    4096  Unused
                    8192  1-opposition object seen at
                          earlier opposition
```

```
                    16384  Critical list numbered object
                    32768  Object is PHA

167 - 194  a        Readable designation

195 - 202  i8       Date of last observation included in
                      orbit solution (YYYYMMDD format)
```

1. Read in the string in columns 21-25 and decompose into the epoch using the instructions here: http://www.cfa.harvard.edu/iau/info/PackedDates.html 2. Read in the mean anomaly from columns 27-35 Convert to radians. 3. Read in the argument of periapsis from lines 38-46. Convert to radians. 4. Read in the longitude of the ascending node from lines 49-57. Convert to radians. 5. Read in the inclination from columns 60-68. Convert to radians. 6. Read in the eccentricity from lines 71-79. 7. Read in the mean motion from lines 81-91. 8. Use the algorithm in section 4.3.1 of the math spec to convert from mean anomaly to true anomaly. 9. Get mu value for central body and convert mean motion to semi-major axis. (I'll provide math for this soon) 10. Use the algorithm in section 4.1.3 to covert from Keplerian elements to cartesian elements.

4.5.2 Atmospheric Density

$$28K_p + 0.03e^{K_p} = A_p + 100\left(1 - e^{(-0.08A_p)}\right) \tag{4.289}$$

Jacchia Roberts

MSISE-90

A. E. Hedin, Extension of the MSIS Thermospheric Model into the Middle and Lower Atmosphere, J. Geophys. Res. 96, 1159, 1991.

Discuss observed vs. adjusted for F10.7 values, also URSI Series D

For testing http://nssdc.gsfc.nasa.gov/space/model/models/msis.html

http://www.agu.org/journals/ja/ja0212/2002JA009430/

go to auxillary material on the left side menu and open the tables-datasets.doc

Other useful models http://nssdc.gsfc.nasa.gov/space/model/

Exponential Atmosphere

4.6 Flux and Geomagnetic Index

The density models discussed above require as inputs solar flux values and geomagnetic data. Those data are provided by several organizations and include historic observations, near term predictions (45 days), and long term predictions (1 value per month for 10+ years). Vallado and Kelso[?] give a good overview of the available data and how it can be conveniently combined from multiple sources into a single coherent Space Weather File (SWF). The SWF file that GMAT reads is identical to Vallado's format for historic data and for the daily predictions. However, for the long-term monthly predictions, the GMAT SWF contains predictions created by Schatten (NEED REFERENCE).

4.6.1 File Overview

The density models in GMAT require as inputs solar flux values and geomagnetic data. Those data are provided by several organizations and include historic observations, near term predictions (45 days), and long term predictions (1 value per month for 30 years). Vallado and Kelso[?] give a good overview of the available data and how it can be conveniently combined from multiple sources into a single coherent Space Weather File (SWF). The SWF file that GMAT reads is identical to Vallado's suggested format for historic

data and for the near-term daily predictions. However, for the long-term monthly predictions, the GMAT SWF contains predictions created by Schatten (NEED REFERENCE).

The SWF is broken down into three sections according to the type of data; historical, daily predicts, and monthly predicts. The first section contains historical data observed by NOAA and the U.S. Air Force. The historical data contains the following values for each day: year, month, day of month, the solar cycle number, 3-hour Kp and Ap values, daily and 81 day averages of both observed and adjusted F10.7 values. The data is provided from Oct. 1 1957 to the current day with the column format described below.

The daily prediction section of the SWF is constructed from data provided by the U.S. Air Force and distributed by NOAA. The predictions contain daily Ap and F10.7 values for 45 days from the present day. 3 Hour values for Ap are not provided and the file assumes the 3 hour values are the same as the daily values for both Ap and Kp.

The long-term monthly section of the file is Schattens 30 year predict containing Ap and F10.7 predictions for Early, Nominal, and Late cycles, as well as Nominal, +2 Sigma, and -2 Sigma values.

4.6.2 Historical (Observed) data

Vallado[?] identifies four files provide by NOAA that can used to assemble historic flux and geomagnetic data:

```
1) ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/KP_AP/yyyy.vm
2) ftp://fdf.gsfc.nasa.gov/generalProducts/database/
3) ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SOLAR_RADIO/FLUX/DAILYPLT.OBS
4) http://www.swpc.noaa.gov/ftpdir/indices/quar_DSD.txt
```

The data in the raw files is used to create a single coherent file containing flux and geomagnetic data from Oct. 1 1957 to the present day. Each row of the file corresponds to a unique day. For each unique row the data is organized into colums as follows.

When converting between observed and adjusted F10.7 values, the following relation is used.

$$F_{10.7}(Obs) = \frac{F_{10.7}(Adj)AU^2}{r_{E/S}^2} \tag{4.290}$$

where

$F_{10.7}(Obs)$	Observed F10.7 value at station on Earth.
$F_{10.7}(Adj)$	Adjusted value F10.7 to 1 AU
AU	Astonomical unit.
$r_{E/S}$	Distance from Earth to Sun.

4.6.3 Near Term Daily Predictions

The

Near term predictions are updated daily and provided at the following location. The file contains a 45 day prediction of Ap and F10.7 values.

```
1) http://www.swpc.noaa.gov/ftpdir/latest/45DF.txt
```

4.6.4 Long Term Monthly Predictions

```
DATATYPE Space Weather File
VERSION 1.0
UPDATED 2010 Sep 21 15:35:06 UTC
#
#  This file contains three sections: historic data, daily predictions for 45 days, and monthly predictions for 30 years.
#  The historic and daily sections are identical in format to that proposed in Ref (1) and made publicly available
#  here:  http://celestrak.com/SpaceData/.  The monthly predictions contained in the file are described in Ref (2).
#
#  REFERENCES:
#
#     (1) Vallado, D., and Kelso, T. S., "Using EOP and Solar Weather Data for Satellite Operations", AIAA/AAS Astrodynamics
#          Specialist Conference, Lake Tahoe, CA, Aug. 7-11, 2005.
#     (2) Vallado, D., and Kelso, T.S.,  Space Weather File, http://celestrak.com/SpaceData/sw20060101.txt.
#     (2) Schatten, K.  NEED REFERENCE
#
# FORMAT(I4,I3,I3,I5,I3,8I3,I4,8I4,I4,F4.1,I2,I4,F6.1,I2,5F6.1)
# ------------------------------------------------------------------------------------------------------------------------------
#                                                                                                    Adj     Adj   Adj   Obs   Obs   Obs
# yy mm dd BSRN ND Kp Kp Kp Kp Kp Kp Kp Kp Sum Ap  Ap  Ap  Ap  Ap  Ap  Ap  Ap  Avg Cp C9 ISN F10.7 Q Ctr81 Lst81 F10.7 Ctr81 Lst81
# ------------------------------------------------------------------------------------------------------------------------------
#
NUM_OBSERVED_POINTS 10
BEGIN OBSERVED
2011 05 17 2426  3 20 20 30 30 20 30 30 30 210   7   7  15  15   7  15  15  15  12        57  92.0 0  96.2 110.1  90.0  94.3 110.0
2011 05 18 2426  4 20 20 10 10 10 20 10 20 120   7   7   4   4   4   7   4   7   5        65  91.0 0  95.8 110.1  88.9  93.9 109.9
2011 05 19 2426  5 10  0 10 10 10 10  0 10  60   4   0   4   4   4   4   0   4   3        36  84.0 0  95.5 110.1  82.1  93.6 109.8
2011 05 20 2426  6 10 10  0  0  0 10 10 10  50   4   4   0   0   0   4   4   4   2        33  84.0 0  95.3 109.9  82.0  93.4 109.7
2011 05 21 2426  7  0 10 10  0 10 10 10 30  80   0   4   4   0   4   4   4  15   4        44  84.0 0  95.1 109.6  82.0  93.1 109.3
2011 05 22 2426  8 20 20  0  0 10 10 10 10  80   7   7   0   0   4   4   4   4   3        47  85.0 0  94.9 109.3  82.9  92.9 108.9
2011 05 23 2426  9 10  0  0 10 20 10 10 20  80   4   0   0   4   7   4   4   7   3        37  84.0 0  94.6 108.9  81.9  92.6 108.5
2011 05 24 2426 10 20 10 10 10 20 20 10 20 120   7   4   4   4   7   7   4   7   5        23  82.0 0  94.2 108.4  80.0  92.1 107.9
2011 05 25 2426 11 10 10 10  0 10 10 10 20  80   4   4   4   0   4   4   4   7   3        23  80.0 0  93.8 107.7  78.0  91.7 107.2
2011 05 26 2426 12 20 20 10  0 20 22 22 22 136   7   7   4   0   7   8   8   8   6            80.0 0  93.3 107.0  77.9  91.2 106.4
END OBSERVED

NUM_DAILY_PREDICTED_POINTS 44
BEGIN DAILY_PREDICTED
2011 05 27 2426 13 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12            80.0    92.9 106.1  77.9  90.8 105.5
2011 05 28 2426 14 30 30 30 30 30 30 30 30 240  15  15  15  15  15  15  15  15  15            80.0    92.6 105.1  77.9  90.4 104.4
2011 05 29 2426 15 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12            80.0    92.3 104.3  77.9  90.1 103.6
2011 05 30 2426 16 24 24 24 24 24 24 24 24 192  10  10  10  10  10  10  10  10  10            80.0    92.0 103.7  77.8  89.8 102.9
2011 05 31 2426 17 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5            80.0    92.1 103.2  77.8  89.8 102.4
```

```
2011 06 01 2426 18 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        80.0    92.1 102.7  77.8  89.8 101.8
2011 06 02 2426 19 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        80.0    92.2 102.3  77.8  89.9 101.4
2011 06 03 2426 20 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    92.1 102.1  87.5  89.8 101.1
2011 06 04 2426 21 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    92.1 102.0  87.4  89.8 101.0
2011 06 05 2426 22 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    92.2 101.8  82.6  89.8 100.7
2011 06 06 2426 23 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        85.0    92.3 101.7  82.5  89.9 100.6
2011 06 07 2426 24 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        85.0    92.4 101.7  82.5  90.0 100.5
2011 06 08 2426 25 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    92.5 101.7  87.4  90.1 100.5
2011 06 09 2426 26 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    92.6 101.7  87.3  90.1 100.5
2011 06 10 2426 27 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    92.7 101.6  87.3  90.2 100.3
2011 06 11 2427  1 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12        90.0    92.9 101.5  87.3  90.4 100.1
2011 06 12 2427  2 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12        95.0    93.0 101.3  92.1  90.5 100.0
2011 06 13 2427  3 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        90.0    93.1 101.1  87.3  90.6  99.7
2011 06 14 2427  4 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    93.3 100.8  82.4  90.7  99.4
2011 06 15 2427  5 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    93.4 100.4  82.4  90.9  99.0
2011 06 16 2427  6 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    93.6 100.1  82.4  91.1  98.6
2011 06 17 2427  7 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    93.9  99.7  82.4  91.3  98.1
2011 06 18 2427  8 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        80.0    94.1  99.2  77.5  91.5  97.6
2011 06 19 2427  9 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        80.0    94.2  98.8  77.5  91.6  97.1
2011 06 20 2427 10 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        80.0    94.5  98.4  77.5  91.9  96.7
2011 06 21 2427 11 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    94.8  98.1  82.3  92.2  96.4
2011 06 22 2427 12 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12        85.0    95.1  97.8  82.3  92.5  96.1
2011 06 23 2427 13 37 37 37 37 37 37 37 37 296  22  22  22  22  22  22  22  22  22        85.0    95.5  97.4  82.3  92.8  95.7
2011 06 24 2427 14 33 33 33 33 33 33 33 33 264  18  18  18  18  18  18  18  18  18        80.0    95.8  97.0  77.4  93.2  95.2
2011 06 25 2427 15 33 33 33 33 33 33 33 33 264  18  18  18  18  18  18  18  18  18        80.0    96.1  96.7  77.4  93.4  94.8
2011 06 26 2427 16 30 30 30 30 30 30 30 30 240  15  15  15  15  15  15  15  15  15        80.0    96.4  96.2  77.4  93.8  94.3
2011 06 27 2427 17 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        80.0    96.8  95.8  77.4  94.1  93.9
2011 06 28 2427 18 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    97.1  95.5  82.2  94.4  93.6
2011 06 29 2427 19 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    97.5  95.3  87.1  94.9  93.4
2011 06 30 2427 20 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    98.0  95.1  87.1  95.3  93.1
2011 07 01 2427 21 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0    98.4  94.9  87.1  95.7  92.9
2011 07 02 2427 22 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        85.0    98.9  94.6  82.2  96.2  92.6
2011 07 03 2427 23 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        85.0    99.3  94.2  82.2  96.6  92.1
2011 07 04 2427 24 22 22 22 22 22 22 22 22 176   8   8   8   8   8   8   8   8   8        85.0    99.8  93.8  82.2  97.1  91.7
2011 07 05 2427 25 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0   100.3  93.3  87.1  97.6  91.2
2011 07 06 2427 26 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0   100.8  92.9  87.1  98.1  90.8
2011 07 07 2427 27 13 13 13 13 13 13 13 13 104   5   5   5   5   5   5   5   5   5        90.0   101.3  92.6  87.1  98.7  90.4
2011 07 08 2428  1 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12        90.0   101.8  92.3  87.1  99.2  90.1
2011 07 09 2428  2 27 27 27 27 27 27 27 27 216  12  12  12  12  12  12  12  12  12        90.0   102.4  92.0  87.1  99.7  89.8
END DAILY_PREDICTED

#          NOMINAL TIMING      EARLY TIMING        LATE TIMING
```

```
#mo. yr.  mean +2sig -2sig ap mean +2sig -2sig ap mean +2sig -2sig ap
NUM_MONTHLY_PREDICTED_POINTS 10
BEGIN MONTHLY_PREDICTED
8 2011   101  121   82   10  112  135   90   11   87  101   73    9
9 2011   103  123   83   10  113  136   90   11   89  104   74    9
10 2011   104  125   84   10  114  138   91   11   91  106   75    9
11 2011   106  127   85   10  115  139   91   11   92  108   76    9
12 2011   107  129   86   10  115  140   92   11   94  110   77    9
1 2012   108  131   87   10  116  141   93   11   96  113   79    9
2 2012   110  132   88   10  117  142   93   11   98  115   80    9
3 2012   111  134   89   10  117  143   94   11   99  117   81    9
4 2012   112  136   90   10  118  144   94   11  101  120   82   10
5 2012   113  137   91   11  119  145   94   11  102  122   83   10
END MONTHLY_PREDICTED
```

```
------------------------------------------------------------------------------------------------------------------------------------
FORMAT(I4,I3,I3,I5,I3,8I3,I4,8I4,I4,F4.1,I2,I4,F6.1,I2,5F6.1)
------------------------------------------------------------------------------------------------------------------------------------
Columns Description
001-004 Year
006-007 Month (01-12)
009-010 Day
012-015 Bartels Solar Rotation Number. A sequence of 27-day intervals counted continuously from 1832 Feb 8.
017-018 Number of Day within the Bartels 27-day cycle (01-27).
020-021 Planetary 3-hour Range Index (Kp) for 0000-0300 UT.
023-024 Planetary 3-hour Range Index (Kp) for 0300-0600 UT.
026-027 Planetary 3-hour Range Index (Kp) for 0600-0900 UT.
029-030 Planetary 3-hour Range Index (Kp) for 0900-1200 UT.
032-033 Planetary 3-hour Range Index (Kp) for 1200-1500 UT.
035-036 Planetary 3-hour Range Index (Kp) for 1500-1800 UT.
038-039 Planetary 3-hour Range Index (Kp) for 1800-2100 UT.
041-042 Planetary 3-hour Range Index (Kp) for 2100-0000 UT.
044-046 Sum of the 8 Kp indices for the day expressed to the nearest third of a unit.
048-050 Planetary Equivalent Amplitude (Ap) for 0000-0300 UT.
052-054 Planetary Equivalent Amplitude (Ap) for 0300-0600 UT.
056-058 Planetary Equivalent Amplitude (Ap) for 0600-0900 UT.
060-062 Planetary Equivalent Amplitude (Ap) for 0900-1200 UT.
064-066 Planetary Equivalent Amplitude (Ap) for 1200-1500 UT.
068-070 Planetary Equivalent Amplitude (Ap) for 1500-1800 UT.
072-074 Planetary Equivalent Amplitude (Ap) for 1800-2100 UT.
076-078 Planetary Equivalent Amplitude (Ap) for 2100-0000 UT.
080-082 Arithmetic average of the 8 Ap indices for the day.
084-086 Cp or Planetary Daily Character Figure. A qualitative estimate of overall level of magnetic activity for the day
determined from the sum of the 8 Ap indices. Cp ranges, in steps of one-tenth, from 0 (quiet) to 2.5 (highly disturbed).
088-088 C9. A conversion of the 0-to-2.5 range of the Cp index to one digit between 0 and 9.
090-092 International Sunspot Number. Records contain the Zurich number through 1980 Dec 31 and the International Brussels
number thereafter.  094-098 10.7-cm Solar Radio Flux (F10.7) Adjusted to 1 AU. Measured at Ottawa at 1700 UT daily from 1947
Feb 14 until 1991 May 31 and measured at Penticton at 2000 UT from 1991 Jun 01 on. Expressed in units of 10-22 W/m2/Hz.
100-100 Flux Qualifier.
0 indicates flux required no adjustment;
1 indicates flux required adjustment for burst in progress at time of measurement;
2 indicates a flux approximated by either interpolation or extrapolation;
3 indicates no observation; and
4 indicates CSSI interpolation of missing data.
102-106 Centered 81-day arithmetic average of F10.7 (adjusted).
108-112 Last 81-day arithmetic average of F10.7 (adjusted).
114-118 Observed (unadjusted) value of F10.7.
```

```
120-124 Centered 81-day arithmetic average of F10.7 (observed).
126-130 Last 81-day arithmetic average of F10.7 (observed).
```

4.6.5 Tank Models

Accurately modelling tank pressure changes is essential for accurate maneuver modelling and reconstruction. The following sections discuss three types of tank models: pressurant tank, regulated fuel tank, and blowdown fuel tank. For each tank there are three models: isothermal, heat transfer, and adiabatic.

The models used in GMAT are based on work by Estey[16] , Hearn[17] and Moran.[18] For each tank, we select a set of state variables that when defined, allow us to determine all remaining properties of the tank. For the state variables, we provide differential equations that describe how the state variables change with respect to time. The number of state variables varies between different tanks, and with the model type such as isothermal and heat transfer.

For each of the three tanks, we develop a heat transfer model, an adiabatic model, and an isothermal model. The heat transfer model is derived using the laws of conservation of energy and the conservation of mass. An adiabatic model is provided by setting the heat transfer rates to zero in the heat transfer model. The isothermal model for each tank is developed separately. Each ot these models is useful for certain classes of maneuvers. Isothermal models are useful for maneuvers with low mass flow rates, adiabatic models are useful for short maneuvers with high mass flow rates. Heat transfer models are useful for long maneuvers with large mass flow rates.

When developing heat transfer models, we'll assume that specific internal energy is given by

$$u = cT \tag{4.291}$$

specific enthalpy for a liquid is given by

$$h_\ell = c_\ell T_\ell \tag{4.292}$$

and specific enthalpy for a gas is given by

$$h_g = T_g(c_g + R_g) \tag{4.293}$$

The notation used in tank model development is shown below. After presenting notation, we present the dynamics model for a pressurant tank.

Nomenclature

A_g, A_ℓ, A_w	= Heat transfer area
c_v, c_g	= Specific heat at constant volume
D	= Tank diameter
d	= Liquid surface diameter
Gr	= Grashof number
h_ℓ, h_v	= Enthalpy
m_g, m_ℓ, m_w, m_v	= Mass
P_g, P_v, P_t	= Pressure
R_v, R_g	= Gas constant
$T_g, T_\ell, T_w, T_v, T_a$	= Temperature
u_g, u_ℓ, u_w, u_v	= Specific internal energy
V_g, V_ℓ, V_t	= Volume
$\dot{W}$	= Work rate
$\dot{Q}_g, \dot{Q}_v, \dot{Q}_l, \dot{Q}_w$	= Heat transfer rate
ν_l, ν_g, ν_v	= Specific volume

Subscripts

a	= Ambient
g	= Pressurant gas
ℓ	= Propellant liquid
t	= Total
v	= Propellant vapor
w	= Tank wall
e	= Exit-flow
i	= In-flow

Pressurant Tank

The pressurant tank model is the simplest of the tank models primarily due to the fact that there is only one substance, the pressurant gas, contained in the tank. In this section, we develop a state space model for pressurant tank dynamics. We choose the state variables to be pressurant gas mass and temperature, m_g and T_g respectively, and tank wall temperature T_w.

In Fig.4.11 we see an illustration of a pressurant tank. We divide the tank into two control volumes: the gas region and the tank wall region. The only mass flow in the system occurs where pressurant gas exits the tank. Heat transfer occurs between the gas and the wall, and the wall and the ambient surroundings.

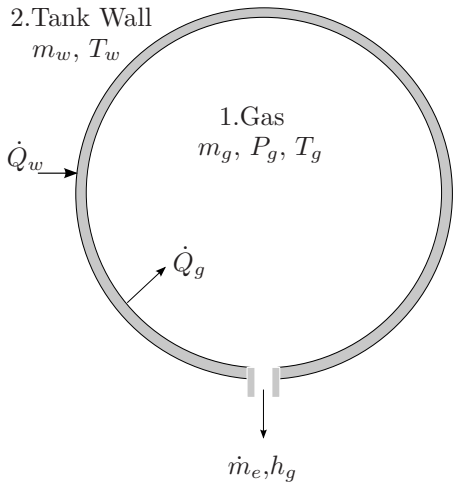


Figure 4.11: Pressurant Tank Diagram

Knowing the volume of the tank and the state variables m_g, T_g, and T_w, we calculate pressure from one of the following two equations of state:

$$P_g = \frac{m_g R_g T_g}{V_g} \tag{4.294}$$

or from the Beattie-Bridgeman Eq.

$$P_g = \frac{R_g T_g}{V_g} + \frac{a_g}{V_g^2} + \frac{b_g}{V_g^3} \tag{4.295}$$

The state variables m_g, T_g, and T_w are described by ordinary differential equations found by applying the first law of thermodynamics and the conservation of mass. The 1st Law applied to the gas control volume yields

$$\frac{d}{dt}\left(m_g u_g\right) = \dot{Q}_g - \dot{m}_e h_g \tag{4.296}$$

The 1st Law applied to the wall control volume yields

$$\frac{d}{dt}\left(m_w u_w\right) = \dot{Q}_w - \dot{Q}_g \tag{4.297}$$

and finally from conservation of mass we obtain

$$\dot{m}_g = -\dot{m}_e \tag{4.298}$$

For these equations to be useful for numerical integration, we need to expand the derivatives, and if necessary, decouple the equations (as we'll see, for the pressurant tank, the equations are not coupled).

Expanding the terms in Eq. (4.296) we have

$$\dot{m}_g c_g T_g + m_g c_g \dot{T}_g = \dot{Q}_g - \dot{m}_e T_g \left(c_g + R_g\right) \tag{4.299}$$

Similarly, expanding Eq. (4.297) we obtain

$$m_w c_w \dot{T}_w = \dot{Q}_w - \dot{Q}_g \tag{4.300}$$

Solving the system of equations yields the following differential equations of state for the pressurant tank heat transfer model.

$$\dot{m}_g = -\dot{m}_e \tag{4.301}$$

$$\dot{T}_g = \frac{1}{m_g c_g}\left(\dot{Q}_g - T_g R_g \dot{m}_e\right) \tag{4.302}$$

$$\dot{T}_w = \frac{1}{m_w c_w}\left(\dot{Q}_w - \dot{Q}_g\right) \tag{4.303}$$

The adiabatic model is obtained by setting the terms $\dot{Q}_g$ and $\dot{Q}_w$ to zero in the above equations. (Note for the adiabatic model there are only two state variables, m_g and T_g, as the wall temperature T_w is removed from the system of equations.) Similarly, the isothermal model is obtained by setting $\dot{T}_g$ and $\dot{T}_w$ to zero. So, for the isothermal model there is only one state variable m_g.

In summary, for the pressurant tank, all models calculate the tank pressure using

$$P_g = \frac{m_g R_g T_g}{V_g}$$

then the specific equations for the heat transfer, adiabatic, and isothermal models, are as follows
Pressurant Tank: Heat Transfer
State Variables: m_g, T_g, T_w

$$\begin{aligned}\dot{m}_g &= -\dot{m}_e \\ \dot{T}_g &= \frac{1}{m_g c_g}\left(\dot{Q}_g - T_g R_g \dot{m}_e\right) \\ \dot{T}_w &= \frac{1}{m_w c_w}\left(\dot{Q}_w - \dot{Q}_w\right)\end{aligned}$$

Pressurant Tank: Adiabtic
State Variables: m_g, T_g

$$\begin{aligned}\dot{m}_g &= -\dot{m}_e \\ \dot{T}_g &= \dot{m}_e \frac{T_g R_g}{m_g c_g}\end{aligned}$$

Pressurant Tank: Isothermal
State Variables: m_g

$$\dot{m}_g = -\dot{m}_e \tag{4.304}$$

Now let's look at a model for a fuel tank operating in blow down mode.

Blowdown Tank

The blowdown tank model is significantly more complex than the pressurant tank model due to the presence of liquid fuel and fuel vapor contained in the tank ullage. In this section, we develop a state space model for a blow down tank. We choose the state variables to be the liquid mass and temperature, m_ℓ and T_ℓ, the gas temperature T_g, and tank wall temperature T_w.

In Fig.4.12 we see an illustration of a blow down tank. We divide the tank into three control volumes: the gas region, the liquid region, and the tank wall region. Mass flow occurs where the pressurant gas exits the tank and at the boundary between the liquid and gas in the form of evaporation. Heat transfer occurs between all three control volumes as well as with the surroundings. In summary, the physical processes modelled for a blow down tank are

1. Vapor pressure is a function of liquid temperature.
2. Liquid density is a function of liquid temperature.
3. Heat transfer between the liquid and gas.
4. Heat transfer between the tank wall and gas.
5. Heat transfer between the tank wall and liquid.
6. Heat transfer between the surroundings and tank. wall.

The assumptions made in the tank model are

1. Pressurant does not dissolve in liquid ($m_g = C$).
2. Vapor and gas temperatures are equal.
3. Vapor and gas volumes are equal.

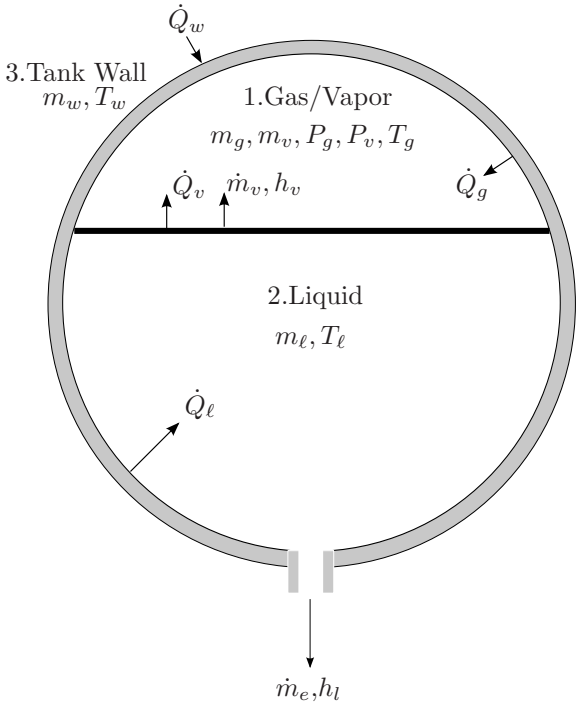


Figure 4.12: Blow Down Tank Diagram

Assume we are given m_g, the tank diameter D, and hence know the total tank volume V_t, and we know the physical constants associated with the liquid and gas (R_g, c_g, ν_g, c_ℓ, $\nu_\ell(T_\ell)$ and $P_v(T_\ell)$). We choose the state variables m_ℓ, T_ℓ, T_g, and T_w, all other tank properties can be calculated from these state variables using the following equations:

$$V_\ell = \nu_\ell(T_\ell) m_\ell \tag{4.305}$$
$$V_g = V_t - V_\ell \tag{4.306}$$
$$P_g = \frac{m_g R_g T_g}{V_g} \tag{4.307}$$
$$P_v = P_v(T_\ell) \tag{4.308}$$
$$m_v = \frac{P_v V_g}{R_v T_g} \tag{4.309}$$
$$P_t = P_v + P_g \tag{4.310}$$

To determine the state equations governing m_ℓ, T_ℓ, T_g, and T_w we apply the 1st law of thermodynamics and the law of conservation of mass. The 1st Law applied to the gas control volume is

$$\frac{d}{dt}\left(m_v u_v + m_g u_g\right) = \dot{Q}_v + \dot{Q}_g - P_t \dot{V}_g + \dot{m}_v h_v \tag{4.311}$$

The 1st Law applied to the liquid control volume is

$$\frac{d}{dt}\left(m_\ell u_\ell\right) = \dot{Q}_\ell - \dot{Q}_v + P_t\dot{V}_g - \dot{m}_v h_{lg} - \dot{m}_e h_\ell \tag{4.312}$$

The 1st Law applied to the wall control volume yields

$$\frac{d}{dt}\left(m_w u_w\right) = \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g \tag{4.313}$$

and finally from conservation of mass:

$$\dot{m}_\ell = -\dot{m}_e - \dot{m}_v \tag{4.314}$$

we also know that

$$\dot{m}_v = \frac{P_v\dot{V}_g}{R_v T_g} - \frac{P_v V_g \dot{T}_g}{R_v T_g^2} \tag{4.315}$$

where we assume that

$$\dot{P}_v \approx 0 \tag{4.316}$$

Equations (4.311) - (4.315) are five equations in five unknowns (m_v, m_ℓ, T_ℓ, T_g, and T_w). Our approach is to use Eq. (4.314) to eliminate $\dot{m}_v$ terms. The result is a system of four equations in four unknowns using Eqs. (4.311), (4.312), (4.313), and (4.315). The result we seek is four decoupled ordinary differential equations for m_ℓ, T_ℓ, T_g, and T_w.

Let's continue with Eq. (4.311). We need to rewrite the equation in terms of $\dot{m}_\ell$ and $\dot{T}_g$ ($\dot{T}_w$ and $\dot{T}_\ell$ don't appear explicitly). Expanding the derivatives assuming $\dot{m}_g = 0$ yields

$$\dot{m}_v c_v T_g + m_v c_v \dot{T}_g + m_g c_g \dot{T}_g = \dot{Q}_v + \dot{Q}_g - P_t\dot{V}_g + \dot{m}_v h_v \tag{4.317}$$

Now, substituting $\dot{m}_v = -\dot{m}_\ell - \dot{m}_e$ and noting that $\dot{V}_g = -\nu_l \dot{m}_\ell$ if we assume

$$\dot{\nu}_\ell = \frac{d\nu_\ell}{dT_\ell}\dot{T}_\ell \approx 0$$

we arrive at

$$\begin{aligned}\left(T_g R_v - P_t\nu_\ell\right)\dot{m}_\ell + \left(m_v c_v + m_g c_g\right)\dot{T}_g = \\ \dot{Q}_v + \dot{Q}_g - \dot{m}_e T_g R_v\end{aligned} \tag{4.318}$$

Now continuing with Eq. (4.312) expanding the derivatives and making similar substitutions as we made previously we obtain

$$\begin{aligned}\dot{m}_\ell c_\ell T_\ell + m_\ell c_\ell \dot{T}_\ell = \dot{Q}_\ell - \dot{Q}_v + P_t(-\nu_\ell \dot{m}_\ell) - \\ (-\dot{m}_\ell - \dot{m}_e)h_v - \dot{m}_e c_\ell T_\ell\end{aligned} \tag{4.319}$$

Grouping terms we obtain

$$\begin{aligned}(c_\ell T_\ell + P_t v_l - h_v)\dot{m}_\ell + (m_\ell c_\ell)\dot{T}_\ell = \\ \dot{Q}_\ell - \dot{Q}_v + \dot{m}_e(h_v - c_\ell T_\ell)\end{aligned} \tag{4.320}$$

For the wall region, described by Eq. (4.313), we arrive at

$$(m_w c_w)\dot{T}_w = \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g \tag{4.321}$$

Finally, by eliminating $\dot{m}_v$ in the Gas Law shown in Eq. (4.315) we obtain

$$-\dot{m}_\ell - \dot{m}_e = \frac{P_v(-\nu_\ell\dot{m}_\ell)}{R_v T_g} - \frac{P_v V_g \dot{T}_g}{R_v T_g^2} \tag{4.322}$$

Grouping terms yields the result

$$\left(1-\frac{P_v\nu_\ell}{R_vT_g}\right)\dot{m}_\ell-\frac{P_vV_g}{R_vT_g^2}\dot{T}_g=-\dot{m}_e \tag{4.323}$$

Equations (4.318), (4.320), (4.321), and (4.323) are four coupled ordinary differential equations that can be decoupled by casting them in matrix form as follows:

$$\begin{pmatrix} A_{11} & 0 & A_{13} & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & 0 & A_{34} \\ A_{41} & & A_{43} & 0 \end{pmatrix}\begin{pmatrix} \dot{m}_\ell \\ \dot{T}_\ell \\ \dot{T}_g \\ \dot{T}_w \end{pmatrix}=\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \tag{4.324}$$

where

$$\begin{aligned} A_{11} &= T_gR_v-P_t\nu_\ell && (4.325)\\ A_{13} &= m_vc_v+m_gc_g && (4.326)\\ A_{21} &= c_\ell T_\ell+P_tv_l-h_v && (4.327)\\ A_{22} &= m_\ell c_\ell && (4.328)\\ A_{34} &= m_wc_w && (4.329)\\ A_{41} &= 1-\nu_\ell/\nu_v && (4.330)\\ A_{43} &= -m_v/T_g && (4.331)\\ b_1 &= \dot{Q}_v+\dot{Q}_g-\dot{m}_eT_gR_v && (4.332)\\ b_2 &= \dot{Q}_\ell-\dot{Q}_v+\dot{m}_e(h_v-c_\ell T_\ell) && (4.333)\\ b_3 &= \dot{Q}_w-\dot{Q}_\ell-\dot{Q}_g && (4.334)\\ b_4 &= -\dot{m}_e && (4.335) \end{aligned}$$

The solution to the equations is

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43}b_1-A_{13}b_4}{A_{11}A_{43}-A_{41}A_{13}} && (4.336)\\ \dot{T}_\ell &= \frac{1}{A_{22}}\left(b_2-A_{21}\frac{A_{43}b_1-A_{13}b_4}{A_{11}A_{43}-A_{41}A_{13}}\right) && (4.337)\\ \dot{T}_g &= \frac{A_{11}b_4-A_{41}b_1}{A_{11}A_{43}-A_{41}A_{13}} && (4.338)\\ \dot{T}_w &= \frac{b_3}{A_{34}} && (4.339) \end{aligned}$$

For the adiabatic model we set all heat transfer rates, $\dot{Q}$, to zero in Eqs. (4.332)-(4.335) and so there are only two state variables as $\dot{T}_w=0$ and so $T_w=$ constant.

Now let's develop equations for an isothermal model of a blow down tank. In the isothermal model, we assume $T_\ell=T_g=T_w=T$. The only state variable that requires a differential equation is m_ℓ. Because T_g, T_ℓ, and hence, P_v are constant, we know that

$$\dot{m}_v=\frac{P_v\dot{V}_g}{R_vT_g} \tag{4.340}$$

Subtituting this result into Eq.(4.314) and solving for $\dot{m}_\ell$ we obtain.

$$\dot{m}_\ell=-\frac{\dot{m}_e}{\left(1-\frac{P_v\nu_\ell}{R_vT}\right)} \tag{4.341}$$

In summary for the heat transfer model for a blow down tank, we choose m_ℓ, T_ℓ, T_g, and T_w are state variables. Eqs. (4.305)-(4.310) are used to calculate the remaining tank properties, and Eqs. (4.336)-(4.339) are used to model the tank states as functions of time.

For the all three models, heat transfer, adiabatic, and isothermal, knowing the state variables m_ℓ, T_ℓ, T_g, and T_w we compute the remaining tank properties using

$$\begin{aligned} V_\ell &= \nu_\ell(T_\ell) m_\ell \\ V_g &= V_t - V_\ell \\ P_g &= \frac{m_g R_g T_g}{V_g} \\ P_v &= P_v(T_\ell) \\ m_v &= \frac{P_v V_g}{R_v T_g} \\ P_t &= P_v + P_g \end{aligned}$$

The models differ in the number of state variables and in the state rate equations. A summary is presented below.

Blow Down Tank: Heat Transfer
State Variables: m_ℓ, T_ℓ, T_g, T_w

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_\ell &= \frac{1}{A_{22}}\left(b_2 - A_{21}\frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}}\right) \\ \dot{T}_g &= \frac{A_{11}b_4 - A_{41}b_1}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_w &= \frac{b_3}{A_{34}} \end{aligned}$$

where A_{ij} and b_i are given by Eqs. (4.325)-(4.335).

Blow Down Tank: Adiabtic
State Variables: m_ℓ, T_ℓ, T_g

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}} \\ \dot{T}_\ell &= \frac{1}{A_{22}}\left(b_2 - A_{21}\frac{A_{43}b_1 - A_{13}b_4}{A_{11}A_{43} - A_{41}A_{13}}\right) \\ \dot{T}_g &= \frac{A_{11}b_4 - A_{41}b_1}{A_{11}A_{43} - A_{41}A_{13}} \end{aligned}$$

where A_{ij} and b_i are given by Eqs. (4.325)-(4.335). Note that all heat flow rates, $\dot{Q}$, are set to zero.

Blow Down Tank: Isothermal
State Variables: m_ℓ

$$\dot{m}_\ell = -\frac{\dot{m}_e}{\left(1 - \frac{P_v \nu_\ell}{R_v T}\right)}$$

Pressure Regulated Tank

The pressure regulated fuel tank model is the most complex tank model supported in GMAT. The model complexity is due to the presence of liquid fuel and fuel vapor contained in the tank ullage, and due to mass and energy transfer from the pressurant tank to the ullage of the regulated fuel tank. In this section, we develop a state space model for a pressure regulated tank. Note, to model a pressure regulated tank using a

heat transfer or adiabitic model, we must simultaneously solve the equations associated with the pressurant tank. For the regulated tank model, we choose the state variables to be the liquid mass and temperature, m_ℓ and T_ℓ, the gas temperature T_g, tank wall temperature T_w, and the incoming pressurant gas mass m_i.

In Fig.4.13 we see an illustration of a pressure regulated tank. Like the blow down tank model, we divide the tank into three control volumes: the gas region, the liquid region, and the tank wall region. Mass flow occurs where the pressurant gas exits the tank, at the boundary between the liquid and gas in the form of evaporation, and from the pressurant tank to the ullage of the regulated tank. Heat transfer occurs between all three control volumes as well as with the surroundings. Hence, the physical processes modelled for a blow down tank are the same as those listed for the blow down tank, with the added process of mass flow from the pressurant tank.

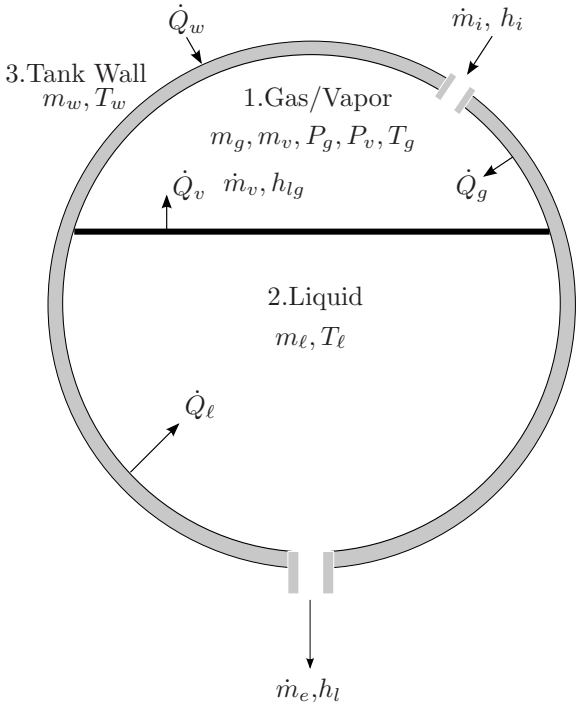


Figure 4.13: Pressure Regulated Tank Diagram

The derivation of the state equations for a pressure regulated tank follows naturally from the derivation of the blow down tank. The only control volume that differs between the two models is the gas/vapor control volume. Applying the 1st Law of thermodynamics to the gas/vapor control volume of the pressure regulated tank gives us

$$\frac{d}{dt}\left(m_v u_v + m_g u_g\right) = \dot{Q}_v + \dot{Q}_g - P_t \dot{V}_g + \dot{m}_v h_v + \dot{m}_p h_p \tag{4.342}$$

Taking the time derivative of the gas law for the gas contained in the tank ullage yields

$$\dot{m}_g = \frac{P_g \dot{V}_g}{R_g T} - \frac{P_g V_g \dot{T}_g}{R_g T_g^2} \tag{4.343}$$

Equations (4.342) and (4.343), together with equations (4.320), (4.321), and (4.323) are a system of 5 equations in 5 unknowns which can be decoupled using simple linear algebra. However, first we must expand Eqs. (4.342) and (4.343) and write them in terms of the state rate derivatives. Expanding Eq. (4.342) we arrive at

$$\begin{aligned}\left(R_v T_g - P_t \nu_\ell\right)\dot{m}_\ell + \left(m_v c_v + m_g c_g\right)\dot{T}_g + \\ \left(c_g T g - h_p\right)\dot{m}_g = \dot{Q}_v + \dot{Q}_g - \dot{m}_e R_v T_g\end{aligned} \tag{4.344}$$

Similarly, for Eq. (4.343) we obtain

$$\frac{\nu_\ell}{\nu_g}\dot{m}_\ell + \frac{m_g}{T_g}\dot{T}_g + \dot{m}_g = 0 \tag{4.345}$$

To integrate the state equations we must decouple the equations and this is easily done by casting the equations in matrix form and solving the system of equations. We can write the equations is state space for as follows

$$\begin{pmatrix} A_{11} & 0 & A_{13} & 0 & A_{15} \\ A_{21} & A_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{34} & 0 \\ A_{41} & 0 & A_{43} & 0 & 0 \\ A_{51} & 0 & A_{43} & 0 & A_{55} \end{pmatrix} \begin{pmatrix} \dot{m}_\ell \\ \dot{T}_\ell \\ \dot{T}_g \\ \dot{T}_w \\ \dot{m}_g \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ 0 \end{pmatrix} \tag{4.346}$$

where the coefficients A_{ij} and b_i are given by

$$\begin{aligned} A_{11} &= T_g R_v - P_t \nu_\ell && (4.347)\\ A_{13} &= m_v c_v + m_g c_g && (4.348)\\ A_{15} &= c_g T_g - h_p && (4.349)\\ A_{21} &= c_\ell T_\ell + P_t v_l - h_{lg} && (4.350)\\ A_{22} &= m_\ell c_\ell && (4.351)\\ A_{34} &= m_w c_w && (4.352)\\ A_{41} &= 1 - \nu_\ell/\nu_v && (4.353)\\ A_{43} &= -m_v/T_g && (4.354)\\ A_{51} &= \nu_\ell/\nu_g && (4.355)\\ A_{53} &= m_g/T_g && (4.356)\\ A_{55} &= 1 && (4.357)\\ b_1 &= \dot{Q}_v + \dot{Q}_g - \dot{m}_e R_v T_g && (4.358)\\ b_2 &= \dot{Q}_\ell - \dot{Q}_v + \dot{m}_e(h_{lg} - c_\ell T_\ell) && (4.359)\\ b_3 &= \dot{Q}_w - \dot{Q}_\ell - \dot{Q}_g && (4.360)\\ b_4 &= -\dot{m}_e && (4.361) \end{aligned}$$

$$\begin{aligned} \dot{m}_\ell &= \frac{A_{55}A_{43}b_1 - b_4 A_{13}A_{55} + b_4 A_{15}A_{53}}{D} && (4.362)\\ \dot{T}_\ell &= \frac{b_2 - A_{21}\dot{m}_\ell}{A_{22}} && (4.363)\\ \dot{T}_g &= \frac{-A_{41}A_{55}b_1 + b_4 A_{11}A_{55} - b_4 A_{51}A_{15}}{D} && (4.364)\\ \dot{T}_w &= \frac{b_3}{A_{34}} && (4.365)\\ \dot{m}_g &= \frac{-b_1 A_{51}A_{43} + b_1 A_{41}A_{53} - b_4 A_{11}A_{53} + b_4 A_{51}A_{13}}{D} && (4.366) \end{aligned}$$

where

$$D = A_{55}A_{43}A_{11} - A_{43}A_{51}A_{15} + A_{41}A_{15}A_{53} - A_{41}A_{13}A_{55} \tag{4.367}$$

For the adiabatic model we set all heat transfer rates, $\dot{Q}$, to zero in Eqs. (4.358)-(4.361). For the adiabatic model there is only four state variables as $\dot{T}_w = 0$ and so $T_w =$ constant.

Now let's develop equations for an isothermal model of a pressure regulated tank. In the isothermal model, we assume $T_\ell = T_g = T_w = T$. The only state variables that require differential equations are m_ℓ and m_g. Because $T_g =$ constant, and hence, $P_v =$ constant, we know that

$$\dot{m}_\ell = -\frac{\dot{m}_e}{\left(1 - \dfrac{P_v \nu_\ell}{R_v T}\right)} \tag{4.368}$$

Similarly, for m_g we obtain

$$\dot{m}_g = \frac{\nu_\ell}{\nu_g} \frac{\dot{m}_e}{\left(1 - \frac{P_v \nu_\ell}{R_v T}\right)} \tag{4.369}$$

Heat Transfer

Heat transfer models are from Ring[19] and Incropera[2] and Pitts[20]

$$\dot{Q} = hA\Delta T \tag{4.370}$$

$$\frac{hL}{k} = Nu = c(Gr_L Pr)^a \tag{4.371}$$

so

$$h = \frac{kc}{L}(Gr_L Pr)^a \tag{4.372}$$

Table 4.10: Dimensionless Heat Transfer Terms[2]

Parameter	Definition	Interpretation
Grashof number (Gr_L)	$\frac{g\beta(T_s - T_\infty)L^3}{\nu^2}$	Measure of the ratio of buoyancy forces to viscous forces.
Prandtl number (Pr)	$\frac{c\mu}{k} = \frac{\mu}{\alpha}$	Ratio of the momentum and thermal diffusivities.
Nusselt number (Nu_L)	$\frac{hL}{k_f}$	Ratio of convection to pure conduction heat transfer.
Reynolds number (Re_L)	$\frac{VL}{\nu}$	Ratio of inertia and viscous forces.

Physiochemical Properties

Hydrazine Properties[21]

$$c = 3084 \text{J/kg/K}$$

$$\rho\ (\text{kg/m}^3) = 1025.6 - 0.87415\ T\ (^o\text{C}) - 0.45283e^{-3}\ T^2\ (^o\text{C})$$

$$\rho\ (\text{kg/m}^3) = 1230.6 - 0.62677\ T\ (^o\text{K}) - 0.45283e^{-3}\ T^2\ (^o\text{K})$$

Some models are from[22]

$$\rho_\ell(T_\ell) = K_1 + K_2 T_\ell + K_3 T_\ell^2\ (\text{kg/m}^3) \tag{4.373}$$

$$\frac{d\rho_\ell}{dT_\ell} = K_2 + 2K_3 T_\ell\ (\text{kg/m}^3) \tag{4.374}$$

$$P_v = C_1 e^{(C_2 + C_3 T_\ell + C_4 T_\ell^2)}\ (\text{N/m}^2) \tag{4.375}$$

$$\frac{dP_v}{dT_\ell} = C_1 ln(C_2 + C_3 T_\ell C_4 T_\ell^2)\,(C_3 + 2C_4 T_\ell) \tag{4.376}$$

Table 4.11: Constants for Density Equations

Const.	N_20_4	$CH_3N_2H_3$
K_1 (kg/m^3)	2066	1105.3
K_2 (kg/m^3/K)	-1.979	-0.9395
K_3 (kg/m^3/K^2)	-4.826e-4	0

Table 4.12: Constants for Vapor Pressure Equations

Const.	N_20_4	$CH_3N_2H_3$
C_1 (kg/m^3)	6895	6895
C_2 (kg/m^3/K)	18.6742	12.4293
C_3 (kg/m^3/K^2)	-5369.57	2543.37
C_4 (kg/m^3/K^2)	194721	0

$$m_d = P_g m_\ell^\alpha \left(\frac{T_\ell}{294}\right)^2 \tag{4.377}$$

$$\begin{aligned}\frac{dm_d}{dt} = m_\ell^\alpha \left(\frac{T_\ell}{294}\right)^2 \dot{P}_g + \alpha P_g m_\ell^{(\alpha-1)} \left(\frac{T_\ell}{294}\right)^2 \dot{m}_\ell \\ +2P_g m_\ell^\alpha \left(\frac{T_\ell}{294}\right) \dot{T}_\ell\end{aligned} \tag{4.378}$$

Table 4.13: Constants for Dissolved Pressurant Equations

Const.	N_20_4	$CH_3N_2H_3$
α	3.335e-11	2.059e-11

Chapter 5

Numerical Integrators

5.1 Runge-Kutta Integrators

The Runge-Kutta integration scheme is a single step method used to solve differential equations for n coupled variables of the form

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

(The superscript in this discussion refers to the variables; hence f^i is the i^{th} variable, and $r^{(n)}$ refers to all n variables.) The method takes an integration step, h, by breaking the interval into several stages (usually of smaller size) and calculating estimates of the integration result at each stage. The later stages use the results of the earlier stages. The cumulative effect of the integration is an approximate total step δt, accurate to a given order in the series expansion of the differential equation, for the state variables $r_i(t + \delta t)$ given the state $r_i(t)$.

The time increment for a given stage is given as a multiple a_i of the total time step desired; thus for the i^{th} stage the interval used for the calculation is $a_i \delta t$; the estimate of the integrated state at this stage is given by

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

where b_{ij} contains a set of coefficients specific to the Runge-Kutta instance being calculated. Given the results of the stage calculations, the total integration step can be calculated using another set of coefficients, c_j and the formula

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

The error control for these propagators is implemented by comparing the results of two different orders of integration. The difference between the two steps provides an estimate of the accuracy of the step; a second set of coefficients corresponding to this second integration scheme can be used to obtain a solution

$$r'^{(n)}(t + \delta t) = r^{(n)}(t) + \sum_{j=1}^{stages} c_j^* k_j^{*(n)}$$

With care, the stage estimates k_j and k_j^* can be selected so that they are the same; in that case, the estimate of the error in the integration $\Delta^{(n)}$ can be written

$$\Delta^{(n)} = \left| \sum_{j=1}^{stages} (c_j - c_j^*) k_j^{(n)} \right|$$

(The difference between the coefficients $c_j - c_j^*$ is the array of error estimate coefficients (ee) in this code.)

Once the estimated error has been calculated, the size of the integration step can be adapted to a size more appropriate to the desired accuracy of the integration. If the step results in a solution that is not accurate enough, the step needs to be recalculated with a smaller step size. Labeling the desired accuracy α and the obtained accuracy ϵ (calculated, for instance, as the largest element of the array Δ), the new step used by the Runge-Kutta integrator is

$$\delta t_{new} = \sigma \delta t \left(\frac{\alpha}{\epsilon}\right)^{1/(m-1)}$$

where m is the order of truncation of the series expansion of the differential equations being solved. The factor σ is a safety factor incorporated into the calculation to avoid unnecessary iteration over attempted steps. Common practice is to set this factor to 0.9; that is the default value used in this implementation.

Similarly, if the step taken does not result in the desired accuracy, you may want to increase the step size parameter for the next integration step. The new estimate for the desired stepsize is given by

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

Sometimes you do not want to increase the stepsize in this manner; for example, you may want to keep the maximum step taken at some fixed value. This implementation provides a mechanism for specifying a maximum allowed step.

Sometimes it is convenient to request steps of a specified size, regardless of the stepsize control algorithm or the calculation of the "best step" described above. This implementation accomplishes that task by taking multiple error controlled steps is necessary to step across the requested interval.

Both of these features are implemented using the boolean flags described in the base class for the integrators. See the documentation for the **Integrator** (p. **??**) class for more information about these flags.

5.1.1 Constructor & Destructor Documentation

RungeKutta::RungeKutta (int *st*, int *order*)

Provides the greatest relative error in the state vector.

This method takes the state vector and calculates the error in each component. The error is then divided by the change in the component. The function returns the largest of the resulting relative errors.

Override this method if you want a different error estimate for the stepsize control. For example, we are using

$$error_i = \left(\frac{\Delta_i(t+\delta t)}{r_i(t+\delta t) - r_i(t)}\right)$$

Another popular approach is to divide the estimated error Δ_i by the norm of the corresponding 3-vector; for instance, divide the error in x by the magnitude of the displacement in position for the step.

5.2 Prince-Dormand Integrators

5.3 Adams Bashforth Moulton

Implementation of the Adams-Bashford-Moulton Predictor-Corrector.

This code implements a fourth-order Adams-Bashford predictor / Adams-Moulton corrector pair to integrate a set of first order differential equations. The algorithm is found at `http://chemical.caeds.eng.uml.edu/onlinec/wh`
or in Bate, Mueller and White, pp. 415-417.

The predictor step extrapolates the next state r_{i+1} of the variables using the the derivative information (f) at the current state and three previous states of the variables, by applying the equation

$$r_{i+1}^{*j} = r_i^j + \frac{h}{24}\left[55 f_n^j - 59 f_{n-1}^j + 37 f_{n-2}^j - 9 f_{n-3}^j\right]$$

The corrector uses derivative information evaluated for this state, along with the derivative information at the original state and two preceding states, to tune this state, giving the final, corrected state:

$$r_{i+1}^{j} = r_{i}^{j} + \frac{h}{24}\left[9f_{n+1}^{*j} + 19f_{n}^{j} - 5f_{n-1}^{j} + 1f_{n-2}^{j}\right]$$

Bate, Mueller and White give the estimated accuracy of this solution to be

$$ee = \frac{19}{270}\left|r_{i+1}^{*j} - r_{i+1}^{j}\right|$$

Method used to fire the step refinement (the corrector phase).

5.4 Bulirsch-Stoer

5.5 Stopping Condition Algorithm

5.6 Integrator Coefficients

Table 5.1: Prince-Dormand 45 Coefficients

a_i	b_{ij}						
0	0						
$\frac{2}{9}$	$\frac{2}{9}$	0					
$\frac{1}{3}$	$\frac{1}{12}$	$\frac{1}{4}$	0				
$\frac{5}{9}$	$\frac{55}{324}$	$-\frac{25}{108}$	$\frac{50}{81}$	0			
$\frac{2}{3}$	$\frac{83}{330}$	$-\frac{13}{22}$	$\frac{61}{66}$	$\frac{9}{110}$	0		
1	$-\frac{19}{28}$	$\frac{9}{4}$	$\frac{1}{7}$	$-\frac{27}{7}$	$\frac{22}{7}$	0	
1	$\frac{19}{200}$	0	$\frac{3}{5}$	$-\frac{243}{400}$	$\frac{33}{40}$	$\frac{7}{80}$	0
c_j	$\frac{19}{200}$	0	$\frac{3}{5}$	$-\frac{243}{400}$	$\frac{33}{40}$	$\frac{7}{80}$	0
e_j	$\frac{19}{200}$ - $\frac{431}{5000}$	0	$\frac{3}{5}$ - $\frac{333}{500}$	$-\frac{243}{400}$ + $\frac{7857}{10000}$	$\frac{33}{40}$ - $\frac{957}{1000}$	$\frac{7}{80}$ - $\frac{193}{2000}$	$\frac{1}{50}$

Table 5.2: Prince-Dormand 56 Coefficients (Warning: There is an error in the original source for these and we have not found the correct coefficients yet!!)

a_i	b_{ij}							
0	0							
$\frac{1}{10}$	$\frac{1}{10}$	0						
$\frac{2}{9}$	$-\frac{2}{81}$	$\frac{20}{81}$	0					
$\frac{3}{7}$	$\frac{615}{1372}$	$-\frac{270}{343}$	$\frac{1053}{1372}$	0				
$\frac{3}{5}$	$\frac{3243}{5500}$	$-\frac{54}{55}$	$\frac{50949}{71500}$	$\frac{4998}{17875}$	0			
$\frac{4}{5}$	$-\frac{26492}{37125}$	$\frac{72}{55}$	$\frac{2808}{23375}$	$-\frac{24206}{37125}$	$\frac{338}{495}$	0		
1	$\frac{5561}{2376}$	$-\frac{35}{11}$	$-\frac{24117}{31603}$	$\frac{899983}{200772}$	$-\frac{5225}{1836}$	$\frac{3925}{4056}$	0	
1	$\frac{465467}{266112}$	$-\frac{2945}{1232}$	$\frac{10513573}{3212352}$	$-\frac{5610201}{14158144}$	$-\frac{424325}{205632}$	$\frac{376225}{454272}$	0	0
c_j	$\frac{61}{864}$	0	$\frac{98415}{321776}$	$\frac{16807}{146016}$	$\frac{1375}{7344}$	$\frac{1375}{5408}$	$-\frac{37}{1120}$	$\frac{1}{10}$
e_j	$\frac{61}{864}$ - $\frac{821}{10800}$	0	$\frac{98415}{321776}$ - $\frac{19683}{71825}$	$\frac{16807}{146016}$ - $\frac{175273}{912600}$	$\frac{1375}{7344}$ - $\frac{395}{3672}$	$\frac{1375}{5408}$ - $\frac{785}{2704}$	$-\frac{37}{1120}$ - $\frac{3}{50}$	$\frac{1}{10}$

Table 5.3: Runge-Kutta-Fehlberg 56 Coefficients

a_i	b_{ij}							
0	0							
$\frac{1}{6}$	$\frac{1}{6}$	0						
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$	0					
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$	0				
$\frac{4}{5}$	$-\frac{8}{5}$	$\frac{144}{25}$	-4	$\frac{16}{25}$	0			
1	$\frac{361}{320}$	$-\frac{18}{5}$	$\frac{407}{128}$	$-\frac{11}{80}$	$\frac{55}{128}$	0		
0	$-\frac{11}{640}$	0	$\frac{11}{256}$	$-\frac{11}{160}$	$\frac{11}{256}$	0	0	
1	$\frac{93}{640}$	$-\frac{18}{5}$	$\frac{803}{256}$	$-\frac{11}{160}$	$\frac{99}{256}$	0	1	0
c_j	$\frac{7}{1408}$	0	$\frac{1125}{2816}$	$\frac{9}{32}$	$\frac{125}{768}$	0	$\frac{5}{66}$	$\frac{5}{66}$
e_j	$\frac{7}{1408}$ - $\frac{31}{384}$	0	0	0	0	$-\frac{5}{66}$	$\frac{5}{66}$	$\frac{5}{66}$

Table 5.4: Prince-Dormand 78 Coefficients

a_i	b_{ij}											
0	0											
$\frac{1}{18}$	$\frac{1}{18}$											
$\frac{1}{12}$	$\frac{1}{48}$	$\frac{1}{16}$										
$\frac{1}{8}$	$\frac{1}{32}$	0	$\frac{3}{32}$									
$\frac{5}{16}$	$\frac{5}{16}$	0	$-\frac{75}{64}$	$\frac{75}{64}$								
$\frac{3}{8}$	$\frac{3}{80}$	0	0	$\frac{3}{16}$	$\frac{3}{20}$							
$\frac{59}{400}$	$\frac{29443841}{614563906}$	0	0	$\frac{77736538}{692538347}$	$-\frac{28693883}{1125000000}$	$\frac{23124283}{1800000000}$						
$\frac{93}{200}$	$\frac{16016141}{946692911}$	0	0	$\frac{61564180}{158732637}$	$\frac{22789713}{633445777}$	$\frac{545815736}{2771057229}$	$-\frac{180193667}{1043307555}$					
$\frac{5490023248}{9719169821}$	$\frac{39632708}{573591083}$	0	0	$-\frac{433636366}{683701615}$	$-\frac{421739975}{2616292301}$	$\frac{100302831}{723423059}$	$\frac{790204164}{839813087}$	$\frac{800635310}{3783071287}$				
$\frac{13}{20}$	$\frac{246121993}{1340847787}$	0	0	$-\frac{37695042795}{15268766246}$	$-\frac{309121744}{1061227803}$	$-\frac{12992083}{490766935}$	$\frac{6005943493}{2108947869}$	$\frac{393006217}{1396673457}$	$\frac{123872331}{1001029789}$			
$\frac{1201146811}{1299019798}$	$-\frac{1028468189}{846180014}$	0	0	$\frac{8478235783}{508512852}$	$\frac{1311729495}{1432422823}$	$-\frac{10304129995}{1701304382}$	$-\frac{48777925059}{3047939560}$	$\frac{15336726248}{1032824649}$	$-\frac{45442868181}{3398467696}$	$\frac{3065993473}{597172653}$		
1	$\frac{185892177}{718116043}$	0	0	$-\frac{3185094517}{667107341}$	$-\frac{477755414}{1098053517}$	$-\frac{703635378}{230739211}$	$\frac{5731566787}{1027545527}$	$\frac{5232866602}{850066563}$	$-\frac{4093664535}{808688257}$	$\frac{3962137247}{1805957418}$	$\frac{65686358}{487910083}$	
1	$\frac{403863854}{491063109}$	0	0	$-\frac{5068492393}{434740067}$	$-\frac{411421997}{543043805}$	$\frac{652783627}{914296604}$	$\frac{11173962825}{925320556}$	$-\frac{13158990841}{6184727034}$	$\frac{3936647629}{1978049680}$	$-\frac{160528059}{685178525}$	$\frac{248638103}{1413531060}$	0
c_j	$\frac{14005451}{335480064}$	0	0	0	0	$-\frac{59238493}{1068277825}$	$\frac{181606767}{758867731}$	$\frac{561292985}{797845732}$	$-\frac{1041891430}{1371343529}$	$\frac{760417239}{1151165299}$	$\frac{118820643}{751138087}$	$-\frac{528747749}{2220607170}$
$\hat{c}_j$	$\frac{13451932}{455176623}$	0	0	0	0	$-\frac{808719846}{976000145}$	$\frac{1757004468}{5645159321}$	$\frac{656045339}{265891186.0}$	$-\frac{3867574721}{1518517206}$	$\frac{465885868}{322736535}$	$\frac{53011238}{667516719}$	$\frac{2.0}{45.0}$

Chapter 6

Measurement Modeling

This chapter gives the formulation for computed measurement values and their partial derivatives, measurement media corrections, measurement error models, and measurement feasibility and editing criteria. We begin with a discussion of the general form of the measurement model which is broken down into several contributions including the model for the ideal observable quantity, media corrections, and measurement errors. The geometry of a generic measurement is presented and the notation used throughout the measurement model formulation is discussed. For each observable type such as two-way range or one-way Doppler we present the general model and then discuss specific details of how these measurements are modeled for specific tracking systems such as TDRSS and DSN. We conclude this chapter with a section on simple geometric measurement models that neglect media corrections, error modeling, and light time corrections.

6.1 General Form of the Measurement Model

Spacecraft tracking measurements are produced by complex interactions between the measurement participants, their associated sensors, and the space environment. Examples of measurement participants include spacecraft with antennas and transponders, tracking stations, quasars, and other celestial bodies such as Earth.

The measurement model in GMAT divides the computed measurement value into three terms

$$y_c = y_i + e_t + \delta_m \tag{6.1}$$

where y_c is the computed measurement value, y_i is the model of the ideal measurement observable quantity including light time correction and sensor delays, e_t is the total stochastic error contribution from all sources including sensor and systematic errors, and δ_m is the sum of media corrections from ionospheric, tropospheric, general relativistic corrections to name a few.

In the next three sections, we discuss each of the contributions to the measurement computed value in detail.

6.1.1 Ideal Observables: Geometry, Coordinate Systems, and Notation

The ideal measurement observable, y_i, accounts for the deterministic portion of the measurement-given the current best estimate. This portion of the computed measurement value contains the model for (1) the specific measurement data type, (2) the dynamics of the participants during the measurement process, and (3) the sequence of events that occur during the taking of a measurement (i.e. light time delays, sensor delays, and averaging intervals). The ideal measurement quantity does not contain stochastic errors such as bias or exponentially correlated noise. Nor does the ideal measurement quantity contain media corrections due to the properties of the space environment such as ionospheric delays or general relativistic corrections.

We use a basic one-way measurement model, shown in Fig. (6.1), to aid in defining our notation. For the purposes of illustration, we have assumed that both participants are spacecraft and that their states are known – being propagated or estimated – with respect to different coordinate systems. The subsequent

mathematical development is general and is valid for measurement processes that involve different participant types such as ground stations.

GMAT allows users to input and output orbit states in numerous coordinate systems. However, internal to GMAT, spacecraft are propagated and estimated with respect to a coordinate system that has a celestial body as the origin and uses the J2000 axis system. Different spacecraft in a single simulation may have different reference celestial bodies depending upon the flight regime of a particular application. Ground station states are defined and estimated in a celestial body fixed system. Below, we express quantities such as the inertial position and velocity of participants and antennas in terms of the coordinate systems in which they are internally (propagated/estimated) represented. These quantities are referenced repeatedly in later sections.

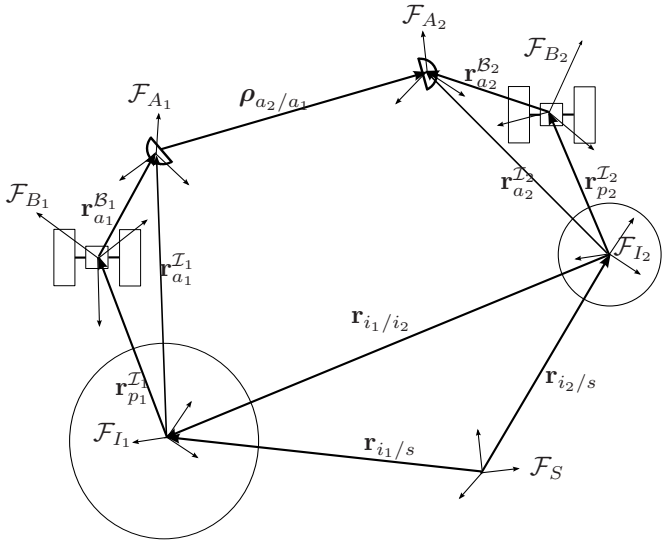


Figure 6.1: Measurement Geometry and Notation

Define the following objects: the initiating participant, $\mathcal{P}_1$, and its attached antenna $\mathcal{A}_1$, the receiving participant, $\mathcal{P}_2$, and its attached antenna $\mathcal{A}_2$. Define the following coordinates systems: $\mathcal{F}_S$ is the solar system barycentric system, $\mathcal{F}_{\mathcal{R}_1}$ is the system in which $\mathcal{P}_1$ is naturally (internally) represented (propagated/estimated), $\mathcal{F}_{\mathcal{B}_1}$ is the body-fixed system of $\mathcal{P}_1$, $\mathcal{F}_{\mathcal{A}_1}$ is the antenna-fixed system for antenna $\mathcal{A}_1$, and $\mathcal{F}_{\mathcal{I}_1}$ is the local celestial-body-centered J2000 system for $\mathcal{P}_1$.

Using the definitions above, the system propagates/estimates spacecraft and ground stations using the representation $\mathbf{r}_p^{\mathcal{R}}$. This means the position of participant p in coordinate system $\mathcal{R}$. For a spacecraft, $\mathcal{R}$ is the celestial-body-centered J2000 system, for a ground station $\mathcal{R}$ is the celestial-body-centered, body-fixed system. The location of antennas are defined with respect to their parent participant's attitude coordinate system and notated $\mathbf{r}_a^{\mathcal{B}}$.

The measurement models require the expressions for the locations of participants $\mathcal{P}_1$ and $\mathcal{P}_2$ and the phase centers of antennas $\mathcal{A}_1$ and $\mathcal{A}_2$ in the local inertial systems $\mathcal{F}_{I_1}$ and $\mathcal{F}_{I_2}$, and in the solar system barycentric coordinate system, $\mathcal{F}_S$. The locations of participants $\mathcal{P}_1$ and $\mathcal{P}_2$ in their local inertial systems are calculated using

$$\mathbf{r}_{p_1}^{\mathcal{I}_1} = \mathbf{R}^{\mathcal{I}/\mathcal{R}_1}\mathbf{r}_{p1}^{\mathcal{R}_1} \tag{6.2}$$

$$\mathbf{r}_{p_2}^{\mathcal{I}_2} = \mathbf{R}^{\mathcal{I}/\mathcal{R}_2}\mathbf{r}_{p2}^{\mathcal{R}_2} \tag{6.3}$$

If $\mathcal{P}_1$ is a spacecraft, $\mathbf{R}^{\mathcal{I}/\mathcal{R}_1} = I_{3x3}$ because spacecraft are represented with respect J2000 axis system (i.e . $\mathcal{F}_{\mathcal{R}_1} = \mathcal{F}_{\mathcal{I}_1}$). If $\mathcal{P}_1$ is a ground station, $\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}$ is the body-fixed rotation matrix for the ground station's reference central body (i.e. for Earth-based ground systems $\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}$ is the rotation matrix from Earth-fixed to Earth-J2000). The expressions for the locations of participants $\mathcal{P}_1$ and $\mathcal{P}_2$ in the solar system barycentric system are

$$\mathbf{r}_{p_1}^{\mathcal{S}} = \mathbf{r}_{p_1}^{\mathcal{I}_1} + \mathbf{r}_{i_1}^{\mathcal{S}} \tag{6.4}$$

$$\mathbf{r}_{p_2}^{S} = \mathbf{r}_{p_2}^{\mathcal{I}_2} + \mathbf{r}_{i_2}^{\mathcal{S}} \tag{6.5}$$

where $\mathbf{r}_{i_2}^{\mathcal{S}}$ is the location of the origin of $\mathcal{F}_{\mathcal{I}_2}$ expressed in $\mathcal{F}_S$. Note that to transform a vector from a local inertial system to the barycentric system only requires a translation because $\mathcal{F}_{\mathcal{I}_1}$ and $\mathcal{F}_{\mathcal{S}}$ both use the J2000 axis system.

The locations of antenna phase centers in local inertial coordinates are

$$\mathbf{r}_{a_1}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}/\mathcal{R}_1}\mathbf{r}_{p1}^{\mathcal{R}_1} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_1}\mathbf{r}_{a_1}^{\mathcal{B}_1} \tag{6.6}$$

$$\mathbf{r}_{a_2}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}/\mathcal{R}_2}\mathbf{r}_{p2}^{\mathcal{R}_2} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_2}\mathbf{r}_{a_2}^{\mathcal{B}_2} \tag{6.7}$$

The locations of antenna phase centers in solar system barycentric coordinates are

$$\mathbf{r}_{a_1}^{S} = \mathbf{r}_{a_1}^{\mathcal{I}} + \mathbf{r}_{i_1}^{\mathcal{S}} \tag{6.8}$$

$$\mathbf{r}_{a_2}^{S} = \mathbf{r}_{a_2}^{\mathcal{I}} + \mathbf{r}_{i_2}^{\mathcal{S}} \tag{6.9}$$

6.1.2 One Way Range Example

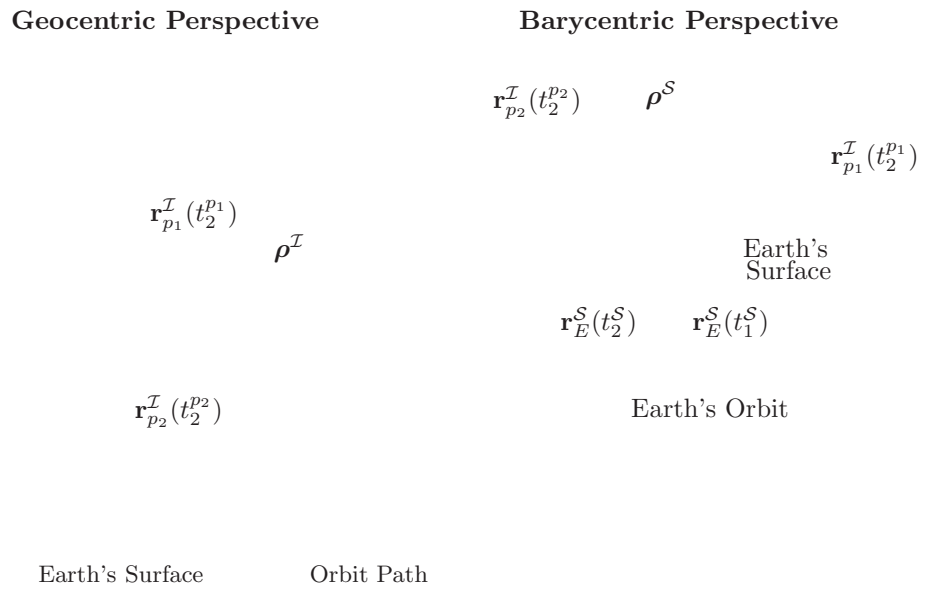


Figure 6.2: Illustration of Range Vector in Geocentric and Barycentric Systems

The range vector as seen from the geocentric inertial observer is

$$\boldsymbol{\rho}^{\mathcal{I}} = \mathbf{r}_{p_2}^{\mathcal{I}}(t_2^{p_2}) - \mathbf{r}_{p_1}^{\mathcal{I}}(t_2^{p_1}) \tag{6.10}$$

The range vector as seen by a barycentric inertial observer is

$$\boldsymbol{\rho}^{\mathcal{S}} = (\mathbf{r}_E^{\mathcal{S}}(t_2^{\mathcal{S}}) + \mathbf{r}_{p_2}^{\mathcal{I}}(t_2^{p_2})) - (\mathbf{r}_E^{\mathcal{S}}(t_1^{\mathcal{S}}) + \mathbf{r}_{p_1}^{\mathcal{I}}(t_2^{p_1}) \tag{6.11}$$

Note the two ranges are not the same

$$\boldsymbol{\rho}^{\mathcal{S}} = \boldsymbol{\rho}^{\mathcal{I}} + (\mathbf{r}_E^{\mathcal{S}}(t_2^{\mathcal{S}}) - \mathbf{r}_E^{\mathcal{S}}(t_1^{\mathcal{S}})) \tag{6.12}$$

6.2 Light-Time Solution

6.2.1 One-Way Light Time

The definition of the one-way light time is the time duration for light to travel from a transmitting antenna to a receiving antenna in a vacuum with no gravity. The center of mass location is used for a spacecraft or ground station that does not have an antenna model. Atmospheric corrections such as Tropospheric and Ionospheric delays, as well as relativistic corrections due to gravity, are additive corrections discussed in detail in later sections.

Define the time duration for the light transit as $\tau = t_{2_a} - t_{1_a}$ where t_{1_a} and t_{2_a} are the signal transmission and receive times at antennas $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. It two participants share the same reference celestial body, the range vector from $\mathcal{A}_1$ to $\mathcal{A}_2$ is given by

$$\boldsymbol{\rho}^{\mathcal{I}} = \mathbf{r}^{\mathcal{I}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{I}}_{a_1}(t_{1_a}) \tag{6.13}$$

where $\mathbf{r}^{\mathcal{I}}_{a_2}(t_{2_a})$ and $\mathbf{r}^{\mathcal{I}}_{a_1}(t_{1_a})$ are calculated using Eqs. (6.6) and (6.7). To solve for τ, we find the solution to the following equation:

$$c\tau - \rho_{a_2/a_1} = 0 \tag{6.14}$$

where $\rho = \|\boldsymbol{\rho}\|$. Depending upon the measurement, we know either time t_{1_a} or t_{2_a}, and solve for the unknown time. In the case where t_{2_a} is known, τ is found using fixed-point iteration on the following equation[10]

$$\tau^{(i+1)} = \frac{1}{c}\|\mathbf{r}^{\mathcal{I}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{I}}_{a_1}(t_{2_a} - \tau^{(i)})\| \tag{6.15}$$

In the case where $t_{1_a}$is known, τ is found using fixed-point iteration on the following equation

$$\tau^{(i+1)} = \frac{1}{c}\|\mathbf{r}^{\mathcal{I}}_{a_2}(t_{1_a} + \tau^{(i)}) - \mathbf{r}^{\mathcal{I}}_{a_1}(t_{1_a})\| \tag{6.16}$$

In either case, the value $\tau^{(0)} = 0$ is used as the initial guess.

If the two participants do not share a common reference celestial body then the range vector is calculated in the solar system barycentric system using

$$\boldsymbol{\rho}^{\mathcal{S}} = \mathbf{r}^{\mathcal{S}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{S}}_{a_1}(t_{1_a}) \tag{6.17}$$

where $\mathbf{r}^{\mathcal{S}}_{a_1}(t_{1_a})$ and $\mathbf{r}^{\mathcal{S}}_{a_2}(t_{2_a})$ are calculated using Eqs. (6.8) and (6.9) respectively. When t_{2_a} is known, τ is found using fixed-point iteration on the following equation[10]

$$\tau^{(i+1)} = \frac{1}{c}\|\mathbf{r}^{\mathcal{S}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{S}}_{a_1}(t_{2_a} - \tau^{(i)})\| \tag{6.18}$$

When $t_{1_a}$is known, τ is found using fixed-point iteration on the following equation

$$\tau^{(i+1)} = \frac{1}{c}\|\mathbf{r}^{\mathcal{S}}_{a_2}(t_{1_a} + \tau^{(i)}) - \mathbf{r}^{\mathcal{S}}_{a_1}(t_{1_a})\| \tag{6.19}$$

6.2.2 Partial Derivatives of the One-Way Light Time

In this section, we derive the partial derivatives of the one-way light time with respect to the solve-for parameters. Partial derivatives of the one-way light time appear in most partial derivatives for computed observations in later sections. The light time solution can be solved in the local inertial system or the solar system barycentric system. Furthermore, the light time solution can be expressed as a time duration or a distance. We begin by formulating the expressions for the partial derivatives of the light time solution in the local inertial system expressed as a distance. Next we formulate the partials of the barycentric light time solution expressed as a distance. Finally, we show the relationship between the partials for the distance-based and time-based formulations.

From the light time solution, we know the range vector in either the local inertial or solar system barycenter as shown below, which are Eqs. (6.17) and (6.13), repeated here for convenience.

$$\boldsymbol{\rho}^{\mathcal{S}} = \mathbf{r}^{\mathcal{S}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{S}}_{a_1}(t_{1_a})$$

$$\boldsymbol{\rho}^{\mathcal{I}} = \mathbf{r}^{\mathcal{I}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{I}}_{a_1}(t_{1_a})$$

If the two participants are referenced to the same celestial body we use the local inertial expression, otherwise we use the solar system barycentric expression. Let's look at the local inertial formulation first. Writing $\boldsymbol{\rho}^{\mathcal{I}}$ in terms of the solve-for parameters we obtain:

$$\boldsymbol{\rho}^{\mathcal{I}} = \left(\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}\mathbf{r}^{\mathcal{R}_2}_{p2} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_2}\mathbf{r}^{\mathcal{B}_2}_{a_2}\right) - \left(\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}\mathbf{r}^{\mathcal{R}_1}_{p1} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_1}\mathbf{r}^{\mathcal{B}_1}_{a_1}\right) \tag{6.20}$$

To simplify the notation, we have assumed that the rotation matrices and position vectors associated with $\mathcal{P}_1$ are evaluated at time t_{1_a} and the rotation matrices and position vectors associated with $\mathcal{P}_2$ are evaluated at time t_{2_a}. The partial derivative of the norm of a vector $\mathbf{a}$ with respect to a dummy variable $\boldsymbol{\chi}$ is given by

$$\frac{\partial a}{\partial \boldsymbol{\chi}} = \frac{\mathbf{a}^T}{a}\frac{\partial \mathbf{a}}{\partial \boldsymbol{\chi}} \tag{6.21}$$

where the superscript "T" is the transpose operator. Define t_m as the measurement time tag and define $\mathbf{r}^{\mathcal{I}_1}_{p_1}(t_m)$ as the position of participant 1,at time t_m, expressed in $\mathcal{F}_{R_1}$.

$$\frac{\partial \rho^{\mathcal{I}}}{\partial \mathbf{r}^{\mathcal{R}_1}_{p_1}(t_m)} = -\frac{(\boldsymbol{\rho}^{\mathcal{I}})^T}{\rho^{\mathcal{I}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}(t_{1_a})\mathbf{A}^{\mathcal{R}_1}_{p_1}(t_{1_a}, t_m) \tag{6.22}$$

$$\frac{\partial \rho^{\mathcal{I}}}{\partial \mathbf{v}^{\mathcal{R}_1}_{p_1}(t_m)} = -\frac{(\boldsymbol{\rho}^{\mathcal{I}})^T}{\rho^{\mathcal{I}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}(t_{1_a})\mathbf{B}^{\mathcal{R}_1}_{p_1}(t_{1_a}, t_m) \tag{6.23}$$

$$\frac{\partial \rho^{\mathcal{I}}}{\partial \mathbf{r}^{\mathcal{R}_2}_{p_2}(t_m)} = \frac{(\boldsymbol{\rho}^{\mathcal{I}})^T}{\rho^{\mathcal{I}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}(t_{2_a})\mathbf{A}^{\mathcal{R}_2}_{p_2}(t_{2_a}, t_m) \tag{6.24}$$

$$\frac{\partial \rho^{\mathcal{I}}}{\partial \mathbf{v}^{\mathcal{R}_2}_{p_2}(t_m)} = \frac{(\boldsymbol{\rho}^{\mathcal{I}})^T}{\rho^{\mathcal{I}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}(t_{2_a})\mathbf{B}^{\mathcal{R}_2}_{p_2}(t_{2_a}, t_m) \tag{6.25}$$

Writing $\partial\boldsymbol{\rho}^{\mathcal{S}}/\partial\boldsymbol{\chi}$ in terms of the solve-for parameters:

$$\boldsymbol{\rho}^{\mathcal{S}} = \left(\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}\mathbf{r}^{\mathcal{R}_2}_{p2} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_2}\mathbf{r}^{\mathcal{B}_2}_{a_2}\right) - \left(\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}\mathbf{r}^{\mathcal{R}_1}_{p1} + \mathbf{R}^{\mathcal{I}/\mathcal{B}_1}\mathbf{r}^{\mathcal{B}_1}_{a_1}\right) + \left(\mathbf{r}^{\mathcal{S}}_{i_1}(t_{2_a}) - \mathbf{r}^{\mathcal{S}}_{i_1}(t_{1_a})\right) \tag{6.26}$$

The partial derivatives are

$$\frac{\partial \rho^{\mathcal{S}}}{\partial \mathbf{r}^{\mathcal{R}_1}_{p_1}(t_m)} = -\frac{(\boldsymbol{\rho}^{\mathcal{S}})^T}{\rho^{\mathcal{S}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}(t_{1_a})\mathbf{A}^{\mathcal{R}_1}_{p_1}(t_{1_a}, t_m) \tag{6.27}$$

$$\frac{\partial \rho^{\mathcal{S}}}{\partial \mathbf{v}^{\mathcal{R}_1}_{p_1}(t_m)} = -\frac{(\boldsymbol{\rho}^{\mathcal{S}})^T}{\rho^{\mathcal{S}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_1}(t_{1_a})\mathbf{B}^{\mathcal{R}_1}_{p_1}(t_{1_a}, t_m) \tag{6.28}$$

$$\frac{\partial \rho^{\mathcal{S}}}{\partial \mathbf{r}^{\mathcal{R}_2}_{p_2}(t_m)} = \frac{(\boldsymbol{\rho}^{\mathcal{S}})^T}{\rho^{\mathcal{S}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}(t_{2_a})\mathbf{A}^{\mathcal{R}_2}_{p_2}(t_{2_a}, t_m) \tag{6.29}$$

$$\frac{\partial \rho^{\mathcal{S}}}{\partial \mathbf{v}^{\mathcal{R}_2}_{p_2}(t_m)} = \frac{(\boldsymbol{\rho}^{\mathcal{S}})^T}{\rho^{\mathcal{S}}}\mathbf{R}^{\mathcal{I}/\mathcal{R}_2}(t_{2_a})\mathbf{B}^{\mathcal{R}_2}_{p_2}(t_{2_a}, t_m) \tag{6.30}$$

In the derivatives above, $\mathbf{A}_{p_1}$ is the upper left 3x3 partition of the state transition matrix for $\mathcal{P}_1$ and $\mathbf{B}_{p_1}$ is the upper right 3x3 partition.

$$\boldsymbol{\Phi}^{\mathcal{R}_1}_{p_1}(t_f, t_i) = \begin{pmatrix} \mathbf{A}^{\mathcal{R}_1}_{p_1}(t_f, t_i) & \mathbf{B}^{\mathcal{R}_1}_{p_1}(t_f, t_i) \\ \mathbf{C}^{\mathcal{R}_1}_{p_1}(t_f, t_i) & \mathbf{C}^{\mathcal{R}_1}_{p_1}(t_f, t_i) \end{pmatrix} \tag{6.31}$$

Above we assume that light time solution is expressed as a distance. Some measurement formulations use the light time solution expressed as a time duration. The time Δt for light to travel a distance ρ (in a vacuum with no gravity) is given by

$$\Delta t = \frac{\rho}{c} = (t_{2_a} - t_{1_a}) \tag{6.32}$$

The partial derivative of Δt with respect to a dummy variable $\boldsymbol{\chi}$ is given by

$$\frac{\partial \Delta t}{\partial \boldsymbol{\chi}} = \frac{1}{c}\frac{\partial \rho}{\partial \boldsymbol{\chi}} \tag{6.33}$$

Partial derivatives of the light time solution expressed in terms of a time duration are calculate using Eq. (6.33) along with the appropriate expression from Eqs.(6.22 - 6.25) and (6.27 - 6.30).

6.3 Computed Value of Two-Way Range

6.3.1 Overview

The computed value of two-way range is a measure of the round trip signal time between a set of two or more participants and uses a single clock on the initiating participant to determine the elapsed time of signal transit. While the basic model is common to many two-way tracking systems, different systems handle time delays and precision differently. Also, if and how the round trip signal time delay is converted to a range or distance unit varies among tracking systems. Below we develop a general formulation of the two-way range observable for two participants and then present how the model applies to different tracking systems including what transponder delays are put on the tracking data file or subtracted from the observed quantity, and how the different systems handle conversion to distance units.

The geometry, definitions, and notation for a two-way measurement are simple extensions of those illustrated in Fig. 6.1. For the initial development, assume we have two participants, each with an attached sensor and antenna. A time-line for a two-participant, two-way measurement is shown in Fig. 6.3 where t_{1_E} is the time the signal is generated by the electronics on the first participant. Time t_{1_T} is the time the signal

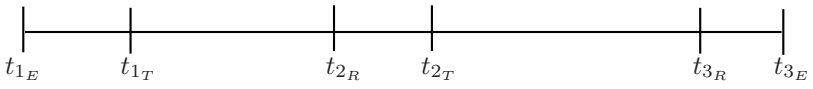


Figure 6.3: Two-Way Measurement Time Line

is transmitted from the antenna on participant 1 where the uplink delay of $\mathcal{P}_1$. The electronics delay of $\mathcal{P}_1$ is defined as τ_1 and such that

$$t_{1_T} = t_{1_E} + \tau_1 \tag{6.34}$$

At time t_{2_R}, the signal arrives at the receiving antenna on $\mathcal{P}_2$. At time t_{2_T} the signal is transmitted by the transmitting antenna on $\mathcal{P}_2$. The transponder delay for $\mathcal{P}_2$ is defined as τ_2 such that

$$t_{2_T} = t_{2_R} + \tau_2 \tag{6.35}$$

Finally, at t_{3_R} the return signal is received at the antenna on $\mathcal{P}_1$ and is stamped and processed at time t_{3_E}. The down-link electronics delay is defined as τ_3 such that

$$t_{3_E} = t_{3_R} + \tau_3 \tag{6.36}$$

According to these definitions, the total elapsed time for a two-participant, two-way range measurement is given by

$$\Delta t = \tau_1 + \frac{\rho_u}{c} + \tau_2 + \frac{\rho_d}{c} + \tau_3 \tag{6.37}$$

where

$$\rho_u = c\,(t_{2_R} - t_{1_T}) = ||\mathbf{r}^C_{2_a}(t_{2_R}) - \mathbf{r}^C_{1_a}(t_{1_T})|| \tag{6.38}$$

and

$$\rho_d = c\,(t_{3_R} - t_{2_T}) = ||\mathbf{r}^C_{1_a}(t_{3_R}) - \mathbf{r}^C_{2}(t_{2_T})|| \tag{6.39}$$

Eqs. (6.38) and (6.39) are solved using the one-way light time algorithm described in Sec. 6.2.1. The coordinate system, $\mathcal{F}_\mathcal{C}$, in which the light time is solved is described in 6.2.1.

6.3.2 NASA Ground Network (STDN) and Universal Space Network (USN)

The time tag for two-way range measurements provided by STDN or USN is t_{3_E}, the time the downlink signal is processed by the electronics at the ground station. The NASA ground network performs system calibration to determine τ_1 and τ_3 and subtracts these values from the observed quantity before writing to the tracking data file. τ_1 and τ_3 values are not provided to the user and therefore cannot be used in the light time correction. Typical values for τ_1 and τ_3 are on the order of microseconds, which for LEO results in about 1 cm of position error in the light time solution. The transponder delay is converted to a distance by multiplying by the speed of light to yield the two-way range measurement, R_2.

$$R_2 = \frac{1}{2}\left(\rho_u + c\tau_2 + \rho_d\right) \tag{6.40}$$

The partial derivatives are shown below:

$$\frac{\partial R_2}{\partial \tau_1} = \text{(not a supported solve-for)} \tag{6.41}$$

$$\frac{\partial R_2}{\partial \tau_3} = \text{(not a supported solve-for)} \tag{6.42}$$

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})}\right) \tag{6.43}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})}\right) \tag{6.44}$$

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})}\right) \tag{6.45}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})}\right) \tag{6.46}$$

$$\tag{6.47}$$

The total range correction, ΔR_2, due to atmospheric effects at the Earth from the troposphere and ionosphere is calculated using

$$\Delta R_2 = \frac{1}{2}\left(\Delta\rho_u + \Delta\rho_d\right) \tag{6.48}$$

where $\Delta\rho_u$ and $\Delta\rho_d$ are the uplink and downlink range corrections respectively. Both $\Delta\rho_u$ and $\Delta\rho_d$ have contributions from the troposphere and ionosphere given by equations (x) and (x) respectively

6.3.3 NASA Space Network (TDRSS)

The geometry for the TDRSS two-way range model is shown in Fig. 6.4. TDRSS provides two-way range observations by using a single relay spacecraft to perform the forward and return link to the user spacecraft. There are four one-way light time transits resulting in four ranges: the uplink range ρ_u, the forward link range, ρ_f, the return link range, ρ_r, and the downlink range, ρ_d. Define the following variables for the time sequence of events. t_{1_E} is the time the signal is generated by the electronics of the initiating ground terminal. Time $t_{1_T} = t_{1_E} + \tau_1$ is the time the signal is transmitted from the antenna at the ground station. At time t_{2_R}, the signal is received at the antenna on the TDRS. At time $t_{2_T} = t_{2_R} + \tau_2$ the signal is broadcast from the forward link antenna to the antenna on the user spacecraft. At t_{3_R} the signal is received at the antenna on the user spacecraft. At time $t_{3_T} = t_{3_R} + \tau_3$ the signal is broadcast from the antenna on the user spacecraft back to the TDRS. At time t_{4_R} the signal is received at the antenna on the TDRS. At time $t_{4_T} = t_{4_R} + \tau_4$ the signal is broadcast from the antenna on the TDRS to the the ground terminal. At time t_{5_R} the signal is received at the antenna on the ground terminal. Finally, at time $t_{5_E} = t_{5_R} + \tau_5$ the signal is processed by the electronics at the ground terminal.

The TDRSS network performs system calibration to determine the uplink delay, τ_1, the TDRSS forward delay, τ_2, the TDRSS return link delay,τ_4, and the down-link delay, τ_5. These values are subtracted from the observed quantity before writing to the tracking data file (Phung[23] *et. al.*, pg. 4-9). The transponder

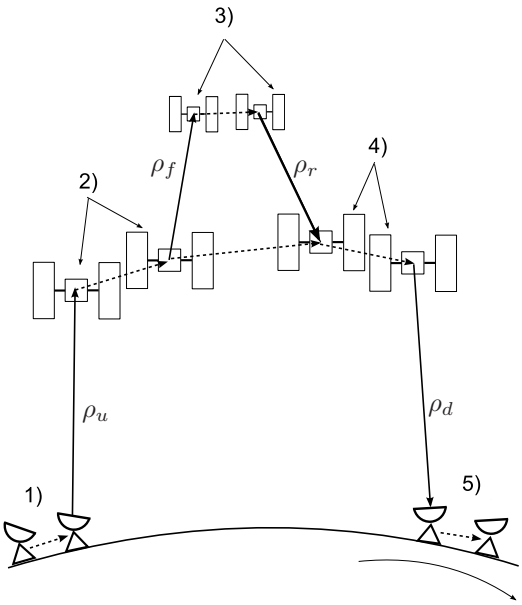


Figure 6.4: TDRSS Two-Way Range Geometry

delay values are not provided to the user and therefore cannot be used in the light time correction. Typical values for those delays are on the order of 0.1 microseconds, which for LEO results in a few mm of position error in the light time solution.

The computed two-way range model for TDRSS is given by (see Long[8] *et. al.* Eq. 7-54):

$$R_2 = \frac{1}{2}\left(\rho_u + \rho_f + \rho_r + \rho_d + c\tau_3\right) \tag{6.49}$$

Eq. (6.49) is solved by backwards signal propagation beginning at the measurement time tag, T_{5_E}. Given these definitions, the four light-time transits in Eq. (6.49) are given below and solved in the order shown. For the purposes of the light-time solutions, we assume that the electronics delays are all zero and drop the subscripts "R", "T", and "E" from the time variables.

$$\rho_d = ||\mathbf{r}^C_{5_a}(t_5) - \mathbf{r}^C_{4_a}(t_4)|| \tag{6.50}$$

$$\rho_r = ||\mathbf{r}^C_{4_a}(t_4) - \mathbf{r}^C_{3_a}(t_3)|| \tag{6.51}$$

$$\rho_f = ||\mathbf{r}^C_{3_a}(t_3) - \mathbf{r}^C_{2_a}(t_2)|| \tag{6.52}$$

$$\rho_u = ||\mathbf{r}^C_{2_a}(t_2) - \mathbf{r}^C_{1_a}(t_1)|| \tag{6.53}$$

The above equations are solved using the one-way light time algorithm discussed in Sec. 6.2.1.

The partial derivatives of the two way range, R_2, are given below. Partials with respect to the user spacecraft state are:

$$\frac{\partial R_2}{\partial \mathbf{r}^{\mathcal{R}_3}_{p_3}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_f}{\partial \mathbf{r}^{\mathcal{R}_3}_{p_3}(t_{5_e})} + \frac{\partial \rho_r}{\partial \mathbf{r}^{\mathcal{R}_3}_{p_3}(t_{5_e})}\right) \tag{6.54}$$

$$\frac{\partial R_2}{\partial \mathbf{v}^{\mathcal{R}_3}_{p_3}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_f}{\partial \mathbf{v}^{\mathcal{R}_3}_{p_3}(t_{5_e})} + \frac{\partial \rho_r}{\partial \mathbf{v}^{\mathcal{R}_3}_{p_3}(t_{5_e})}\right) \tag{6.55}$$

$$\tag{6.56}$$

Partials with respect to the ground terminal state are (note: p_1 and p_5 are the same participant):

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{5_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_5}^{\mathcal{R}_5}(t_{5_e})}\right) \tag{6.57}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_5}^{\mathcal{R}_5}(t_{5_e})}\right) \tag{6.58}$$

$$\tag{6.59}$$

Partials with respect to the TDRS state are (note: p_2 and p_4 are the same participant):

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{5_e})} + \frac{\partial \rho_f}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{5_e})} + \frac{\partial \rho_r}{\partial \mathbf{r}_{p_4}^{\mathcal{R}_4}(t_{5_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_4}^{\mathcal{R}_4}(t_{5_e})}\right) \tag{6.60}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{5_e})} = \frac{1}{2}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{5_e})} + \frac{\partial \rho_f}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{5_e})} + \frac{\partial \rho_r}{\partial \mathbf{v}_{p_4}^{\mathcal{R}_4}(t_{5_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_4}^{\mathcal{R}_4}(t_{5_e})}\right) \tag{6.61}$$

$$\tag{6.62}$$

6.3.4 NASA Deep Space Network (DSN)

The NASA Deep Space Network performs system calibration to determine τ_1 and τ_3 and these values are provided in the records on the tracking data file (Moyer,[24]pg. 11-12). The ideal value for the two-way measurement time duration is given by

$$\Delta t_2 = \tau_1 + \frac{\rho_u}{c} + \tau_2 + \frac{\rho_d}{c} + \tau_3 \tag{6.63}$$

It is worth at this time comparing the above equation to Eq. 11-7 of Moyer. In Moyer's formulation RLT_{12} and RLT_{23} are the general relativistic corrections to the light transit time between antennas. These are treated as media corrections in GMAT, and so do not appear in 6.63. The terms of the form $(ET - TAI)_{t_3}$ are also treated as corrections that can be optionally applied and when implemented will be discussed in the media corrections section. The same is true of the sensor corrections terms of the form Δ_A and Δ_{sc}. Finally, the term labelled R_c in Moyer is the range bias (in meters). In GMAT, this term is handled via the measurement error model.

The DSN converts the observed value in units of time to a range unit by integrating a conversion factor, F, that is dependent upon the uplink station's carrier frequency (Moyer,[24]pg. 13-64). For an S-band uplink transmitter, the conversion factor F is

$$F = \frac{1}{2}F_T(S) \tag{6.64}$$

where $F_T(S)$ may be defined by a frequency ramp table described in section xxx, or could be a constant. For S-Band, one range unit is equivalent to two cycles of the transmitter frequency. For X-Band uplink transmitters, the conversion factor is given by

$$F = \frac{11}{75}F_T(\text{X,HEV}) \tag{6.65}$$

for stations prior to conversion to Block 5 exciters (BVE), and for a BVE transmitter F is given by

$$F = \frac{221}{741\text{x}2}F_T(\text{S,BVE}) \tag{6.66}$$

The measurement time duration, Δt, is converted to a distance unit by integrating the conversion factor, F, over the duration of the precision light time solution (Moyer,[24]pg. 13-72).

$$R_2 = \int_{t_1}^{t_3} F\ dt \tag{6.67}$$

In the case of constant frequency transmission, the range conversion simplifies to

$$R_2 = F\Delta t_2 \tag{6.68}$$

$$\frac{\partial R_2}{\partial \tau_1} = \text{(not a supported solve-for)} \tag{6.69}$$

$$\frac{\partial R_2}{\partial \tau_2} = \text{(not a supported solve-for)} \tag{6.70}$$

$$\frac{\partial R_2}{\partial \tau_3} = \text{(not a supported solve-for)} \tag{6.71}$$

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})} = \frac{F}{c}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{3_e})}\right) \tag{6.72}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})} = \frac{F}{c}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{3_e})}\right) \tag{6.73}$$

$$\frac{\partial R_2}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})} = \frac{F}{c}\left(\frac{\partial \rho_u}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{3_e})}\right) \tag{6.74}$$

$$\frac{\partial R_2}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})} = \frac{F}{c}\left(\frac{\partial \rho_u}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})} + \frac{\partial \rho_d}{\partial \mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{3_e})}\right) \tag{6.75}$$

$$\tag{6.76}$$

where the partials of the form $\partial \rho_u / \partial \mathbf{v}_{p_2}^{\mathcal{R}_2}$ are provided in Sec. (6.2.2).

The total range correction, ΔR_2, due to atmospheric effects at the Earth from the troposphere and ionosphere is calculated using

$$\Delta R_2 = \frac{1}{2}\left(\Delta \rho_u + \Delta \rho_d\right) \tag{6.77}$$

where $\Delta \rho_u$ and $\Delta \rho_d$ are the uplink and downlink range corrections respectively. Both $\Delta \rho_u$ and $\Delta \rho_d$ have contributions from the troposphere and ionosphere given by equations (x) and (x) respectively.

6.4 Computed Value of Averaged Two-Way Doppler

Computed values of two-way Doppler measurements are measures of the average Doppler shift over some time interval. The average is used because it is not physically possible to take instantaneous Doppler measurements. The Doppler shifted received frequency, F_R is

$$F_R = F_T\left(1 - \frac{\dot{\rho}}{c}\right) \tag{6.78}$$

where F_T is the transmitted or reference frequency, $\dot{\rho}$ is the range rate, and c is the speed of light. The instantaneous doppler shift, ΔF, is given

$$\Delta F = -F_T \frac{\dot{\rho}}{c} \tag{6.79}$$

Many systems use a constant transmission frequency over the Doppler averaging interval. In this case, for a two-participant Doppler measurement, the average Doppler shift $\bar{\Delta F}$ is given by

$$\bar{\Delta F} = \frac{1}{\Delta T_a}\int_{t_o}^{t_f} -F_T \frac{\dot{\rho}}{c} dt = -\frac{F_T}{c\Delta T_a}\left(\rho(t_f) - \rho(t_o)\right) \tag{6.80}$$

where ΔT_a is the averaging interval given by $t_f - t_o$.

Many tracking systems provide two-way doppler observations. The primary differences in the observables are how the integration in Eq. (6.80) is performed, the frequency used as the reference frequency, and the measurement time tag. For example, STDN, USN, and TRDSS all uses constant values of F_T, while DSN often uses ramped values. TDRSS uses T_f as the measurement time tag t_t, while DSN uses the center of the averaging interval $t_o + \Delta t/2$ as the measurement time tag.

6.5 NASA Deep Space Network (DSN)

Fig. 6.5 shows a time-line for an averaged two-way doppler measurement. $t^s_{1_E}$ is the time the initial signal is generated by the electronics on participant 1 (superscript "s" is described shortly). At time $t^s_{1_E} + \tau_1$ the signal is broadcast from the antenna on participant 1. At time $t^s_{2_R}$ the signal is received at the antenna on participant 2. At time $t^s_{2_T}$ the signal is rebroadcast by the antenna on participant 2. At time $t^s_{3_R}$, the signal is received at the antenna at the participant performing the Doppler count. At time $t^s_{3_E}$, the signal is processed by the electronics on participant 1 and the phase counting begins. Phase counting continues for an interval ΔT_a in local station time.

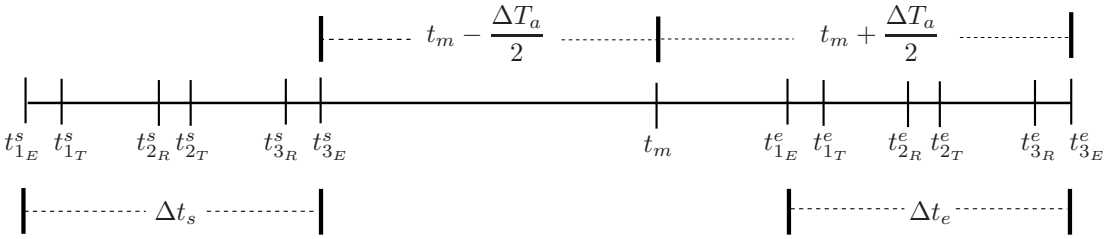


Figure 6.5: Two-Way Doppler Measurement Time Line

The computed value of the two-way average Doppler is calculated by differencing two round-trip light times. Define the starting round trip light time as Δt_s which has a time tag of $t^s_{3_E}$ and define the ending round-trip light time as Δt_e which has a time tag of $t^e_{3_E}$. The time tag, t_m, for a DSN two-way Doppler measurement is the center of the averaging interval which results in the following:

$$t^e_{3_E} = t_m + \frac{\Delta T_a}{2} \tag{6.81}$$

$$t^s_{3_E} = t_m - \frac{\Delta T_a}{2} \tag{6.82}$$

Δt_s and Δt_e are computed as described in Sec. 6.3.4. The two-way Doppler observable provided by the DSN is the negative of the classical Doppler frequency shift (see discussion in Moyer[24] pg. 13-26 and Eq. 13-47)

$$\Delta \bar{F} = \frac{M_T F_T}{\Delta T_a} \left(\Delta t_e - \Delta t_s \right) \tag{6.83}$$

where M_T is the transponder turn-around ratio and F_T is the transmitted frequency.

6.6 Computed Values of Optical Angles Observables

The computed values of optical azimuth and elevation are calculated using the one-way transit of reflected light from one participant to an optical sensor on the second participant. The mathematical model shares much in common with the one-way range model and the geometry for optical ra/dec is illustrated in Fig. (6.1).

Again, for the purposes of illustration, we have assumed that both participants are spacecraft and that their states are known – being propagated or estimated – with respect to different coordinate systems. The subsequent mathematical development is general and is valid for measurement processes that involve different participant types. Optical angles are calculated using the range vector corrected for light time delays as described in Sec. (6.2.1). To determine the angles, we first represent the range vector in the appropriate coordinate system using either Eqs. (6.13) or (6.17) which are repeated below for convenience

$$\boldsymbol{\rho}^{\mathcal{I}}_d = \mathbf{r}^{\mathcal{I}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{I}}_{a_1}(t_{1_a})$$

$$\boldsymbol{\rho}^{\mathcal{S}}_d = \mathbf{r}^{\mathcal{S}}_{a_2}(t_{2_a}) - \mathbf{r}^{\mathcal{S}}_{a_1}(t_{1_a})$$

Eq. (6.13) is used if both participants are referenced to the same celestial body and Eq. (6.17) is used otherwise. The only difference between $\mathcal{F}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{I}}$ is a translation so $\mathbf{R}^{\mathcal{O}/\mathcal{I}} = \mathbf{R}^{\mathcal{O}/\mathcal{S}}$. As a result, we can write

$$\boldsymbol{\rho}^{\mathcal{O}}_d = \mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a}) \boldsymbol{\rho}^{\mathcal{I}}_d \tag{6.84}$$

The rotation matrix $\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})$ is determined by the sensor type and type of angle measurement. If the receiving participant is a sensor on a ground station, then $\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})$ is the rotation from J2000 to station topocentric coordinates at time t_{2_a}. If the receiving participant is a sensor on a spacecraft, then $\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})$ is the rotation from inertial to body axes at time t_{2_a}.

Given the range vector, the elevation is calculated using

$$\sin(\delta) = \frac{\rho_z^{\mathcal{O}}}{\|\boldsymbol{\rho}^{\mathcal{O}}\|} \tag{6.85}$$

where $\rho_z^{\mathcal{O}}$ is the third component of $\boldsymbol{\rho}^{\mathcal{O}}$. If $\rho_x^{\mathcal{O}} \geq 1e-10$ and $\rho_y^{\mathcal{O}} \geq 1e-10$ the azimuth is calculated using

$$\tan\beta = \frac{\rho_y^{\mathcal{O}}}{-\rho_x^{\mathcal{O}}} \tag{6.86}$$

otherwise the azimuth is undefined, and the measurement is infeasible.

The elevation partial derivatives have the general form

$$\frac{\partial\delta}{\partial\chi} = \frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - \hat{\boldsymbol{\rho}}^{\mathcal{O}}\left(\hat{\boldsymbol{\rho}}^{\mathcal{O}}\right)^T\right)\frac{\partial\boldsymbol{\rho}_d^{\mathcal{O}}}{\partial\chi} \tag{6.87}$$

where $\mathbf{z} = [0 \;\; 0 \;\; 1]^T$ and χ is a dummy variable. The partial derivatives of elevation with respect to the individual solve-for parameters are:

$$\frac{\partial\delta}{\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})} = \frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - \hat{\boldsymbol{\rho}}^{\mathcal{O}}(\hat{\boldsymbol{\rho}}^{\mathcal{O}})^T\right)\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})\frac{\partial\boldsymbol{\rho}_d^{\mathcal{I}}}{\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})} \tag{6.88}$$

$$\frac{\partial\delta}{\partial\mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{2_e})} = \mathbf{0}_{1x3} \tag{6.89}$$

$$\frac{\partial\delta}{\partial\mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{2_e})} = \frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - \hat{\boldsymbol{\rho}}^{\mathcal{O}}(\hat{\boldsymbol{\rho}}^{\mathcal{O}})^T\right)\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})\frac{\partial\boldsymbol{\rho}_d^{\mathcal{I}}}{\partial\mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{2_e})} \tag{6.90}$$

$$\frac{\partial\delta}{\partial\mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{2_e})} = \mathbf{0}_{1x3} \tag{6.91}$$

$$\tag{6.92}$$

where partials of the form $\partial\boldsymbol{\rho}_d^{\mathcal{I}}/\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})$ are given in Sec. 6.2.2.

The azimuth partial derivatives have the general form

$$\frac{\partial\beta}{\partial\zeta} = \frac{1}{\sec^2\beta}\frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}}\left(\frac{\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}\hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}} - \mathbf{I}_{3x3}\right)\frac{\partial\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}}{\partial\zeta} \tag{6.93}$$

where ζ is a dummy variable, $\mathbf{x} = [1 \;\; 0 \;\; 0]^T$, and $\mathbf{y} = [0 \;\; 1 \;\; 0]^T$. The partial derivatives of the azimuth with respect to the solve-for variables are:

$$\frac{\partial\beta}{\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})} = \frac{1}{\sec^2\beta}\frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}}\left(\frac{\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}\hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}} - \mathbf{I}_{3x3}\right)\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})\frac{\partial\boldsymbol{\rho}_d^{\mathcal{I}}}{\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})} \tag{6.94}$$

$$\frac{\partial\beta}{\partial\mathbf{v}_{p_1}^{\mathcal{R}_1}(t_{2_e})} = \mathbf{0}_{1x3} \tag{6.95}$$

$$\frac{\partial\beta}{\partial\mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{2_e})} = \frac{1}{\sec^2\beta}\frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}}\left(\frac{\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}\hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T\hat{\boldsymbol{\rho}}_d^{\mathcal{O}}} - \mathbf{I}_{3x3}\right)\mathbf{R}^{\mathcal{O}/\mathcal{I}}(t_{2_a})\frac{\partial\boldsymbol{\rho}_d^{\mathcal{I}}}{\partial\mathbf{r}_{p_2}^{\mathcal{R}_2}(t_{2_e})} \tag{6.96}$$

$$\frac{\partial\beta}{\partial\mathbf{v}_{p_2}^{\mathcal{R}_2}(t_{2_e})} = \mathbf{0}_{1x3} \tag{6.97}$$

$$\frac{\partial\delta}{\partial\mathcal{O}_b} = 1 \tag{6.98}$$

where partials of the form $\partial\boldsymbol{\rho}_d^{\mathcal{I}}/\partial\mathbf{r}_{p_1}^{\mathcal{R}_1}(t_{2_e})$ are given in Sec. 6.2.2.

6.7 Geometric Measurements

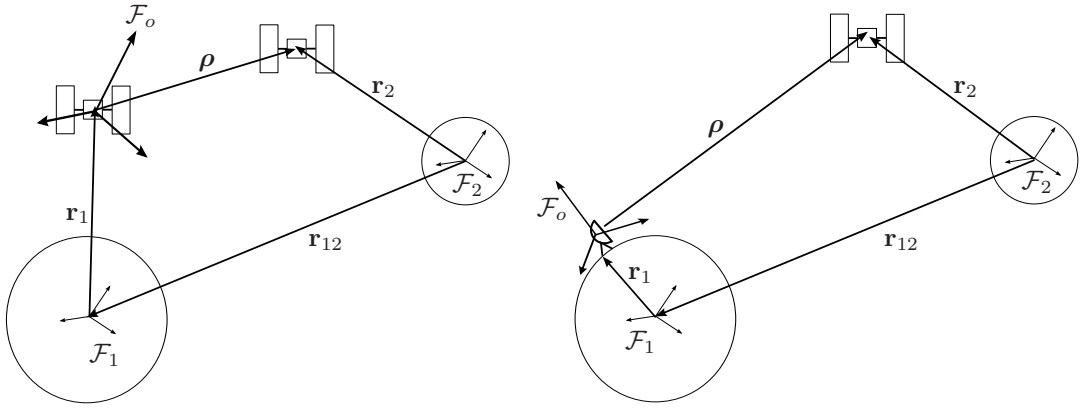
GMAT supports several geometric measurement models including range, range rate, azimuth/elevation pairs, and right ascension/declination pairs. Geometric measurement models are based purely on kinematics and do not model real-world phenomenon such as light time delay, sensor delays, or atmospheric effects. Hence, these models are primarily used in error analysis and tracking data scheduling.

The general form of the geometric measurement model is given by

$$\mathcal{O}_c = \mathbf{f}_k\left(\mathbf{r}_1(t), \dot{\mathbf{r}}_1(t), \mathbf{r}_2(t), \dot{\mathbf{r}}_2(t)\right) + \mathcal{O}_b \tag{6.99}$$

and illustrated in Fig. 6.6 where

$\mathbf{f}_k$	The kinematic model
$\mathbf{r}_1$	Location of participant 1
$\dot{\mathbf{r}}_1$	Velocity of participant 1
$\mathbf{r}_2$	Location of participant 2
$\dot{\mathbf{r}}_2$	Velocity of participant 2
$\mathbf{r}_{12}$	Vector from origin of $\mathcal{F}_2$ to origin of $\mathcal{F}_1$
t	Measurement time tag
$\mathcal{O}_b$	Measurement bias
$\mathcal{F}_1$	Coordinate system in which participant 1 is expressed
$\mathcal{F}_2$	Coordinate system in which participant 2 is expressed
$\mathcal{F}_o$	Coordinate system in which the observation is expressed
$\boldsymbol{\rho}$	Range vector
ρ	Range
$\dot{\rho}$	Range rate
β	Azimuth angle
δ	Elevation angle
α	Right ascension
δ	Declination



(a) Vehicle to Vehicle Measurement

(b) Vehicle to Station Measurement

Figure 6.6: Geometric Measurements

There are two basic types of geometric measurement illustrated in Fig. 6.6: vehicle to vehicle in figure (a), and vehicle to station in figure(b). The primary difference between those two measurement types are the coordinate systems in which the measurement values are expressed, and the coordinate systems in which the participant's states are commonly represented. For example, tracking station locations are usually expressed in body-centered body-fixed systems, while vehicle states are usually expressed in body-centered inertial.

The mathematical models presented below for geometric measurements are valid for any combination of coordinate systems of the participants and measurement values. The defaults for each measurement type are chosen according to the convention most commonly used in practice. The models ensure that the proper transformations are performed when, for example, a ground station and vehicle are not defined with respect to the same central body or axis system.

6.7.1 Geometric Range

The geometric range measurement is simply the geometric distance between the two measurement participants. Referring to Fig. 6.6, assume that the participant labeled "2" is a space vehicle, and the participant numbered "1" may be either a space vehicle or a ground station. In general, the range vector is

$$\boldsymbol{\rho} = \mathbf{r}_2 - \mathbf{r}_{21} - \mathbf{r}_1 + \mathcal{O}_b \tag{6.100}$$

The inertial representation of $\boldsymbol{\rho}$ is used to evaluate the range measurement.

$$[\boldsymbol{\rho}]_I = \mathbf{R}_{I,2}\,[\mathbf{r}_2]_2 - [\mathbf{r}_{12}]_I - \mathbf{R}_{I,1}\,[\mathbf{r}_1]_1 + \mathcal{O}_b \tag{6.101}$$

In the above equation, $\mathcal{F}_2$ is always the J2000 axes system centered at the second spacecraft's reference central body When the first participant is a ground station, $\mathcal{F}_1$ is the central-body-fixed system of the ground station's central body. When the first participant is a spacecraft, $\mathcal{F}_1$ is the J2000 axes system centered at the spacecraft's reference central body. The range measurement is computed using.

$$\rho = \| [\boldsymbol{\rho}]_I \| \tag{6.102}$$

The geometric range partial derivatives are provided below. We assume the solve for parameters are expressed in their associated reference frame (i.e. $\mathbf{r}_1$ is solved for in $\mathcal{F}_1$). Partial derivatives are provided with respect to the position and velocity of both participants as well as the measurement bias.

$$\frac{\partial \rho}{\partial [\mathbf{r}_1]_1} = -[\hat{\boldsymbol{\rho}}]_I^T\,\mathbf{R}_{I,1} \tag{6.103}$$

$$\frac{\partial \rho}{\partial [\mathbf{v}_1]_1} = \mathbf{0}_{1x3} \tag{6.104}$$

$$\frac{\partial \rho}{\partial [\mathbf{r}_2]_2} = [\hat{\boldsymbol{\rho}}]_I^T\,\mathbf{R}_{I,2} \tag{6.105}$$

$$\frac{\partial \rho}{\partial [\mathbf{v}_2]_2} = \mathbf{0}_{1x3} \tag{6.106}$$

$$\frac{\partial \rho}{\partial \mathcal{O}_b} = 1 \tag{6.107}$$

6.7.2 Geometric Range Rate

The geometric range rate measurement is the component of velocity of participant 2 with respect to participant 1, expressed in the observation coordinate system, along the range vector, also expressed in the measurement coordinate system:

$$\dot{\rho} = [\dot{\boldsymbol{\rho}}]_o^T\,[\hat{\boldsymbol{\rho}}]_o + \mathcal{O}_b \tag{6.108}$$

where $[\boldsymbol{\rho}]_o$ is given by **??** and $[\dot{\boldsymbol{\rho}}]_o$ is given by:

$$[\dot{\boldsymbol{\rho}}]_o = \mathbf{R}_{o,2}\,[\dot{\mathbf{r}}_2]_2 + \dot{\mathbf{R}}_{o,2}\,[\mathbf{r}_2]_2 - \mathbf{R}_{o,J_{2k}}\,[\dot{\mathbf{r}}_{12}]_{J_{2k}} - \dot{\mathbf{R}}_{o,J_{2k}}\,[\mathbf{r}_{12}]_{J_{2k}} - \mathbf{R}_{o1}\,[\dot{\mathbf{r}}_1]_1 - \dot{\mathbf{R}}_{o1}\,[\mathbf{r}_1]_1 + \mathcal{O}_b \tag{6.109}$$

The geometric range rate partial derivatives have the general form

$$\frac{\partial \dot{\rho}}{\partial \zeta} = \frac{[\dot{\boldsymbol{\rho}}]_o^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o\,[\hat{\boldsymbol{\rho}}]_o^T\right)\frac{\partial\,[\boldsymbol{\rho}]_o}{\partial \zeta} + [\hat{\boldsymbol{\rho}}]_o^T\,\frac{\partial\,[\dot{\boldsymbol{\rho}}]_o}{\partial \zeta} + \frac{\mathcal{O}_b}{\partial \zeta} \tag{6.110}$$

where ζ is a dummy variable. We assume the solve for parameters are expressed in their associated reference frame (i.e. $\mathbf{r}_1$ is solved for in $\mathcal{F}_1$).

$$\frac{\partial \dot{\rho}}{\partial [\mathbf{r}_1]_1} = -\frac{[\dot{\boldsymbol{\rho}}]_o^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o [\hat{\boldsymbol{\rho}}]_o^T\right)\mathbf{R}_{o,1} - [\hat{\boldsymbol{\rho}}]_o^T \dot{\mathbf{R}}_{o,1} \tag{6.111}$$

$$\frac{\partial \dot{\rho}}{\partial [\dot{\mathbf{r}}_1]_1} = -[\hat{\boldsymbol{\rho}}]_o^T \mathbf{R}_{o,1} \tag{6.112}$$

$$\frac{\partial \dot{\rho}}{\partial [\mathbf{r}_2]_2} = \frac{[\dot{\boldsymbol{\rho}}]_o^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o [\hat{\boldsymbol{\rho}}]_o^T\right)\mathbf{R}_{o,2} + [\hat{\boldsymbol{\rho}}]_o^T \dot{\mathbf{R}}_{o,2} \tag{6.113}$$

$$\frac{\partial \dot{\rho}}{\partial [\mathbf{v}_2]_2} = [\hat{\boldsymbol{\rho}}]_o^T \mathbf{R}_{o,2} \tag{6.114}$$

$$\frac{\partial \dot{\rho}}{\partial \mathcal{O}_b} = 1 \tag{6.115}$$

6.7.3 Geometric Az/El

The azimuth angle is the angle between an observer's local North and a line of sight vector to a space object. The elevation angle is a measure of the angle between the local horizon plane of the observer and the line of sight vector to a space object. *For Az/El measurements, $\mathcal{F}_o$, is the topocentric system ground station, and $\mathcal{F}_{J_{2k}}$ for inter-spacecraft measurements.* The geometric azimuth and elevation model is calculated using vector geometry and ignores signal propagation, atmospheric distortion, and other error sources.

Elevation is calculated using

$$\sin(\delta) = \frac{[\rho_z]_o}{\| [\boldsymbol{\rho}]_o \|} \tag{6.116}$$

where $[\rho]_z$ is the third component of $[\boldsymbol{\rho}]_o$ which is given by Eq. **??**. Provided that $|[\rho_x]_o| >= 1e-8$, the azimuth is calculated using

$$\tan\beta = \frac{[\rho_y]_o}{-[\rho_x]_o} \tag{6.117}$$

When $|[\rho_x]_o| < 1e-8$ the measurement is infeasible. The geometric elevation partial derivatives have the general form

$$\frac{\partial \delta}{\partial \zeta} = \frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o [\hat{\boldsymbol{\rho}}]_o^T\right)\frac{\partial [\boldsymbol{\rho}]_o}{\partial \zeta} \tag{6.118}$$

where $\mathbf{z} = [0 \;\; 0 \;\; 1]^T$ and ζ is a dummy variable. The partial derivatives of elevation with respect to the individual solve-for parameters are:

$$\frac{\partial \delta}{\partial [\mathbf{r}_1]_1} = -\frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o [\hat{\boldsymbol{\rho}}]_o^T\right)\mathbf{R}_{o,1} \tag{6.119}$$

$$\frac{\partial \delta}{\partial [\dot{\mathbf{r}}_1]_1} = \mathbf{0}_{1x3} \tag{6.120}$$

$$\frac{\partial \delta}{\partial [\mathbf{r}_2]_2} = \frac{1}{\cos\delta}\frac{\hat{\mathbf{z}}^T}{\rho}\left(\mathbf{I}_{3x3} - [\hat{\boldsymbol{\rho}}]_o [\hat{\boldsymbol{\rho}}]_o^T\right)\mathbf{R}_{o,2} \tag{6.121}$$

$$\frac{\partial \delta}{\partial [\mathbf{v}_2]_2} = \mathbf{0}_{1x3} \tag{6.122}$$

$$\frac{\partial \delta}{\partial \mathcal{O}_b} = 1 \tag{6.123}$$

The geometric azimuth partial derivatives have the general form

$$\frac{\partial \beta}{\partial \zeta} = -\frac{1}{\sec^2\beta}\frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T [\boldsymbol{\rho}]_o}\left(\mathbf{I}_{3x3} + \frac{[\boldsymbol{\rho}]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T [\boldsymbol{\rho}]_o}\right)\frac{\partial [\boldsymbol{\rho}]_o}{\partial \zeta} \tag{6.124}$$

where ζ is a dummy variable, $\mathbf{x} = [1 \quad 0 \quad 0]^T$, and $\mathbf{y} = [0 \quad 1 \quad 0]^T$. The partial derivatives of the azimuth with respect to the solve-for variables are:

$$\frac{\partial \beta}{\partial \left[\mathbf{r}_1\right]_1} = \frac{1}{\sec^2 \beta} \frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \left(\mathbf{I}_{3x3} + \frac{\left[\boldsymbol{\rho}\right]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \right) \mathbf{R}_{o,1} \tag{6.125}$$

$$\frac{\partial \beta}{\partial \left[\dot{\mathbf{r}}_1\right]_1} = \mathbf{0}_{1x3} \tag{6.126}$$

$$\frac{\partial \beta}{\partial \left[\mathbf{r}_2\right]_2} = -\frac{1}{\sec^2 \beta} \frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \left(\mathbf{I}_{3x3} + \frac{\left[\boldsymbol{\rho}\right]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \right) \mathbf{R}_{o,2} \tag{6.127}$$

$$\frac{\partial \beta}{\partial \left[\mathbf{v}_2\right]_2} = \mathbf{0}_{1x3} \tag{6.128}$$

$$\frac{\partial \delta}{\partial \mathcal{O}_b} = 1 \tag{6.129}$$

6.7.4 Geometric RA/Dec

The declination angle is the angle between the line of site vector expressed in $\mathcal{F}_o$, and the x_y plane of $\mathcal{F}_o$. The right ascension is the angle between the $\hat{\mathbf{x}}$ axis of $\mathcal{F}_o$ and the projection of the line of site vector in the $x-y$ plane of $\mathcal{F}_o$. *For RA/Dec measurements, $\mathcal{F}_o$, is the celestial body's body-fixed frame for ground station, and $\mathcal{F}_{J_{2k}}$ for inter-spacecraft measurements.*The geometric right ascension and declination are calculated using vector geometry and signal propagation, atmospheric distortion, and other error sources are not included.

The declination is calculated using

$$\sin(\delta) = \frac{\left[\rho_z\right]_o}{\| \left[\boldsymbol{\rho}\right]_o \|} \tag{6.130}$$

where $[\rho]_z$ is the third component of $[\boldsymbol{\rho}]_o$ which is given by Eq. **??**. Provided that $|[\rho_x]_o| >= 1e-8$, the right ascension is calculated using

$$\tan \alpha = \frac{\left[\rho_y\right]_o}{-\left[\rho_x\right]_o} \tag{6.131}$$

When $|[\rho_x]_o| < 1e-8$, the measurement is infeasible.

The geometric declination partial derivatives have the general form

$$\frac{\partial \delta}{\partial \zeta} = \frac{1}{\cos \delta} \frac{\hat{\mathbf{z}}^T}{\rho} \left(\mathbf{I}_{3x3} - \left[\hat{\boldsymbol{\rho}}\right]_o \left[\hat{\boldsymbol{\rho}}\right]_o^T \right) \frac{\partial \left[\boldsymbol{\rho}\right]_o}{\partial \zeta} \tag{6.132}$$

where $\mathbf{z} = [0 \quad 0 \quad 1]^T$ and ζ is a dummy variable. The partial derivatives of declination with respect to the individual solve-for parameters are:

$$\frac{\partial \delta}{\partial \left[\mathbf{r}_1\right]_1} = -\frac{1}{\cos \delta} \frac{\hat{\mathbf{z}}^T}{\rho} \left(\mathbf{I}_{3x3} - \left[\hat{\boldsymbol{\rho}}\right]_o \left[\hat{\boldsymbol{\rho}}\right]_o^T \right) \mathbf{R}_{o,1} \tag{6.133}$$

$$\frac{\partial \delta}{\partial \left[\dot{\mathbf{r}}_1\right]_1} = \mathbf{0}_{1x3} \tag{6.134}$$

$$\frac{\partial \delta}{\partial \left[\mathbf{r}_2\right]_2} = \frac{1}{\cos \delta} \frac{\hat{\mathbf{z}}^T}{\rho} \left(\mathbf{I}_{3x3} - \left[\hat{\boldsymbol{\rho}}\right]_o \left[\hat{\boldsymbol{\rho}}\right]_o^T \right) \mathbf{R}_{o,2} \tag{6.135}$$

$$\frac{\partial \delta}{\partial \left[\mathbf{v}_2\right]_2} = \mathbf{0}_{1x3} \tag{6.136}$$

$$\frac{\partial \delta}{\partial \mathcal{O}_b} = 1 \tag{6.137}$$

The geometric right ascension partial derivatives have the general form

$$\frac{\partial \alpha}{\partial \zeta} = -\frac{1}{\sec^2 \alpha} \frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \left(\mathbf{I}_{3x3} + \frac{\left[\boldsymbol{\rho}\right]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \right) \frac{\partial \left[\boldsymbol{\rho}\right]_o}{\partial \zeta} \tag{6.138}$$

where ζ is a dummy variable, $\mathbf{x} = [1 \;\; 0 \;\; 0]^T$, and $\mathbf{y} = [0 \;\; 1 \;\; 0]^T$. The partial derivatives of the right ascension with respect to the solve-for variables are:

$$\frac{\partial \alpha}{\partial \left[\mathbf{r}_1\right]_1} = \frac{1}{\sec^2 \alpha} \frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \left(\mathbf{I}_{3x3} + \frac{\left[\boldsymbol{\rho}\right]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \right) \mathbf{R}_{o,1} \tag{6.139}$$

$$\frac{\partial \alpha}{\partial \left[\dot{\mathbf{r}}_1\right]_1} = \mathbf{0}_{1x3} \tag{6.140}$$

$$\frac{\partial \alpha}{\partial \left[\mathbf{r}_2\right]_2} = -\frac{1}{\sec^2 \alpha} \frac{\hat{\mathbf{y}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \left(\mathbf{I}_{3x3} + \frac{\left[\boldsymbol{\rho}\right]_o \hat{\mathbf{x}}^T}{\hat{\mathbf{x}}^T \left[\boldsymbol{\rho}\right]_o} \right) \mathbf{R}_{o,2} \tag{6.141}$$

$$\frac{\partial \alpha}{\partial \left[\mathbf{v}_2\right]_2} = \mathbf{0}_{1x3} \tag{6.142}$$

$$\frac{\partial \delta}{\partial \mathcal{O}_b} = 1 \tag{6.143}$$

6.8 Measurement Error Modeling

The general model of a measurement error is as follows:

$$e = b + v \tag{6.144}$$

where b models the systematic errors, and v models the measurement noise. We assume that the measurement noise is a discrete sequence of uncorrelated random numbers. Variables such as v are known as random variables, and the next subsection describes how to model them. Subsequent subsections describe models for the systematic errors.

The discussion of systematic errors treats such errors as scalar quantities to simplify the exposition; generalization to the vector case is straightforward. Note that if the measurement is non-scalar, but the errors in the component measurements are independent of one another, then we can model each measurement independently, so modeling the biases as vector is not required. If the measurement errors are not independent, then many estimators require that we apply a transformation to the data prior to processing so that the data input to the estimator have independent measurement errors; the next subsection describes some ways to accomplish this transformation.

6.8.1 Models and Realizations of Random Variables

A continuous random variable is a function that maps the outcomes of random events to the real line. Realizations of random variables are thus real numbers. A vector of n random variables maps outcomes of random events to $\mathcal{R}^n$. For our purposes, random variables will always be associated with a probability density function that indicates the likelihood that a realization occurs within a particular interval of the real line, or within a particular subspace of $\mathcal{R}^n$ for the vector case. Currently, all of our models assume that this density is the normal or Gaussian density. For the vector case, the normal probability density function is

$$f(x) = \frac{1}{2\pi|P|}\mathrm{e}^{-\frac{1}{2}(x-\mu)'P^{-1}(x-\mu)} \tag{6.145}$$

where μ is a vector of mean values for each component of x, and P is a matrix that contains the variances of each component of x along its diagonal, and the covariances between each component as its off-diagonal components. The covariances indicate the degree of correlation between the random variables composing x. The matrix P is thus called the variance-covariance matrix, which we will hereafter abbreviate to just "covariance matrix," or "covariance." Since the normal density is completely characterized by its mean and covariance, we will use the following notation as a shorthand to describe normally-distributed random vectors:

$$x \sim N(\mu, P) \tag{6.146}$$

Thus, the model for the measurement noise is

$$v \sim N(0, R) \tag{6.147}$$

For the scalar case, or for the vector case when the covariance is diagonal, we may directly generate realizations of a normally-distributed random vector from normal random number generators available in most software libraries. If P has non-zero off-diagonal elements, we must model the specified correlations when we generate realizations. If P is strictly positive definite, we can factor it as follows:

$$P = SS' \tag{6.148}$$

where S is a triangular matrix known as a Cholesky factor; this can be viewed as a "matrix square root." The Cholesky factorization is available in many linear algebra libraries. We can then use S to generate correlated realizations of x as follows. Let z be a normally-distributed random vector of the same dimension as x, with zero mean and unit variance, that is

$$z \sim N(0, I) \tag{6.149}$$

Then, with

$$x = Sz \tag{6.150}$$

we can generate properly correlated realizations of x. We can also use a Cholesky factorization of the measurement noise covariance R, if R is non-diagonal, to transform correlated measurements into uncorrelated auxiliary measurements for cases in which the estimator cannot handle correlated measurement data.

If P is only non-negative definite, i.e. $P \geq 0$ rather than $P > 0$ as above, the Cholesky factorization does not exist. In this case, since P's eigenvalues are real and distinct, it has a diagonal factorization:

$$P = VDV' \tag{6.151}$$

where V is a matrix of eigenvectors and D is a diagonal matrix of eigenvalues. Then, with z as above,

$$x = V\sqrt{D}z \tag{6.152}$$

where $\sqrt{D}$ implies taking the square roots of each diagonal element.

6.8.2 Zero-Input Bias State Models

The simplest non-zero measurement error consists only of measurement noise. The next simplest class of measurement errors consists of biases which are either themselves constant, or are the integrals of constants. We can view such biases as the output of a system which has zero inputs, and which may have internal states. In the sequel, we will consider cases where there are random inputs to the system.

In cases were the bias is the output of a system with internal states, the estimator may treat the internal states as solve-for or consider parameters. In such cases, the estimator requires a measurement partials matrix. Otherwise, the "measurement partial" is just $H = \partial b/\partial b = 1$.

Random Constant

The simplest type of systematic error is a constant bias on the measurement. There are two types of such biases: deterministic constants, which are truly constant for all time, and random constants, which are constant or very nearly so over a particular time of interest. For example, each time a sensor is power-cycled, a bias associated with it may change in value, but so long as the sensor remains powered on, the bias will not change.

In some cases, we may have reason to believe that a particular systematic error source truly is a deterministic bias, but due to limited observability, we do not have knowledge of its true value. In such cases, we may view our estimate of the bias as a random constant, and its variance as a measure of the imprecision of our knowledge.

Thus, we may view all constants that could be solve-for or consider parameters in orbit determination as random constants. Our model for a random constant is

$$\dot{b}(t) = 0,\ b(t_o) \sim N(0, p_{bo}). \tag{6.153}$$

Since $b(t)$ is a zero-mean constant, its mean is zero for all time, and its covariance is constant for all time as well. Thus, to simulate a realization of the random constant, we need only generate a random number according to $N(0, p_{bo})$, as the previous subsection described.

Random Ramp

The random ramp model assumes that the rate of change of the bias is itself a random constant; thus the random ramp model is

$$\ddot{b}(t) = 0,\ \dot{b}(t_o) \sim N(0, p_{\dot{b}o}). \tag{6.154}$$

Thus, the initial condition $\dot{b}(t_o)$ is a random constant. For a pure random ramp, the initial condition on $b(t_o)$ and its covariance are taken to be zero, but an obvious and common generalization is to allow $b(t_o)$ to also be a random constant.

It is convenient to write this model as a first-order vector system as follows:

$$\begin{bmatrix}\dot{b}(t)\\ \ddot{b}(t)\end{bmatrix} = \begin{bmatrix}\dot{b}(t)\\ \dot{d}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}b(t)\\ d(t)\end{bmatrix} \tag{6.155}$$

$$\dot{x}(t) = A(t)x(t) \tag{6.156}$$

The resulting output equation is

$$e = \begin{bmatrix}1 & 0\end{bmatrix}x + v \tag{6.157}$$

$$= Hx + v \tag{6.158}$$

Note that the ensemble of realizations of $x(t)$ has zero-mean for all time. The covariance evolves in time according to

$$P_x(t) = \Phi(t - t_o)P_{xo}\Phi'(t - t_o) \tag{6.159}$$

where

$$\Phi(t) = \begin{bmatrix}1 & t\\ 0 & 1\end{bmatrix} \text{ and } P_{xo} = \begin{bmatrix}p_{bo} & 0\\ 0 & p_{\dot{b}o}\end{bmatrix} \tag{6.160}$$

which we can also write in recursive form as

$$P_x(t + \Delta t) = \Phi(\Delta t)P_x(t)\Phi'(\Delta t) \tag{6.161}$$

Thus, we can generate realizations of the random ramp with either $x(t) \sim N(0, P_x(t))$ or recursively from

$$x(t + \Delta t) = \Phi(\Delta t)x(t) \tag{6.162}$$

Note that $\|P_x\|$ becomes infinite as t^2 becomes infinite. This could lead to an overflow of the representation of P_x in a computer program if the propagation time is large, and could also lead to the representation of P_x losing either its symmetry and/or its positive definiteness due to roundoff and/or truncation.

Higher-Order Derivatives of Random Constants

In principle, a random constant may be associated with any derivative of the bias in a straightforward extension of the models above. In practice, it is rare to need more than two derivatives. Conventional terminology does not appear in the literature for derivatives of higher order than the random ramp. The slope of the bias is most commonly described as the "bias drift," so that a "drift random ramp" would be one way to describe a bias whose second derivative is a random constant. The measurement partials matrix needs to be accordingly padded with trailing zeros for the derivatives of the bias in such cases.

6.8.3 Single-Input Bias State Models

The simplest non-constant systematic errors are systems with a single input that is a random process. We can think of a random process as the result of some kind of limit in which the intervals between an uncorrelated sequence of random variables get infinitesimally small. In this limit, each random increment instantaneously perturbs the sequence, so that the resulting process is continuous but non-differentiable. We call this kind of a random input "process noise."

Although such random processes are non-differentiable, there are various techniques for generalizing the concept of integration so that something like integrals of the process noise exist, and hence so do the differentials that appear under the integral signs. It turns out that so long as any coefficients of the process noise are non-random, these differentials behave for all practical purposes as if they were differentiable.

Random Walk

The random walk is the simplest random process of the type described above. In terms of the "formal derivatives" mentioned above, the random walk model for a measurement bias is

$$\dot{b}(t) = w(t),\ w(t) \sim N(0, q\delta(t - s)) \tag{6.163}$$

The input noise process on the right hand side is known as "white noise," and the Dirac delta function that appears in the expression for its variance indicates that the white noise process consists of something like an infinitely-tightly spaced set of impulses. The term q that appears along with the delta function is the intensity of each impulse[1]. The initial condition $b(t_o)$ is an unbiased random constant. Since $b(t_o)$ and $w(t)$ are zero-mean, then $b(t)$ is also zero-mean for all time. The variance of b evolves in time according to

$$p_b(t) = p_{bo} + q(t - t_o) \tag{6.164}$$

which we can also write in recursive form as

$$p_b(t + \Delta t) = p_b(t) + q\Delta t \tag{6.165}$$

Thus, to generate a realization of the random walk at time t, we need only generate a random number according to $N(0, p_b(t))$. Equivalently, we could also generate realizations of $w_\Delta(t) \sim N(0, q\Delta t)$, and recursively add these discrete noise increments to the bias as follows:

$$b(t + \Delta t) = b(t) + w_\Delta(t) \tag{6.166}$$

Note that p_b becomes infinite as t becomes infinite. This could lead to an overflow of the representation of p_b in a computer program if both the propagation time and q are large.

Random Run

The random run model assumes that the rate of change of the bias is itself a random walk; thus the random run model is

$$\ddot{b}(t) = w(t),\ w(t) \sim N(0, q\delta(t - s)) \tag{6.167}$$

The initial condition $\dot{b}(t_o)$ is a random constant. For a pure random run, the initial condition on $b(t_o)$ and its covariance are taken to be zero, but an obvious and common generalization is to allow $b(t_o)$ to also be a random constant.

It is convenient to write this model as a first-order vector system as follows:

$$\begin{bmatrix} \dot{b}(t) \\ \ddot{b}(t) \end{bmatrix} = \begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b(t) \\ d(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \tag{6.168}$$

$$\dot{x}(t) = A(t)x(t) + b(t)w(t) \tag{6.169}$$

The measurement partial is the same as for the random ramp. The initial condition $x(t_o)$ is an unbiased random constant. Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time. The covariance evolves in time according to

$$P_x(t) = \Phi(t - t_o)P_{xo}\Phi'(t - t_o) + Q_\Delta(t - t_o) \tag{6.170}$$

where

$$\Phi(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \text{ and } P_{xo} = \begin{bmatrix} p_{bo} & 0 \\ 0 & p_{\dot{b}o} \end{bmatrix} \tag{6.171}$$

and

$$Q_\Delta(t) = q \begin{bmatrix} t^3/3 & t^2/2 \\ t^2/2 & t \end{bmatrix} \tag{6.172}$$

which we can also write in recursive form as

$$P_x(t + \Delta t) = \Phi(\Delta t)P_x(t)\Phi'(\Delta t) + Q_\Delta(\Delta t) \tag{6.173}$$

Thus, we can generate realizations of the random run with either $x(t) \sim N(0, P_x(t))$ or recursively from

$$x(t + \Delta t) = \Phi(\Delta t)x(t) + w_\Delta(t) \tag{6.174}$$

[1]Another way to imagine the input sequence, in terms of a frequency domain interpretation, is that it is a noise process whose power spectral density, q, is non-zero at all frequencies, which implies infinite bandwidth.

where $w_\Delta(t) \sim N(0, Q_\Delta(\Delta t))$. Note that a Cholesky decomposition of $Q_\Delta(t)$ is

$$\sqrt[C]{Q_\Delta(t)} = \begin{bmatrix} \sqrt{3t^3}/3 & 0 \\ \sqrt{3t}/2 & \sqrt{t}/2 \end{bmatrix} \tag{6.175}$$

Note that $\|P_x\|$ becomes infinite as t^3 becomes infinite. This could lead to an overflow of the representation of P_x in a computer program if both the propagation time and q are large, and could also lead to the representation of P_x losing either its symmetry and/or its positive definiteness due to roundoff and/or truncation.

Higher-Order Derivatives of Random Walks

In principle, a random walk may be associated with any derivative of the bias in a straightforward extension of the models above. In practice, it is rare to need more than two derivatives. Conventional terminology does not appear in the literature for derivatives of higher order than the random run. A "drift random run" would be one way to describe a bias whose second derivative is a random walk. Below, we will refer to such a model as a "random zoom."

First-Order Gauss-Markov

The first-order Gauss-Markov (FOGM) process is one of the simplest random processes that introduces time correlation between samples. In terms of a frequency domain interpretation, we can view it as white noise passed through a low-pass filter. Since such noise, often called "colored noise," has finite bandwidth, it is physically realizable, unlike white noise. In the notation of formal derivatives, the FOGM model is

$$\dot{b}(t) = -\frac{1}{\tau} b(t) + w(t), \tag{6.176}$$

where, as with the random walk, $b(t_o) \sim N(0, P_{bo})$, and $w(t) \sim N(0, q\delta(t-s))$. The time constant, τ, also known as the "half-life," gives the correlation time, or the time over which the intensity of the time correlation will fade to half its value.

Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time. The covariance evolves in time according to

$$p_b(t) = \mathrm{e}^{-\frac{2}{\tau}(t-t_o)} p_{bo} + q_\Delta(t - t_o) \tag{6.177}$$

where

$$q_\Delta(t - t_o) = \frac{q\tau}{2}\left(1 - \mathrm{e}^{-\frac{2}{\tau}(t-t_o)}\right) \tag{6.178}$$

which we can also write in recursive form as

$$p_b(t + \Delta t) = \mathrm{e}^{-\frac{2\Delta t}{\tau}} p_b(t) + q_\Delta(\Delta t) \tag{6.179}$$

Thus, to generate a realization of the random walk at time t, we need only generate a random number according to $N(0, p_b(t))$. Equivalently, we could also generate realizations of $w_\Delta(t) \sim N(0, q_\Delta(\Delta t))$, and recursively add these discrete noise increments to the bias as follows:

$$b(t + \Delta t) = \mathrm{e}^{-\frac{\Delta t}{\tau}} b(t) + w_\Delta(t) \tag{6.180}$$

Note that p_b approaches a finite steady-state value of $q\tau/2$ as t becomes infinite. We can choose the parameters of the FOGM so that this steady-state value avoids any overflow of the representation of p_b in a computer program.

Integrated First-Order Gauss-Markov Model

As with the random walk and random constant models, any number of derivatives of the bias may be associated with a FOGM process. However, integation of the FOGM destroys its stability. For example, the singly integrated first-order Gauss-Markov model is given by

$$\begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix} \begin{bmatrix} b(t) \\ d(t) \end{bmatrix} + \begin{bmatrix} 0 \\ w(t) \end{bmatrix}, \tag{6.181}$$

which leads to the following state transition matrix,

$$\Phi(t) = \begin{bmatrix} 1 & \tau\left(1-e^{-t/\tau}\right) \\ 0 & e^{-t/\tau} \end{bmatrix}, \tag{6.182}$$

and process noise covariance,

$$Q_\Delta(t) = \frac{q\tau}{2}\begin{bmatrix} \tau^2\left\{\left(1-e^{-2t/\tau}\right)^2 + 2t/\tau + 4\left(1-e^{-t/\tau}\right)\right\} & \tau\left(1-e^{-t/\tau}\right)^2 \\ \tau\left(1-e^{-t/\tau}\right)^2 & \left(1-e^{-2t/\tau}\right) \end{bmatrix}. \tag{6.183}$$

Clearly, this is an unstable model, as the bias variance increases linearly with elapsed time. If a Gauss-Markov model is desired because of its stability properties, the following second-order model is available.

Second-Order Gauss-Markov

The model for a second-order Gauss-Markov random process is

$$\ddot{b}(t) = -2\zeta\omega_n\dot{b}(t) - \omega_n^2 b(t) + w(t),\ w(t) \sim N(0, q\delta(t-s)) \tag{6.184}$$

The initial conditions $b(t_o)$ and $\dot{b}(t_o)$ are random constants. It is convenient to write this model as a first-order vector system as follows:

$$\begin{bmatrix} \dot{b}(t) \\ \ddot{b}(t) \end{bmatrix} = \begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}\begin{bmatrix} b(t) \\ d(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \tag{6.185}$$

$$\dot{x}(t) = A(t)x(t) + b(t)w(t) \tag{6.186}$$

The measurement partial is the same as for the random ramp. The initial condition $x(t_o)$ is an unbiased random constant. Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time.

The covariance evolves in time according to according to

$$P_x(t) = \Phi(t-t_o)P_{xo}\Phi'(t-t_o) + Q_\Delta(t-t_o) \tag{6.187}$$

which we can also write in recursive form as

$$P_x(t+\Delta t) = \Phi(\Delta t)P_x(t)\Phi'(\Delta t) + Q_\Delta(\Delta t) \tag{6.188}$$

Thus, we can generate realizations of the random run with either $x(t) \sim N(0, P_x(t))$ or recursively from

$$x(t+\Delta t) = \Phi(\Delta t)x(t) + w_\Delta(t) \tag{6.189}$$

where $w_\Delta(t) \sim N(0, Q_\Delta(\Delta t))$.

For the underdamped case ($\zeta < 1$), the state transition matrix and discrete process noise covariance are given by[2]:

$$\Phi(t) = \frac{\mathrm{e}^{-\zeta\omega_n t}}{w_d}\begin{bmatrix} (\omega_d\cos\omega_d t + \zeta\omega_n\sin\omega_d t) & \sin\omega_d t \\ -\omega_n^2\sin\omega_d t & (\omega_d\cos\omega_d t - \zeta\omega_n\sin\omega_d t) \end{bmatrix} \tag{6.190}$$

and

$$Q_\Delta^{(1,1)}(t) = \frac{q}{4\zeta\omega_n^3}\left[1 - \frac{\mathrm{e}^{-2\zeta\omega_n t}}{w_d^2}(\omega_d^2 + 2\zeta\omega_n\omega_d\cos\omega_d t\sin\omega_d t + 2\zeta^2\omega_n^2\sin^2\omega_d t)\right] \tag{6.191}$$

$$Q_\Delta^{(2,2)}(t) = \frac{q}{4\zeta\omega_n}\left[1 - \frac{\mathrm{e}^{-2\zeta\omega_n t}}{w_d^2}(\omega_d^2 - 2\zeta\omega_n\omega_d\cos\omega_d t\sin\omega_d t + 2\zeta^2\omega_n^2\sin^2\omega_d t)\right] \tag{6.192}$$

$$Q_\Delta^{(2,1)}(t) = Q_\Delta^{(1,2)}(t) = \frac{q}{2\omega_d^2}\mathrm{e}^{-2\zeta\omega_n t}\sin^2\omega_d t \tag{6.193}$$

[2] M. C. Wang and G. E. Uhlenbeck. On the theory of brownian motion ii. In N. Wax, editor, *Selected Papers on Noise and Stochastic Processes*, pages 113–132. Dover, 1954.

where $\omega_d = \omega_n\sqrt{1-\zeta^2}$. In the over-damped case ($\zeta > 1$), replace sin and cos with sinh and cosh, respectively. In the critically-damped case,

$$\Phi(t) = \begin{bmatrix} \mathrm{e}^{-\omega_n t}(1+\omega_n t) & t\mathrm{e}^{-\omega_n t} \\ -\omega_n^2 t\mathrm{e}^{-\omega_n t} & \mathrm{e}^{-\omega_n t}(1-\omega_n t) \end{bmatrix} \tag{6.194}$$

and

$$Q_\Delta^{(1,1)}(t) = \frac{q}{4\omega_n^3}\left[1-\mathrm{e}^{-2\omega_n t}(1+2\omega_n t+2\omega_n^2 t^2)\right] \tag{6.195}$$

$$Q_\Delta^{(2,2)}(t) = \frac{q}{4\omega_n}\left[1-\mathrm{e}^{-2\omega_n t}(1-2\omega_n t+2\omega_n^2 t^2)\right] \tag{6.196}$$

$$Q_\Delta^{(2,1)}(t) = Q_\Delta^{(1,2)}(t) = \frac{qt^2}{2}\mathrm{e}^{-2\omega_n t} \tag{6.197}$$

Note that for any damping ratio, $\|P_x\|$ remains finite, since as $t \to \infty$,

$$P_x(t\to\infty) = \frac{q}{4\zeta\omega_n}\begin{bmatrix} 1/\omega_n^2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.198}$$

Thus, the ratio of the steady-state standard deviations of x and $\dot{x}$ will be

$$\frac{\sigma_d}{\sigma_b} = \omega_n, \tag{6.199}$$

and these are related to the power spectral density by

$$q = 4\zeta\frac{\sigma_d^3}{\sigma_b}. \tag{6.200}$$

Hence, we can choose the parameters of the SOGM so that we avoid any overflow, loss of symmetry and/or positive definiteness of P_x due to roundoff and/or truncation.

6.8.4 Multi-Input Bias State Models

We may combine any of the above models to create multi-input bias models; for example the bias could be a second-order Gauss-Markov, and the bias rate could be a first-order Gauss-Markov. In practice, the most useful have been found to be the following.

Bias and Drift Random Walks (Random Walk + Random Run)

A common model for biases in clocks, gyros, and accelerometers is that the bias is driven by both its own white noise input, and also by the integral of the white noise of its drift. Such models derive from observations that the error magnitudes of these devices depend on the time scale over which the device is observed. They are often characterized by Allan deviation specifications, which may be heuristically associated with the white noise power spectral densities. The model is as follows:

$$\begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} b(t) \\ d(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} w_b(t) \\ w_d(t) \end{bmatrix} \tag{6.201}$$

$$\dot{x}(t) = A(t)x(t) + B(t)w(t) \tag{6.202}$$

The measurement partial is the same as for the random ramp. The initial condition $x(t_o)$ is an unbiased random constant. Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time. The covariance evolves in time according to

$$P_x(t) = \Phi(t-t_o)P_{xo}\Phi'(t-t_o) + Q_\Delta(t-t_o) \tag{6.203}$$

where

$$\Phi(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \text{ and } P_{xo} = \begin{bmatrix} p_{bo} & 0 \\ 0 & p_{\dot{b}o} \end{bmatrix} \tag{6.204}$$

and

$$Q_\Delta(t) = \begin{bmatrix} q_b t + q_d t^3/3 & q_d t^2/2 \\ q_d t^2/2 & q_d t \end{bmatrix} \tag{6.205}$$

which we can also write in recursive form as

$$P_x(t+\Delta t) = \Phi(\Delta t) P_x(t) \Phi'(\Delta t) + Q_\Delta(\Delta t) \tag{6.206}$$

Thus, we can generate realizations of the random run with either $x(t) \sim N(0, P_x(t))$ or recursively from

$$x(t+\Delta t) = \Phi(\Delta t) x(t) + w_\Delta(t) \tag{6.207}$$

where $w_\Delta(t) \sim N(0, Q_\Delta(\Delta t))$. Note that a Cholesky decomposition of $Q_\Delta(t)$ is

$$\overset{C}{\sqrt{Q_\Delta(t)}} = \begin{bmatrix} \sqrt{q_b t + q_d t^3/12} & 0 \\ \sqrt{q_d t^3}/2 & \sqrt{q_d t} \end{bmatrix} \tag{6.208}$$

Note that $\|P_x\|$ becomes infinite as t^3 becomes infinite. This could lead to an overflow of the representation of P_x in a computer program if both the propagation time and q are large, and could also lead to the representation of P_x losing either its symmetry and/or its positive definiteness due to roundoff and/or truncation.

Bias, Drift, and Drift Rate Random Walks (Random Walk + Random Run + Random Zoom)

Another model for biases in very-high precision clocks, gyros, and accelerometers is that the bias is driven by two integrals of white noise in addition to its own white noise input. Such models are often characterized by Hadamard deviation specifications, which may be heuristically associated with the white noise power spectral densities. The model is as follows:

$$\begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \\ \ddot{d}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b(t) \\ d(t) \\ \dot{d}(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_b(t) \\ w_d(t) \\ w_{\dot{d}}(t) \end{bmatrix} \tag{6.209}$$

$$\dot{x}(t) = A(t)x(t) + B(t)w(t) \tag{6.210}$$

The resulting output equation is

$$e = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x + v \tag{6.211}$$

$$= Hx + v \tag{6.212}$$

The initial condition $x(t_o)$ is an unbiased random constant. Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time. The covariance evolves in time according to

$$P_x(t) = \Phi(t - t_o) P_{xo} \Phi'(t - t_o) + Q_\Delta(t - t_o) \tag{6.213}$$

where

$$\Phi(t) = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \text{ and } P_{xo} = \begin{bmatrix} p_{bo} & 0 & 0 \\ 0 & p_{do} & 0 \\ 0 & 0 & p_{\dot{d}o} \end{bmatrix} \tag{6.214}$$

and

$$Q_\Delta(t) = \begin{bmatrix} q_b t + q_d t^3/3 + q_{\dot{d}} t^5/5 & q_d t^2/2 + q_{\dot{d}} t^4/8 & q_{\dot{d}} t^3/6 \\ q_d t^2/2 + q_{\dot{d}} t^4/8 & q_d t + q_{\dot{d}} t^3/3 & q_{\dot{d}} t^2/2 \\ q_{\dot{d}} t^3/6 & q_{\dot{d}} t^2/2 & q_{\dot{d}} t \end{bmatrix} \tag{6.215}$$

which we can also write in recursive form as

$$P_x(t+\Delta t) = \Phi(\Delta t) P_x(t) \Phi'(\Delta t) + Q_\Delta(\Delta t) \tag{6.216}$$

Thus, we can generate realizations of the random run with either $x(t) \sim N(0, P_x(t))$ or recursively from

$$x(t + \Delta t) = \Phi(\Delta t)x(t) + w_\Delta(t) \tag{6.217}$$

where $w_\Delta(t) \sim N(0, Q_\Delta(\Delta t))$. Note that a Cholesky decomposition of $Q_\Delta(t)$ is

$$\sqrt[C]{Q_\Delta(t)} = \begin{bmatrix} \sqrt{q_b t + q_d t^3/12 + q_d t^5/720} & 0 & 0 \\ t/2\sqrt{q_d t + q_d t^3/12} & \sqrt{q_d t + q_d t^3/12} & 0 \\ t^2/6\sqrt{q_d t} & t/2\sqrt{q_d t} & \sqrt{q_d t} \end{bmatrix} \tag{6.218}$$

Note that $\|P_x\|$ becomes infinite as t^3 becomes infinite. This could lead to an overflow of the representation of P_x in a computer program if both the propagation time and q are large, and could also lead to the representation of P_x losing either its symmetry and/or its positive definiteness due to roundoff and/or truncation.

Bias and Drift Coupled First- and Second-Order Gauss-Markov

The following model provides a stable alternative to the "Random Walk + Random Run" model[3]. Its transient response can be tuned to approximate the Random Walk + Random Run model, and its stable steady-state response can be used to avoid computational issues with long propagation times. The model is as follows.

$$\begin{bmatrix} \dot{b}(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} -1/\tau & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} b(t) \\ d(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_b(t) \\ w_d(t) \end{bmatrix} \tag{6.219}$$

$$\dot{x}(t) = A(t)x(t) + B(t)w(t) \tag{6.220}$$

The measurement partial is the same as for the random ramp. The initial condition $x(t_o)$ is an unbiased random constant. Since $x(t_o)$ and $w(t)$ are zero-mean, then $x(t)$ is also zero-mean for all time. The covariance evolves in time according to

$$P_x(t) = \Phi(t - t_o)P_{xo}\Phi'(t - t_o) + Q_\Delta(t - t_o) \tag{6.221}$$

where

$$\Phi(t) = \frac{e^{at}}{b} \begin{bmatrix} b\cos bt + (a + 2\zeta\omega_n)\sin bt & \sin bt \\ -\omega_n^2 \sin bt & b\cos bt + (a + \beta)\sin bt \end{bmatrix} \tag{6.222}$$

where

$$\beta = 1/\tau, \tag{6.223}$$

$$a = -\frac{1}{2}\left(\beta + 2\zeta\omega_n\right), \tag{6.224}$$

$$b = \sqrt{\omega_d^2 + \beta\zeta\omega_n - \frac{1}{4}\beta^2}, \tag{6.225}$$

$$\omega_d = \omega_n\sqrt{1 - \zeta^2}, \tag{6.226}$$

and we assume that $b^2 > 0$. Let

$$c = -\frac{\beta}{2} + \zeta\omega_n;$$

[3] R. Carpenter and T. Lee. A stable clock error model using coupled first- and second-order gauss-markov processes. In *Astrodynamics 2008*, Advances in the Astronautical Sciences. Univelt, 2008.

then, the process noise covariance is given by the following:

$$\begin{aligned} Q_{\Delta}^{(1,1)}(t) &= q_b \left[\frac{e^{2at}-1}{4a}\left(1+\frac{c^2}{b^2}\right) + \frac{e^{2at}\sin 2bt}{4(a^2+b^2)}\left(\frac{b^2-c^2+2ac}{b}\right) \right. \\ &\quad \left. + \frac{e^{2at}\cos 2bt - 1}{4(a^2+b^2)}\left(\frac{ab^2-ac^2+2b^2c}{b^2}\right)\right] \\ &\quad + \frac{q_d}{b^2}\left(\frac{e^{2at}-1}{4a} - \frac{e^{2at}(b\sin 2bt + a\cos 2bt) - a}{4(a^2+b^2)}\right) \end{aligned} \tag{6.227}$$

$$\begin{aligned} Q_{\Delta}^{(2,2)}(t) &= q_d \left[\frac{e^{2at}-1}{4a}\left(1+\frac{c^2}{b^2}\right) + \frac{e^{2at}\sin 2bt}{4(a^2+b^2)}\left(\frac{b^2-c^2+2ac}{b}\right) \right. \\ &\quad \left. + \frac{e^{2at}\cos 2bt - 1}{4(a^2+b^2)}\left(\frac{ab^2-ac^2+2b^2c}{b^2}\right)\right] \\ &\quad + \frac{q_b\omega_n^4}{b^2}\left(\frac{e^{2at}-1}{4a} - \frac{e^{2at}(b\sin 2bt + a\cos 2bt) - a}{4(a^2+b^2)}\right) \end{aligned} \tag{6.228}$$

$$\begin{aligned} Q_{\Delta}^{(1,2)}(t) &= \frac{q_b\omega_n^2}{b^2}\left[\frac{c}{4a}\left(1-e^{2at}\right) + \frac{e^{2at}\left[(bc-ab)\sin 2bt + (ac-b^2)\cos 2bt\right] - (ac-b^2)}{4(a^2+b^2)}\right] \\ &\quad + \frac{q_d}{b^2}\left[\frac{c}{4a}\left(1-e^{2at}\right) + \frac{e^{2at}\left[(ab+bc)\sin 2bt + (ac-b^2)\cos 2bt\right] - (ac-b^2)}{4(a^2+b^2)}\right]. \end{aligned} \tag{6.229}$$

Examining the solution given above, we see that the parameter a governs the rate of decay of all of the exponential terms. Therefore, we define the "rise time" as that interval within which the transient response of the covariance will reach a close approximation to the above steady-state value; thus, we define the rise time as follows:

$$t_r = -\frac{3}{a}. \tag{6.230}$$

Next, we note that all of the trigonometric terms are modulated by $2b$; thus we may view this value as a characteristic damped frequency of the coupled system. The period of the oscillation, Π, is then

$$\Pi = \pi/b \tag{6.231}$$

In the limit as $t \to \infty$, all the exponential terms in the analytical solution die out, so that the steady-state value of the covariance simplifies to:

$$P(\infty) = -\frac{1}{4a(a^2+b^2)}\begin{bmatrix} q_d + (2a^2+b^2+c^2-2ac)q_b & \beta(q_d+q_b\omega_n^2) \\ \beta(q_d+q_b\omega_n^2) & (2a^2+b^2+c^2+2ac)q_d + q_b\omega_n^4 \end{bmatrix} \tag{6.232}$$

6.9 Measurement Editing and Feasibility Criteria

6.9.1 Line of Site Test

This test checks to see if a celestial body obstructs the signal path between two objects. There are three cases that can occur in this test: two spacecraft (case 1), a spacecraft and a participant on the surface of a celestial body (case 2), and two participants on the surfaces of different celestial bodies (case 3). Below we address each of these cases starting with case 1.

For all three cases, assume the signal is generated by participant 1 at time t_1 and, if no obstruction occurs, the signal is received by participant 2 at time t_2. Define the location of the first participant at time t_1, expressed in $\mathcal{F}_{\mathcal{I}_1}$ as $\mathbf{r}_1^{\mathcal{I}_1}(t_1)$ which is calculated using Eq. (6.6). Define the location of participant 1 at time t_1, expressed in $\mathcal{F}_{\mathcal{S}}$, as $\mathbf{r}_1^{\mathcal{S}}(t_1)$ which is calculated using Eq. (6.8). Similarly, $\mathbf{r}_2^{\mathcal{I}_2}(t_2)$ and $\mathbf{r}_2^{\mathcal{S}}(t_2)$ are calculated using Eqs. (6.7) and (6.9) respectively.

Two Spacecraft

6.9.2 Height of Ray Path

6.9.3 Line of Sight

The Line of Sight (LOS) test is a visibility test for inter-spacecraft measurements. The algorithm presented here is based on Vallado[25] (pp. 307-311) with slight modifications to include light time correction when applicable (i.e. for all measurements except geometric measurements). This test first checks to see if the intersection of the perpendicular distance vector, $\mathbf{d}(\tau_{min})$, with the ray path, $\boldsymbol{\rho}$, lies in between the two participants. It then checks to see if the ray path height, h, is above the minimum allowable ray path altitude h_{min}.

The geometry for the LOS test is shown in Fig. (6.7). The variables in Fig. 6.7 are defined as follows.

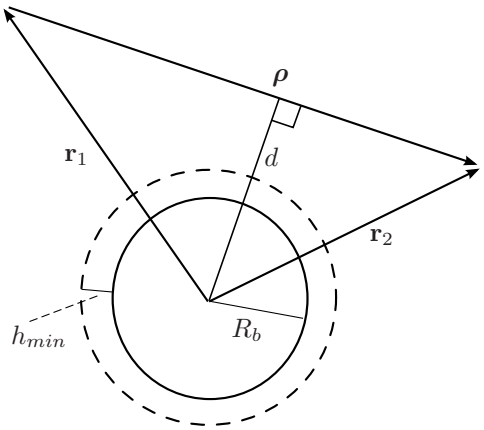


Figure 6.7: Height of Ray Path Geometry

$\mathbf{r}_1(t_1)$	Position vector of participant 1 at time of signal transmission, t_1
$\mathbf{r}_2(t_2)$	Position vector of participant 2 at time of signal reception, t_2
$\boldsymbol{\rho}$	Vector from location of particpant 1 at signal transmission to location of participant 2 at signal reception
ρ	Magnitude of $\boldsymbol{\rho}$
R_b	Central body radius
h_{min}	Minimum acceptable ray path height
τ	Ray path parameter defined as the fraction of the unit distance along $\boldsymbol{\rho}$
$\mathbf{d}(\tau)$	Position vector of τ
τ_{min}	Value of τ minimizing the distance between $\boldsymbol{\rho}$ and the central body
$d(\tau_{min})$	Distance from central body origin to ray path (measured perpendicular)

The ray path vector, $\boldsymbol{\rho}$, is computed from

$$[\boldsymbol{\rho}]_1 = [\mathbf{r}_2(t_2)]_1 - [\mathbf{r}_1(t_1)]_1 \tag{6.233}$$

where the quantities $\mathbf{r}_2(t_2)$ and $[\mathbf{r}_1(t_1)]_1$ are determined during light time correction. The ray path parameter, τ, is a unitless value indicating a point along $\boldsymbol{\rho}$. The only value of τ that we care about here is τ_{min}, which is computed from

$$\tau_{min} = \frac{[\mathbf{r}_2(t_2)]_1 \cdot [\boldsymbol{\rho}]_1}{\rho^2} \tag{6.234}$$

The position vector of τ, $\mathbf{d}(\tau)$, is simply

$$\mathbf{d}(\tau) = [\mathbf{r}_2(t_2)]_1 - [\boldsymbol{\rho}]_1 \tau \tag{6.235}$$

Finally, the criteria for measurement feasibility is when the following statement is true

$$T^2 - T > 0 \text{ or } d(\tau_{min})^2 - R_b^2 >= 0 \tag{6.236}$$

For geometric measurements, $t_1 = t_2$. For measurements involving participants about different central bodies, the LOS test is performed twice, once for each central body.

6.9.4 Horizon Angle Test

The horizon angle test checks to see if a space-based observer is above the local horizon for a ground-based observer. The space-based observer can be a spacecraft or ground based observer on another celestial body. The geometry for this test is shown in Fig. 6.8 where the nomenclature is defined below.

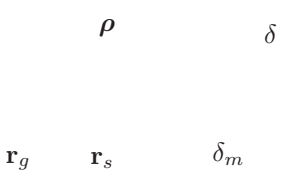


Figure 6.8: Height Angle Test Geometry

$\mathbf{r}_g$	Position vector of the ground-based observer at time t_g
t_g	Time the signal is at the receiving electronics of the ground-based observer
$\mathbf{r}_s$	Position vector of the space-based observer at time t_s
t_s	Time the signal is at the receiving electronics of the space-based observer
$\boldsymbol{\rho}$	Vector from location of ground-based observer to space-based observer of participant 2 at signal reception
δ_m	Minimum elevation angle above local horizon of ground-based observer

From inspection of Fig. 6.8, a space-based observer is above the local horizon of a ground-based observer if

$$\delta \geq \delta_m \tag{6.237}$$

From manipulation of the inner product we know

$$\sin\delta = \frac{\boldsymbol{\rho}\cdot\mathbf{r}_g}{\rho r_g} \tag{6.238}$$

where $\boldsymbol{\rho} = \mathbf{r}_s - \mathbf{r}_g$. Finally, the horizon check is true if the following statement is satisfied and false otherwise:

$$\frac{\boldsymbol{\rho}\cdot\mathbf{r}_g}{\rho r_g} > \sin\delta_m \tag{6.239}$$

6.9.5 Range Limit Test

6.9.6 Range Rate Limit Test

6.9.7 Solar Exclusion Angle Test

Chapter 7

Mathematics in GMAT Scripting

7.1 Basic Operators

7.2 Math Functions

7.2.1 max

`[maxX] = max(X)`

`X` is an nxm array. `maxX` is a 1xm row vector containing the maximum value in each column of `X`.

7.2.2 min

`[minX] = min(X)`

`X` is an nxm array. `minX` is a 1xm array containing the minumum value contained in each row of `X`.

7.2.3 abs

`[absX] = abs(X)`

`X` is an nxm array. `absX` is a nxm array where each component is the absolute value of the corresponding component of `X`.

7.2.4 mean

`[meanX] = mean(X)`

`X` is an nxm array. `meanX` is a 1xm row vector containing the mean of each column of `X`.

7.2.5 dot

`[dotp] = dot(vec1,vec2)`

The `dot` function calculates the dot (scalar) product of two vectors. `vec1` and `vec2` must both be vectors with the same length. `dotp` is the scalar product.

7.2.6 cross

`[crossp] = cross(vec1,vec2)`

The `cross` function calculates the cross product of two vectors. `vec1` and `vec2` must both be vectors with the same length. `crossp` is the cross product.

7.2.7 norm

```
[normv] = norm(vec)
```

The `norm` function calculates the 2-norm of a vector. `vec` must both be a vector. `normv` is the root-sum-square of the components of `vec`.

7.2.8 det

```
[detX] = norm(X)
```

The `det` function calculates the determinant of a matrix. `X` is an *nxn* array. `detX` is the determinant of `X`.

7.2.9 inv

```
[invX] = inv(X)
```

The `inv` function returns the inverse of a matrix. `X` must be a square matrix.

7.2.10 eig

7.2.11 sin, cos, tan

7.2.12 asin, acos, atan, atan2

7.2.13 sinh, cosh, tanh

7.2.14 asinh, acosh, atanh

7.2.15 transpose

7.2.16 DegToRad

7.2.17 RadToDeg

7.2.18 log

7.2.19 log10

7.2.20 exp

7.2.21 sqrt

Chapter 8

Solvers

8.1 Differential Correction

8.2 Broyden's Method

8.3 Newton's Method

8.4 Matlab fmincon

The user first creates a solver and names it. An example is

```
Create fminconOptimizer SPQfmincon
```

The user creates an optimization sequence by issuing an optimize command, followed by the name of the optimizer to use

```
Optimize SQPfmincon
EndOptimize
```

8.5 The Vary Command

The user defines the independent variables by the vary command,

Table 8.1: Available Commands in an fmincon Loop

Value	Command
X_i	`Vary`
Upper Bound on X_i	`Vary`
Lower Bound on X_i	`Vary`
Nondimensionalization Factor 1	`Vary`
Nondimensionalization Factor 2	`Vary`
Nonlinear constraint function	`NonLinearConstraint`
Linear constraint function	`LinearConstraint`
Cost Function	`OptimizerName.Cost =`

Chapter 9

Events

In GMAT, events allow a user to determine when station contacts, eclipses, or spacecraft-to-spacecraft occur among others. In general, events are dependent upon the orbit dynamics and time dependent parameters, and therefore can only be determined during or after orbit propagation. The implementation of Events requires GMAT to find the roots of parametric functions of time. The roots of the parametric equation are the event times, in the case of a discrete event, or define the event boundaries, in the case of an interval event.

In this chapter, we'll look at how GMAT calculates the roots of event functions and hence locates both discrete and interval events. This includes two subproblems. The first is determining if a root/event has occurred during a propagation step. The second, is determining the numerical value of the root. We begin by looking at the mathematical definition of an event function in GMAT.

9.1 Overview of Events

We begin by defining two types of events: (1) discrete events that occur at an instant of time, such as the epoch of an impulsive maneuver, and (2) an interval event that occurs over some finite time span, such as a spacecraft eclipse. Events are located using root finding methods.

Let's proceed using an example event. Assume we wish to determine when a tracking station will next be able to take measurements of a spacecraft's range and Doppler. The conditions that must be satisfied, called the event functions, are that the spacecraft must be 5 degrees above the horizon, and that the relative range rate of the spacecraft must be less than 3 km/s.

1) Feasible 2) If not feasible, what condition causes infeasiblity 3) Ordered for efficiency 4) Event location of for discrete event

$$\Delta t_e < \Delta t_s \tag{9.1}$$

An Event Function in GMAT has three outputs. The general form of an Event Function is

$$[\ \mathbf{f}, \mathbf{d}, \mathbf{p}\ ] = \mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) \tag{9.2}$$

where t is the current time, $\mathbf{x}(t)$ is a vector of time dependent parameters such as spacecraft states, and $\mathbf{C}$ is a vector of constants. $\mathbf{f}$ is vector of function values at t, $\mathbf{d}$ is a vector describing the or sign change we wish to track that occurs at the root, and $\mathbf{p}$ is a vector that tells GMAT whether a root is possible or not. Let's talk about some of the output variables in more detail.

For efficiency and convenience, the user can calculate several different function values, f, inside of a single event function, $\mathbf{F}$. This is useful when several functions require similar yet expensive calculations. GMAT allows the user to pass back a vector of function values in the output parameter $\mathbf{f}$, where the components of $\mathbf{f}$ are simply the values of the different functions f, or

$$\mathbf{f} = [\ f_1(t, \mathbf{x}(t), \mathbf{C}), f_2(t, \mathbf{x}(t), \mathbf{C}) ... f_n(t, \mathbf{x}(t), \mathbf{C})\ ]^T \tag{9.3}$$

The output parameter **d** allows the user to define which type of roots for GMAT to calculate. For example, in some cases we might only be interested in roots that occur when the function changes from a negative value to a positive value. In other situations we may only be interested in roots that occur when the function passes from positive to negative. Finally, we may be interested in both types of roots. **d** is a vector that has the same number of elements as **f**, and the first element of **d** corresponds to the first element of **f** and so on. Table 9.1 summarizes the allowable choices for components of **d** and the action GMAT will take depending upon the selection.

Table 9.1: Allowable Values for **d** in Event Function Output

Value	Action
d = 1	Find roots when the function is moving in the positive direction.
d = -1	Find roots when the function is moving in the negative direction.
d = 2	Find both types of roots

The last output variable in the Event Function output, **p**, is a flag that allows the user to tell GMAT whether or not a root is possible. If a component of **p** is zero, then GMAT will not attempt to try to find a root of the corresponding function in **f**. This flag is included to improve the efficiency of the algorithm. It is often possible to perform a few simple calculations to determine if a root is possible or not. For example, let's assume an event function is written to track Earth shadow crossings and that the function is positive when a spacecraft is not in Earth's shadow, and negative when it is in Earth shadow. It is a relatively simple calculation to determine if a spacecraft is on the day-side of Earth by taking the dot product of the Sun vector and the spacecraft's position vector. If the quantity is positive, there is no need to continue calculating the actual function value.

Now that we've looked at the definitions of the inputs and outputs of an Event Function, let's look at some different approaches to finding the roots of an event function.

9.2 Issues in Locating Zero Crossings

Before discussing the practical issues in finding roots of Event Functions, let's take a look at a hypothetical function to illustrate some of the issues that must be addressed. Figure 9.1 shows a sample event function. The smooth line represents the locus of points of the function itself, and the large "X" marks represent the function values at the integration time steps. The smaller tick marks indicate the function values at the internal integrator stages, which may be available if we use a dense output numerical integrator.

In general, we don't have continuous time expressions for the inputs to event functions. We only know the inputs to Event Functions at discrete points in time, so we only know the Event Function values at discrete points in time. Since these discrete times come from the numerical integration of a differential equation, we can only calculate Event Function values at the integration time steps, or at the internal stages if the information is available. This fact can cause a significant problem because an Event Function may vary rapidly and the discrete times at which we know the Event Function may not give an accurate picture of the function.

Let's consider a few ways in which we can determine if a root has occurred, given a set of times and Event Function values. The most obvious method is to simply look for sign changes in the function values. If the function changes sign, then we know we have bracketed a root. This approach will incorrectly conclude that a zero crossing did not occur, if there is an even number of zero crossings between two function values. Another approach is to fit a polynomial through the data, and see if the polynomial has any real roots. While this approach may be more accurate than looking for sign changes in some cases, it still does not guarantee a zero crossing is missed. A third approach might be to force the integrator to select step sizes based on the rate of change of the Event Function. We will not investigate this method further here though.

In short, there is no way to guarantee that a root crossing is missed. However, by having an understanding of the Event Function, having control over the maximum integration step size, and having access to the internal integrator stages when using a dense output integrator, we can do a acceptably good job determining

when zero-crossings occur. Once a zero-crossing is identified, there are well known ways to calculate the actual root value.

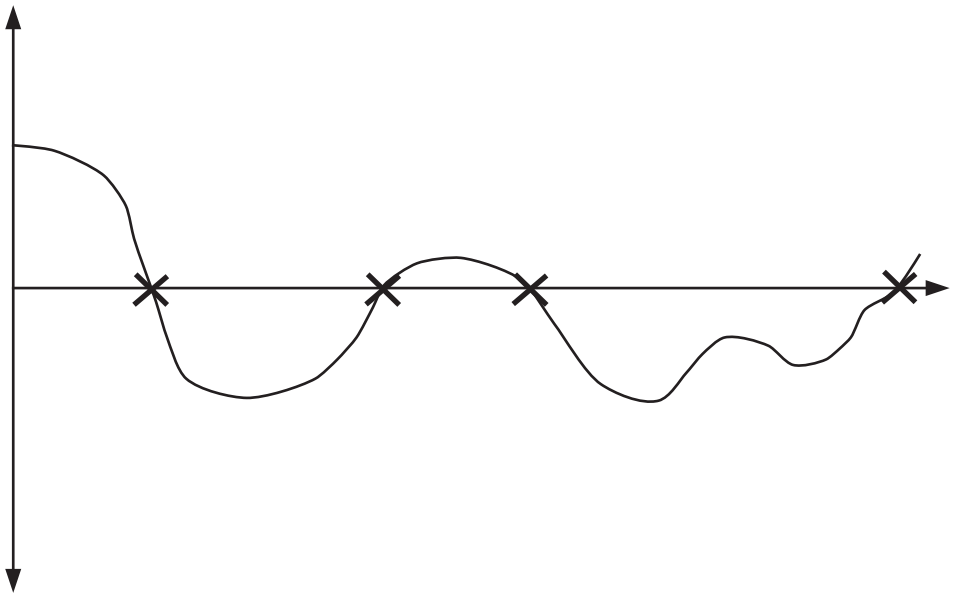


Figure 9.1: Sample Event Function Output

$$\mathbf{F} = \mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) \tag{9.4}$$

We want to find all t such that

$$\mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) = 0 \tag{9.5}$$

9.3 Root Finding Options in GMAT

In implementing a root finding approach, we need to balance accuracy and the need to find every root, with speed and performance. One way to do this is to allow the user to select between different approaches depending upon the accuracy needed for a particular application. The user has several controls to tell GMAT how to determine if a zero crossing has occurred, and how to calculate the numerical value of a root if one has been detected. Let's look at the choices implemented in GMAT, and discuss some options that can be included if a more robust method is required.

The first group of controls available to the user are related to how or if GMAT tries to determine if a root has occurred. The user can provide a flag in the output of an Event Function that tells GMAT whether it is possible that a root has occurred during the last integration step. This flag is notated as **p** and is discussed in section 9.1. If an element of **p** is zero, then GMAT will not use more sophisticated and therefore more computationally intensive methods to determine if a zero crossing for the particular component of the event function has occurred. **p** can be either zero or one, and can change value during propagation. If **p** changes from zero to one, GMAT begins using a root checking method specified by the user to determine if a zero crossing has occurred, and begins storing function data in case it is needed to interpolate a root location.

The second control that determines if a zero crossing has occurred is called `RootCheckMethod` in the GMAT script language. There are several RootCheckMethod options available and the user can currently select between `FunctionSignChange` and `PolynomialFit`. If the user selects `FunctionSignChange`, then GMAT looks for sign changes in the function output to determine if a zero crossing has occurred. If the user selects `PolynomialFit`, then GMAT fits a polynomial to the Event Function data, and checks to see if the polynomial has any real roots. If the polynomial has real roots, then a zero crossing has occurred. The

type of polynomial GMAT uses in `RootCheckMethod` is the same as it uses in `RootFindingMethod` and is discussed in more detail below.

If a zero crossing is detected, there are many ways to determine the numerical value of the root. The user can select between the different methods by using the `RootSolvingMethod` option. The two methods currently implemented in GMAT are called `QuadraticPolynomial` and `CubicSpline` in the GMAT script language. As the name suggests, if the user selects `QuadraticPolynomial`, then GMAT uses the last three function values to create a quadratic polynomial. Then, the quadratic equation is used to determine the root locations. Similarly, if the user selects `CubicSpline`, GMAT constructs a cubic spline and then uses interpolation to find the root value.

Allowing the options above requires that care is taken in designing an algorithm to track events. In the next section we discuss some of the issues that must be addressed in the Event Function algorithm, and present a flow chart that describes the algorithm in detail

9.4 Algorithm for Event Functions

Table 9.2: Variables in Event Function Algorithm

Variable	Definition
n_r	Number of data points required to use the requested `RootSolvingMethod` option
n_c	Number of data points required to use the requested `RootCheckMethod` option
f	A vector of function values provided by the user defined Event Function
N	The length of **f**, which is the number function values contained in the output of a user defined Event Function.
d	A vector of flags (length N) that defines which type of roots to track. (negative to positive, positive to negative, or both)
p	A vector of flags (length N) that tells whether or not a zero crossing is possible. A component of **p** is one if a root is possible, otherwise it is zero.
Startup	A vector of flags of lenght N. The components of **Startup** correspond to the components of **f**. A component of **Startup** is one, if there is less than max(n_r,n_c) data points saved for use in root finding. Otherwise, a component of **Startup** is zero.

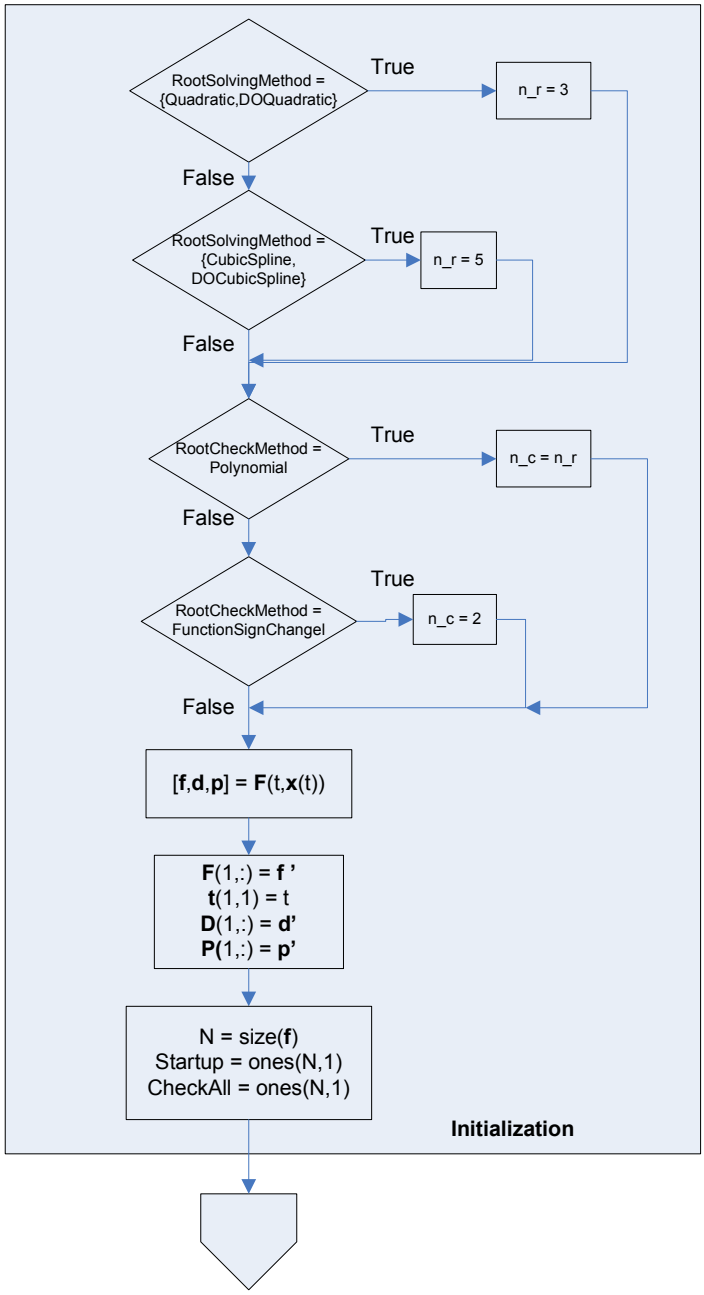


Figure 9.2: Initializations for the Event Location Algorithm

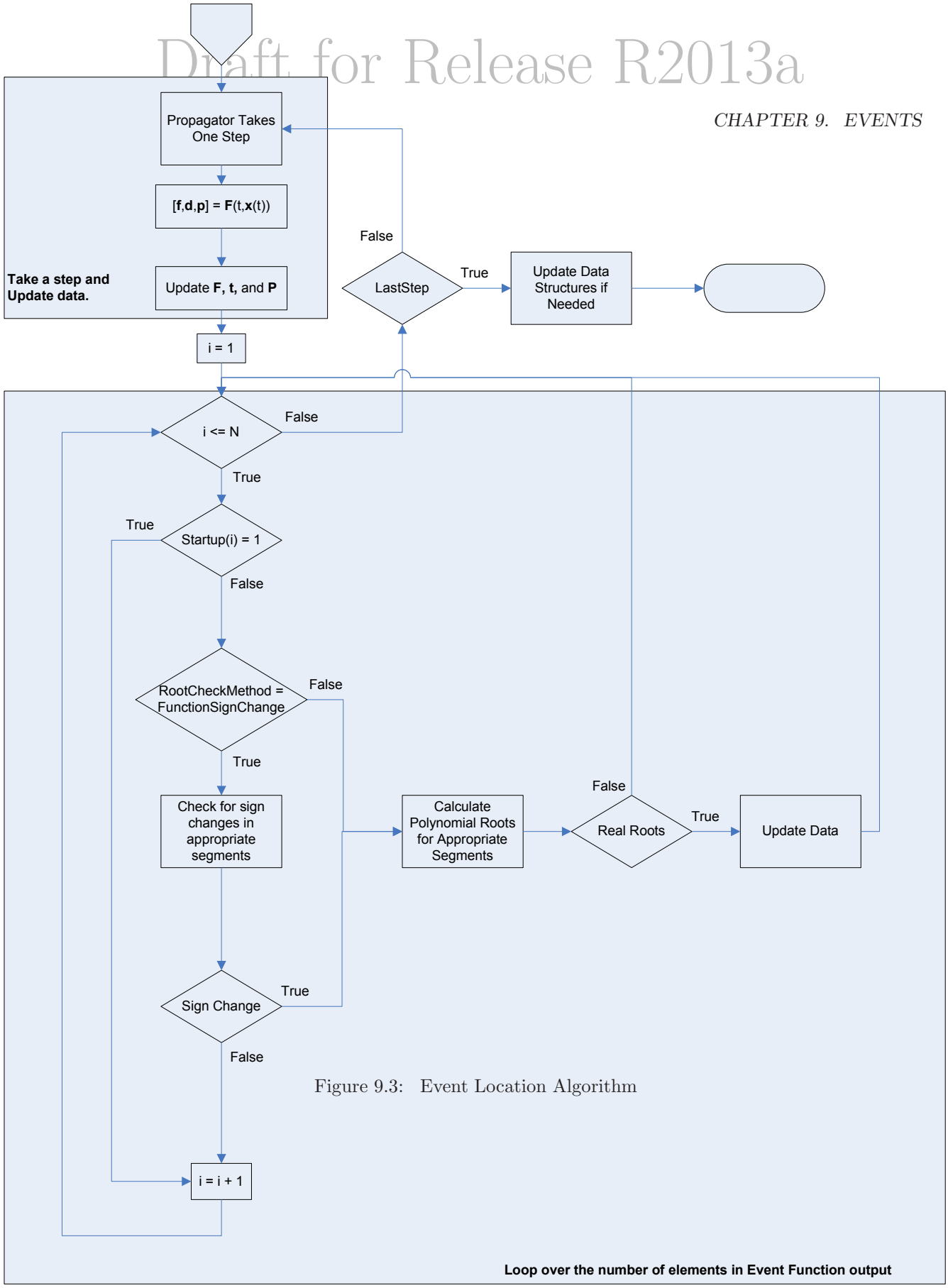


Figure 9.3: Event Location Algorithm

Chapter 10

Graphics

10.1 Ground Track Plotting

A ground track plot displays spherical latitude and longitude over a Poincare (unwrapped cylinder) projection of a body's surface geography. The algormithm used to generate a ground track plot ensures that line wrapping at the plot boundaries is handled correctly, and that when wrapping occurs, the plot is interpolated to the plot boundary.

Longitude is computed by converting position and velocity of the object to the body fixed system of the ground track plot central body. Define x, y, and z as the body fixed coordinates of the object. The longitude is calculated using

$$\lambda = atan2(y, x) \tag{10.1}$$

and the latitude is calculate using

$$\phi = asin\frac{z}{r} \tag{10.2}$$

where

$$r = \sqrt{x^2 + y^2 + z^2} \tag{10.3}$$

There are three special cases to consider when plotting a new point on a ground track: (1) the new point wraps off the right-hand side of the plot, (2) the new point wraps off the left-had side of the plot, and (3) the new point does not wrap off either plot boundary. When wrapping oocurs as in cases (1) and (2), the plot algorithm must perform "Pen Up" and "Pen Down" commands to avoid connecting points on opposite ends of the plot with a spurious straight line. Secondly, when wrapping occurs, the system must interpolate the line segments to plot boundaries.

The test to determine the case depends upon whether the object is moving clockwise or counterclockwise in the body fixed system of the ground track plot central body. The orbit direction is determined by evaluating the z component of orbit angular momentum expressed in the body fixed system. Define a variable d that is positive for clockwise motion in the body fixed system (the spacecraft is moving to the right on a ground track plot), and negative for counterclockwise motion. (spacecraft is moving to the left on a ground track plot), and 0 for no motion. The variable d is computed as follows

$$d = sign(x\dot{y} - \dot{x}y) \tag{10.4}$$

$$m = \frac{\phi_i - \phi_{i-1}}{\lambda_i - \lambda_{i-1}} \tag{10.5}$$

Input: $\lambda_i, \lambda_{i-1}, \phi_i, \phi_{i-1}, d_i, d_{i-1}$
Output: Updated Ground Track Plot
$m\lambda_i^+ = mod(\lambda_i, 2\pi)$;
$m\lambda_{i-1}^+ = mod(\lambda_{i-1}, 2\pi)$;
$m\lambda_i^- = mod(\lambda_i, -2\pi)$;
$m\lambda_{i-1}^- = mod(\lambda_{i-1}, -2\pi)$;
% New point wraps off RHS border;
if $d_i = d_{i-1} = 1$ *And* $m\lambda_{i-1}^+ < \pi$ *And* $m\lambda_i^+ > \pi$ **then**
  $m = \dfrac{\phi_i - \phi_{i-1}}{m\lambda_i^+ - m\lambda_{i-1}^+}$;
  $\phi_b = m(\pi - m\lambda_i^+) + \phi_i$;
  Plot line segment from $(\lambda_{i-1}, \phi_{i-1})$ to (π, ϕ_b);
  Plot line segment from $(-\pi, \phi_b)$ to (λ_i, ϕ_i);
% New point wraps off LHS border;
else if $d_i = d_{i-1} = -1$ *And* $m\lambda_i^- < -\pi$ *And* $m\lambda_{i-1}^- > -\pi$ **then**
  $m = \dfrac{\phi_i - \phi_{i-1}}{m\lambda_i^- - m\lambda_{i-1}^-}$;
  $\phi_b = m(-\pi - m\lambda_i^-) + \phi_i$;
  Plot line segment from $(\lambda_{i-1}, \phi_{i-1})$ to $(-\pi, \phi_b)$;
  Plot line segment from (π, ϕ_b) to (λ_i, ϕ_i);
% New does not wrap off plot border ;
else
  Plot line segment from $(\lambda_{i-1}, \phi_{i-1})$ to (λ_i, ϕ_i);
end

Algorithm 1: Algorithm for Updating A Ground Track Plot

10.2 Footprint and Limb Computation

10.2.1 Overview

Computing an instrument footprint or the Earth Limb as viewed from a spacecraft involves essentially three related problems: (1) Determining the intersection of a given line with a given ellipsoid, (2) determination of a line tangent to a known ellipsoid, and (3) determining how to select the points along the footprint or limb curve to provide a smooth plot in the graphics. These problems are related and discussed in order below starting with the problem of computing the intersection of a line and an ellipsoid.

10.2.2 Intersection of Line and Ellipsoid

Begin by defining a ray $\boldsymbol{\ell}$ such that

$$\boldsymbol{\ell} = \mathbf{p} + \alpha\hat{\mathbf{d}} \tag{10.6}$$

where $\mathbf{p}$ (in this context the usually location of a spacecraft) is the the starting location of ray $\boldsymbol{\ell}$, $\hat{\mathbf{d}}$ is the unit vector in the direction of the ray, and α is the distance from coordinates $\mathbf{p}$ in the direction. The equation for a tri-axial ellipoid is defined as

$$\frac{x^2}{R_x^2} + \frac{y^2}{R_y^2} + \frac{z^2}{R_z^2} - 1 = 0 \tag{10.7}$$

where R_x, R_y, and R_z are the ellipsoid radii in the x, y, and z directions respectively. For simplicity of notation, we'll assume all coordinates such x, y, and z, and ray $\boldsymbol{\ell}$ are expressed in the body coordinates of the ellipsoid. Subtituting Eq. (10.6) into Eq. (10.7) results in

$$\frac{\boldsymbol{\ell}^T\hat{\mathbf{i}}}{R_x^2} + \frac{\boldsymbol{\ell}^T\hat{\mathbf{j}}}{R_y^2} + \frac{\boldsymbol{\ell}^T\hat{\mathbf{k}}}{R_z^2} - 1 = 0 \tag{10.8}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{i}}$, and $\hat{\mathbf{i}}$ are unit vectors in the x, y, and z directions respectively. Define α^* as the value of α that simulteoulsy satisfies both the equation for ray $\boldsymbol{\ell}$ and the equation for the triaxial ellipoid. Expanding

Eq. (10.8) and grouping terms by powers of α^* yields

$$\alpha^{*2}\left(\frac{d_x^2}{R_x^2}+\frac{d_y^2}{R_y^2}+\frac{d_z^2}{R_z^2}\right)+2\alpha^*\left(\frac{d_x p_x}{R_x^2}+\frac{d_y p_y}{R_y^2}+\frac{d_z p_z}{R_z^2}\right)+\left(\frac{p_x^2}{R_x^2}+\frac{p_y^2}{R_y^2}+\frac{p_z^2}{R_z^2}-1\right)=0 \tag{10.9}$$

This is a quadratic equation and the solution is

$$\alpha^* = \frac{-B \pm \sqrt{B^2-4AC}}{2A} \tag{10.10}$$

where

$$A = \left(\frac{d_x^2}{R_x^2}+\frac{d_y^2}{R_y^2}+\frac{d_z^2}{R_z^2}\right) \tag{10.11}$$

$$B = 2\left(\frac{d_x p_x}{R_x^2}+\frac{d_y p_y}{R_y^2}+\frac{d_z p_z}{R_z^2}\right) \tag{10.12}$$

$$C = \left(\frac{p_x^2}{R_x^2}+\frac{p_y^2}{R_y^2}+\frac{p_z^2}{R_z^2}-1\right) \tag{10.13}$$

There are three physical types of solutions for α^* given by Eq. (10.10). If $(B^2-4AC)<0$ then ray $\boldsymbol{\ell}$ does not intersect the ellipoid. If $(B^2-4AC)=0$ then ray $\boldsymbol{\ell}$ is tangent to the ellipsoid (we'll revisit this relation when computing the limb curve). In this case,

$$\boldsymbol{\ell}^* = \mathbf{p} - \frac{B}{2A}\hat{\mathbf{d}} \tag{10.14}$$

Finally, if $(B^2-4AC)>0$ then ray $\boldsymbol{\ell}$ intersects the ellipsoid at two locations. For graphics purposes, we require the smaller value of α^* given by Eq. (10.10). If we define $\boldsymbol{\alpha}^*$ as a vector containing the two solutions for α corresponding to the two intersection points, then, the equation for the nearest intersection point, $\boldsymbol{\ell}^*$ is given by

$$\boldsymbol{\ell}^* = \mathbf{p} + min(\boldsymbol{\alpha}^*)\hat{\mathbf{d}} \tag{10.15}$$

10.2.3 Determining the Limb Region

10.2.4 Selecting Points for Accurate Graphics

Chapter 11

Numerical Algorithms

11.1 Lagrange Interpolation

Below we describe the algorithm used for lagrange interpolation. This includes specifying how points from the known data are chosen for interpolation, and how interpolation is performed for multiple function values (i.e. position and velocity) at the desired interpolation point.

Assume we have m functions to interpolate and that each function is known at ℓ values of the independent variable x (in ephemeris interpolation this is time) as illustrated in Table 11.1. Given some point, say x', we desire the interpolated values of the functions, or $\mathbf{f}' \approx \mathbf{P}(x')$ where P is the Lagrange interpolating polynomial.

Table 11.1: Example of Data for interpolation

x	$\mathbf{f}_1$	$\mathbf{f}_2$	...	$\mathbf{f}_m$
x_1	$f_1(x_1)$	$f_2(x_1)$	...	$f_m(x_1)$
x_2	$f_1(x_2)$	$f_2(x_2)$	...	$f_m(x_2)$
x_3	$f_1(x_3)$	$f_2(x_3)$	...	$f_m(x_3)$
x_4	$f_1(x_4)$	$f_2(x_4)$	...	$f_m(x_4)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x_{\ell-1}$	$f_1(x_{\ell-1})$	$f_2(x_{\ell-1})$	...	$f_m(x_{\ell-1})$
x_ℓ	$f_1(x_\ell)$	$f_2(x_\ell)$	...	$f_m(x_\ell)$

Before proceeding we define the following variables:

$\mathbf{x}$	m_x x 1 Known values of the independent variable (monotonically increasing or decreasing)
x'	Value of independent variable a the required interpolation point
$\mathbf{f}_i$	m_x x m_f Array of dependent variable values for the i^{th} function
$\mathbf{f}'$	1 x m_f Array of interpolated function values at x'
n	Order of interpolation
m_x	Number of points in $\mathbf{x}$ and $\mathbf{f}_i$
m_f	Number of functions to be interpolated

For interpolation to be feasible, several conditions must be satisfied. First, x' must lie between the minimum and maximum value of $\mathbf{x}$, or at the boundaries (otherwise we would be extrapolating):

$$x' \geq \min(\mathbf{x}) \tag{11.1}$$

and

$$x' \leq \max(\mathbf{x}) \tag{11.2}$$

Second, there must be enough data points in points in $\mathbf{x}$ to support interpolation to the desired order. If the requested interpolation is of order n, then we require $n+1$ data points to perform the interpolation. Hence the following criteria must be met:

$$m_x \geq n+1 \tag{11.3}$$

If the requested interpolation is feasible, we must choose the subset of $\mathbf{x}$ to use for interpolating the data at x'. Define $x(q)$ as the q^{th} element of $\mathbf{x}$. Given x', we choose q to minimize the difference between x' and the mean of $x(q)$ and $x(q+n)$, or:

$$\min_q \left| \frac{x(q+n)+x(q)}{2} - x' \right| \qquad (q \in 1,2,3,...m_x) \tag{11.4}$$

Choosing q in this way places x' as near to the center of the interpolation interval as possible.

The standard formula for the Lagrange interpolating polynomial for a single function is

$$P(x) = \sum_{k=1}^{n} P_j(x) \tag{11.5}$$

where $P_j(x)$ is given by

$$P_j(x) = y_j \prod_{k=1,k\neq j}^{n} \frac{x - x_j}{x_j - x_k} \tag{11.6}$$

Lagrange interpolation is an efficient method when interpolating multiple data sets available at the same independent variable points. This is due to the fact that the product term in Eq. (11.6) is only a function of the values of the independent variables and the desired interpolation point and not on the function being interpolated. As a result, we can evaluate the product term one time and use it for many functions (which is what is done when interpolating an ephemeris file). The algorithm for implementing Lagrangian interpolation for multiple functions is shown below.

Input: $\mathbf{x}, \mathbf{f}, x', q, n$
Output: $\mathbf{f}'$
$\mathbf{f}' = \mathbf{0}_{(1 \text{ x } m_f)}$;
for $i = q$ *to* $q+n$ **do**
  $\mathbf{prod} = f(i,:)$ % The i^{th} row of $\mathbf{f}$;
  for $j = q$ *to* $q+n$ **do**
    if $i \neq j$ **then**
      % The product in Eq. (11.6);
      $\mathbf{prod} = \mathbf{prod} \cdot \dfrac{x' - x(j)}{x(i) - x(j)}$;
    end
  end
  % The summation in Eq. (11.5);
  $\mathbf{f}' = \mathbf{f}' + \mathbf{prod}$;
end

Algorithm 2: Algorithm for Lagrangian Interpolation

11.2 Quadratic Polynomial Interpolation

We are given three data points defined by a vector of independent variables

$$\mathbf{x} = [\ x_1\ x_2\ x_3\ ]^T \tag{11.7}$$

and a vector of corresponding dependent variables

$$\mathbf{y} = [\ y_1\ y_2\ y_3\ ]^T \tag{11.8}$$

we wish to find a quadratic polynomial that fits the data such that

$$y = Ax^2 + Bx + C \tag{11.9}$$

We begin by forming the system of linear equations

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \tag{11.10}$$

We can solve for the coefficients using

$$A = \frac{\begin{vmatrix} y_1 & x_1 & 1 \\ y_2 & x_2 & 1 \\ y_3 & x_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}} \tag{11.11}$$

$$B = \frac{\begin{vmatrix} x_1^2 & y_1 & 1 \\ x_2^2 & y_2 & 1 \\ x_3^2 & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}} \tag{11.12}$$

$$C = y_3 - Ax_3^2 - Bx_3 \tag{11.13}$$

11.3 Cubic Spline (Not-a-Knot) Interpolation

We are given five data points defined by a vector of independent variables

$$\mathbf{x} = [\ x_1\ x_2\ x_3\ x_4\ x_5\ ]^T \tag{11.14}$$

and a vector of corresponding dependent variables

$$\mathbf{y} = [\ y_1\ y_2\ y_3\ y_4\ y_5\ ]^T \tag{11.15}$$

we wish to find the four cubic polynomials, $i = 1, 2, 3, 4$, such that

$$p_i = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{11.16}$$

where the values for x_i are known from the inputs.

To calculate the coefficients a_i, b_i, c_i, and d_i we start by calculating the eight quantities

$$h_i = x_{i+1} - x_i \tag{11.17}$$

$$\Delta_i = \frac{y_{i+1} - y_i}{h_i} \tag{11.18}$$

Next we solve the following system of linear equations

$$\mathbf{AS} = \mathbf{B} \tag{11.19}$$

where the components of $\mathbf{A}$ are given by

$$A_{11} = 2h_2 + h_1 \tag{11.20}$$
$$A_{12} = 2h_1 + h_2 \tag{11.21}$$
$$A_{13} = 0 \tag{11.22}$$
$$A_{21} = 0 \tag{11.23}$$
$$A_{22} = h_3 + 2h_4 \tag{11.24}$$
$$A_{23} = 2h_3 + h_4 \tag{11.25}$$
$$A_{31} = \frac{h_2^2}{h_1 + h_2} \tag{11.26}$$
$$A_{32} = \frac{h_1 h_2}{(h_1 + h_2)} + 2(h_2 + h_3) + \frac{h_3 h_4}{(h_3 + h_4)} \tag{11.27}$$
$$A_{33} = \frac{h_3^2}{h_3 + h_4} \tag{11.28}$$

the components of $\mathbf{B}$ are

$$B_{11} = 6(\Delta_2 - \Delta_1) \tag{11.29}$$
$$B_{21} = 6(\Delta_4 - \Delta_3) \tag{11.30}$$
$$B_{31} = 6(\Delta_3 - \Delta_2) \tag{11.31}$$

and $\mathbf{S} = [\ S_1\ S_3\ S_5\ ]^T$. We can solve for the components of $\mathbf{S}$ using Cramer's Rule as follows

$$S_1 = \frac{\begin{vmatrix} B_{11} & A_{12} & A_{13} \\ B_{21} & A_{22} & A_{23} \\ B_{31} & A_{32} & A_{33} \end{vmatrix}}{|\mathbf{A}|} \tag{11.32}$$

$$S_3 = \frac{\begin{vmatrix} A_{11} & B_{11} & A_{13} \\ A_{21} & B_{21} & A_{23} \\ A_{31} & B_{31} & A_{33} \end{vmatrix}}{|\mathbf{A}|} \tag{11.33}$$

$$S_5 = \frac{B_{31} - A_{31}S_1 - A_{32}S_3}{A_{33}} \tag{11.34}$$

Now we can calculate S_2 and S_4 using

$$S_2 = \frac{h_2 S_1 + h_1 S_3}{h_1 + h_2}; \tag{11.35}$$

$$S_4 = \frac{h_4 S_3 + h_3 S_5}{h_3 + h_4}; \tag{11.36}$$

Finally, the coefficients for the i^{th} cubic polynomial are given by

$$a_i = \frac{S_{i+1} - S_i}{6h_i} \tag{11.37}$$
$$b_i = \frac{S_i}{2} \tag{11.38}$$
$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \tag{11.39}$$
$$d_i = y_i \tag{11.40}$$

11.4 Root Location using Brent's Method

NOTE: The implementation used in GMAT's even locators follows the Algorithm in Numerical Recipes 3rd edition which slightlyl different than the algorithm presented below.

Brent's[26] method is a root finding algorithm that takes advantage of superlinear convergence properties of the secant method and inverse quadratic interpolation, and the robustness of the bisection method. The algorithm is a modification of Dekker's method which combines the bisection method and the secant method.[27]

The algorithm requires as inputs to real numbers that bound the desired root.

f	Function whose root is desired
a	Initial guess for independent variable (a and b must bound the desired root)
b	Initial guess for independent variable (a and b must bound the desired root)
tol	Convergence tolerance.
N	Maximum number of iterations
x_o	Root value
i	Number of iterations
ϵ	Machine precision

Input: f, a, b, tol, N,
Output: x_o, i
$f_a = f(a)$; $f_b = f(b)$;
if $f_a \cdot f_b > 0$; error and return; **end** ;
$c = a$; $f_c = f_a$; $d = b - a$; $e = d$;
if $|f_c| \leq |f_b|$ **then**
  $a = b$; $b = c$; $c = a$; $f_a = f_b$; $f_b = f_c$; $f_c = f_a$;
end
$t = 2\epsilon|b| + tol$;
$m = 0.5(c - b)$;
$i = 1$;
while $|m| >= t$ *and* $|f_b| >= t$ *and* $i \leq N$ **do**
  if $|e| \leq t$ *or* $|f_a| < |fb|$ **then**
    $d = m$; $e = m$;
  else
    $s = f_b/f_a$;
    if $a = c$ **then**
      $p = 2.0 \cdot m \cdot s$;
      $q = 1.0 - s$;
    else
      $q = f_a/f_c$; $r = f_b/f_c$;
      $p = s(2.0 \cdot m \cdot q(q - r) - (b - a)(r - 1.0))$;
      $q = (q - 1.0)(r - 1.0)(s - 1.0)$;
    end
    if $p >= 0.0$ **then**
      $q = -q$;
    else
      $p = -p$;
    end
    $s = e$;
    $e = d$;
    if $(2 \cdot p < 3.0 \cdot m \cdot q - |t \cdot q|)$ *or* $(p \leq |0.5 \cdot s \cdot q|)$ **then**
      $d = p/q$;
    else
      $d = m$; $e = m$;
    end
  end
  $a = b$; $f_a = f_b$;
  if $|d| \geq t$ **then**
    $b = b + d$;
  else
    if $m \geq 0.0$ **then**
      $b = b + t$;
    else
      $b = b - t$;
    end
  end
  $f_b = f(b)$;
  if $(\ f_b > 0.0$ *and* $f_c > 0\ )$ *or* $(\ f_b < 0.0$ *and* $f_c < 0\ )$ **then**
    $c = a$; $f_c = fa$; $d = b - a$; $e = d$;
  else
    if $|f_c| \leq |f_b|$ **then**
      $a = b$; $b = c$; $c = a$; $f_a = f_b$; $f_b = f_c$; $f_c = f_a$;
    end
  end
  $t = 2\epsilon|b| + tol$; $m = 0.5(c - b)$; $i = i + 1$;
end
$x_o = b$;

Algorithm 3: Brent's Method for Root Finding

Chapter 12

MathSpecAppendices

12.1 Vector Identities

$$\frac{\partial \mathbf{a}^T\mathbf{a}}{\partial x} = 2\mathbf{a}^T\frac{\partial \mathbf{a}}{\partial x} \tag{12.1}$$

$$\frac{\partial a}{\partial x} = \frac{\partial}{\partial x}\left(\mathbf{a}^T\mathbf{a}\right)^{1/2} = \frac{\mathbf{a}^T}{a}\frac{\partial \mathbf{a}}{\partial x} \tag{12.2}$$

$$\frac{\partial a^{-1}}{\partial x} = \frac{\partial}{\partial x}\left(\mathbf{a}^T\mathbf{a}\right)^{-1/2} = -\frac{\mathbf{a}^T}{a^3}\frac{\partial \mathbf{a}}{\partial x} \tag{12.3}$$

$$\frac{\partial}{\partial x}\left(\frac{\mathbf{a}}{a^3}\right) = \frac{1}{a^3}\frac{\partial \mathbf{a}}{\partial x} - 3\frac{\mathbf{a}\mathbf{a}^T}{a^5}\frac{\partial \mathbf{a}}{\partial x} \tag{12.4}$$

Bibliography

[1] Seidelmann, P. K., "Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000," *Celestial Mechanics and Dynamical Astronomy*, Vol. 82, No. 1, 2002, pp. 83 – 111.

[2] Frank P. Incropera, Davit P. Dewitt, T. L. B. and Lavine, A. S., *Fundamentals of Heat and Mass Transfer*, John Wiley and Sons, Inc., NJ, 2007.

[3] Vallado, D. A., *Fundamentals of Astrodynamics and Applications, 2nd Ed.*, Microcosm Press, El Segundo, CA, 2001.

[4] P. Kenneth Seidelmann, E., *Explanatory Supplement to the Astronomical Almanac*, Univesity Science Books, Mill Valley, CA, 1992.

[5] P. K. Seidelmann, T. F., "Why New Time Scales?" *Astronomy and Astrophysics*, Vol. 265, 1992, pp. 833 – 838.

[6] Kaplan, G. H., "The IAU Resolutions on Astronomical Reference Systems, Time Scales, and Earth Rotation Models Explanation and Implementation," *United States Naval Observatory Circular No. 179*, 2005.

[7] V. Coppola, J. S. and Vallado, D., "THE IUA 2000A and IAU 2006 Precession-Nutation Theories and Their Implementation," *Somewhere*.

[8] Long, A. and *et. al.*, J. C., *Goddard Trajectory Determination System Mathematical Theory, Revision 1*, Computer Sciences Corporation, Lanham-Seabrook, MD, 1989.

[9] Kizner, W., "A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories," *Jet Propulsion Laboratory External Publication*, , No. 674, 1959.

[10] Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods, Applications*, Springer, Berlin, Germany, 2005.

[11] Szebehely, V., *Theory of Orbits*, Academic Press Inc., New York, NewYork, 1967.

[12] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, New York, 1997.

[13] Pines, S., "A Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, No. 11, 1973, pp. 1508 – 1511.

[14] John B. Lundberg, B. E. S., "Recursion Formulas of Legender Functions for Use with Nonsingular Geopotential Models," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 1, 1988, pp. 31 – 38.

[15] C. Huang, J.C. Ries, B. T. M. W., "Relativisitic Effects for Near-Earth Satellite Orbit Determination," *Celestial Mechanics and Dynamical Astronomy*.

[16] Paul N. Estey, David H. Lewis Jr., M. C., "Prediction of Propellant Tank Pressure History Using State Space Methods," *Journal of Spacecraft and Rockets*, Vol. 20, No. 1, 1983, pp. 49 – 54.

[17] Hearn, H. C., "Development and Validation of Fluid/Thermodynamic Models for Spacecraft Propulsion Systems," *Journal of Propulsion and Power*, Vol. 17, No. 3, 2001, pp. 527 – 533.

[18] Michael J. Moran, H. S., *Fundamentals of Engineering Thermodynamics*, John Wiley and Sons, NY, 1992.

[19] Eliot Ring, E., *Rocket Propellant and Pressurization Systems*, Prentice Hall, Inc., NJ, 1964.

[20] Donald R. Pitts, L. E. S., *Shaum's Outlines, Heat Transfer 2nd Ed.*, McGraw Hill, NY, 1998.

[21] *Hydrazine Handbook*, Rocket Research Company.

[22] A. Ricciardi, E. P., "Prediction of the Performance and Thermodynamic Conditions of Bipropellant Propulsion System During its Lifetime," *AIAA/SAE/ASEE 23rd Joint Propulsion Conference*, June.

[23] 1980.

[24] Moyer, T., *Formulation for Observed and Computed Values of Deep Space Network Data Types for Navigation*, Jet Propulsion Laboratory, Pasadena, CA, 2000.

[25] Vallado, D. A., *Fundamentals of Astrodynamics and Applications, 3rd Ed.*, Microcosm Press, Hawthorne, CA, 2007.

[26] Brent, R., *Algorithms for Minimization without Derivatives*, Prentice Hall, Inc., NJ, 1973.

[27] Dekker, T., "Finding a zero by means of successive linear interpolation," *Constructive Aspects of the Fundamental Theorem of Algebra*.

Index