Draft: Work in Progress

General Mission Analysis Tool (GMAT) Mathematical Specifications

DRAFT

NASA Goddard Space Flight Center
Greenbelt RD
Greenbelt, MD 20771

Draft: Work in Progress

Contents

1 Introduction **9**

2 Time **11**
2.1 Time Systems . . . 11
2.1.1 Atomic Time: TAI and A.1 . . . 11
2.1.2 Universal Time: UTC and UT1 . . . 11
2.1.3 Dynamic Time: TT, TDB and TCB . . . 12
2.2 Time Formats . . . 12
2.2.1 Julian Date and Modified Julian Date . . . 13
2.2.2 Gregorian Date . . . 13
2.3 Conclusions . . . 14

3 Coordinate Systems **15**
3.1 General Coordinate SystemTransformations . . . 15
3.2 Pseudo-Rotating CoordinateSystems . . . 17
3.3 The $\mathcal{F}_{J_{2k}}$ Inertial System and FK5 Reduction . . . 17
3.3.1 Overview of FK5 Reduction . . . 17
3.3.2 Precession Calculations . . . 19
3.3.3 Nutation Calculations . . . 19
3.3.4 Sidereal Time Calculations . . . 21
3.3.5 Polar Motion Calculations . . . 21
3.4 Deriving $\mathbf{R}_{J_{2k},i}$ and $\dot{\mathbf{R}}_{J_{2k},i}$ for Various Coordinate Systems . . . 21
3.4.1 Equator System . . . 21
3.4.2 MJ2000 Ecliptic (MJ2000Ec) . . . 22
3.4.3 True of Epoch Equator (TOEEq) . . . 22
3.4.4 Mean of Epoch Equator (MOEEq) . . . 23
3.4.5 True of Date Equator (TODEq) . . . 23
3.4.6 Mean of Date Equator (MODEq) . . . 23
3.4.7 Mean of Date Ecliptic (MODEc) . . . 24
3.4.8 True of Date Ecliptic (TODEc) . . . 24
3.4.9 Celestial Body Fixed . . . 24
3.4.10 Body Inertial . . . 25
3.4.11 Object Referenced . . . 26
3.4.12 Geocentric Solar Ecliptic (GSE) . . . 28
3.4.13 Geocentric Solar Magnetic (GSM) . . . 28
3.5 Appendix 1: Derivatives of ObjectReferenced Unit Vectors . . . 29

4 Calculation Objects **33**
4.1 Spacecraft State Representations . . . 33
4.1.1 Definitions . . . 33
4.1.2 Cartesian State to Keplerian Elements . . . 33
4.1.3 Keplerian Elements to CartesianState . . . 36
4.1.4 Equinoctial Elements to CartesianState . . . 37
4.1.5 Cartesian State to EquinoctialElements . . . 38
4.1.6 Cartesian State to SphericalAZFPA State . . . 38
4.1.7 SphericalAZFPA State to Cartesian State . . . 39
4.1.8 Cartesian State to SphericalRADEC State . . . 40

Draft: Work in Progress

4.1.9 SphericalRADEC State to Cartesian State . . . 40
4.1.10 Keplerian or Cartesian, to Modified Keplerian Elements . . . 40
4.1.11 Modified Keplerian Elements to Keplerian Elements . . . 40
4.2 Simple Parameters . . . 41
4.2.1 A1Gregorian . . . 41
4.2.2 A1ModJulian . . . 41
4.2.3 Altitude . . . 42
4.2.4 AOP . . . 42
4.2.5 Apoapsis . . . 42
4.2.6 AZI . . . 43
4.2.7 BdotT and BdotR . . . 43
4.2.8 BetaAngle . . . 44
4.2.9 BVectorAngle and BVectorMag . . . 44
4.2.10 C3Energy . . . 44
4.2.11 DEC . . . 44
4.2.12 DECV . . . 45
4.2.13 ECC . . . 45
4.2.14 FPA . . . 45
4.2.15 EA . . . 45
4.2.16 Energy . . . 45
4.2.17 HMAG . . . 45
4.2.18 HX,HY, and HZ . . . 46
4.2.19 HA . . . 46
4.2.20 INC . . . 46
4.2.21 Latitude . . . 46
4.2.22 Longitude . . . 47
4.2.23 LST . . . 47
4.2.24 MA . . . 48
4.2.25 MHA . . . 48
4.2.26 MM . . . 48
4.2.27 OrbitPeriod . . . 48
4.2.28 PercentShadow . . . 48
4.2.29 RA . . . 49
4.2.30 RAV . . . 49
4.2.31 RAAN . . . 50
4.2.32 RadApo . . . 50
4.2.33 RadPer . . . 50
4.2.34 RMAG . . . 50
4.2.35 SemilatusRectum . . . 50
4.2.36 SMA . . . 50
4.2.37 TA . . . 51
4.2.38 TAIModJulian . . . 51
4.2.39 TTModJulian . . . 51
4.2.40 TTGregorian . . . 51
4.2.41 Umbra and Penumbra . . . 51
4.2.42 UTCModJulian . . . 52
4.2.43 VelApoapsis . . . 52
4.2.44 VelPeriapsis . . . 53
4.2.45 VMAG . . . 53
4.3 Other Calculations . . . 53
4.3.1 MA to TA . . . 53
4.3.2 EA to TA . . . 53
4.3.3 HA to TA . . . 54
4.4 Libration Points . . . 54
4.5 Barycenter . . . 56

5 Dynamics Modelling **57**

Draft: Work in Progress

5.1 Equations of Motion . . . 57
5.1.1 Orbit State Equations . . . 57
5.1.2 State Transition Matrix Equations . . . 58
5.1.3 Multiple Spacecraft Propgation and Coupled Propagation of the Equations of Motion . . . 58
5.2 Force Modelling . . . 58
5.2.1 Non-Spherical Gravity . . . 58
5.2.2 n-Body Point Mass Gravity . . . 60
5.2.3 Atmospheric Drag . . . 62
5.2.4 Solar Radiation Pressure . . . 62
5.2.5 Spacecraft Thrust . . . 62
5.3 Environment Modelling . . . 62
5.3.1 Celestial Body Ephemeris . . . 62
5.3.2 Analytic Ephemeris Model . . . 62
5.3.3 Atmospheric Density . . . 62
5.3.4 Jacchia Roberts . . . 62

6 Attitude **63**
6.1 Attitude Propagation . . . 63
6.1.1 Internal State Representation and Attitude Initial Conditions . . . 63
6.1.2 Kinematic Attitude Propagation . . . 63
6.2 Attitude Parameterizationsand Conversions . . . 65
6.2.1 Conversion: Quaternions to DCM . . . 65
6.2.2 Conversion: DCM to Quaternions . . . 65
6.2.3 Conversion: DCM to EulerAxis/Angle . . . 65
6.2.4 Conversion: Euler Axis/Angle to DCM . . . 66
6.2.5 Conversion: Euler Angles to DCM . . . 66
6.2.6 Conversion: DCM to Euler Angles . . . 66
6.2.7 Conversion: Angular Velocity toEuler Angles Rates . . . 66
6.2.8 Conversion: Euler Angles Rates to Angular Velocity . . . 67
6.2.9 Conversion: Quaternions to Euler Angles . . . 67
6.2.10 Conversion: Euler Angles to Quaternions . . . 67
6.3 Appendix 1 . . . 67

7 Numerical Integrators **71**
7.1 Runge-Kutta Integrators . . . 71
7.1.1 Constructor & Destructor Documentation . . . 72
7.2 Prince-Dormand Integrators . . . 72
7.3 Adams Bashforth Moulton . . . 72
7.4 Bulirsch-Stoer . . . 72
7.5 Stopping Condition Algorithm . . . 72
7.6 Integrator Coefficients . . . 72

8 Mathematics in GMAT Scripting **75**
8.1 Basic Operators . . . 75
8.2 Math Functions . . . 75
8.2.1 max . . . 75
8.2.2 min . . . 75
8.2.3 abs . . . 75
8.2.4 mean . . . 75
8.2.5 dot . . . 75
8.2.6 cross . . . 75
8.2.7 norm . . . 75
8.2.8 det . . . 75
8.2.9 inv . . . 75
8.2.10 eig . . . 76
8.2.11 sin, cos, tan . . . 76
8.2.12 asin, acos, atan, atan2 . . . 76

Draft: Work in Progress

8.2.13 sinh, cosh, tanh . . . 76
8.2.14 asinh, acosh, atanh . . . 76
8.2.15 transpose . . . 76
8.2.16 DegToRad . . . 76
8.2.17 RadToDeg . . . 76
8.2.18 log . . . 76
8.2.19 log10 . . . 76
8.2.20 exp . . . 76
8.2.21 sqrt . . . 76

9 Solvers **77**
9.1 Differential Correction . . . 77
9.2 Broyden's Method . . . 77
9.3 Newton's Method . . . 77
9.4 Matlab fmincon . . . 77
9.5 The Vary Command . . . 77

10 Event Functions **79**
10.1 Event Function Mathematical Definition . . . 79
10.2 Issues in Locating Zero Crossings . . . 80
10.3 Root Finding Options in GMAT . . . 80
10.4 Algorithm for Event Functions . . . 81
10.5 Appendix 1: Root Finding Algorithms . . . 84
10.6 Quadratic Polynomial . . . 84
10.7 Cubic Spline (Not-a-Knot) . . . 84

11 Spacecraft Model **87**
11.1 Orbit . . . 87
11.2 Attitude . . . 87
11.3 Ballistics and Mass . . . 87
11.4 Actuators . . . 87
11.4.1 Thrust and Impulse Models . . . 87
11.5 Sensors . . . 88
11.6 Tanks . . . 88

12 MathSpecAppendices **91**
12.1 Vector Identities . . . 91

Draft: Work in Progress

List of Figures

3.1 Illustration of a Translating and Rotating Coordinate System 16
3.2 General Coordinate System Transformation Approach in GMAT 18
3.3 Inertial Motion of Earth's Spin Axis 19
3.4 Intermediate Transformations and Coordinate Systems in FK5 Reduction 20
3.5 IAU Definition of Pole and Meridian Locations for Planets and Moons 23
3.6 Diagram of an Object Referenced Coordinate System 27
3.7 Coordinate System Transformation Algorithm 32

4.1 The Keplerian Elements 34
4.2 The Spherical Elements 36
4.3 Geometry of the B-Plane as Seen From a Viewpoint Perpendicular to the B-Plane 43
4.4 The B-Vector as Seen From a Viewpoint Perpendicular to Orbit Plane 43
4.5 Geocentric and Geodetic Latitude 46
4.6 Local Sidereal Time Geometry 47
4.7 Shadow Geometry 48
4.8 Occultation Geometry in Calculation of PercentShadow 49
4.9 Geometry of Umbra and Penumbra Regions 52
4.10 Geometry of Libration Points 55
4.11 Location of Libration Points 56

5.1 N-Body Illustration 60

10.1 Sample Event Function Output 81
10.2 Initializations for the Event Location Algorithm 82
10.3 Event Location Algorithm 83

Draft: Work in Progress

List of Tables

3.1 Recommended Values for Pole and Prime Meridian Locations of the Sun and Planets[4] 30
3.2 Recommended Values for Pole and Prime Meridian Locations of Luna[4] 31

4.1 The Cartesian State 35
4.2 The Keplerian Elements (also see Fig. 4.1) 35
4.3 Keplerian Elements for Special Cases 35
4.4 The Equinoctial Elements 38
4.5 The Spherical Elements 39
4.6 The Modified Keplerian Elements 41
4.7 Location of Libration Points in RLP Frame, with the Origin at the Primary Body 55

6.1 Rotation Matrices for 12 Unique Euler Angle Rotation Sequences 68
6.2 Kinematics of Euler Angle Rotation Sequences 69
6.3 Computation of Euler Angles from DCM 70

7.1 Prince-Dormand 45 Coefficients 73
7.2 Prince-Dormand 56 Coefficients 73
7.3 Runge-Kutta-Fehlberg 56 Coefficients 74

9.1 Available Commands in an fmincon Loop 77

10.1 Allowable Values for **d** in Event Function Output 79
10.2 Variables in Event Function Algorithm 81

11.2 Default Fuel Tank Parameters 88
11.1 Default Thrust and Specific Impulse Coefficients 89

Draft: Work in Progress

Draft: Work in Progress

Chapter 1

Introduction

Draft: Work in Progress

Draft: Work in Progress

Chapter 2

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.

2.1 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.

2.1.1 Atomic Time: TAI and A.1

Atomic time (AT) is a highly accurate time system which is independent of the rotation of the Earth.[1] 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.

$$A.1 = TAI + 0.0343817sec \tag{2.1}$$

where

$$TAI = UTC + \Delta AT \tag{2.2}$$

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.

2.1.2 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

$$UTC = UT1 - \Delta UT1 \tag{2.3}$$

Draft: Work in Progress

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.

2.1.3 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:

$$TT = TAI + 32.184 \text{ sec} \tag{2.4}$$

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

$$TDB \approx TT + \underbrace{0.001658 \sin M_E + 0.00001385 \sin 2M_E}_{\text{units of seconds}} \tag{2.5}$$

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

$$M_E \approx 357.5277233 + 35,999.05034 T_{TT} \tag{2.6}$$

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

$$T_{TT} = \frac{JD_{TT} - 2,451,545.0}{36,525} \tag{2.7}$$

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[1] (pp. 195-198) and Seidelmann[2] (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.[2] 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:

$$TCB - TDB = L_B(JD - 2443144.5)86400 \tag{2.8}$$

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.[3] 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.[4] Note that Seidelmann[4] 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.

2.2 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

Draft: Work in Progress

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 : 00.000	TT
01 Jan 2000	11 : 59 : 27.815986276	TAI
01 Jan 2000	11 : 59 : 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.

2.2.1 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[2] (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.[2] 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[1]

$$JD = 367Y - Int\left(\frac{7\left(Y + Int\left(\frac{M+9}{12}\right)\right)}{4}\right) + Int\left(\frac{275M}{9}\right) + D + 1,721,013.5 + \frac{\frac{\frac{s}{60}+m}{60}+H}{24} \tag{2.9}$$

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

$$MJD = JD - 2,430,000.0 \tag{2.10}$$

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:

$$\text{JDay} = 367Y - Int\left(\frac{7\left(Y + Int\left(\frac{M+9}{12}\right)\right)}{4}\right) + Int\left(\frac{275M}{9}\right) + D + 1,721,013.5 - 2,430,000.0 \tag{2.11}$$

$$\text{PartofDay} = \frac{\frac{\frac{s}{60}+m}{60}+H}{24} \tag{2.12}$$

$$\text{MJD} = (\text{JDay} - 2,430,000.0) + \text{PartofDay} \tag{2.13}$$

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.

2.2.2 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 MJD is taken from Vallado[1] 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

Draft: Work in Progress

$Year = Year - 1$

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

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

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

$\tau = (Days - DayofYr)24$

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

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

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

2.3 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.

Draft: Work in Progress

Chapter 3

Coordinate Systems

There are numerous coordinate systems used in space mission analysis, that when used appropriately can greatly simplify the work and yield insight that is not obvious otherwise. Some examples are equatorial and ecliptic systems, and rotating coordinate systems based on the relative motion of two bodies such as the Earth and Moon. GMAT has the capability to calculate many parameters in different coordinate systems, and these parameters can then be used in plots, reports, solvers, control flow statements and stopping conditions to name a few.

In this chapter we investigate how GMAT performs coordinate system transformations, and how different coordinate systems are defined. We begin by defining some notation. Next, we look at how to transform a vector and its first derivative from one coordinate system to another when the coordinate systems are translating and rotating with respect to one another. Finally, we look at each coordinate system defined in GMAT and discuss how to find its rotation matrix and the first derivative of the rotation matrix to rotate to the J2000 coordinate system.

3.1 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.

Let's start by defining some notation. In Fig. 3.1, we see an illustration of a point o drawn with respect to two coordinate systems $\mathcal{F}_i$ and $\mathcal{F}_f$. The vector $\mathbf{r}_i^o$ is the position vector of point o with respect to frame $\mathcal{F}_i$. The vector $\mathbf{r}_f^o$ is the position vector of point o with respect to $\mathcal{F}_f$. $\mathbf{r}_{fi}$ is the vector from the origin of $\mathcal{F}_i$ to $\mathcal{F}_f$. Let's define the rotation matrix that rotates from $\mathcal{F}_i$ to $\mathcal{F}_f$ as $\mathbf{R}_{fi}$. Finally, let's define the angular velocity $\boldsymbol{\omega}_{fi}$ as the angular velocity of $\mathcal{F}_i$ with respect to $\mathcal{F}_f$. To simplify the notation, we assume that a vector is expressed in the frame denoted by the right-most subscript. If we need to express a vector in another coordinate system, we use curly brackets. As an example, $\{\mathbf{r}_i^o\}_f$ is the position of point o, with respect to frame $\mathcal{F}_i$, expressed in frame $\mathcal{F}_f$. In summary, we have

$\mathbf{r}_i^o$	Position vector of point o w/r/t frame $\mathcal{F}_i$ expressed in $\mathcal{F}_i$
$\{\mathbf{r}_i^o\}_f$	Position vector of point o w/r/t frame $\mathcal{F}_i$ expressed in $\mathcal{F}_f$
$\mathbf{r}_{fi}$	Position vector from origin of $\mathcal{F}_i$ to origin of $\mathcal{F}_f$ expressed in $\mathcal{F}_f$
$\mathbf{R}_{fi}$	Rotation matrix from frame $\mathcal{F}_i$ to $\mathcal{F}_f$
$\boldsymbol{\omega}_{fi}$	Angular velocity of frame $\mathcal{F}_i$ w/r/t $\mathcal{F}_f$, expressed in frame $\mathcal{F}_i$
$\{\boldsymbol{\omega}_{fi}\}_f$	Angular velocity of frame $\mathcal{F}_i$ w/r/t $\mathcal{F}_f$, expressed in frame $\mathcal{F}_f$

To further simplify the notation, let's drop the superscript "o" from $\mathbf{r}_i^o$ and $\mathbf{r}_f^o$. Then, from inspection of Figure 3.1 we can write

$$\mathbf{r}_f = \underbrace{\mathbf{R}_{fi}\mathbf{r}_i}_{Rot.} + \underbrace{\mathbf{r}_{if}}_{Trans.} \tag{3.1}$$

Equation (3.1) 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. (3.1) 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. (3.1) with respect to time.

$$\frac{d\mathbf{r}_f}{dt} = \frac{d\mathbf{R}_{fi}\mathbf{r}_i}{dt} + \frac{d\mathbf{r}_{if}}{dt} \tag{3.2}$$

Draft: Work in Progress

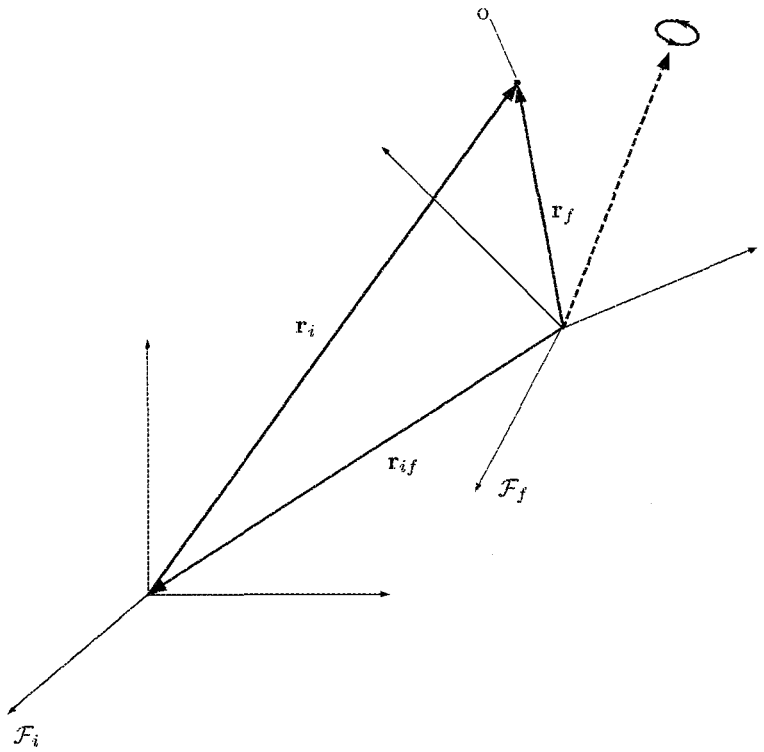

Figure 3.1: Illustration of a Translating and Rotating Coordinate System

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

$$\dot{\mathbf{r}}_f = \dot{\mathbf{R}}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i + \dot{\mathbf{r}}_{if} \tag{3.3}$$

In Eq. (3.3) 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

$$\dot{\mathbf{R}}_{fi} = \mathbf{R}_{fi}\omega_{fi}^{\times} = \{\omega_{fi}^{\times}\}_f\mathbf{R}_{fi} \tag{3.4}$$

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

$$\boldsymbol{\omega}^x = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \tag{3.5}$$

In summary, using Eq. (3.4) to transform a derivative vector from $\mathcal{F}_i$ to $\mathcal{F}_f$ we can use any of the following three equations:

$$\dot{\mathbf{r}}_f = \underbrace{\mathbf{R}_{fi}\omega_{fi}^{\times}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{3.6}$$

$$\dot{\mathbf{r}}_f = \underbrace{\{\omega_{fi}^{\times}\}_f\mathbf{R}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{3.7}$$

$$\dot{\mathbf{r}}_f = \underbrace{\dot{\mathbf{R}}_{fi}\mathbf{r}_i + \mathbf{R}_{fi}\dot{\mathbf{r}}_i}_{Rot.} + \underbrace{\dot{\mathbf{r}}_{if}}_{Trans.} \tag{3.8}$$

We choose between Eqs. (3.6), (3.7), or (3.8) 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 $\mathcal{F}_{J2k}$, 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. 3.2. 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

Draft: Work in Progress

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. (3.1) and (3.8) we see that

$$\{\mathbf{r}_i\}_1 = \mathbf{R}_{J_{2k},i}\mathbf{r}_i \tag{3.9}$$

$$\{\dot{\mathbf{r}}_i\}_1 = \dot{\mathbf{R}}_{J_{2k},i}\mathbf{r}_i + \mathbf{R}_{J_{2k},i}\dot{\mathbf{r}}_i \tag{3.10}$$

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.3.2 we can see that

$$\{\mathbf{r}_i\}_1 = \{\mathbf{r}_{Ri}\}_{J_{2k}} + \{\mathbf{r}_{fR}\}_{J_{2k}} + \{\mathbf{r}_f\}_2 \tag{3.11}$$

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

$$\{\mathbf{r}_f\}_2 = \{\mathbf{r}_i\}_1 - \{\mathbf{r}_{Ri}\}_{J_{2k}} - \{\mathbf{r}_{fR}\}_{J_{2k}} \tag{3.12}$$

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 .(3.12) to obtain

$$\{\mathbf{v}_f\}_2 = \{\mathbf{v}_i\}_1 - \{\mathbf{v}_{Ri}\}_{J_{2k}} - \{\mathbf{v}_{fR}\}_{J_{2k}} \tag{3.13}$$

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.

$$\mathbf{r}_f = \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{3.14}$$

$$\dot{\mathbf{r}}_f = \dot{\mathbf{R}}_{f,J_{2k}}\{\mathbf{r}_f\}_2 + \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{3.15}$$

3.2 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. (3.9) and (3.14). 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. (3.10) and (3.15) become

$$\left\{\frac{d\mathbf{r}_i}{dt}\right\}_1 = \mathbf{R}_{J_{2k},i}\frac{d\mathbf{r}_i}{dt} \tag{3.16}$$

and

$$\frac{d\mathbf{r}_f}{dt} = \mathbf{R}_{f,J_{2k}}\{\mathbf{v}_f\}_2 \tag{3.17}$$

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 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.

$$\mathbf{R}_{i,J_{2k}} = \mathbf{R}^{-1}_{J_{2k},i} = \mathbf{R}^T_{J_{2k},i} \tag{3.18}$$

and

$$\dot{\mathbf{R}}_{if} = \dot{\mathbf{R}}^{-1}_{J_{2k},i} = \dot{\mathbf{R}}^T_{J_{2k},i} \tag{3.19}$$

3.3 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.

3.3.1 Overview of FK5 Reduction

The inertial system most commonly used in astrodynamics as of this writing is the FK5 system. We call this

Draft: Work in Progress

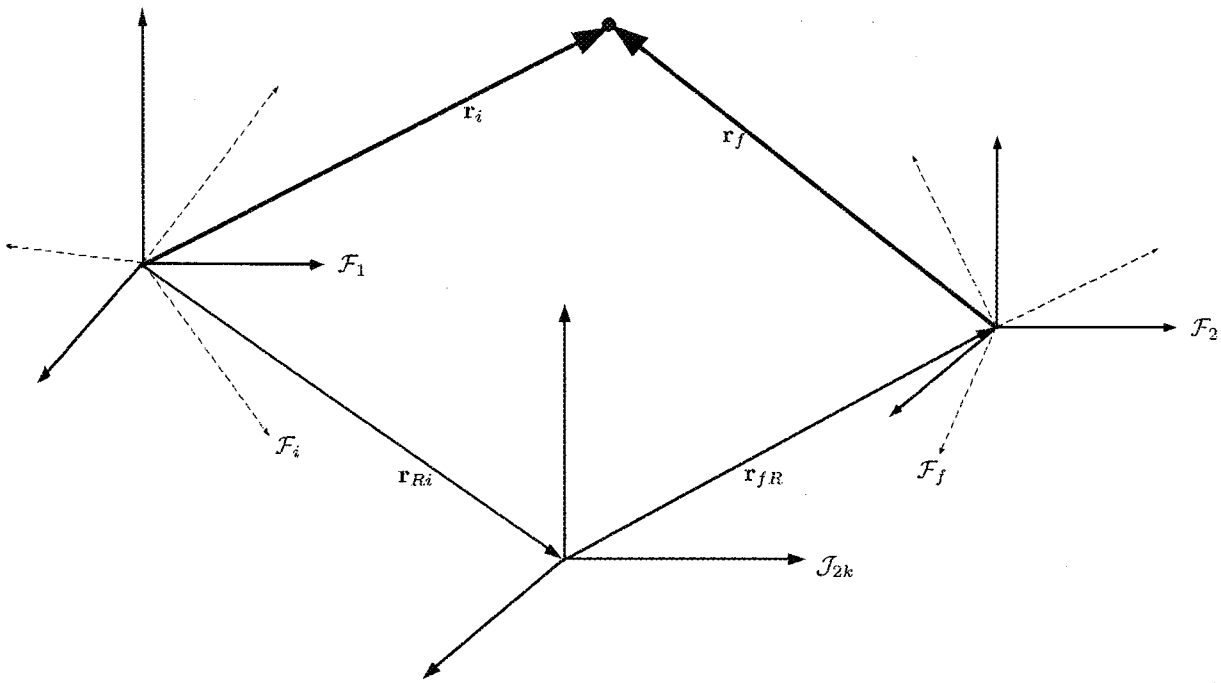

Figure 3.2: General Coordinate System Transformation Approach in GMAT

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.[2] The motion is illustrated in Fig. 3.3. 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.

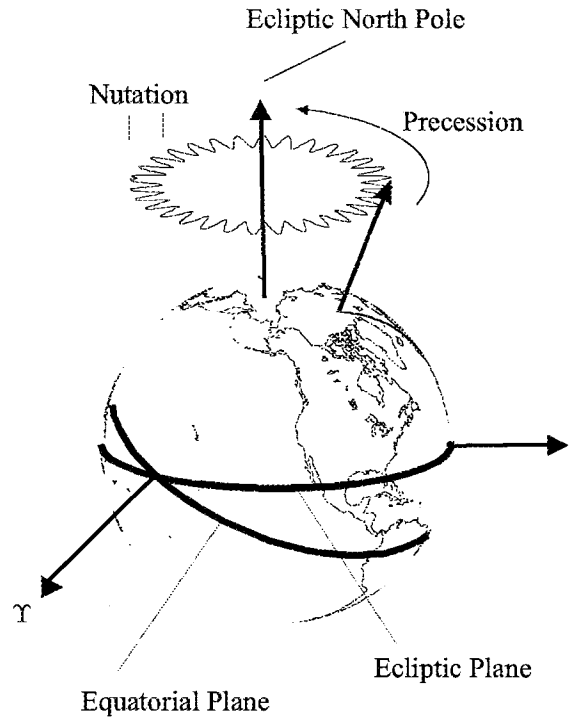

Figure 3.3: Inertial Motion of Earth's Spin Axis

Sidereal Time

3.3.2 Precession Calculations

$$JD_{TDB} \approx JD_{TT} \tag{3.20}$$

$$T_{TDB} = \frac{JD_{TDB} - 2,451,545.0}{36525} \tag{3.21}$$

$$\zeta = 2306.2181'' T_{TDB} + 0.30188 T_{TDB}^2 + 0.017998 T_{TDB}^3 \tag{3.22}$$

$$\Theta = 2004.3109'' T_{TDB} - 0.42665 T_{TDB}^2 - 0.041833 T_{TDB}^3 \tag{3.23}$$

$$z = 2306.2181'' T_{TDB} + 1.09468 T_{TDB}^2 + 0.018203 T_{TDB}^3 \tag{3.24}$$

$$\mathbf{P} = \mathbf{R}_3(-z)\mathbf{R}_2(\Theta)\mathbf{R}_3(-\zeta) \tag{3.25}$$

3.3.3 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

Draft: Work in Progress

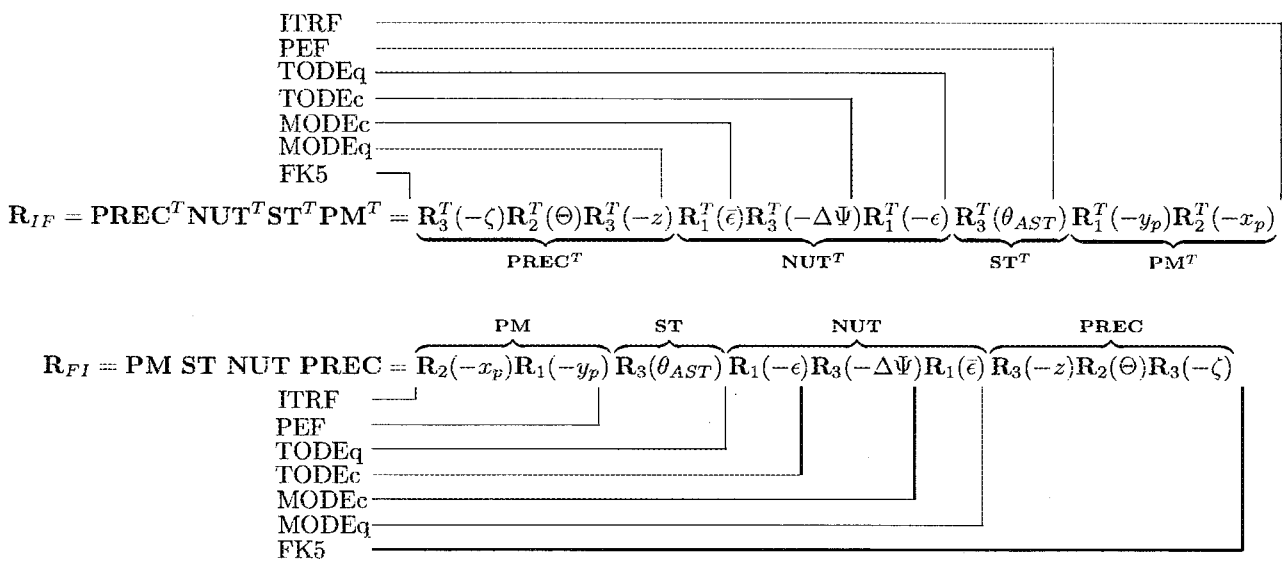

Figure 3.4: Intermediate Transformations and Coordinate Systems in FK5 Reduction

$$\bar{\epsilon} = 84381.448 - 46.8150T_{TDB} - 0.00059T_{TDB}^2 + 0.001813T_{TDB}^3 \tag{3.26}$$

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:

$$l = 134.96340251^\circ + (1717915923.2178T_{TDB} + 31.8792T_{TDB}^2 + 0.051635T_{TDB}^3 - 0.00024470T_{TDB}^4)'' \tag{3.27}$$

$$l' = 357.52910918^\circ + (129596581.0481T_{TDB} - 0.5532T_{TDB}^2 - 0.000136T_{TDB}^3 - 0.00001149T_{TDB}^4)'' \tag{3.28}$$

$$F = 93.27209062^\circ + (1739527262.8478T_{TDB} - 12.7512T_{TDB}^2 + 0.001037T_{TDB}^3 + 0.00000417T_{TDB}^4)'' \tag{3.29}$$

$$D = 297.85019547^\circ + (1602961601.2090T_{TDB} - 6.3706T_{TDB}^2 + 0.006593T_{TDB}^3 - 0.00003169T_{TDB}^4)'' \tag{3.30}$$

$$\Omega = 125.04455501^\circ + (-6962890.2665T_{TDB} + 7.4722T_{TDB}^2 + 0.007702T_{TDB}^3 - 0.00005939T_{TDB}^4)'' \tag{3.31}$$

1980 Nutation Theory

$$\Delta\Psi_{1980} = \sum_{i=1}^{106}(A_i + B_iT_{TDB})\sin a_p \tag{3.32}$$

$$\Delta\epsilon_{1980} = \sum_{i=1}^{106}(C_i + D_iT_{TDB})\cos a_p \tag{3.33}$$

$$a_p = a_{1i}l + a_{2i}l' + a_{3i}F + a_{4i}D + a_{5i}\Omega \tag{3.34}$$

$$\mathbf{N} = \mathbf{R}_1(-\epsilon)\mathbf{R}_3(-\Delta\Psi)\mathbf{R}_1(\bar{\epsilon}) \tag{3.35}$$

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,

Draft: Work in Progress

according to the 1996 IERS theory:

$$\Delta\Psi_{1996} = \sum_{i=1}^{263} (A_i + B_i T_{TDB}) \sin a_p + E_i \cos a_p \quad (3.36)$$

$$\Delta\epsilon_{1996} = \sum_{i=1}^{263} (C_i + D_i T_{TDB}) \cos a_p + F_i \sin a_p \quad (3.37)$$

$$a_p = a_{1i} M_\circ + a_{2i} M_\odot + a_{3i} u_{M\circ} + a_{4i} D_\odot + a_{5i} \Omega_\circ \quad (3.38)$$

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:

$$\Delta\Psi_{pl} = \sum_{i=1}^{118} (A_i + B_i T_{TDB}) \sin a_{pl} \quad (3.39)$$

$$\Delta\epsilon_{pl} = \sum_{i=1}^{118} (C_i + D_i T_{TDB}) \cos a_{pl} \quad (3.40)$$

$$\begin{aligned}
a_{pl} = \; & a_{1i}\lambda_V + a_{2i}\lambda_E + a_{3i}\lambda_M + a_{4i}\lambda_J + a_{5i}\lambda_S + \\
& a_{6i} p_a + a_{7i} D + a_{8i} F + a_{9i} l + a_{10i} \Omega \quad (3.41)
\end{aligned}$$

$$\Delta\Psi = \Delta\Psi_{1996} + \underbrace{\Delta\Psi_{pl}}_{optional} \quad (3.42)$$

$$\Delta\epsilon = \Delta\epsilon_{1996} + \underbrace{\Delta\epsilon_{pl}}_{optional} \quad (3.43)$$

$$\epsilon = \bar{\epsilon} + \Delta\epsilon \quad (3.44)$$

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

$$\mathbf{N} = \mathbf{R}_1(-\epsilon)\mathbf{R}_3(-\Delta\Psi)\mathbf{R}_1(\bar{\epsilon}) \quad (3.45)$$

3.3.4 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:

$$\begin{aligned}
\theta_{GMST} = & 1.00965822615e6 + 4.746600277219299e10 T_{UT1} \\
& + 1.396560 T_{UT1}^2 + 9.3e-5 T_{UT1}^3 \quad \text{(arcseconds)} \quad (3.46)
\end{aligned}$$

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

$$EQ_{equinox} = \Delta\Psi \cos\epsilon + 0.00264'' \sin\Omega + 0.000063'' \sin 2\Omega \quad (3.47)$$

If the Julian date falls on or before 2450449.5 we use

$$EQ_{equinox} = \Delta\Psi \cos\epsilon \quad (3.48)$$

$$\theta_{AST} = \theta_{GMST} + EQ_{equinox} \quad (3.49)$$

$$\mathbf{ST} = \mathbf{R}_3(\theta_{AST}) \quad (3.50)$$

3.3.5 Polar Motion Calculations

$$\mathbf{PM} = \mathbf{R}_2(-x_p)\mathbf{R}_1(-y_p) \quad (3.51)$$

3.4 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.

3.4.1 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.

Draft: Work in Progress

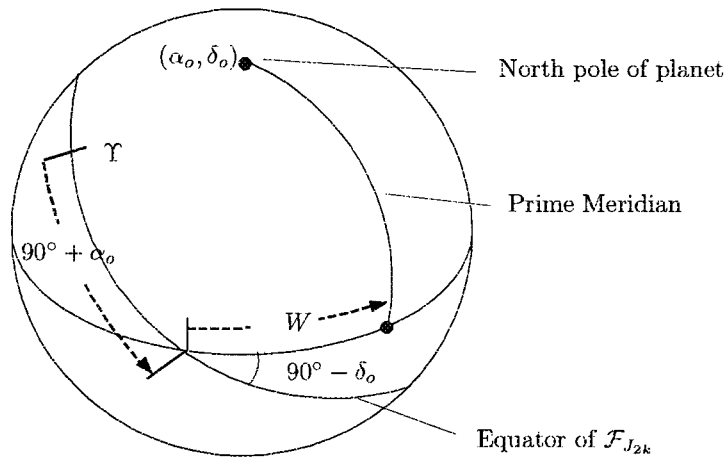

Figure 3.5: IAU Definition of Pole and Meridian Locations for Planets and Moons

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. 3.3.3 and 3.3.2 . Because t_o is fixed for a TOEEq system, the derivative of $\mathbf{R}_{Ii}$ is identically equal to the zero matrix.

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.61}$$

3.4.4 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

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o) \tag{3.62}$$

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. 3.3.2. Because t_o is fixed for a MOEEq system, the derivative of $\mathbf{R}_{Ii}$ is the zero matrix.

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.63}$$

3.4.5 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[1] Fig. 3-29).

$$\mathbf{R}_{Ii} = \mathbf{N}^T(t_o)\mathbf{P}^T(t_o) \tag{3.64}$$

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

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.65}$$

3.4.6 Mean of Date Equator (MODEq)

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

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o) \tag{3.66}$$

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

Draft: Work in Progress

and can be found in Vallado[1] 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

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.67}$$

3.4.7 Mean of Date Ecliptic (MODEc)

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

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o)\mathbf{R}_1^T(\bar{\epsilon}) \tag{3.68}$$

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[1] pgs. 214 - 219. $\bar{\epsilon}$ is given by Vallado,[1] Eq. (3-52). For a more complete discussion, you can also refer to Seidelmann,[2] 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

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.69}$$

3.4.8 True of Date Ecliptic (TODEc)

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

$$\mathbf{R}_{Ii} = \mathbf{P}^T(t_o)\mathbf{R}_1^T(\bar{\epsilon})\mathbf{R}_3^T(-\Delta\Psi) \tag{3.70}$$

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,[1] Eq. (3-52). $\Delta\Psi$ is given by Eq.(3-62) in Vallado.[1] For a more complete discussion, you can also refer to Seidelmann,[2] 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

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.71}$$

3.4.9 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.

$$\mathbf{R}_{Ii} = \mathbf{P}^T\mathbf{N}^T\mathbf{S}\mathbf{T}^T\mathbf{P}\mathbf{M}^T \tag{3.72}$$

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

$$\dot{\mathbf{R}}_{Ii} = \mathbf{P}^T\mathbf{N}^T\dot{\mathbf{S}\mathbf{T}}^T\mathbf{P}\mathbf{M}^T \tag{3.73}$$

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

$$\dot{\mathbf{S}\mathbf{T}} = \begin{pmatrix} -\omega_e \sin\theta_{AST} & \omega_e \cos\theta_{AST} & 0.0 \\ -\omega_e \cos\theta_{AST} & -\omega_e \sin\theta_{AST} & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.74}$$

and ω_e is given by

$$\omega_e = 7.29211514670698e^{-5}\left(1 - \frac{LOD}{86400}\right) \tag{3.75}$$

Note that the 2nd edition of Vallado[1] has inconsistencies in Eqs. (3.72) and (3.73) and they are discussed in the errata to the 2nd edition. We have modified equations Eqs. (3.72) and (3.73) 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[4] *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

$$\mathbf{R}_{Ii} = \mathbf{R}_3^T(90^\circ + \alpha_o)\mathbf{R}_1^T(90^\circ - \delta_o)\mathbf{R}_3^T(W) \tag{3.76}$$

For the derivative we assume that

$$\frac{d}{dt}\left(\mathbf{R}_3^T(90^\circ + \alpha_o)\right) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.77}$$

and

$$\frac{d}{dt}\left(\mathbf{R}_1^T(90^\circ - \delta_o)\right) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.78}$$

Draft: Work in Progress

$$\frac{d}{dt}\left(\mathbf{R}_3^T(W)\right) = \begin{pmatrix} -\dot{W}\sin(W) & -\dot{W}\cos(W) & 0.0 \\ \dot{W}\cos(W) & -\dot{W}\sin(W) & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.79}$$

where $\dot{W}$ is the time derivative of W for the given body. Note, Seidelmann[4] does not provide the values for $\dot{W}$ so we include them in Table 3.1. In summary

$$\dot{\mathbf{R}} = \mathbf{R}_3^T(90^\circ + \alpha_o)\mathbf{R}_1^T(90^\circ - \delta_o)\frac{d}{dt}\mathbf{R}_3^T(W) \tag{3.80}$$

3.4.10 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[4] 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:

$$\dot{\mathbf{R}}_{Ii} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.81}$$

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:

$$\alpha = \alpha_o(T_{J2000}) \tag{3.82}$$

$$\delta = \delta_o(T_{J2000}) \tag{3.83}$$

Where $T_{J2000} = 2451544.9999999990686$ TDB and the equations for α_o and δ_o are given by Seidelmann[4] and reproduced in Table 3.1. Finally, the rotation matrix is calculated using

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3^T(90^\circ + \alpha)\mathbf{R}_1^T(90^\circ - \delta) \tag{3.84}$$

The result is a rotation matrix, that is time invariant, for each celestial body. The individual matrices are shown below.

For the Sun

$$\begin{aligned} R_{11} &= 0.9606338208497771 \\ R_{21} &= 0.2778176780544363 \\ R_{31} &= 0 \\ R_{12} &= -0.2494239059807128 \\ R_{22} &= 0.8624542595398803 \\ R_{32} &= 0.4404093156676427 \\ R_{13} &= 0.1223534934723278 \\ R_{23} &= -0.4230720836476433 \\ R_{33} &= 0.8977971010607901 \end{aligned} \tag{3.85}$$

For Mercury

$$\begin{aligned} R_{11} &= 0.9815938660446788 \\ R_{21} &= 0.1909803187332696 \\ R_{31} &= 0 \\ R_{12} &= -0.1677571842642237 \\ R_{22} &= 0.8622324234816705 \\ R_{32} &= 0.4779254910806334 \\ R_{13} &= 0.09127436261733374 \\ R_{23} &= -0.4691287304711406 \\ R_{33} &= 0.8784003785150228 \end{aligned} \tag{3.86}$$

For Venus

$$\begin{aligned} R_{11} &= 0.9988399975085459 \\ R_{21} &= 0.04815245972043356 \\ R_{31} &= 0 \\ R_{12} &= -0.04437694044018298 \\ R_{22} &= 0.9205233405740161 \\ R_{32} &= 0.3881590738545506 \\ R_{13} &= 0.01869081416890205 \\ R_{23} &= -0.3877088083617988 \\ R_{33} &= 0.9215923900425705 \end{aligned} \tag{3.87}$$

For the Earth

$$\begin{aligned} R_{11} &= 1 \\ R_{12} &= 0 \\ R_{13} &= 0 \\ R_{21} &= 0 \\ R_{22} &= 1 \\ R_{23} &= 0 \\ R_{31} &= 0 \\ R_{32} &= 0 \\ R_{33} &= 1 \end{aligned} \tag{3.88}$$

Draft: Work in Progress

For the Moon

$$\begin{aligned}
R_{11} &= 0.998496505205088 \\
R_{21} &= -0.0548154092680678 \\
R_{31} &= 0 \\
R_{12} &= 0.0499357293985327 \\
R_{22} &= 0.909610125238044 \\
R_{32} &= 0.412451018902689 \\
R_{13} &= -0.0226086714041825 \\
R_{23} &= -0.411830900942613 \\
R_{33} &= 0.91097977859343
\end{aligned} \tag{3.89}$$

For Mars

$$\begin{aligned}
R_{11} &= 0.6732521982472343 \\
R_{21} &= 0.7394129276360177 \\
R_{31} &= 0 \\
R_{12} &= -0.5896387605430038 \\
R_{22} &= 0.5368794307891334 \\
R_{32} &= 0.6033958972853944 \\
R_{13} &= 0.4461587269353554 \\
R_{23} &= -0.4062376142607542 \\
R_{33} &= 0.7974417791532832
\end{aligned} \tag{3.90}$$

For Jupiter

$$\begin{aligned}
R_{11} &= 0.9994209020316729 \\
R_{21} &= -0.03402735050216735 \\
R_{31} &= 0 \\
R_{12} &= 0.0307100286015568 \\
R_{22} &= 0.9019874904580493 \\
R_{32} &= 0.430668621100356 \\
R_{13} &= -0.01465451212046692 \\
R_{23} &= -0.4304192217768545 \\
R_{33} &= 0.9025101322420253
\end{aligned} \tag{3.91}$$

For Saturn

$$\begin{aligned}
R_{11} &= -0.6506284356468798 \\
R_{21} &= 0.7593962330217962 \\
R_{31} &= 0 \\
R_{12} &= -0.7545700815904118 \\
R_{22} &= -0.6464935305479939 \\
R_{32} &= 0.1125615694996715 \\
R_{13} &= 0.08547883186107168 \\
R_{23} &= 0.07323575787752883 \\
R_{33} &= 0.9936447519469775
\end{aligned} \tag{3.92}$$

For Uranus

$$\begin{aligned}
R_{11} &= 0.9755767334083795 \\
R_{21} &= -0.2196589111150187 \\
R_{31} &= 0 \\
R_{12} &= -0.05749969269512644 \\
R_{22} &= -0.2553748540714755 \\
R_{32} &= 0.9651308042166816 \\
R_{13} &= -0.2119995815377986 \\
R_{23} &= -0.9415591572895125 \\
R_{33} &= -0.2617680858165513
\end{aligned} \tag{3.93}$$

For Neptune

$$\begin{aligned}
R_{11} &= 0.8717809455009272 \\
R_{21} &= 0.4898958900230839 \\
R_{31} &= 0 \\
R_{12} &= -0.3337976487639454 \\
R_{22} &= 0.5940005535292547 \\
R_{32} &= 0.7319443094160927 \\
R_{13} &= 0.3585765089087282 \\
R_{23} &= -0.6380951021167846 \\
R_{33} &= 0.6813644604126335
\end{aligned} \tag{3.94}$$

For Pluto

$$\begin{aligned}
R_{11} &= 0.7311155947298647 \\
R_{21} &= 0.6822536091093958 \\
R_{31} &= 0 \\
R_{12} &= -0.1077863335431382 \\
R_{22} &= 0.1155058299435205 \\
R_{32} &= 0.9874413955017208 \\
R_{13} &= 0.6736854558650673 \\
R_{23} &= -0.7219338031331282 \\
R_{33} &= 0.1579857286263988
\end{aligned} \tag{3.95}$$

3.4.11 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. 3.6 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 **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 **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

Draft: Work in Progress

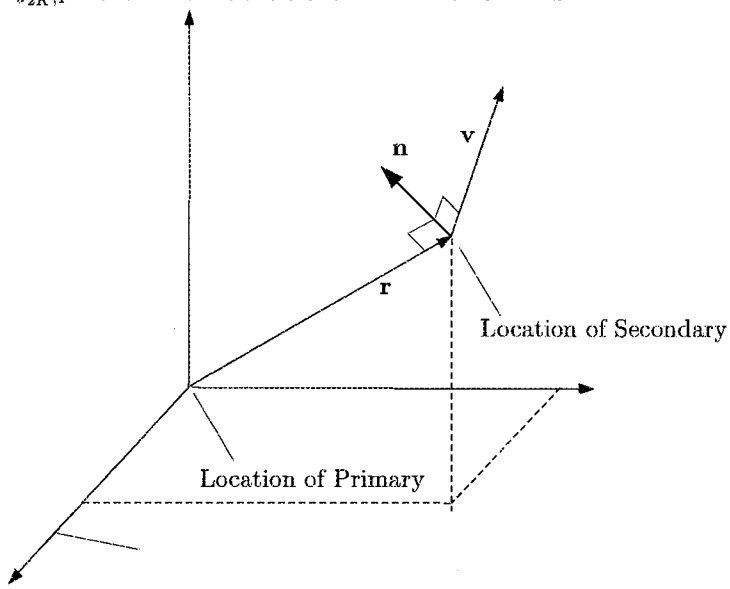

Figure 3.6: Diagram of an Object Referenced Coordinate System

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 $\dot{\mathbf{R}}_{Ii}$.

$$\mathbf{R}_{Ii} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{bmatrix} \tag{3.96}$$

$$\dot{\mathbf{R}}_{Ii} = \begin{bmatrix} \dot{\hat{\mathbf{x}}} & \dot{\hat{\mathbf{y}}} & \dot{\hat{\mathbf{z}}} \end{bmatrix} \tag{3.97}$$

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

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \tag{3.98}$$

and

$$\dot{\hat{\mathbf{z}}} = \dot{\hat{\mathbf{x}}} \times \hat{\mathbf{y}} + \hat{\mathbf{x}} \times \dot{\hat{\mathbf{y}}} \tag{3.99}$$

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

$$\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}} \tag{3.100}$$

and

$$\dot{\hat{\mathbf{x}}} = \dot{\hat{\mathbf{y}}} \times \hat{\mathbf{z}} + \hat{\mathbf{y}} \times \dot{\hat{\mathbf{z}}} \tag{3.101}$$

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

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{3.102}$$

and

$$\dot{\hat{\mathbf{y}}} = \dot{\hat{\mathbf{z}}} \times \hat{\mathbf{x}} + \hat{\mathbf{z}} \times \dot{\hat{\mathbf{x}}} \tag{3.103}$$

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:

$$\hat{\mathbf{r}} = \frac{\mathbf{r}}{\|\mathbf{r}\|} = \frac{\mathbf{r}}{r} \tag{3.104}$$

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{\mathbf{v}}{v} \tag{3.105}$$

$$\hat{\mathbf{n}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \tag{3.106}$$

$$\dot{\hat{\mathbf{r}}} = \frac{\mathbf{v}}{r} - \frac{\hat{\mathbf{r}}}{r}(\hat{\mathbf{r}} \cdot \mathbf{v}) \tag{3.107}$$

$$\dot{\hat{\mathbf{v}}} = \frac{\mathbf{a}}{v} - \frac{\hat{\mathbf{v}}}{v}(\hat{\mathbf{v}} \cdot \mathbf{a}) \tag{3.108}$$

$$\dot{\hat{\mathbf{n}}} = \frac{\mathbf{r} \times \mathbf{a}}{n} - \frac{\hat{\mathbf{n}}}{n}(\mathbf{r} \times \mathbf{a} \cdot \hat{\mathbf{n}}) \tag{3.109}$$

Draft: Work in Progress

3.4.12 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

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_{sun}}{\|\mathbf{r}_{sun}\|} \tag{3.110}$$

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

$$\hat{\mathbf{z}} = \frac{\mathbf{r}_{sun} \times \mathbf{v}_{sun}}{\|\mathbf{r}_{sun} \times \mathbf{v}_{sun}\|} \tag{3.111}$$

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

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{3.112}$$

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

$$\mathbf{R}_{Ii} = [\ \hat{\mathbf{x}} \quad \hat{\mathbf{y}} \quad \hat{\mathbf{z}}\ ] \tag{3.113}$$

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

$$\frac{d\hat{\mathbf{x}}}{dt} = \frac{\mathbf{v}_{sun}}{r_{sun}} - \hat{\mathbf{x}}\left(\hat{\mathbf{x}} \cdot \frac{\mathbf{v}_{sun}}{r_{sun}}\right) \tag{3.114}$$

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

$$\frac{d\hat{\mathbf{z}}}{dt} \approx \mathbf{0} \tag{3.115}$$

$$\frac{d\hat{\mathbf{y}}}{dt} = \hat{\mathbf{z}} \times \frac{d\hat{\mathbf{x}}}{dt} \tag{3.116}$$

$$\dot{\mathbf{R}}_{Ii} = \begin{bmatrix} \frac{d\hat{\mathbf{x}}}{dt} & \frac{d\hat{\mathbf{y}}}{dt} & \frac{d\hat{\mathbf{z}}}{dt} \end{bmatrix} \tag{3.117}$$

3.4.13 Geocentric Solar Magnetic (GSM)

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_{sun}}{\|\mathbf{r}_{sun}\|} \tag{3.118}$$

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.

$$\lambda_d = 70.1^\circ \text{ W} \tag{3.119}$$

$$\phi_d = 78.6^\circ \text{ N} \tag{3.120}$$

The dipole vector in the Earth Fixed system is simply

$$\{\mathbf{r}_d\}_F = \begin{pmatrix} \cos\phi_d \cos(-\lambda_d) \\ \cos\phi_d \sin(-\lambda_d) \\ \sin\phi_d \end{pmatrix} \tag{3.121}$$

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

$$\{\mathbf{r}_d\}_I = R_{IF}\{\mathbf{r}_d\}_F \tag{3.122}$$

Then, the y-axis is defined as

$$\hat{\mathbf{y}} = \frac{\{\mathbf{r}_d\}_I \times \hat{\mathbf{x}}}{\|\{\mathbf{r}_d\}_I \times \hat{\mathbf{x}}\|} \tag{3.123}$$

the z-axis is defined as

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \tag{3.124}$$

and

$$\mathbf{R}_{Ii} = [\ \hat{\mathbf{x}} \quad \hat{\mathbf{y}} \quad \hat{\mathbf{z}}\ ] \tag{3.125}$$

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

$$\frac{d\hat{\mathbf{x}}}{dt} = \frac{\mathbf{v}_{sun}}{r_{sun}} - \hat{\mathbf{x}}\left(\hat{\mathbf{x}} \cdot \frac{\mathbf{v}_{sun}}{r_{sun}}\right) \tag{3.126}$$

Let's define

$$\mathbf{y} = (R_{IF}\{\mathbf{r}_d\}_F) \times \hat{\mathbf{x}} \tag{3.127}$$

and

$$y = \|(R_{IF}\{\mathbf{r}_d\}_F) \times \hat{\mathbf{x}}\| \tag{3.128}$$

then

$$\frac{d\mathbf{y}}{dt} = \dot{\mathbf{y}} = \left(\dot{R}_{IF}\{\mathbf{r}_d\}_F\right) \times \hat{\mathbf{x}} + (R_{IF}\{\mathbf{r}_d\}_F) \times \frac{\hat{\mathbf{x}}}{dt} \tag{3.129}$$

Now we can write

$$\frac{d\hat{\mathbf{y}}}{dt} = \dot{\hat{\mathbf{y}}} = \frac{\dot{\mathbf{y}}}{y} - \hat{\mathbf{y}}\left(\hat{\mathbf{y}} \cdot \frac{\dot{\mathbf{y}}}{y}\right) \tag{3.130}$$

Finally,

$$\dot{\hat{\mathbf{z}}} = \dot{\hat{\mathbf{x}}} \times \hat{\mathbf{y}} + \hat{\mathbf{x}} \times \dot{\hat{\mathbf{y}}} \tag{3.131}$$

and

$$\dot{\mathbf{R}}_{Ii} = \begin{bmatrix} \frac{d\hat{\mathbf{x}}}{dt} & \frac{d\hat{\mathbf{y}}}{dt} & \frac{d\hat{\mathbf{z}}}{dt} \end{bmatrix} \tag{3.132}$$

Draft: Work in Progress

3.5 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:

$$\mathbf{n} = \mathbf{r} \times \mathbf{v} \tag{3.133}$$

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

$$\frac{d\mathbf{n}}{dt} = \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) = \underbrace{\frac{d\mathbf{r}}{dt} \times \mathbf{v}}_{0} + \mathbf{r} \times \frac{d\mathbf{v}}{\partial t} \tag{3.134}$$

so we know that

$$\boxed{\frac{d\mathbf{n}}{dt} = \mathbf{r} \times \mathbf{a}} \tag{3.135}$$

The next useful derivative is

$$\frac{dn}{dt} = \frac{d\|\mathbf{n}\|}{dt} = \frac{d}{dt}\left(\mathbf{n}^T\mathbf{n}\right)^{1/2} = \frac{\mathbf{n}^T}{n}\frac{d\mathbf{n}}{dt} \tag{3.136}$$

So we can write

$$\boxed{\frac{dn}{dt} = \frac{\mathbf{n}}{n} \cdot (\mathbf{r} \times \mathbf{a})} \tag{3.137}$$

The following two derivatives are also useful

$$\frac{dr}{dt} = \frac{d\|\mathbf{r}\|}{dt} = \frac{d}{dt}(\mathbf{r}^T\mathbf{r})^{1/2} = \frac{\mathbf{v} \cdot \mathbf{r}}{r} \tag{3.138}$$

$$\boxed{\frac{dr}{dt} = \frac{\mathbf{v} \cdot \mathbf{r}}{r}} \tag{3.139}$$

$$\frac{dv}{dt} = \frac{d\|\mathbf{v}\|}{dt} = \frac{d}{dt}(\mathbf{v}^T\mathbf{v})^{1/2} = \frac{\mathbf{v} \cdot \mathbf{a}}{v} \tag{3.140}$$

so we can write

$$\boxed{\frac{dv}{dt} = \frac{\mathbf{v} \cdot \mathbf{a}}{v}} \tag{3.141}$$

$$\boxed{\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}} \tag{3.142}$$

$$\boxed{\hat{\mathbf{r}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}} \tag{3.143}$$

$$\boxed{\hat{\mathbf{n}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|}} \tag{3.144}$$

The time derivatives are derived as follows.

$$\dot{\hat{\mathbf{r}}} = \frac{\partial \hat{\mathbf{r}}}{\partial t} = \frac{\partial}{\partial t}\left(\mathbf{r}r^{-1}\right) = \frac{\mathbf{v}}{r} - \frac{\mathbf{r}}{r^2}\left(\frac{\mathbf{r} \cdot \mathbf{v}}{r}\right) \tag{3.145}$$

which can be rewritten as

$$\boxed{\frac{d\hat{\mathbf{r}}}{dt} = \frac{\mathbf{v}}{r} - \frac{\hat{\mathbf{r}}}{r}(\hat{\mathbf{r}} \cdot \mathbf{v})} \tag{3.146}$$

$$\dot{\hat{\mathbf{v}}} = \frac{\partial \hat{\mathbf{v}}}{\partial t} = \frac{\partial}{\partial t}\left(\mathbf{v}v^{-1}\right) = \frac{\mathbf{a}}{v} - \frac{\mathbf{v}}{v^2}\left(\frac{\mathbf{v} \cdot \mathbf{a}}{v}\right) \tag{3.147}$$

which can be rewritten as

$$\boxed{\dot{\hat{\mathbf{v}}} = \frac{\mathbf{a}}{v} - \frac{\hat{\mathbf{v}}}{v}(\hat{\mathbf{v}} \cdot \mathbf{a})} \tag{3.148}$$

Finally,

$$\dot{\hat{\mathbf{n}}} = \frac{d}{dt}\left(\mathbf{n}n^{-1}\right) = \frac{\mathbf{r} \times \mathbf{a}}{n} - \frac{\mathbf{n}}{n^3}(\mathbf{r} \times \mathbf{a} \cdot \mathbf{n}) \tag{3.149}$$

$$\boxed{\dot{\hat{\mathbf{n}}} = \frac{\mathbf{r} \times \mathbf{a}}{n} - \frac{\hat{\mathbf{n}}}{n}(\mathbf{r} \times \mathbf{a} \cdot \hat{\mathbf{n}})} \tag{3.150}$$

Draft: Work in Progress

Table 3.1: Recommended Values for Pole and Prime Meridian Locations of the Sun and Planets[4]

Name	Values
Sun	$\alpha_o = 286.13^\circ$ (deg) $\delta_o = 63.87^\circ$ (deg) $W = 84.10^\circ + 14.1844000^\circ d$ (deg) $\dot{W} = 14.1844000^\circ$ (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	$\alpha_o = 313.02$ $\delta_o = 9.09$ $W = 236.77 - 56.3623195d$ $\dot{W} = -56.3623195$

Draft: Work in Progress

Table 3.2: Recommended Values for Pole and Prime Meridian Locations of Luna[4]

Name	Values
Luna	
	$\alpha_o = 269.9949 + 0.0031T - 3.8787 \sin E1 - 0.1204 \sin E2 + 0.0700 \sin E3 - 0.0172 \sin E4 + 0.0072 \sin E6 - 0.0052 \sin E10 - 0.0043 \sin E13$
	$\delta_o = 66.5392 + 0.0130T + 1.5419 \cos E1 + 0.0239 \cos E2 - 0.0278 \cos E3 + 0.0068 \cos E4 - 0.00292 \cos E6 + 0.0009 \cos E7 + .0008 \cos E10 - 0.0009 \cos E13$
	$W = 38.3213 + 13.17635815d - 1.4 \times 10^{-12}d^2 + 3.5610 \sin E1 + 0.1208 \sin E2 - 0.0642 \sin E3 + 0.0158 \sin E4 + 0.0252 \sin E5 - 0.0066 \sin E6 - 0.0047 \sin E7 - 0.0046 \sin E8 + 0.0028 \sin E9 + 0.0052 \sin E10 + 0.0040 \sin E11 + 0.0019 \sin E12 - 0.0044 \sin E13$
	$\dot{W} = +13.17635815 - 2.8 \times 10^{-12}d - .18870 \cos E1 - .01280 \cos E2 - .835 \cos E3 + .211 \cos E4 + .0248 \cos E5 - .17 \cos E6 - .061 \cos E7 - .0015 \cos E8 + .0049 \cos E9 - .00083 \cos E10 + .00001 \cos E11 + .00031 \cos E12 - .057 \cos E13$
where	$E1 = 125.045 - 0.0529921d$ $E2 = 250.089 - 0.1059842d$ $E3 = 260.008 + 13.0120009d$ $E4 = 176.625 + 13.3407154d$ $E5 = 357.529 + 0.9856003d$ $E6 = 311.589 + 26.4057084d$ $E7 = 134.963 + 13.0649930d$ $E8 = 276.617 + 0.3287146d$ $E9 = 34.226 + 1.7484877d$ $E10 = 15.134 - 0.1589763d$ $E11 = 119.743 + 0.0036096d$ $E12 = 239.961 + 0.1643573d$ $E13 = 25.053 + 12.9590088d$

Draft: Work in Progress

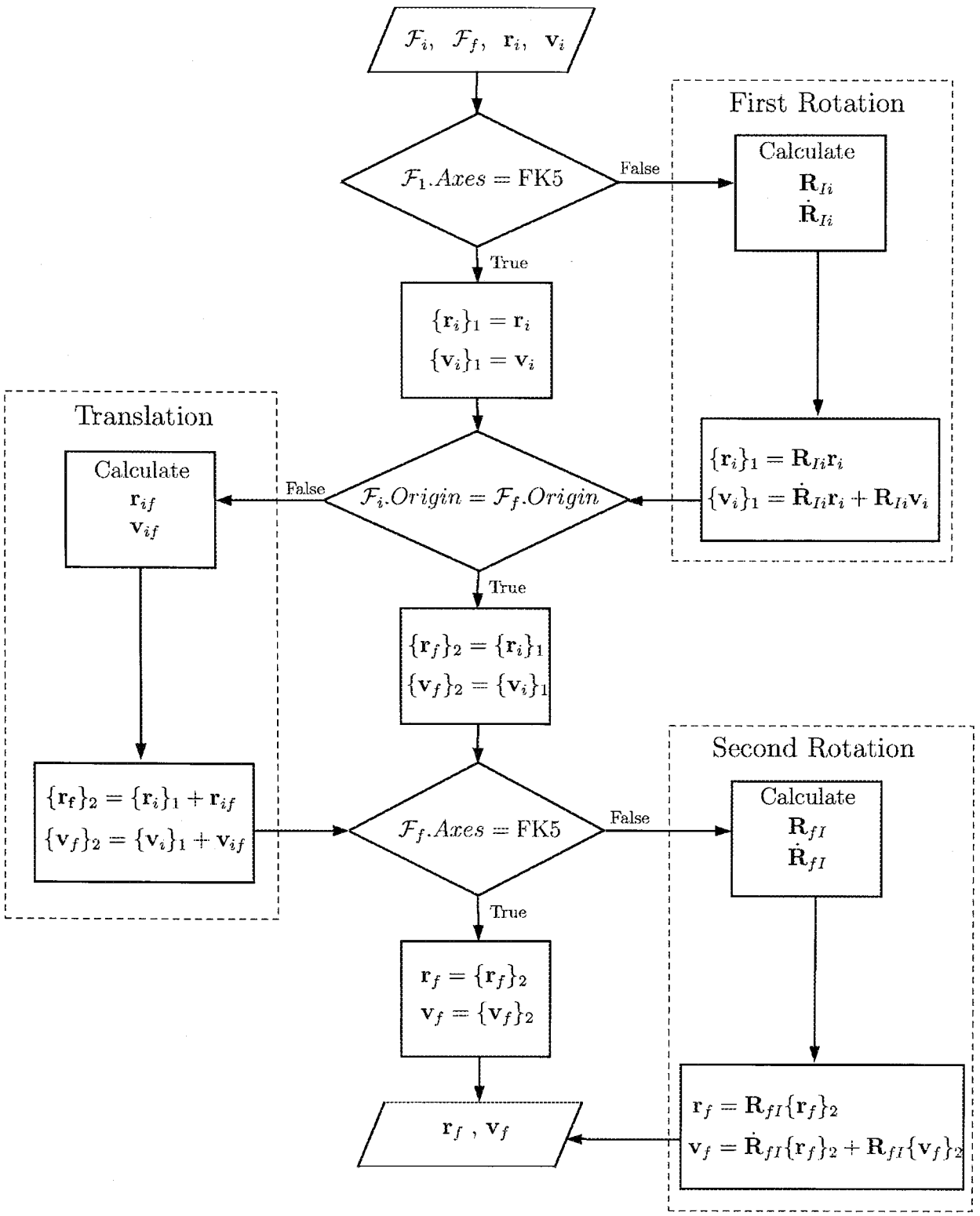

Figure 3.7: Coordinate System Transformation Algorithm

* See Sec. 3.2 for Treatment of Pseudo-Rotating Coordinate Systems

Draft: Work in Progress

Chapter 4

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.

4.1 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.

4.1.1 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 4.2 and illustrated in Fig. 4.1. Sections 4.1.2 and 4.1.3 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 4.1.

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 4.4.

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

4.1.2 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

Draft: Work in Progress

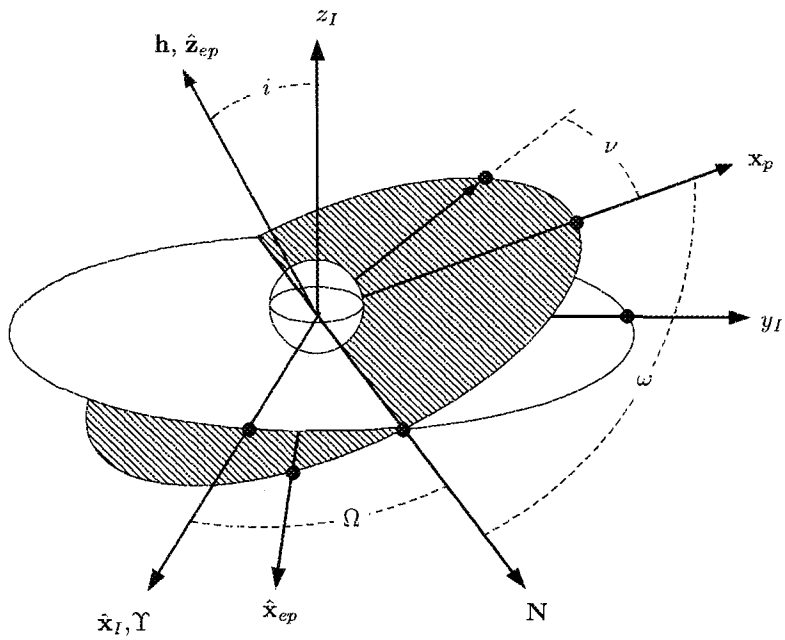

Figure 4.1: The Keplerian Elements

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.

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.1}$$

$$h = \|\mathbf{h}\| \tag{4.2}$$

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

$$\mathbf{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times \mathbf{h} \tag{4.3}$$

$$n = \|n\| \tag{4.4}$$

The orbit eccentricity and energy are calculated using

$$r = \|\mathbf{r}\| \tag{4.5}$$

$$v = \|\mathbf{v}\| \tag{4.6}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r} \cdot \mathbf{v})\mathbf{v}}{\mu} \tag{4.7}$$

$$e = \|\mathbf{e}\| \tag{4.8}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{4.9}$$

Recall that for parabolic orbits, the semimajor axis is infinite and the energy is zero. GMAT checks to see if the orbit is near parabolic and returns an error message in this case. The following error check avoids the possibility of a divide by zero:

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

$$a = -\frac{\mu}{2\xi} \tag{4.10}$$

otherwise, report error and return. The error reported is: "Warning: GMAT does not support parabolic orbits in conversion from cartesian to Keplerian state".

If the orbit is not parabolic according to the above definition, then we continue and calculate the inclination.

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{4.11}$$

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

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{4.12}$$

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

$$\omega = \cos^{-1}\left(\frac{\mathbf{n} \cdot \mathbf{e}}{ne}\right) \tag{4.13}$$

Draft: Work in Progress

Table 4.1: 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

Table 4.2: The Keplerian Elements (also see Fig. 4.1)

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 $i \leq 90°$ 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 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 4.3: 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.

Draft: Work in Progress

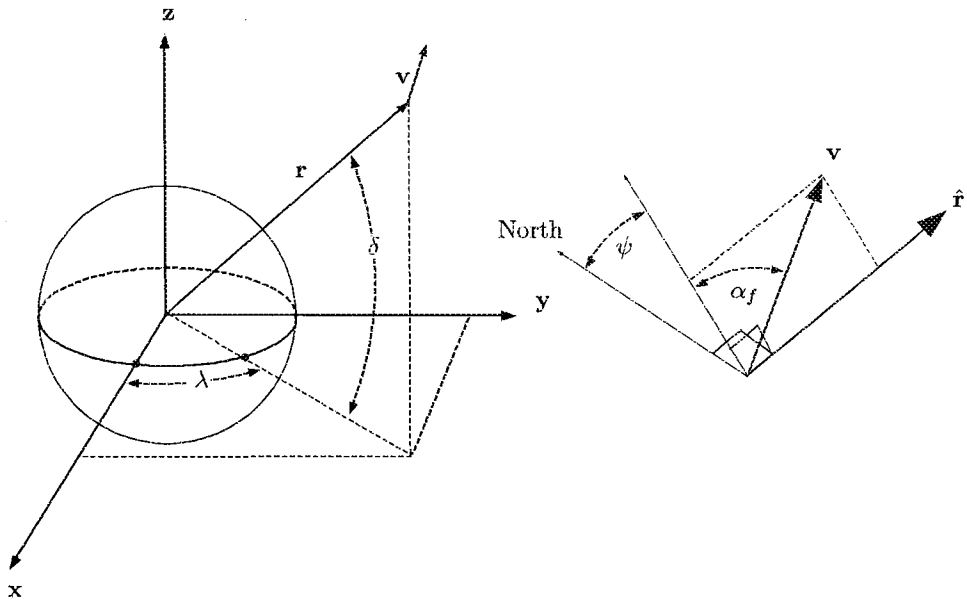

Figure 4.2: The Spherical Elements

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

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) \tag{4.14}$$

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

$$\Omega = 0 \tag{4.15}$$

$$\omega = \cos^{-1}\frac{e_x}{e} \tag{4.16}$$

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

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{r}}{er}\right) \tag{4.17}$$

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

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{4.18}$$

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

$$\omega = 0 \tag{4.19}$$

$$\nu = \cos^{-1}\left(\frac{\mathbf{n}\cdot\mathbf{r}}{nr}\right) \tag{4.20}$$

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

$$\Omega = 0 \tag{4.21}$$

$$\omega = 0 \tag{4.22}$$

$$\nu = \cos^{-1}\left(\frac{r_x}{r}\right) \tag{4.23}$$

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.

4.1.3 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 4.1 and 4.2 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 $\mathbf{v}$

We begin by checking that the magnitude of the position vector is not infinite. This avoids the possibility of a division by zero.

if $(1 + e\cos\nu < \text{1e-30})$, then display error and return: "Warning: Radius is near infinite in keplerian to cartesian conversion"

Draft: Work in Progress

If their is not a divide by zero issue we continue by calculating the semilatus rectum, and the radius.

$$p = a(1 - e^2); \tag{4.24}$$

$$r = \frac{p}{1 + e\cos\nu} \tag{4.25}$$

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

$$x = r\left(\cos(\omega + \nu)\cos\Omega - \cos i \sin(\omega + \nu)\sin\Omega\right) \tag{4.26}$$

$$y = r\left(\cos(\omega + \nu)\sin\Omega + \cos i \sin(\omega + \nu)\cos\Omega\right) \tag{4.27}$$

$$z = r\left(\sin(\omega + \nu)\sin i\right) \tag{4.28}$$

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

$$\dot{x} = \sqrt{\frac{\mu}{p}}\left(\cos\nu + e\right)\left(-\sin\omega\cos\Omega - \cos i \sin\Omega\cos\omega\right) - \sqrt{\frac{\mu}{p}}\sin\nu\left(\cos\omega\cos\Omega - \cos i\sin\Omega\sin\omega\right) \tag{4.29}$$

$$\dot{y} = \sqrt{\frac{\mu}{p}}\left(\cos\nu + e\right)\left(-\sin\omega\sin\Omega + \cos i\cos\Omega\cos\omega\right) - \sqrt{\frac{\mu}{p}}\sin\nu\left(\cos\omega\sin\Omega + \cos i\cos\Omega\sin\omega\right) \tag{4.30}$$

$$\dot{z} = \sqrt{\frac{\mu}{p}}\left[(\cos\nu + e)\sin i\cos w - \sin\nu\sin i\sin\omega\right] \tag{4.31}$$

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

4.1.4 Equinoctial Elements to Cartesian State

The equinoctial elements used in GMAT are defined in Table 4.4. The algorithm to convert from equinoctial elements to the cartesian state was taken from the GTDS Mathematical Theory.[5]

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 true longitude F. The equation relating the two is transcendental:

$$\lambda = F + h\cos F - k\sin F \tag{4.32}$$

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

$$F(i+1) = F(i) - \frac{f(F)}{f'(F)} \tag{4.33}$$

where

$$f(F) = F + h\cos(F) - k\sin(F) - \lambda \tag{4.34}$$

$$f'(F) = 1 - h\sin(F) - k\cos(F) \tag{4.35}$$

Once the true longitude is calculated, we continue with

$$\beta = \frac{1}{1 + \sqrt{1 - h^2 - k^2}} \tag{4.36}$$

$$n = \frac{mu}{a^3} \tag{4.37}$$

$$r = a(1 - k\cos F - h\sin f) \tag{4.38}$$

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

$$X_1 = a\left[(1 - h^2\beta)\cos F + hk\beta\sin F - k\right] \tag{4.39}$$

$$Y_1 = a\left[(1 - k^2\beta)\sin F + hk\beta\cos F - h\right] \tag{4.40}$$

$$\dot{X}_1 = \frac{na^2}{r}\left[hk\beta\cos F - (1 - h^2\beta)\sin F\right] \tag{4.41}$$

$$\dot{Y}_1 = \frac{na^2}{r}\left[(1 - k^2\beta)\cos F - hk\beta\sin F\right] \tag{4.42}$$

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

$$\mathbf{r} = X_1\hat{\mathbf{f}} + Y_1\hat{\mathbf{g}} \tag{4.43}$$

$$\mathbf{v} = \dot{X}_1\hat{\mathbf{f}} + \dot{Y}_1\hat{\mathbf{g}} \tag{4.44}$$

where

$$\begin{bmatrix}\hat{\mathbf{f}} & \hat{\mathbf{g}} & \hat{\mathbf{w}}\end{bmatrix} = \frac{1}{1 + p^2 + q^2}\ \mathbf{Q} \tag{4.45}$$

and

$$\mathbf{Q} = \begin{pmatrix} 1 - p^2 + q^2 & 2pqj & 2p \\ 2pq & \left(1 + p^2 - q^2\right)j & -2q \\ -2pj & 2q & \left(1 - p^2 - q^2\right)j \end{pmatrix} \tag{4.46}$$

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

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

Draft: Work in Progress

Table 4.4: 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.

4.1.5 Cartesian State to Equinoctial Elements

The equinoctial elements used in GMAT are defined in Table 4.4. The algorithm to convert from the cartesian state to the equinoctial elements was taken from the GTDS Mathematical Theory.[5]

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

Fine: a, h, k, p, q, λ, and μ

We begin by calculating the semimajor axis and the eccentricity vector.

$$r = \|\mathbf{r}\| \tag{4.47}$$

$$v = \|\mathbf{v}\| \tag{4.48}$$

$$a = \frac{1}{\frac{2}{r} - \frac{v^2}{\mu}} \tag{4.49}$$

$$\mathbf{e} = -\frac{\mathbf{r}}{r} - \frac{(\mathbf{r} \times \mathbf{v}) \times \mathbf{v}}{\mu} \tag{4.50}$$

The angular momentum vector is

$$\hat{\mathbf{h}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \tag{4.51}$$

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

$$f_x = 1 - \frac{h_x^2}{1 + h_z^j} \tag{4.52}$$

$$f_y = -\frac{h_x h_y}{1 + h_z^j} \tag{4.53}$$

$$f_z = -h_x^j \tag{4.54}$$

where j is described in Section 4.1.4.

$$\hat{\mathbf{g}} = \hat{\mathbf{h}} \times \hat{\mathbf{f}} \tag{4.55}$$

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

$$h = \mathbf{e} \cdot \hat{\mathbf{g}} \tag{4.56}$$

$$k = \mathbf{e} \cdot \hat{\mathbf{f}} \tag{4.57}$$

$$p = \frac{h_x}{1 + h_z^j} \tag{4.58}$$

$$q = -\frac{h_y}{1 + h_z^j} \tag{4.59}$$

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

$$X_1 = \mathbf{r} \cdot \hat{\mathbf{f}} \tag{4.60}$$

$$Y_1 = \mathbf{r} \cdot \hat{\mathbf{g}} \tag{4.61}$$

and

$$\cos F = k + \frac{\left(1 - k^2\beta\right) X_1 - hk\beta Y_1}{a\sqrt{1 - h^2 - k^2}} \tag{4.62}$$

$$\sin F = h + \frac{\left(1 - h^2\beta\right) Y_1 - hk\beta X_1}{a\sqrt{1 - h^2 - k^2}} \tag{4.63}$$

$$F = \tan_2^{-1}\left(\frac{\sin F}{\cos F}\right) \tag{4.64}$$

where β is given by Eq. 4.36. The mean longitude is computed using the generalized Kepler equation

$$\lambda = F + h\cos f - k\sin F \tag{4.65}$$

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

4.1.6 Cartesian State to SphericalAZFPA State

The spherical state, with azimuth, α_f, and flight path angle, ψ, is described in Table 4.5 and Fig. 4.2. 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

Draft: Work in Progress

Table 4.5: The Spherical Elements

Symbol	Name	Description
r	r	Magnitude of the position vector, $\|\mathbf{r}\|$
λ	Right Ascension	The between the projection of $\mathbf{r}$ into the $x-y$ and the x-axis. Measured counterclockwise.
δ	Declination	The angle between $\mathbf{r}$ and the $x-y$ plane measured in the plane formed by $\mathbf{r}$ and $\hat{\mathbf{z}}$.
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 $\mathbf{r}$.

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

$$r = \|\mathbf{r}\| \tag{4.66}$$

$$\lambda = \tan_2^{-1}(y, x) \tag{4.67}$$

$$\delta = \sin^{-1}(\frac{z}{r}) \tag{4.68}$$

The magnitude of the velocity vector is simply

$$v = \|\mathbf{v}\| \tag{4.69}$$

We calculate the vertical flight path angle, psi, using

$$\psi = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{v}}{rv}\right) \tag{4.70}$$

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

$$\hat{\mathbf{x}} = \begin{pmatrix} \cos(\delta)\cos(\lambda) \\ \cos(\delta)\sin(\lambda) \\ \sin(\delta) \end{pmatrix} \tag{4.71}$$

$$\hat{\mathbf{y}} = \begin{pmatrix} \cos(\lambda+\pi/2) \\ \sin(\lambda+\pi/2) \\ 0 \end{pmatrix} \tag{4.72}$$

$$\hat{\mathbf{z}} = \begin{pmatrix} -\sin(\delta)\cos(\lambda) \\ -\sin(\delta)\sin(\lambda) \\ \cos(\delta) \end{pmatrix} \tag{4.73}$$

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

$$\mathbf{R}_{\ell i} = [\ \hat{\mathbf{x}}\ \ \hat{\mathbf{y}}\ \ \hat{\mathbf{z}}\ ]^T \tag{4.74}$$

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

$$\mathbf{v}' = \mathbf{R}_{\ell i}\mathbf{v} \tag{4.75}$$

Finally, we calculate the azimuth angle using

$$\alpha_f = \tan_2^{-1}(v'_y, v'_z) \tag{4.76}$$

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.

4.1.7 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

$$x = r\cos\delta\cos\lambda \tag{4.77}$$

$$y = r\cos\delta\sin\lambda \tag{4.78}$$

$$z = r\sin\delta \tag{4.79}$$

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

$$\mathbf{v} = v\left[\cos(\psi)\hat{\mathbf{x}} + \sin(\psi)\sin(\alpha_f)\hat{\mathbf{y}} + \sin(\psi)\cos(\alpha_f)\hat{\mathbf{z}}\right] \tag{4.80}$$

where, $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{x}}$ are given in Eqs. (4.71), (4.72), and (4.73) respectively. Breaking down Eq. (4.80) into components gives us

$$v_x = v[\cos\psi\cos\delta\cos\lambda - \sin\psi(\sin\alpha_f\sin\lambda + \cos\alpha_f\sin\delta\cos\lambda)] \tag{4.81}$$

$$v_y = v[\cos\psi\cos\delta\sin\lambda + \sin\psi(\sin\alpha_f\cos\lambda - \cos\alpha_f\sin\delta\sin\lambda)] \tag{4.82}$$

$$v_z = v[\cos\psi\sin\delta + \sin\psi\cos\alpha_f cos\delta] \tag{4.83}$$

Draft: Work in Progress

4.1.8 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. 4.1.6. The algorithm to calculate λ_v and δ_v is shown below.

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

Find: r, λ, δ, v, λ_v, and δ_v

To calculate r, λ, δ, and v we use Eqs. (4.66), (4.67), (4.68), and (4.69) respectively. The right ascension of velocity, λ_v, and declination of velocity, δ_v, are calculated using

$$\lambda_v = \tan_2^{-1}(v_y, v_x) \tag{4.84}$$

$$\delta_v = \sin^{-1}(\frac{v_z}{v}) \tag{4.85}$$

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.

4.1.9 SphericalRADEC State to Cartesian State

This transformation is similar to the conversion presented in Sec 4.1.7. The primary difference is how the velocity is represented.

Given: r, λ, δ, v, λ_v, and δ_v

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

The position components are calculated using Eqs. (4.77), (4.78), and (4.79). The velocity components are calculated using

$$v_x = v\cos\lambda\cos\delta \tag{4.86}$$

$$v_y = v_x\tan\lambda \tag{4.87}$$

$$v_z = v\sin\delta \tag{4.88}$$

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.

4.1.10 Keplerian or Cartesian, to Modified Keplerian Elements

The modified Keplerian elements, described in Table 4.6, 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. 4.1.3. Knowing the Keplerian elements, we calculate r_a and r_p using

$$r_a = a(1+e) \tag{4.89}$$

$$r_p = a(1-e) \tag{4.90}$$

Now let's look at the inverse transformation.

4.1.11 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 right 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

$$e = \frac{1-\frac{r_p}{r_a}}{1+\frac{r_p}{r_a}} \tag{4.91}$$

The semimajor axis is calculated using

$$a = \frac{r_p}{1-e} \tag{4.92}$$

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.

Draft: Work in Progress

Table 4.6: 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.

4.2 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.

4.2.1 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.

4.2.2 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.

Draft: Work in Progress

4.2.3 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. 4.2.21. 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

$$r_{xy} = \sqrt{x^2 + y^2} \tag{4.93}$$

$$e^2 = 2f - f^2 \tag{4.94}$$

$$h = \frac{r_{xy}}{\cos(\phi_{gd})} - \frac{R}{\sqrt{1 - e^2 \sin^2 \phi_{gd}}}; \tag{4.95}$$

4.2.4 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.[1]

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,

$$\omega = \cos^{-1}\left(\frac{e_x}{e}\right) \tag{4.96}$$

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

$$\mathbf{n} = [\,0 \;\; 0 \;\; 1\,]^T \times \mathbf{h}$$

$$\omega = \cos^{-1}\frac{\mathbf{n}\cdot\mathbf{e}}{\|\mathbf{n}\|\|\mathbf{e}\|}$$

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

4.2.5 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 $\mathbf{v}$

Draft: Work in Progress

- $\mathcal{F}_2$ 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,

$$A = \mathbf{r} \cdot \mathbf{v} \tag{4.97}$$

4.2.6 AZI

Description: `AZI` is the azimuth angle of a spacecraft, as shown in Fig. 4.1 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. 4.1.6. There is little benefit using a routine that calculates only α_f and not ψ.

4.2.7 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.[6]

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.

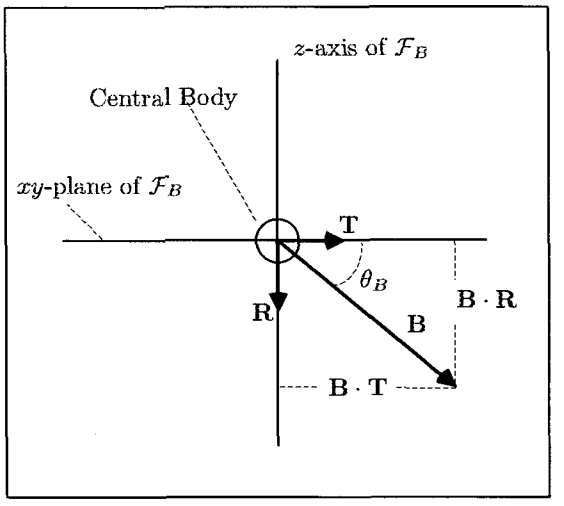

Figure 4.3: Geometry of the B-Plane as Seen From a Viewpoint Perpendicular to the B-Plane

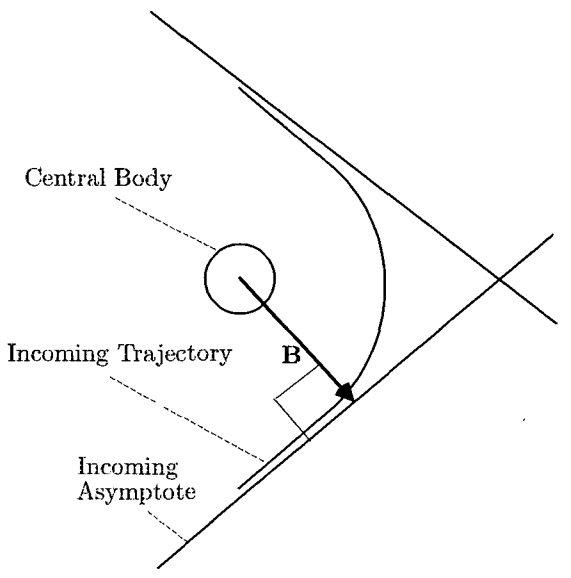

Figure 4.4: The B-Vector as Seen From a Viewpoint Perpendicular to Orbit Plane

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

Draft: Work in Progress

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

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

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:

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

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 $\mathcal{F}_2$

$$A = \mathbf{r}\cdot\mathbf{v} \tag{4.98}$$

4.2.8 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.

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

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

$$\beta = \sin^{-1}\left(\hat{\mathbf{h}}\cdot\hat{\mathbf{r}}_{s\oplus}\right) \tag{4.99}$$

- $\mathbf{r}_\oplus$: 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.

4.2.9 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

$$\|\mathbf{B}\| = \sqrt{B_T^2 + B_R^2} \tag{4.100}$$

$$\theta_B = \tan^{-1}\frac{B_R}{B_T} \tag{4.101}$$

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

4.2.10 C3Energy

Given: a, and μ

Find: C_3

$$C_3 = -\frac{\mu}{a} \tag{4.102}$$

Comment: a is calculated from the satellite cartesian state as shown in Section 4.1.2, and μ is associated with the specified central body.

4.2.11 DEC

Description: `DEC` is the declination of a spacecraft, as shown in Fig. 4.2 using the symbol δ.

Dependency: Coordinate System.

Draft: Work in Progress

Given: $\mathbf{r}$, $\mathbf{v}$ and $\mathcal{F}$

Find: δ

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

$$r = \|\mathbf{r}\| \tag{4.103}$$

$$\delta = \sin^{-1}(\frac{z}{r}) \tag{4.104}$$

4.2.12 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,

$$\delta_v = \sin^{-1}(\frac{v_z}{v}) \tag{4.105}$$

4.2.13 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.[1]

Dependency: Central Body.

Given: $\mathbf{r}$, $\mathbf{v}$, and μ (Central Body)

Find: e

$$r = \|\mathbf{r}\| \tag{4.106}$$

$$v = \|\mathbf{v}\| \tag{4.107}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r}\cdot\mathbf{v})\mathbf{v}}{\mu} \tag{4.108}$$

$$e = \|\mathbf{e}\| \tag{4.109}$$

4.2.14 FPA

Description: FPA is the orbit vertical Flight Path Angle as as shown in Fig. 4.2 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,

$$\psi = \cos^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{v}}{rv}\right) \tag{4.110}$$

4.2.15 EA

Given: ν, e

Find: E

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

Otherwise,

$$\sin(E) = \frac{\sqrt{1-e^2}\sin(\nu)}{1+e\cos\nu} \tag{4.111}$$

$$\cos(E) = \frac{e+\cos\nu}{1+e\cos\nu} \tag{4.112}$$

$$E = \text{atan2}(\sin E, \cos E) \tag{4.113}$$

4.2.16 Energy

Description: Energy is the orbit energy.

Dependency: Central Body.

Given: $\mathbf{r}$, $\mathbf{v}$, and central body.

Find: ξ

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,

$$r = \mathbf{r} \tag{4.114}$$

$$v = \mathbf{v} \tag{4.115}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{4.116}$$

4.2.17 HMAG

Description: HMAG is the magnitude of the orbit angular momentum.

Dependency: Central Body.

Given: $\mathbf{r}$, $\mathbf{v}$, and central body.

Find: h

Draft: Work in Progress

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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.117}$$

$$h = \|\mathbf{h}\| \tag{4.118}$$

4.2.18 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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = [\ h_x \quad h_y \quad h_z\ ]^T \tag{4.119}$$

4.2.19 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,

$$\sinh(H) = \frac{\sin(\nu)\sqrt{e^2-1}}{1+e\cos\nu} \tag{4.120}$$

$$\cosh(H) = \frac{e+\cos\nu}{1+e\cos\nu} \tag{4.121}$$

$$H = \tanh^{-1}\left(\frac{\sinh H}{\cosh H}\right) \tag{4.122}$$

4.2.20 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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.123}$$

$$h = \mathbf{h} \tag{4.124}$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{4.125}$$

4.2.21 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.[1]

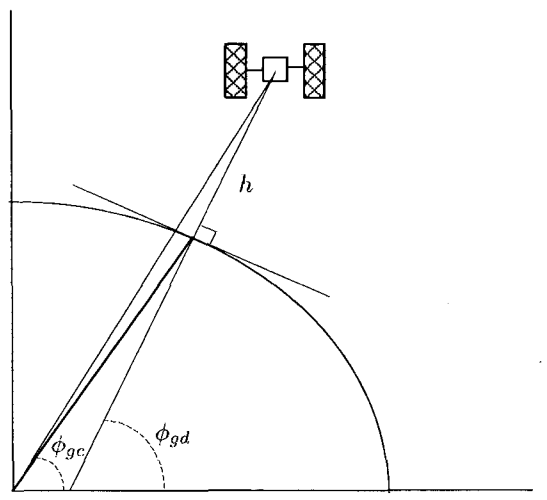

Figure 4.5: Geocentric and Geodetic Latitude

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.

Draft: Work in Progress

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

$$r_{xy} = \sqrt{x^2 + y^2} \tag{4.126}$$

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

$$\phi_{gd} \approx \text{atan2}(z, r_{xy}); \tag{4.127}$$

$$e^2 = 2f - f^2 \tag{4.128}$$

Set $\delta = 1.0$ to initialize the loop, then,

While ($\delta > 10^{-7}$)

$$\phi' = \phi_{gd} \tag{4.129}$$

$$\phi_{gd} = \text{atan2}\left(z + \frac{Re^2 \sin^2 \phi_{gd}}{\sqrt{1 - e^2 \sin \phi_{gd}}}, r_{xy}\right) \tag{4.130}$$

$$\delta = |\phi_{gd} - \phi'| \tag{4.131}$$

EndWhile

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

4.2.22 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,

$$\phi = \tan_2^{-1}(y, x); \tag{4.132}$$

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

4.2.23 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. 4.6, where $\mathcal{F}_I$ is the body's equatorial inertial system (as described in Sec. 3.4.1), $\mathcal{F}_F$ is the body's fixed system (as described in Sec. 3.4.9). 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. 3.4.1) of selected central body.
- $\mathcal{F}_F$ is the central body's fixed system (as described in Sec. 3.4.9)
- 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)

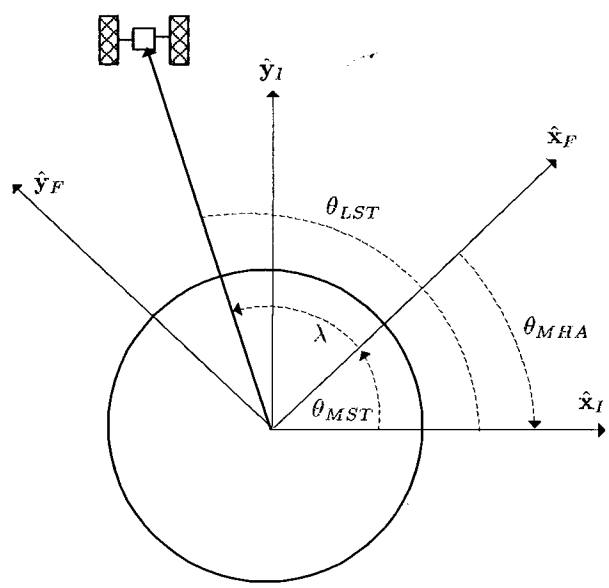

Figure 4.6: Local Sidereal Time Geometry

We begin by calculating λ using the algorithm described in Sec. 4.2.22. 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.

$$T_{UT1} = \frac{t_{ut1} - 21544.5}{36525} \tag{4.133}$$

Draft: Work in Progress

$$\theta_{MST} = 67310.54841^s + (876600^h \left(\frac{3600s}{1h}\right) + 8640184.812866)T_{UT1} + 0.093104T_{UT1}^2 - 6.2 \times 10^{-6}T_{UT1}^3 \tag{4.134}$$

4.2.24 MA

Given: ν, e

Find: M

If e < (1 - $1e^{-11}$) then calculate E using algorithm in Sec. 4.2.15. Then M is calculated using

$$M = E - e \sin E \tag{4.135}$$

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. 4.2.19. Then M is calculated using

$$M = e \sinh H - H \tag{4.136}$$

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".

4.2.25 MHA

4.2.26 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

$$n = \sqrt{\frac{\mu}{a^3}} \tag{4.137}$$

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

$$n = \sqrt{-\frac{\mu}{a^3}} \tag{4.138}$$

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

$$n = 2\sqrt{\mu} \tag{4.139}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 4.1.2, and μ is associated with the specified central body.

4.2.27 OrbitPeriod

Given: a, and μ

Find: T

If a < 0, then $T = 0$, return.

Otherwise,

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{4.140}$$

Comment: a is calculated from the satellite cartesian state as shown in Section 4.1.2, and μ is associated with the specified central body.

4.2.28 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[7] pgs. 80-83.

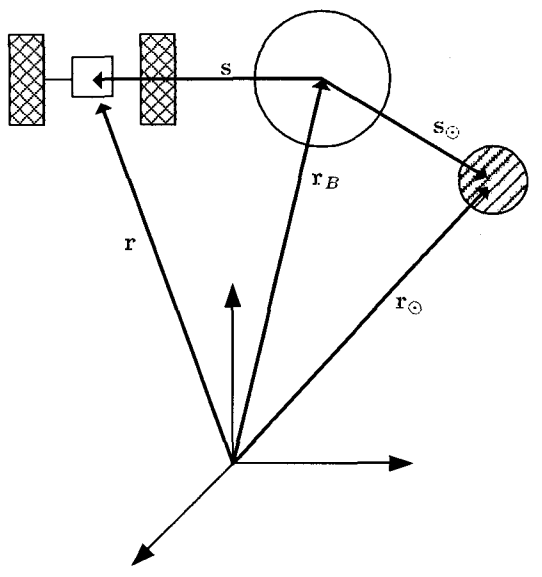

Figure 4.7: Shadow Geometry

- $R_\odot$ = Radius of the Sun
- R_B = Radius of occulting body
- $R'_\odot$ = Apparent radius of the Sun

Draft: Work in Progress

- 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

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

$$\mathbf{s} = \mathbf{r} - \mathbf{r}_B \tag{4.141}$$

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

$$\mathbf{s}_\odot = \mathbf{r}_\odot - \mathbf{r}_B \tag{4.142}$$

(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

$$R'_\odot = \sin^{-1}\frac{R_\odot}{\|\mathbf{r}_\odot - \mathbf{r}\|} \tag{4.143}$$

$$R'_B = \sin^{-1}\frac{R_B}{\|\mathbf{r} - \mathbf{r}_B\|} \tag{4.144}$$

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

$$D' = \cos^{-1}\left(\frac{-\mathbf{s}^T(\mathbf{r}_\odot - \mathbf{r})}{s\|\mathbf{r}_\odot - \mathbf{r}\|}\right) \tag{4.145}$$

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

$$p = 0; \tag{4.146}$$

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

$$p = 100; \tag{4.147}$$

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. 4.8.

$$A = R'^2_\odot \cos^{-1}\left(\frac{c_1}{R'_\odot}\right) + R'^2_B \cos^{-1}\left(\frac{D' - c_1}{R'_B}\right) - D'c_2 \tag{4.148}$$

where

$$c_1 = \frac{D'^2 + R'^2_\odot - R'^2_B}{2D'} \tag{4.149}$$

and

$$c_2 = \sqrt{R'^2_\odot - c_1^2} \tag{4.150}$$

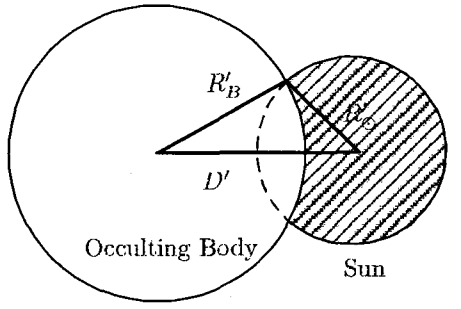

Figure 4.8: Occultation Geometry in Calculation of PercentShadow

The percent shadow can be calculated using

$$p = 100\frac{A}{\pi R'^2_\odot} \tag{4.151}$$

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

$$p = 100\frac{R'^2_B}{R'^2_\odot} \tag{4.152}$$

4.2.29 RA

Description: `RA` is the right ascension of a spacecraft, as shown in Fig. 4.2 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,

$$\lambda = \tan_2^{-1}(y, x) \tag{4.153}$$

4.2.30 RAV

Description: `RAV` is the right ascension of velocity of a spacecraft, as shown in Fig. 4.2 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,

$$\lambda_v = \tan_2^{-1}(v_y, v_x) \tag{4.154}$$

Draft: Work in Progress

4.2.31 RAAN

Description: `RAAN` is the right ascension of the ascending node as shown in Fig. 4.1 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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.155}$$

$$h = \|\mathbf{h}\| \tag{4.156}$$

$$\mathbf{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times \mathbf{h} \tag{4.157}$$

$$n = \|n\| \tag{4.158}$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{4.159}$$

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

$$\Omega = \cos^{-1}\left(\frac{n_x}{n}\right) \tag{4.160}$$

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

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

$$\Omega = 0 \tag{4.161}$$

4.2.32 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,

$$r_a = a(1 + e) \tag{4.162}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 4.1.2.

4.2.33 RadPer

Given: a, and e

Find: r_p

$$r_p = a(1 - e) \tag{4.163}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 4.1.2.

4.2.34 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,

$$r = \|\mathbf{r}\| \tag{4.164}$$

4.2.35 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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.165}$$

$$h = \|\mathbf{h}\| \tag{4.166}$$

$$p = \frac{h^2}{\mu} \tag{4.167}$$

4.2.36 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.[1]

Dependency: Central Body.

Given: $\mathbf{r}$, $\mathbf{v}$, and μ (Central Body)

Find: a

$$r = \|\mathbf{r}\| \tag{4.168}$$

$$v = \|\mathbf{v}\| \tag{4.169}$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{4.170}$$

Draft: Work in Progress

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

$$a = -\frac{\mu}{2\xi} \tag{4.171}$$

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

4.2.37 TA

Description: `TA` is the orbit true anomaly as shown in Fig. 4.1 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,

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.172}$$

$$h = \|\mathbf{h}\| \tag{4.173}$$

$$\mathbf{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times \mathbf{h} \tag{4.174}$$

$$n = \|n\| \tag{4.175}$$

$$r = \|\mathbf{r}\| \tag{4.176}$$

$$v = \|\mathbf{v}\| \tag{4.177}$$

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r} \cdot \mathbf{v})\mathbf{v}}{\mu} \tag{4.178}$$

$$e = \|\mathbf{e}\| \tag{4.179}$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \tag{4.180}$$

There are three special cases, and they are treated differently.

Special Case 1: Elliptic Orbit

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

$$\nu = \cos^{-1}\left(\frac{\mathbf{e} \cdot \mathbf{r}}{er}\right) \tag{4.181}$$

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

$$\nu = \cos^{-1}\left(\frac{\mathbf{n} \cdot \mathbf{r}}{nr}\right) \tag{4.182}$$

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

$$\nu = \cos^{-1}\left(\frac{r_x}{r}\right) \tag{4.183}$$

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

4.2.38 TAIModJulian

Description: `TAIModJulian` is the epoch in the TAI time system, expressed in the modified Julian date format. See Sec. 2.1.1 and 2.2.1 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

$$TAI = A1 - 0.0343817\text{sec} \tag{4.184}$$

4.2.39 TTModJulian

Description: `TTModJulian` is the epoch in the TT time system, expressed in the modified Julian date format. See Sec. 2.1.3 and 2.2.1 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

$$TT = A1 - 0.0343817\text{sec} + 32.184\text{sec} \tag{4.185}$$

4.2.40 TTGregorian

Description: `TTGregorian` is the epoch in the TT time system, expressed in the Gregorian date format. See Sec. 2.1.3 and 2.2.2 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. (4.185). Then, knowing the epoch in the TT time system in the modified Julian date format, we use the algorithm in Sec. 2.2.1 to obtain the Gregorian date.

4.2.41 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[7] 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.

Draft: Work in Progress

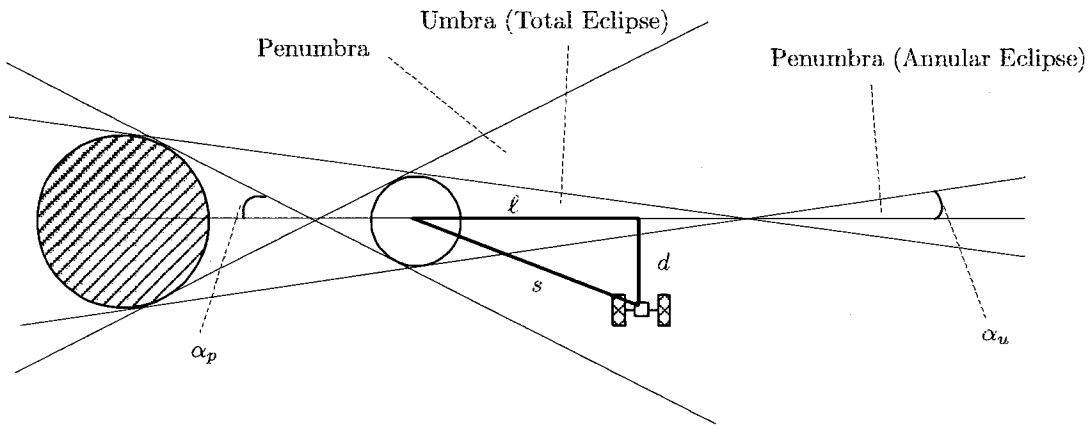

Figure 4.9: Geometry of Umbra and Penumbra Regions

For definitions of see Sec. 4.2.28.

$$\ell = \frac{-\mathbf{s}^T \mathbf{s}_\odot}{s_\odot} \tag{4.186}$$

$$d = \sqrt{s^2 - l^2} \tag{4.187}$$

$$\sin \alpha_p = \frac{R_\odot + R_B}{s_\odot} \tag{4.188}$$

$$\sin \alpha_u = \frac{R_\odot - R_B}{s_\odot} \tag{4.189}$$

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

$$r_p = \tan \alpha_p \left(\ell + \frac{R_B}{\sin \alpha_p} \right) \tag{4.190}$$

$$r_u = \tan \alpha_u \left(\ell - \frac{R_B}{\sin \alpha_u} \right) \tag{4.191}$$

Finally, if $\ell \geq 0$

$$d_p = d - r_p \tag{4.192}$$

$$d_u = d - |r_u| \tag{4.193}$$

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

$$d_p = |d - r_p| \tag{4.194}$$

$$d_u = |d - |r_u|| \tag{4.195}$$

4.2.42 UTCModJulian

Description: `UTCModJulian` is the epoch in the UTC time system, expressed in the modified Julian date format. See Sec. 2.1.2 and 2.2.1 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

$$UTC = A1 - 0.0343817\text{sec} - \Delta AT \tag{4.196}$$

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

4.2.43 VelApoapsis

Given: a, e, and μ

Find: v_a

If e > (1 - $1e^{-12}$) then $v_a = 0$.

Otherwise,

$$v_a = \sqrt{\frac{\mu}{a}\left(\frac{1-e}{1+e}\right)} \tag{4.197}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 4.1.2, and μ is associated with the specified central body.

Draft: Work in Progress

4.2.44 VelPeriapsis

Given: a, e, and μ

Find: v_p

$$v_p = \sqrt{\frac{\mu}{a}\left(\frac{1+e}{1-e}\right)} \tag{4.198}$$

Comment: a and e are calculated from the satellite cartesian state as shown in Section 4.1.2, and μ is associated with the specified central body.

4.2.45 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,

$$v = \|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{4.199}$$

4.3 Other Calculations

4.3.1 MA to TA

Description: This algorithm shows how to calculate ν given M and e and is taken from Vallado.[1]

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. 4.3.2

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. 4.3.3

4.3.2 EA to TA

Description: This algorithm shows how to calculate ν given E and e and is taken from Vallado.[1]

Given: E and e.

Find: ν

$$\sin\nu = \frac{\sqrt{1-e^2}\sin(E)}{1 - e\cos E} \tag{4.200}$$

$$\cos\nu = \frac{\cos E - e}{1 - e\cos E} \tag{4.201}$$

$$\nu = \text{atan2}(\sin\nu, \cos\nu) \tag{4.202}$$

Draft: Work in Progress

4.3.3 HA to TA

Description: This algorithm shows how to calculate ν given H and e and is taken from Vallado.[1]

Given: H and e.

Find: ν

$$\sin\nu = -\frac{\sqrt{e^2-1}\sinh(H)}{1-e\cosh H} \tag{4.203}$$

$$\cos\nu = \frac{\cosh H - e}{1-e\cosh H} \tag{4.204}$$

$$\nu = \text{atan2}(\sin\nu, \cos\nu) \tag{4.205}$$

4.4 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 4.10 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. 4.10. 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. 4.11. 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:

$$\begin{aligned}\gamma_1^5 - (3-\mu^*)\gamma_1^4 + (3-2\mu^*)\gamma_1^3 - \mu^*\gamma_1^2 \\ + 2\mu^*\gamma_1 - \mu^* = 0 \quad \text{(For L1)}\end{aligned} \tag{4.206}$$

$$\begin{aligned}\gamma_2^5 + (3-\mu^*)\gamma_2^4 + (3-2\mu^*)\gamma_2^3 - \mu^*\gamma_2^2 \\ - 2\mu^*\gamma_2 - \mu^* = 0 \quad \text{(For L2)}\end{aligned} \tag{4.207}$$

$$\begin{aligned}\gamma_3^5 + (2+\mu^*)\gamma_3^4 + (1+2\mu^*)\gamma_3^3 - (1-\mu^*)\gamma_3^2 \\ - 2(1-\mu^*)\gamma_3 - (1-\mu^*) = 0 \quad \text{(For L3)}\end{aligned} \tag{4.208}$$

where

$$\mu^* = \frac{m_2}{m_1+m_2} \tag{4.209}$$

Equations (4.206)-(4.208) do not have exact analytic solutions.Szebehely[8] notes that they are most easily solved using an iterative method with the following as the initial guesses:

$$\gamma_1 = \gamma_2 = \left(\frac{\mu^*}{3(1-\mu^*)}\right)^{1/3} \tag{4.210}$$

$$\gamma_3 = 1 \tag{4.211}$$

GMAT uses the Newton-Raphson method to solve for the roots of the equations by iterating on

$$\gamma(i+1) = \gamma(i) - \frac{F(\gamma(i))}{F'(\gamma(i))} \tag{4.212}$$

until the the difference $|\gamma(i+1) - \gamma(i)| < 10^{-8}$. The derivative $F'(\gamma)$ for each libration point is shown below.

$$\begin{aligned}F'(\gamma) = 5\gamma_1^4 - 4(3-\mu^*)\gamma_1^3 + 3(3-2\mu^*)\gamma_1^2 \\ - 2\mu^*\gamma_1 + 2\mu^* \quad \text{(For L1)}\end{aligned} \tag{4.213}$$

$$\begin{aligned}F'(\gamma) = 5\gamma_2^4 + 4(3-\mu^*)\gamma_2^3 + 3(3-2\mu^*)\gamma_2^2 \\ - 2\mu^*\gamma_2 - 2\mu^* \quad \text{(For L2)}\end{aligned} \tag{4.214}$$

$$\begin{aligned}F'(\gamma) = 5\gamma_3^4 + 4(2+\mu^*)\gamma_3^3 + 3(1+2\mu^*)\gamma_3^2 \\ - 2(1-\mu^*)\gamma_3 - 2(1-\mu^*) \quad \text{(For L3)}\end{aligned} \tag{4.215}$$

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 ith libration point can be expressed in the rotating system with the origin centered on the primary body as

$$\mathbf{r}^i = r_s\,[x_i \quad y_i \quad 0]^T \tag{4.216}$$

Draft: Work in Progress

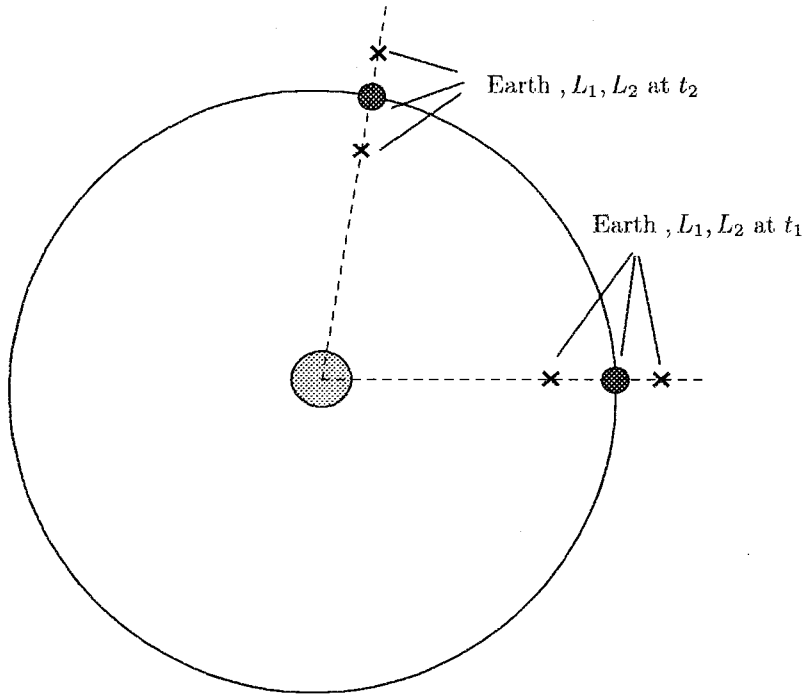

Figure 4.10: Geometry of Libration Points

Table 4.7: 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	$-\sqrt{3}/2$

where

$$r_s = \|\mathbf{r}_s\| \tag{4.217}$$

The velocity of the ith libration point can be expressed in the rotating system with the origin centered on the primary body as

$$\mathbf{v}^i = \frac{\mathbf{v}_s \cdot \mathbf{r}_s}{r_s} \begin{bmatrix} x_i & y_i & 0 \end{bmatrix}^T \tag{4.218}$$

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:

$$\mathbf{R}^{Ii} = \begin{pmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 \\ \hat{x}_2 & \hat{y}_2 & \hat{z}_2 \\ \hat{x}_3 & \hat{y}_3 & \hat{z}_3 \end{pmatrix} \tag{4.219}$$

and

$$\dot{\mathbf{R}}^{Ii} = \begin{pmatrix} \dot{\hat{x}}_1 & \dot{\hat{y}}_1 & \dot{\hat{z}}_1 \\ \dot{\hat{x}}_2 & \dot{\hat{y}}_2 & \dot{\hat{z}}_2 \\ \dot{\hat{x}}_3 & \dot{\hat{y}}_3 & \dot{\hat{z}}_3 \end{pmatrix} \tag{4.220}$$

where

$$\hat{\mathbf{x}} = \frac{\mathbf{r}_s}{r_s} \tag{4.221}$$

$$\hat{\mathbf{z}} = \frac{\mathbf{r}_s \times \mathbf{v}_s}{\|\mathbf{r}_s \times \mathbf{v}_s\|} \tag{4.222}$$

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{4.223}$$

and

$$\dot{\hat{\mathbf{x}}} = \dot{\hat{\mathbf{r}}}_s = \frac{\mathbf{v}_s}{r_s} - \frac{\hat{\mathbf{r}}_s}{r_s}(\hat{\mathbf{r}}_s \cdot \mathbf{v}_s) \tag{4.224}$$

$$\dot{\hat{\mathbf{z}}} = \frac{\mathbf{r}_s \times \mathbf{a}_s}{\|\mathbf{r}_s \times \mathbf{v}_s\|} - \frac{\hat{\mathbf{z}}}{\|\mathbf{r}_s \times \mathbf{v}_s\|}(\mathbf{r}_s \times \mathbf{a}_s \cdot \hat{\mathbf{z}}) \tag{4.225}$$

$$\dot{\hat{\mathbf{y}}} = \dot{\hat{\mathbf{z}}} \times \hat{\mathbf{x}} + \hat{\mathbf{z}} \times \dot{\hat{\mathbf{x}}} \tag{4.226}$$

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:

$$\mathbf{r} = \mathbf{R}^{Ii}\mathbf{r}^i \tag{4.227}$$

$$\mathbf{v} = \dot{\mathbf{R}}^{Ii}\mathbf{r}^i + \mathbf{R}^{Ii}\mathbf{v}^i \tag{4.228}$$

Draft: Work in Progress

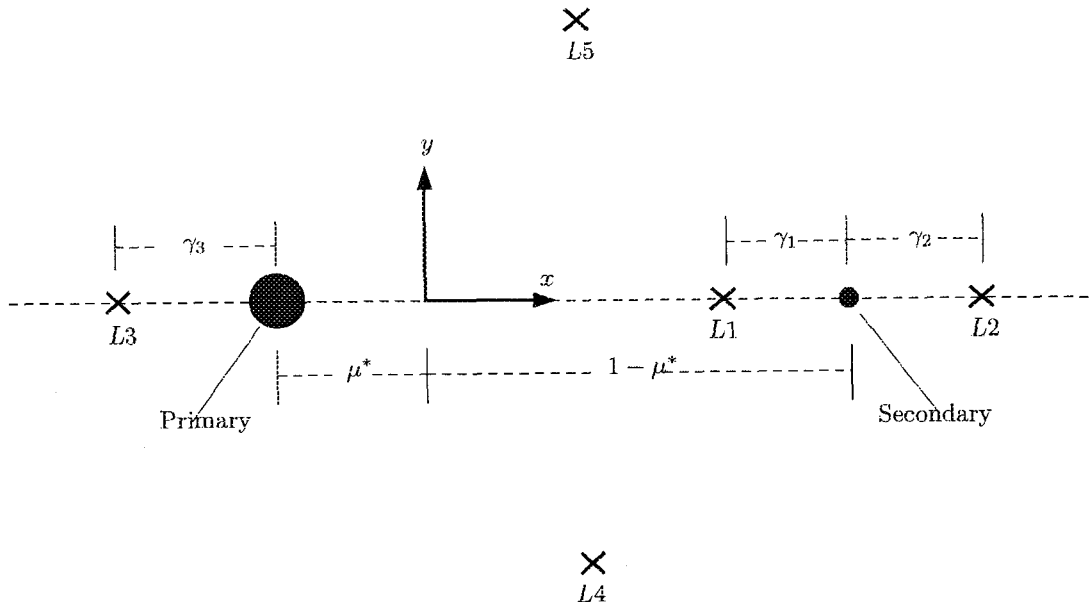

Figure 4.11: Location of Libration Points

4.5 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 ith body with respect to a common reference system, then we can calculate the barycenter of the system using

$$\mathbf{r}_b = \frac{\sum_{i=1}^{n} m_i \mathbf{r}_i}{\sum_{i=1}^{n} m_i} \tag{4.229}$$

Similarly, we can calculate the velocity of the barycenter using the following equation

$$\mathbf{v}_b = \frac{\sum_{i=1}^{n} m_i \mathbf{v}_i}{\sum_{i=1}^{n} m_i} \tag{4.230}$$

Draft: Work in Progress

Chapter 5

Dynamics Modelling

One of the fundamental capabilities of GMAT is to model the motion of spacecraft in many different flight regimes. The flight regimes, such as low Earth, or Libration Points, are determined by which forces and perturbations dominate the dynamics. In this chapter we present how GMAT models the dynamics of spacecraft in motion. We discuss how GMAT calculates many different types of forces including mulitiple non-spherical gravity perturbations, third-body effects, and atmospheric drag among others. We being by looking at the general form of the equations of motion.

5.1 Equations of Motion

5.1.1 Orbit State Equations

Let's begin by defining the position and velocity of a spacecraft with respect to the central body of integration as $\mathbf{r}$ and $\mathbf{v}$. From Newton's Second Law we know that

$$m\frac{d^2\mathbf{r}}{d^2t} = \sum F \tag{5.1}$$

which says that the mass, times the acceleration, is equal to the sum of the forces. Solving for the acceleration gives us the second order differential equation

$$\frac{d^2\mathbf{r}}{d^2t} = \sum \frac{F}{m} \tag{5.2}$$

GMAT has the capability to model many different types of accelerations experienced by spacecraft in orbit. If we include all of the possible forces GMAT can model in the summation on the left hand side of Eq. (5.2), then we would have

$$\begin{aligned}\frac{d^2\mathbf{r}}{dt^2} = &-\frac{\mu}{r^3}\mathbf{r} + \nabla\phi^o_{sj} + G\sum_{\substack{k=1\\k\neq j}}^{n_b} m_k\left(\frac{\mathbf{r}_{ks}}{r^3_{ks}} - \frac{\mathbf{r}_{kj}}{r^3_{kj}}\right)\\ &+\sum_{\substack{k=1\\k\neq j}}^{n_b}\left(\nabla\phi^o_{ks} + \nabla\phi^o_{kj}\right) + \frac{\dot{m}_s}{m}\frac{d\mathbf{r}}{dt}\\ &-\frac{1}{2}\rho v^2_{rel}\frac{C_dA}{m_s}\hat{\mathbf{v}}_{rel} + \frac{P_{SR}C_RA_\odot}{m_s}\hat{\mathbf{r}}_{s\odot}\end{aligned} \tag{5.3}$$

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	$G\sum_{\substack{k=1\\k\neq j}}^{n_b} m_k\left(-\frac{\mathbf{r}_{kj}}{r^3_{kj}}\right)$
Third Body Direct Nonspherical	$\sum_{\substack{k=1\\k\neq j}}^{n_b}\left(\nabla\phi^o_{ks}\right)$
Third Body Indirect Nonspherical	$\sum_{\substack{k=1\\k\neq j}}^{n_b}\left(\nabla\phi^o_{kj}\right)$
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	$\frac{P_{SR}C_RA_\odot}{m_s}\hat{\mathbf{r}}_{s\odot}$

In general, Eq. (5.3) does not have an analytic solution

Draft: Work in Progress

so GMAT uses numerical integration to find approximate, although very accurate, solutions. GMAT uses first order numerical integrators, so we must take the three second order differential equations in Eq. (5.3) and convert them to six first order equations. So, we define a new variable $\mathbf{x}$ such that

$$\mathbf{x} = \begin{bmatrix}\mathbf{r}^T & \mathbf{v}^T\end{bmatrix}^T = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{5.4}$$

then taking the derivative we arrive at

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{r}}^T & \dot{\mathbf{v}}^T\end{bmatrix}^T = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \ddot{x} \quad \ddot{y} \quad \ddot{z}]^T \tag{5.5}$$

5.1.2 State Transition Matrix Equations

$$\dot{\mathbf{\Phi}}(t, t_o) = \tilde{\mathbf{A}}\mathbf{\Phi}(t, t_o) \tag{5.6}$$

where

$$\tilde{\mathbf{A}} = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} \tag{5.7}$$

subject to the initial conditions

$$\mathbf{\Phi}(t_o, t_o) = \mathbf{I}_{6\times 6} \tag{5.8}$$

If we define $\mathbf{x}$ as

$$\mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} \tag{5.9}$$

then

$$\dot{\mathbf{x}} = \begin{pmatrix} \mathbf{v} \\ \mathbf{a} \end{pmatrix} \tag{5.10}$$

Now we can write $\tilde{\mathbf{A}}$ as

$$\tilde{\mathbf{A}} = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial \mathbf{v}}{\partial \mathbf{r}} & \frac{\partial \mathbf{v}}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{a}}{\partial \mathbf{r}} & \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \end{pmatrix} \tag{5.11}$$

For convenience, lets use the following notation

$$\mathbf{A} = \frac{\partial \mathbf{v}}{\partial \mathbf{r}} \tag{5.12}$$

$$\mathbf{B} = \frac{\partial \mathbf{v}}{\partial \mathbf{v}} \tag{5.13}$$

$$\mathbf{C} = \frac{\partial \mathbf{a}}{\partial \mathbf{r}} \tag{5.14}$$

$$\mathbf{D} = \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \tag{5.15}$$

5.1.3 Multiple Spacecraft Propgation and Coupled Propagation of the Equations of Motion

5.2 Force Modelling

5.2.1 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

$$\mathbf{a}_{cb} = \nabla U \tag{5.16}$$

where U is the gravitational potential. The potential for a nonspherical body comes from the solution to Laplace's equation:

$$\nabla^2 U = 0 \tag{5.17}$$

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

$$\nabla U = \frac{\partial U}{\partial r}\mathbf{u}_r + \frac{1}{r}\frac{\partial U}{\partial \phi}\mathbf{u}_\phi + \frac{1}{r\cos\phi}\frac{\partial U}{\partial \lambda}\mathbf{u}_\lambda \tag{5.18}$$

We see that there are two singularities in Eq. (5.18). 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[9] developed a uniform expression of the gravitational potential that avoids the singularity at the poles:

$$U = \frac{\mu}{r}\left[1 + \sum_{n=1}^{\infty}\left(\frac{R_\otimes}{r}\right)^n \sum_{m=0}^{n} A_{nm}(u)[C_{nm}cos(m\lambda)\cos^m\phi + S_{nm}sin(m\lambda)\cos^m\phi]\right] \tag{5.19}$$

Examining this form of the potential it is easy that there is not a singularity at the poles when taking the gradient

Draft: Work in Progress

in spherical coordinates. Pines rewrites Eq. (5.19) as

$$U = \frac{\mu}{r}\left[1 + \sum_{n=1}^{\infty}\left(\frac{R_{\oplus}}{r}\right)^n \sum_{m=0}^{n} A_{nm}(u)[C_{nm}r_m(s,t) + S_{nm}i_m(s,t)]\right] \tag{5.20}$$

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

$$r_0 = 1, \qquad r_1 = s, \qquad i_0 = 0, \qquad i_1 = t$$
$$r_m = sr_{m-1} - ti_{m-1}, \qquad i_m = si_{m-1} + tr_{m-1}$$

The coefficients $A_{nm}(u)$ are called "derived" Legendre functions and are given by

$$A_{nm}(u) = \frac{d^m}{du^m}(P_n(u)) \tag{5.21}$$

where we know from Rodrigues'[10] formula that

$$P_{n0}(u) = P_n(u) = \frac{1}{2^n n!}\frac{d^n}{du^n}(u^2-1)^n \tag{5.22}$$

and

$$P_{nm}(u) = (1-u^2)^{m/2}\frac{d^m}{du^m}P_n(u) \tag{5.23}$$

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.[10] 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

$$\bar{P}_{nm}\bar{C}_{nm} = P_{nm}C_{nm} \qquad \bar{P}_{nm}\bar{S}_{nm} = P_{nm}S_{nm} \tag{5.24}$$

GMAT uses the normalization factor N_{nm} given by

$$N_{nm} = \left[\frac{(n-m)!(2n+1)!}{(n+m)!}\right]^{1/2} \tag{5.25}$$

The non-dimensional spherical harmonic coefficients and Legendre functions are

$$\bar{P}_{nm} = N_{nm}P_{nm} \qquad \bar{C}_{nm} = \frac{C_{nm}}{N_{nm}} \qquad \bar{S}_{nm} = \frac{S_{nm}}{N_{nm}} \tag{5.26}$$

The derived Legendre polynomials are normalized using

$$\bar{A}_{nm} = N_{nm}A_{nm} \tag{5.27}$$

where $\bar{A}_{nm}$ are the normalized Legendre polynomials. Lundberg[10] 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

$$\bar{A}_{nm} = u\left[\frac{(2n+1)(2n-1)}{(n-m)(n+m)}\right]^{1/2}\bar{A}_{n-1,m} - \left[\frac{(2n+1)(n-m-1)(n+m-1)}{(2n-3)(n+m)(n-m)}\right]^{1/2}\bar{A}_{n-2,m} \tag{5.28}$$

The recursive algorithm is started using

$$\bar{A}_{11} = \sqrt{3}\cos\phi \tag{5.29}$$
$$\bar{A}_{nn} = \cos\phi\sqrt{\frac{2n+1}{2n}}\bar{A}_{n-1,n-1} \tag{5.30}$$

The above equations are normalized using Eq. (5.27) and used in

The acceleration due to nonspherical gravity can be written as

$$\mathbf{a}_g = \left(\frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u}\right)\hat{\mathbf{r}} + \begin{pmatrix}\frac{1}{r}\frac{\partial U}{\partial s} & \frac{1}{r}\frac{\partial U}{\partial t} & \frac{1}{r}\frac{\partial U}{\partial u}\end{pmatrix}^T \tag{5.31}$$

To simplify the partial derivatives in Eq. (5.31), Pines defines some intermediate variables as follows

$$\begin{aligned}\rho_0 &= \mu/r \\ \rho_1 &= \rho\rho_0 \\ \rho_n &= \rho\rho_{n-1} \quad \text{for } n > 1\end{aligned} \tag{5.32}$$

Using Lundberg's nondimensionalization approach, we can write

$$\begin{aligned}\bar{D}_{nm}(s,t) &= \bar{C}_{nm}r_m(s,t) + \bar{S}_{nm}i_m(s,t) \\ \bar{E}_{nm}(s,t) &= \bar{C}_{nm}r_{m-1}(s,t) + \bar{S}_{nm}i_{m-1}(s,t) \\ \bar{F}_{nm}(s,t) &= \bar{S}_{nm}r_{m-1}(s,t) - \bar{C}_{nm}i_{m-1}(s,t) \\ \bar{G}_{nm}(s,t) &= \bar{C}_{nm}r_{m-2}(s,t) + \bar{S}_{nm}i_{m-2}(s,t) \\ \bar{H}_{nm}(s,t) &= \bar{S}_{nm}r_{m-2}(s,t) - \bar{C}_{nm}i_{m-2}(s,t)\end{aligned} \tag{5.33}$$

The partial derivatives in Eq. (5.31) can be written as

$$\frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u} = -\sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\oplus}}\sum_{m=0}^{n} c_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{D}_{nm} \tag{5.34}$$

Draft: Work in Progress

$$\frac{1}{r}\frac{\partial U}{\partial s} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n}\bar{A}_{nm}(u)m\bar{E}_{nm} \tag{5.35}$$

$$\frac{1}{r}\frac{\partial U}{\partial t} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n}\bar{A}_{nm}(u)m\bar{F}_{nm} \tag{5.36}$$

$$\frac{1}{r}\frac{\partial U}{\partial s} = \sum_{n=0}^{\infty}\frac{\rho_{n+1}}{R_{\otimes}}\sum_{m=0}^{n}c_{n,m+1}\bar{A}_{n,m+1}(u)\bar{D}_{nm} \tag{5.37}$$

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:

$$a_{11} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}m(m-1)\bar{A}_{nm}\bar{G}_{nm} \tag{5.38}$$

$$a_{12} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}m(m-1)\bar{A}_{nm}\bar{H}_{nm} \tag{5.39}$$

$$a_{13} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}mc_{n,m+1}\bar{A}_{n,m+1}\bar{E}_{nm} \tag{5.40}$$

$$a_{14} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}mc_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{E}_{nm} \tag{5.41}$$

$$a_{23} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}mc_{n,m+1}\bar{A}_{n,m+1}\bar{F}_{nm} \tag{5.42}$$

$$a_{24} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}mc_{n+1,m+1}\bar{A}_{n+1,m+1}\bar{F}_{nm} \tag{5.43}$$

$$a_{33} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}c_{n,m+2}\bar{A}_{n,m+2}\bar{D}_{nm} \tag{5.44}$$

$$a_{34} = -\sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}c_{n+1,m+2}\bar{A}_{n+1,m+2}\bar{D}_{nm} \tag{5.45}$$

$$a_{44} = \sum_{n=0}^{\infty}\frac{\rho_{n+2}}{R_{\otimes}^2}\sum_{m=0}^{n}c_{n+2,m+2}\bar{A}_{n+2,m+2}\bar{D}_{nm} \tag{5.46}$$

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,

$$\mathbf{C}_g = \frac{\partial \mathbf{a}_g}{\partial \mathbf{r}} \tag{5.47}$$

where $\mathbf{C}_g$ is a symmetric matrix with components given by

$$\begin{aligned} c_{11} &= a_{11} + s^2a_{44} + a_4/r + 2sa_{14} && (5.48)\\ c_{12} &= c_{21} = a_{12} + sta_{44} + sa_{24} + ta_{14} && (5.49)\\ c_{13} &= c_{31} = a_{13} + sua_{44} + sa_{34} + ua_{14} && (5.50)\\ c_{22} &= -a_{11} + t^2a_{44} + a_4/r + 2ta_{24} && (5.51)\\ c_{23} &= c_{32} = a_{23} + tua_{44} + ua_{24} + ta_{34} && (5.52)\\ c_{33} &= a_{33} + u^2a_{44} + a_4/r + 2*u*a_{34} && (5.53) \end{aligned}$$

Note that

$$a_4 = \frac{\partial U}{\partial r} - \frac{s}{r}\frac{\partial U}{\partial s} - \frac{t}{r}\frac{\partial U}{\partial t} - \frac{u}{r}\frac{\partial U}{\partial u} \tag{5.54}$$

and is given in Eq. (5.34).

5.2.2 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.5.1. Assume the jth body is the central body of the integration.

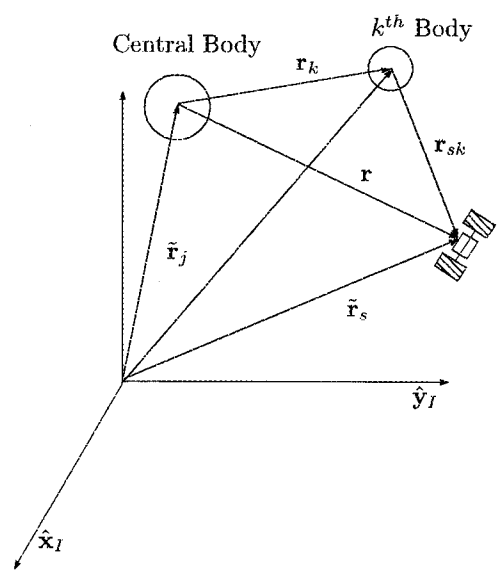

Figure 5.1: N-Body Illustration

- $\tilde{\mathbf{r}}_s$ 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 kth gravitational body with respect a hypothesized inertial frame.

Draft: Work in Progress

- $\mathbf{r}$ 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 kth gravitational body with respect to the central body.

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.5.1 we see that

$$\tilde{\mathbf{r}}_j + \mathbf{r} = \tilde{\mathbf{r}}_s \tag{5.55}$$

By reordering and taking the second derivative with respect to time we obtain

$$\ddot{\mathbf{r}} = \ddot{\tilde{\mathbf{r}}}_s - \ddot{\tilde{\mathbf{r}}}_j \tag{5.56}$$

We can apply Newton's 2nd Law to the spacecraft and obtain

$$m_s \ddot{\tilde{\mathbf{r}}}_s = \sum_{k=1}^{n} F_k = G \sum_{k=1}^{n} \frac{m_s m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} (\mathbf{r}_k - \mathbf{r}) \tag{5.57}$$

where $(\mathbf{r}_k - \mathbf{r})$ is a vector from the spacecraft to the kth body, m_s is the mass of the spacecraft, and m_k is the mass of the kth body. We can write $\ddot{\tilde{\mathbf{r}}}_s$ as simply

$$\ddot{\tilde{\mathbf{r}}}_s = G \sum_{k=1}^{n} \frac{m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} (\mathbf{r}_k - \mathbf{r}) \tag{5.58}$$

We can apply Newton's 2nd Law to the jth body and obtain

$$m_j \ddot{\tilde{\mathbf{r}}}_j = \frac{G m_s m_j}{r^3} \mathbf{r} + G \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{m_j m_k}{\|\mathbf{r}_k\|^3} \mathbf{r}_k \tag{5.59}$$

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

$$\ddot{\tilde{\mathbf{r}}}_j = \frac{G m_s}{r^3} \mathbf{r} + G \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{m_k}{\|\mathbf{r}_k\|^3} \mathbf{r}_k \tag{5.60}$$

Substituting Eq. (5.58) and (5.60) into (5.56) we get

$$\ddot{\mathbf{r}} = G \sum_{k=1}^{n} \frac{m_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} (\mathbf{r}_k - \mathbf{r}) - \frac{G m_s}{r^3} \mathbf{r} - G \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{m_k}{\|\mathbf{r}_k\|^3} \mathbf{r}_k \tag{5.61}$$

Finally, collecting terms yields

$$\mathbf{a}_{pm} = \ddot{\mathbf{r}} = \underbrace{-\frac{\mu_j}{r^3} \mathbf{r}}_{1} + G \sum_{\substack{k=1 \\ k \neq j}}^{n} m_k \left(\underbrace{\frac{\mathbf{r}_k - \mathbf{r}}{\|\mathbf{r}_k - \mathbf{r}\|^3}}_{2} - \underbrace{\frac{\mathbf{r}_k}{\|\mathbf{r}_k\|^3}}_{3} \right) \tag{5.62}$$

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 kth body on the spacecraft. The third type of terms are called indirect. They account for the force of the kth 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,

$$\mathbf{A}_{pm} = \mathbf{D}_{pm} = \mathbf{0}_{3\times 3} \tag{5.63}$$

We also know that

$$\mathbf{B}_{pm} = \mathbf{I}_{3\times 3} \tag{5.64}$$

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

$$\frac{\partial}{\partial \mathbf{r}} \left(-\frac{\mu_j}{r^3} \mathbf{r} \right) = -\frac{\mu_j}{r^3} \mathbf{I}_3 + 3\mu_j \frac{\mathbf{r}\mathbf{r}^T}{r^5} \tag{5.65}$$

Similarly, applying Eq. (12.4) to the direct terms we see that

$$\frac{\partial}{\partial \mathbf{r}} \left(\sum_{\substack{k=1 \\ k \neq j}}^{n} \mu_k \frac{\mathbf{r}_k - \mathbf{r}}{\|\mathbf{r}_k - \mathbf{r}\|^3} \right) = -\sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} \mathbf{I}_3 + 3 \sum_{\substack{k=1 \\ k \neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5} \right) \tag{5.66}$$

Finally, the derivative of the indirect terms are zero and we have

$$\mathbf{C}_{pm} = \underbrace{-\frac{\mu_j}{r^3} \mathbf{I}_3 + 3\mu_j \frac{\mathbf{r}\mathbf{r}^T}{r^5}}_{1} \underbrace{-\sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} \mathbf{I}_3 + 3 \sum_{\substack{k=1 \\ k \neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5} \right)}_{2} \tag{5.67}$$

Combining similar terms we can express the result as

$$\mathbf{C}_{pm} = -\left(\frac{\mu_j}{r^3} + \sum_{\substack{k=1 \\ k \neq j}}^{n} \frac{\mu_k}{\|\mathbf{r}_k - \mathbf{r}\|^3} \right) \mathbf{I}_3 + 3 \left(\mu_j \frac{\mathbf{r}\mathbf{r}^T}{r^5} + \sum_{\substack{k=1 \\ k \neq j}}^{n} \mu_k \left(\frac{(\mathbf{r}_k - \mathbf{r})(\mathbf{r}_k - \mathbf{r})^T}{(\|\mathbf{r}_k - \mathbf{r})\|^5} \right) \right) \tag{5.68}$$

Draft: Work in Progress

5.2.3 Atmospheric Drag

$$\mathbf{a}_d = -\frac{1}{2}\rho v_{rel}^2 \frac{C_d A}{m_s}\hat{\mathbf{v}}_{rel} \tag{5.69}$$

where

$$\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega}_{\otimes} \times \mathbf{r} \tag{5.70}$$

where $\boldsymbol{\omega}_{\otimes}$ is the Earth's angular velocity vector in the FK5 system.

GMAT does not currently support calculating the STM using drag. The components of the sensitivity matrix $\tilde{\mathbf{A}}$ contain derivatives of the atmospheric density with respect to position. These derivatives are non trivial for most density models and are not currently included in GMAT.

5.2.4 Solar Radiation Pressure

$$\mathbf{a}_s = -P_{SR}\frac{C_R A}{m_s}\hat{\mathbf{s}} \tag{5.71}$$

where $\hat{\mathbf{s}}$ is a unitized vector pointing from the spacecraft to the sun

$$\mathbf{s} = \mathbf{r}_s - \mathbf{r} \tag{5.72}$$

where $\mathbf{r}_s$ is the Sun's position vector and $\mathbf{r}$ is the spacecrafts position vector.

$$\mathbf{A}_s = \mathbf{D}_s = \mathbf{0}_{3\times 3} \tag{5.73}$$

$$\mathbf{B}_s = \mathbf{I}_{3\times 3} \tag{5.74}$$

$$\mathbf{C}_s = P_{SR}\frac{C_R A}{m_s}\left(\frac{1}{s^3}\mathbf{I}_3 - 3\frac{\mathbf{s}\mathbf{s}^T}{s^5}\right) \tag{5.75}$$

where

$$s = \|\mathbf{s}\| \tag{5.76}$$

5.2.5 Spacecraft Thrust

5.3 Environment Modelling

5.3.1 Celestial Body Ephemeris

5.3.2 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.

5.3.3 Atmospheric Density

$$28K_p + 0.03e^{K_p} = A_p + 100\left(1 - e^{(-0.08A_p)}\right) \tag{5.77}$$

5.3.4 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

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

Solar Radiation Pressure

Draft: Work in Progress

Chapter 6

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 the users to use several common attitude parameterizations including quaternions, Euler angles, the Direction Cosine Matrix (DCM), Euler angle rates, and the angular velocity vector. Given an initial attitude state, GMAT can propagate the attitude using one of several kinematic attitude propagation models.

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 attitude modes. We conclude the chapter with a discussion of how GMAT converts between different attitude parameterizations.

6.1 Attitude Propagation

Given a set of initial conditions that define the attitude, GMAT can propagate the attitude using several methods. Currently, GMAT only supports kinematic 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. This section presents the different Kinematic attitude modes, and how GMAT calculates the attitude state in each mode. Let's begin by looking at the internal attitude state representation and how the user can define initial conditions.

6.1.1 Internal State Representation and Attitude Initial Conditions

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 orientation depending upon 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 DCM that represents the rotation from the inertial system to the body axes as the attitude orientation parameterization. In the future, when 6 degree of freedom attitude modelling is implemented, GMAT will use the quaternion that represents the rotation from the inertial system to the body axes. GMAT uses the angular velocity of the body with respect to the inertial frame, expressed in the body frame, $\{\omega_{IB}\}_B$, 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 orientation with respect to $\mathcal{F}_i$ using Euler angles, the DCM, or quaternions. The user can define the body rate with respect to $\mathcal{F}_i$ by defining the angular velocity in $\mathcal{F}_i$, $\{\omega_{IB}\}_i$, 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.

6.1.2 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

Draft: Work in Progress

attitude equations of motion, but rather calculates the attitude based on the geometrical definition provided by the user. There are several Kinematic modes to choose from. 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 supplies two pieces of information. They first specify a coordinate system in which to fix the attitude, $\mathcal{F}_i$. $\mathcal{F}_i$ can be any of the default coordinate systems or any user defined coordinate system. Secondly, the user specifies how the body axis system, $\mathcal{F}_B$ is oriented with respect to $\mathcal{F}_i$ by defining $\mathbf{R}_{Bi}$ or an equivalent parameterization. With this information, GMAT calculates the rotation from the inertial to the body axes and the angular velocity of the body with respect to the inertial frame, expressed in the body frame, $\{\boldsymbol{\omega}_{IB}\}_B$.

GMAT calculates the rotation matrix from $\mathcal{F}_i$ to $\mathcal{F}_B$, $\mathbf{R}_{Bi}$, from the initial conditions provided by the user. For CSFixed mode, $\mathbf{R}_{Bi}$ is constant and is stored for use in the equations below. Knowing $\mathbf{R}_{Bi}$, we can calculate the rotation matrix from the inertial frame to the body frame, $\mathbf{R}_{BI}$, using the following equation

$$\mathbf{R}_{BI} = \mathbf{R}_{Bi}\mathbf{R}_{iI} \tag{6.1}$$

$\mathbf{R}_{iI}$ is the rotation matrix from $\mathcal{F}_I$ 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. 3.

To calculate $\{\boldsymbol{\omega}_{IB}\}_B$, we start from Eulers' equation:

$$\dot{\mathbf{R}}_{BI} = -\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B\mathbf{R}_{BI} \tag{6.2}$$

where

$$\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{6.3}$$

and $\{\boldsymbol{\omega}_{IB}\}_B$ is the rotation of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$, expressed in $\mathcal{F}_B$. Solving Eq. 6.2 for $\{\boldsymbol{\omega}^{\times}_{IB}\}_B$ we obtain

$$\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B = -\dot{\mathbf{R}}_{BI}\mathbf{R}_{BI}^T \tag{6.4}$$

Taking the derivative of Eq. (6.1) with respect to time yields

$$\dot{\mathbf{R}}_{BI} = \mathbf{R}_{Bi}\dot{\mathbf{R}}_{iI} \tag{6.5}$$

because by definition, for the CSFixed mode, $\dot{\mathbf{R}}_{Bi} = \mathbf{0}$. Substituting Eq. (6.5) into Eq. (6.4) we obtain

$$\{\boldsymbol{\omega}^{\times}{}_{IB}\}_B = -\mathbf{R}_{Bi}\dot{\mathbf{R}}_{iI}\mathbf{R}_{BI}^T \tag{6.6}$$

where $\mathbf{R}_{Bi}$ is known from user input, and $\mathbf{R}_{BI}$ is known from Eq. (6.1) . GMAT knows how to calculate $\dot{\mathbf{R}}_{iI}$ for all allowable $\mathcal{F}_i$ and details are contained in Ch. 3.

In summary, in CSFixed mode, Eq.(6.1) is used to calculate $\mathbf{R}_{BI}$, and Eq. (6.6) is used to calculate $\{\boldsymbol{\omega}_{IB}\}_B$. If another attitude parameterization is required, GMAT uses the algorithms in Sec. 6.2 to transform from $\mathbf{R}_{BI}$ and $\{\boldsymbol{\omega}_{IB}\}_B$ to the required parameterization. Now let's look at the spinning spacecraft mode.

Spinning Spacecraft Mode

In spinnning 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 determined from the attitude initial conditions, the angular velocity vector, and the elapsed time from the initial spacecraft epoch. Let's take a closer look at the calculations.

In the spinning spacecraft mode, the user provides three pieces of information. They first choose a coordinate system, $\mathcal{F}_i$, in which to define the initial conditions. Secondly, they define the initial orientation with respect to $\mathcal{F}_i$ by providing $\mathbf{R}_{Bi}$ or an equivalent parameterzation that is then converted to the DCM. 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 $\mathbf{R}_{BI}(t)$ where t is an arbitrary epoch, we begin by calculating $\mathbf{R}_{B_oI}$ where $\mathbf{R}_{B_oI} = \mathbf{R}_{BI}(t_o)$. We calculate $\mathbf{R}_{B_oI}$ using

$$\mathbf{R}_{B_oI} = \mathbf{R}_{Bi}\mathbf{R}_{iI}(t_o) \tag{6.7}$$

where $\mathbf{R}_{Bi}$ comes from user provided data, and $\mathbf{R}_{iI}(t_o)$ is calculated by GMAT and is dependent upon $\mathcal{F}_i$. See Ch. 3 for details on how GMAT calculates $\mathbf{R}_{iI}$ for all allowable coordinate systems in GMAT.

Before calculating $\mathbf{R}_{BI}(t)$ we must determine the spin axis in the body frame, $\{\boldsymbol{\omega}_{IB}\}_B$. The user provides $\{\boldsymbol{\omega}_{IB}\}_i$. In spinning mode we assume the spin axis direction is constant in inertial space and in the body frame so $\{\boldsymbol{\omega}_{IB}\}_B(t) = \{\boldsymbol{\omega}_{IB}\}_B(t_o) = \{\boldsymbol{\omega}_{IB}\}_B$. We can find the spin axis in the body frame using $\mathbf{R}_{Bi}$ as follows

$$\{\boldsymbol{\omega}_{IB}\}_B = \mathbf{R}_{Bi}\{\boldsymbol{\omega}_{IB}\}_i \tag{6.8}$$

Once calculated, GMAT saves $\mathbf{R}_{B_oI}$ and $\{\boldsymbol{\omega}_{IB}\}_B$ for use in calculating the attitude orientation and rate at other epochs.

GMAT calculates $\mathbf{R}_{BI}(t)$ using the Euler axis/angle rotation algorithm in Sec. 6.2. The Euler axis is simply the unitized angular velocity vector or,

$$\mathbf{a} = \frac{\{\boldsymbol{\omega}_{IB}\}_B}{\omega_{IB}} \tag{6.9}$$

Draft: Work in Progress

where

$$\omega_{IB} = \|\{\boldsymbol{\omega}_{IB}\}_B\| \tag{6.10}$$

The Euler angle ϕ is calculated using

$$\phi(t) = \omega_{IB}(t - t_o) \tag{6.11}$$

where t is the current epoch, and t_o is the spacecraft's initial epoch. Let's define the rotation matrix that results from the Euler axis/angle rotation using $\mathbf{a}$ and $\phi(t)$, as $\mathbf{R}_{BB_o}(t)$. We can calculate $\mathbf{R}_{BI}(t)$ using

$$\mathbf{R}_{BI}(t) = \mathbf{R}_{BB_o}(t)\mathbf{R}_{B_oI} \tag{6.12}$$

To summarize, in spinning mode the user provides $\mathbf{R}_{Bi}$ and $\{\boldsymbol{\omega}_{IB}\}_i$. GMAT assumes that that the spin axis direction is constant, and uses the Euler axis/angle method to propagate the attitude to find $\mathbf{R}_{BI}$.

Now let's look at how GMAT performs conversions between the different attitude parameterizations.

6.2 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, 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 DCM.

6.2.1 Conversion: Quaternions to DCM

Given: $\mathbf{q}$, q_4

Find: $\mathbf{R}$

Name: *ToCosineMatrix*

$$\mathbf{q} = (q_1 \quad q_2 \quad q_3)^T \tag{6.13}$$

$$\mathbf{q}^\times = \begin{pmatrix} 0 & -q3 & q2 \\ q3 & 0 & -q1 \\ -q2 & q1 & 0 \end{pmatrix} \tag{6.14}$$

$$c = \frac{1}{q_1^2 + q_2^2 + q_3^2 + q_4^2} \tag{6.15}$$

$$\mathbf{R} = c\left[(q_4^2 - \mathbf{q}^T\mathbf{q})\mathbf{I}_3 + 2\mathbf{q}\mathbf{q}^T - 2q_4\mathbf{q}^\times\right] \tag{6.16}$$

6.2.2 Conversion: DCM to Quaternions

Given: $\mathbf{R}$

Find: $\mathbf{q}$, q_4

Define following vector

$$\mathbf{v} = [\,R_{11} \quad R_{22} \quad R_{33} \quad \text{trace}(\mathbf{R})\,] \tag{6.17}$$

Define i_m as the index of the maximum component of $\mathbf{v}$

if $i_m = 1$

$$\mathbf{q}'' = \begin{pmatrix} 2v_{i_m} + 1 - \text{trace}(\mathbf{R}) \\ R_{12} + R_{21} \\ R_{13} + R_{31} \\ R_{23} - R_{32} \end{pmatrix} \tag{6.18}$$

if $i_m = 2$

$$\mathbf{q}'' = \begin{pmatrix} R_{21} + R_{12} \\ 2v_{i_m} + 1 - \text{trace}(\mathbf{R}) \\ R_{23} + R_{32} \\ R_{31} - R_{13} \end{pmatrix} \tag{6.19}$$

if $i_m = 3$

$$\mathbf{q}'' = \begin{pmatrix} R_{31} + R_{13} \\ R_{32} + R_{23} \\ 2v_{i_m} + 1 - \text{trace}(\mathbf{R}) \\ R_{12} - R_{21} \end{pmatrix} \tag{6.20}$$

if $i_m = 4$

$$\mathbf{q}'' = \begin{pmatrix} R_{23} - R_{32} \\ R_{31} - R_{13} \\ R_{12} - R_{21} \\ 1 + \text{trace}(\mathbf{R}) \end{pmatrix} \tag{6.21}$$

We normalize $\mathbf{q}''$ using

$$\mathbf{q}' = \frac{\mathbf{q}''}{\|\mathbf{q}''\|} \tag{6.22}$$

Finally,

$$\mathbf{q} = [\,q_1' \quad q_2' \quad q_3'\,]^T \tag{6.23}$$

and

$$q_4 = q_4' \tag{6.24}$$

6.2.3 Conversion: DCM to Euler Axis/Angle

Given: $\mathbf{R}$

Find: $\mathbf{a}$, ϕ

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{6.25}$$

$$\phi = \cos^{-1}\left(\frac{1}{2}\left(\text{trace}(\mathbf{R}) - 1\right)\right) \tag{6.26}$$

Draft: Work in Progress

$$\mathbf{a} = \frac{1}{2\sin\phi}\begin{pmatrix} R_{23} - R_{32} \\ R_{31} - R_{13} \\ R_{12} - R_{21} \end{pmatrix} \tag{6.27}$$

If $\|\sin\phi\| < 10^{-14}$, then we assume

$$\mathbf{a} = [\,1 \;\; 0 \;\; 0\,]^T \tag{6.28}$$

Note that if $\|\sin\phi\| < 10^{-14}$ then $\cos\phi \approx 1$ and we arrive at a DCM of $\mathbf{I}_3$.

6.2.4 Conversion: Euler Axis/Angle to DCM

Given: $\mathbf{a}$, ϕ

Find: $\mathbf{R}$

$$\mathbf{a}^\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{6.29}$$

$$\mathbf{R} = \cos\phi\mathbf{I}_3 + (1-\cos\phi)\mathbf{a}\mathbf{a}^T - \sin\phi\mathbf{a}^\times \tag{6.30}$$

6.2.5 Conversion: Euler Angles to DCM

Given: Sequence order (i.e. 123, 121, ... 313), θ_1, θ_2, θ_3

Find: $\mathbf{R}$

We'll give an example for a 321 rotation, and then present results for the remaining 11 Euler angle sequences. First, let's define $\mathbf{R}_3(\theta_1)$, $\mathbf{R}_2(\theta_2)$, and $\mathbf{R}_1(\theta_3)$.

$$\mathbf{R}_3(\theta_1) = \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.31}$$

$$\mathbf{R}_2(\theta_2) = \begin{pmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix} \tag{6.32}$$

$$\mathbf{R}_1(\theta_3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{pmatrix} \tag{6.33}$$

Now we can write

$$\mathbf{R}_{321} = \mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1) = \\ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_3 & s_3 \\ 0 & -s_3 & c_3 \end{pmatrix}\begin{pmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{pmatrix}\begin{pmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.34}$$

where $c_1 = \cos\theta_1$, $s_1 = \sin\theta_1$ etc. We can rewrite $\mathbf{R}_{321}$ as

$$\mathbf{R}_{321} = \begin{pmatrix} c_2c_1 & c_2s_1 & -s_2 \\ -c_3s_1 + s_3s_2c_1 & c_3c_1 + s_3s_2s_1 & s_3c_2 \\ s_3s_1 + c_3s_2c_1 & -s_3c_1 + c_3s_2s_1 & c_3c_2 \end{pmatrix} \tag{6.35}$$

The approach is similar for the remaining 11 Euler angle sequences. Rather than derive the DCM matrices for the remaining 11 sequences, we present them in Table 6.1.

6.2.6 Conversion: DCM to Euler Angles

Given: Sequence order (i.e. 123, 121, ... 313), $\mathbf{R}$

Find: θ_1, θ_2, θ_3

We'll give an example for a 321 rotation, and then present results for the remaining 11 Euler angle sequences. Examining, Eq. (6.35), we see that

$$\frac{R_{21}}{R_{11}} = \frac{\cos\theta_2\sin\theta_1}{\cos\theta_2\cos\theta_1} \tag{6.36}$$

From this we can see that

$$\theta_1 = \tan^{-1}\frac{R_{21}}{R_{11}} \tag{6.37}$$

Further inspection of Eq. (6.35) shows us that

$$\theta_2 = \sin^{-1} R_{13} \tag{6.38}$$

At first glance, we may choose to calculate θ_3 using $\theta_3 = \tan^{-1}(R_{23}/R_{33})$. However, in the case that $\theta_2 = 90°$, this would result in the indeterminate case, $\theta_3 = \tan^{-1}(R_{23}/R_{33}) = \tan^{-1}(0/0)$. An improved method, found in the ADEAS mathematical specifications document, is to determine θ_3 using

$$\theta_3 = \tan^{-1}\left(\frac{R_{31}\sin\theta_1 - R_{32}\cos\theta_1}{-R_{21}\sin\theta_1 + R_{22}\cos\theta_1}\right) \tag{6.39}$$

Substituting values from Eq. (6.35) into Eq. (6.39), and using abbreviated notation, we see that

$$\theta_3 = \tan^{-1}\left(\frac{s_1(s_3s_1 + c_3s_2c_1) - c_1(-s_3c_1 + c_3s_2s_1)}{s_1(c_3s_1 - s_3s_2c_1) + c_1(c_3c_1 + s_3s_2s_1)}\right) \tag{6.40}$$

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 6.3.

Note: All $\tan^{-1}$ use a quadrant check (equaivalent to atan2) to make sure the the correct quadrant is chosen.

6.2.7 Conversion: Angular Velocity to Euler Angles Rates

Given: Sequence (i.e. 123, 121, ... 313), θ_2, θ_3 $\boldsymbol{\omega}$

Find: $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = \mathbf{S}^{-1}(\theta_2, \theta_3)\boldsymbol{\omega} \tag{6.41}$$

Draft: Work in Progress

$\mathbf{S}^{-1}(\theta_2, \theta_3)$ is dependent upon the Euler sequence. Table 6.2 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.

6.2.8 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: $\boldsymbol{\omega}$

$$\boldsymbol{\omega} = \mathbf{S}(\theta_2, \theta_3) \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} \tag{6.42}$$

$\mathbf{S}(\theta_2, \theta_3)$ is dependent upon the Euler sequence. Table 6.2 contains the different expressions for $\mathbf{S}^{-1}(\theta_2, \theta_3)$ for each of the 12 unique Euler sequences.

6.2.9 Conversion: Quaternions to Euler Angles

Given: $\mathbf{q}$, q_4, Euler Sequence

Find: θ_1, θ_2, and θ_3

There is not a direct transformation to convert from the quaternions to the Euler Angles. GMAT first converts from the quaternion to the DCM using the algorithm in Sec. 6.2.1. The DCM is then used to calculate the Euler Angles for the given Euler angle sequence using the algorithm in Sec. 6.2.6.

6.2.10 Conversion: Euler Angles to Quaternions

Given: θ_1, θ_2, and θ_3, Euler Sequence

Find: $\mathbf{q}$, q_4

There is not a direct transformation to convert from Euler Angles to quaternions. GMAT first converts from the Euler Angles to the DCM using the algorithm in Sec. 6.2.5. The DCM is then used to calculate the quaternions using the algorithm in Sec. 6.2.2.

6.3 Appendix 1

Draft: Work in Progress

Table 6.1: Rotation Matrices for 12 Unique Euler Angle Rotation Sequences

$$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_3c_2 & c_3s_2s_1 + s_3c_1 & -c_3s_2c_1 + s_1s_3 \\ -s_3c_2 & -s_3s_2s_1 + c_3c_1 & s_3s_2c_1 + c_3s_1 \\ s_2 & -c_2s_1 & c_2c_1 \end{pmatrix}$$

$$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_3c_2 & c_3s_2c_1 + s_1s_3 & c_3s_2s_1 - s_3c_1 \\ -s_2 & c_2c_1 & c_2s_1 \\ s_3c_2 & s_3s_2c_1 - c_3s_1 & s_3s_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_2c_1 & s_2 & -c_2s_1 \\ -c_3s_2c_1 + s_3s_1 & c_3c_2 & c_3s_2s_1 + s_3c_1 \\ s_3s_2c_1 + c_3s_1 & -s_3c_2 & -s_3s_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_1 + s_3s_2s_1 & s_3c_2 & -c_3s_1 + s_3s_2c_1 \\ -s_3c_1 + c_3s_2s_1 & c_3c_2 & s_3s_1 + c_3s_2c_1 \\ c_2s_1 & -s_2 & c_2c_1 \end{pmatrix}$$

$$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_1 - s_3s_2s_1 & c_3s_1 + s_3s_2c_1 & -s_3c_2 \\ -c_2s_1 & c_2c_1 & s_2 \\ s_3c_1 + c_3s_2s_1 & s_3s_1 - c_3s_2c_1 & c_3c_2 \end{pmatrix}$$

$$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_2c_1 & c_2s_1 & -s_2 \\ -c_3s_1 + s_3s_2c_1 & c_3c_1 + s_3s_2s_1 & s_3c_2 \\ s_3s_1 + c_3s_2c_1 & -s_3c_1 + c_3s_2s_1 & c_3c_2 \end{pmatrix}$$

$$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_2 & s_2s_1 & -s_2c_1 \\ s_3s_2 & c_3c_1 - s_3c_2s_1 & c_3s_1 + s_3c_2c_1 \\ c_3s_2 & -s_3c_1 - c_3c_2s_1 & -s_3s_1 + c_3c_2c_1 \end{pmatrix}$$

$$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1) = \begin{pmatrix} c_2 & s_2c_1 & s_2s_1 \\ -c_3s_2 & c_3c_2c_1 - s_3s_1 & c_3c_2s_1 + s_3c_1 \\ s_3s_2 & -s_3c_2c_1 - c_3s_1 & -s_3c_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_1 - s_3c_2s_1 & s_3s_2 & -c_3s_1 - s_3c_2c_1 \\ s_2s_1 & c_2 & s_2c_1 \\ s_3c_1 + c_3c_2s_1 & -c_3s_2 & -s_3s_1 + c_3c_2c_1 \end{pmatrix}$$

$$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1) = \begin{pmatrix} c_3c_2c_1 - s_3s_1 & c_3s_2 & -c_3c_2s_1 - s_3c_1 \\ -s_2c_1 & c_2 & s_2s_1 \\ s_3c_2c_1 + c_3s_1 & s_3s_2 & -s_3c_2s_1 + c_3c_1 \end{pmatrix}$$

$$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_1 - s_3c_2s_1 & c_3s_1 + s_3c_2c_1 & s_3s_2 \\ -s_3c_1 - c_3c_2s_1 & -s_3s_1 + c_3c_2c_1 & c_3s_2 \\ s_2s_1 & -s_2c_1 & c_2 \end{pmatrix}$$

$$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1) = \begin{pmatrix} c_3c_2c_1 - s_3s_1 & c_3c_2s_1 + s_3c_1 & -c_3s_2 \\ -s_3c_2c_1 - c_3s_1 & -s_3c_2s_1 + c_3c_1 & s_3s_2 \\ s_2c_1 & s_2s_1 & c_2 \end{pmatrix}$$

Draft: Work in Progress

Table 6.2: Kinematics of Euler Angle Rotation Sequences

Euler Sequence	$\mathbf{S}(\theta_2,\theta_3)$	$\mathbf{S}^{-1}(\theta_2,\theta_3)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c3c2 & s3 & 0 \\ -s3c2 & c3 & 0 \\ s2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} c3/c2 & -s3/c2 & 0 \\ s3 & c3 & 0 \\ -s2c3/c2 & s3s2/c2 & 1 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c3c2 & -s3 & 0 \\ -s2 & 0 & 1 \\ s3c2 & c3 & 0 \end{pmatrix}$	$\begin{pmatrix} c3/c2 & 0 & s3/c2 \\ -s3 & 0 & c3 \\ s2c3/c2 & 1 & s3s2/c2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s2 & 0 & 1 \\ c3c2 & s3 & 0 \\ -s3c2 & c3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & c3/c2 & -s3/c2 \\ 0 & s3 & c3 \\ 1 & -s2c3/c2 & s3s2/c2 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s3c2 & c3 & 0 \\ c3c2 & -s3 & 0 \\ -s2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} s3/c2 & c3/c2 & 0 \\ c3 & -s3 & 0 \\ s3s2/c2 & s2c3/c2 & 1 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -s3c2 & c3 & 0 \\ s2 & 0 & 1 \\ c3c2 & s3 & 0 \end{pmatrix}$	$\begin{pmatrix} -s3/c2 & 0 & c3/c2 \\ c3 & 0 & s3 \\ s3s2/c2 & 1 & -s2c3/c2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -s2 & 0 & 1 \\ s3c2 & c3 & 0 \\ c3c2 & -s3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & s3/c2 & c3/c2 \\ 0 & c3 & -s3 \\ 1 & s3s2/c2 & s2c3/c2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c2 & 0 & 1 \\ s3s2 & c3 & 0 \\ c3s2 & -s3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & s3/s2 & c3/s2 \\ 0 & c3 & -s3 \\ 1 & -s3c2/s2 & -c3c2/s2 \end{pmatrix}$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\begin{pmatrix} c2 & 0 & 1 \\ -c3s2 & s3 & 0 \\ s3s2 & c3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -c3/s2 & s3/s2 \\ 0 & s3 & c3 \\ 1 & c3c2/s2 & -s3c2/s2 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} s3s2 & c3 & 0 \\ c2 & 0 & 1 \\ -c3s2 & s3 & 0 \end{pmatrix}$	$\begin{pmatrix} s3/s2 & 0 & -c3/s2 \\ c3 & 0 & s3 \\ -s3c2/s2 & 1 & c3c2/s2 \end{pmatrix}$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\begin{pmatrix} c3s2 & -s3 & 0 \\ c2 & 0 & 1 \\ s3s2 & c3 & 0 \end{pmatrix}$	$\begin{pmatrix} c3/s2 & 0 & s3/s2 \\ -s3 & 0 & c3 \\ -c3c2/s2 & 1 & -s3c2/s2 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} s3s2 & c3 & 0 \\ c3s2 & -s3 & 0 \\ c2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} s3/s2 & c3/s2 & 0 \\ c3 & -s3 & 0 \\ -s3c2/s2 & -c3c2/s2 & 1 \end{pmatrix}$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\begin{pmatrix} -c3s2 & s3 & 0 \\ s3s2 & c3 & 0 \\ c2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -c3/s2 & s3/s2 & 0 \\ s3 & c3 & 0 \\ c3c2/s2 & -s3c2/s2 & 1 \end{pmatrix}$

Draft: Work in Progress

Table 6.3: Computation of Euler Angles from DCM

Euler Sequence	Euler Angle Computations		
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(-R_{32}/R_{33})$	$\theta_2 = \sin^{-1}(R_{31})$	$\theta_3 = \tan^{-1}\left(\frac{R_{13}\sin\theta_1 + R_{12}\cos\theta_1}{R_{23}\sin\theta_1 + R_{22}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(R_{23}/R_{22})$	$\theta_2 = \sin^{-1}(-R_{21})$	$\theta_3 = \tan^{-1}\left(\frac{R_{12}\sin\theta_1 - R_{13}\cos\theta_1}{-R_{32}\sin\theta_1 + R_{33}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(-R_{13}/R_{11})$	$\theta_2 = \sin^{-1}(R_{12})$	$\theta_3 = \tan^{-1}\left(\frac{R_{21}\sin\theta_1 + R_{23}\cos\theta_1}{R_{31}\sin\theta_1 + R_{33}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(R_{31}/R_{33})$	$\theta_2 = \sin^{-1}(-R_{32})$	$\theta_3 = \tan^{-1}\left(\frac{R_{23}\sin\theta_1 - R_{21}\cos\theta_1}{-R_{13}\sin\theta_1 + R_{11}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(-R_{21}/R_{22})$	$\theta_2 = \sin^{-1}(R_{23})$	$\theta_3 = \tan^{-1}\left(\frac{R_{32}\sin\theta_1 + R_{31}\cos\theta_1}{R_{12}\sin\theta_1 + R_{11}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(R_{12}/R_{11})$	$\theta_2 = \sin^{-1}(-R_{13})$	$\theta_3 = \tan^{-1}\left(\frac{R_{31}\sin\theta_1 - R_{32}\cos\theta_1}{-R_{21}\sin\theta_1 + R_{22}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(R_{12}/(-R_{13}))$	$\theta_2 = \cos^{-1}(R_{11})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{33}\sin\theta_1 - R_{32}\cos\theta_1}{R_{23}\sin\theta_1 + R_{22}\cos\theta_1}\right)$
$\mathbf{R}_1(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_1(\theta_1)$	$\theta_1 = \tan^{-1}(R_{13}/(R_{12}))$	$\theta_2 = \cos^{-1}(R_{11})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{22}\sin\theta_1 + R_{23}\cos\theta_1}{-R_{32}\sin\theta_1 + R_{33}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(R_{21}/(R_{23}))$	$\theta_2 = \cos^{-1}(R_{22})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{33}\sin\theta_1 + R_{31}\cos\theta_1}{-R_{13}\sin\theta_1 + R_{11}\cos\theta_1}\right)$
$\mathbf{R}_2(\theta_3)\mathbf{R}_3(\theta_2)\mathbf{R}_2(\theta_1)$	$\theta_1 = \tan^{-1}(R_{23}/(-R_{21}))$	$\theta_2 = \cos^{-1}(R_{22})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{11}\sin\theta_1 - R_{13}\cos\theta_1}{R_{31}\sin\theta_1 + R_{33}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_1(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(R_{31}/(-R_{32}))$	$\theta_2 = \cos^{-1}(R_{33})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{22}\sin\theta_1 - R_{21}\cos\theta_1}{R_{12}\sin\theta_1 + R_{11}\cos\theta_1}\right)$
$\mathbf{R}_3(\theta_3)\mathbf{R}_2(\theta_2)\mathbf{R}_3(\theta_1)$	$\theta_1 = \tan^{-1}(R_{32}/(R_{31}))$	$\theta_2 = \cos^{-1}(R_{33})$	$\theta_3 = \tan^{-1}\left(\frac{-R_{11}\sin\theta_1 + R_{12}\cos\theta_1}{-R_{21}\sin\theta_1 + R_{22}\cos\theta_1}\right)$

Draft: Work in Progress

Chapter 7

Numerical Integrators

7.1 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 ith 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 ith 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

Draft: Work in Progress

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

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.

7.1.1 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.

7.2 Prince-Dormand Integrators

7.3 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/white/ma` 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:

$$r_{i+1}^j = r_i^j + \frac{h}{24}\left[9 f_{n+1}^{*j} + 19 f_n^j - 5 f_{n-1}^j + 1 f_{n-2}^j\right]$$

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).

7.4 Bulirsch-Stoer

7.5 Stopping Condition Algorithm

7.6 Integrator Coefficients

Draft: Work in Progress

Table 7.1: 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 7.2: 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	$\frac{98415}{321776} - \frac{19683}{71825}$	$\frac{16807}{146016} - \frac{175273}{912600}$	$\frac{1375}{7344} - \frac{395}{3672}$	$\frac{1375}{5408} - \frac{785}{2704}$	$-\frac{37}{1120} - \frac{3}{50}$	$\frac{1}{10}$

Draft: Work in Progress

Table 7.3: 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	$-\frac{5}{66}$	$\frac{5}{66}$	$\frac{5}{66}$

Draft: Work in Progress

Chapter 8

Mathematics in GMAT Scripting

8.1 Basic Operators

8.2 Math Functions

8.2.1 max

```
[maxX] = max(X)
```

X is an nxm array. maxX is a 1xm row vector containing the maximum value in each column of X.

8.2.2 min

```
[minX] = min(X)
```

X is an nxm array. minX is a 1xm array containing the minumum value contained in each row of X.

8.2.3 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.

8.2.4 mean

```
[meanX] = mean(X)
```

X is an nxm array. meanX is a 1xm row vector containing the mean of each column of X.

8.2.5 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.

8.2.6 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.

8.2.7 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.

8.2.8 det

```
[detX] = norm(X)
```

The det function calculates the determinant of a matrix. X is an nxn array. detX is the determinant of X.

8.2.9 inv

```
[invX] = inv(X)
```

The inv function returns the inverse of a matrix. X must be a square matrix.

Draft: Work in Progress

8.2.10 eig

8.2.11 sin, cos, tan

8.2.12 asin, acos, atan, atan2

8.2.13 sinh, cosh, tanh

8.2.14 asinh, acosh, atanh

8.2.15 transpose

8.2.16 DegToRad

8.2.17 RadToDeg

8.2.18 log

8.2.19 log10

8.2.20 exp

8.2.21 sqrt

Draft: Work in Progress

Chapter 9

Solvers

9.1 Differential Correction

9.2 Broyden's Method

9.3 Newton's Method

9.4 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
```

9.5 The Vary Command

The user defines the independent variables by the vary command,

Table 9.1: 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 =`

Draft: Work in Progress

Draft: Work in Progress

Chapter 10

Event Functions

Event functions in GMAT allow a user to determine when different types of events occur such as station contacts, eclipse events, or spacecraft-to-spacecraft line of sight. In general, these events are dependent upon the orbit state and other time dependent parameters, and therefore can only be determined during or after orbit propagation. The implementation of Event Functions requires GMAT to find the roots of a parametric function of time. The roots of the parametric equation are the event times.

In this chapter, we'll look at how GMAT calculates the roots of Event Functions. For the implementation in GMAT, this includes two subproblems. The first is determining if a root has occurred during a propagation step. The second, is determining the numerical value of the root. There is a trade between robustness and performance, so we'll look at several options provided in GMAT that allow the user to select between a more robust yet slower approach, or a fast but less robust approach. Let's begin my looking at the mathematical definition of an event function in GMAT.

10.1 Event Function Mathematical Definition

An Event Function in GMAT has three outputs. The general form of an Event Function is

$$[\ \mathbf{f}, \mathbf{d}, \mathbf{p}\ ] = \mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) \tag{10.1}$$

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

$$\mathbf{f} = [\ f_1(t, \mathbf{x}(t), \mathbf{C}), f_2(t, \mathbf{x}(t), \mathbf{C}) ... f_n(t, \mathbf{x}(t), \mathbf{C})\ ]^T \tag{10.2}$$

The output parameter $\mathbf{d}$ 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. $\mathbf{d}$ is a vector that has the same number of elements as $\mathbf{f}$, and the first element of $\mathbf{d}$ corresponds to the first element of $\mathbf{f}$ and so on. Table 10.1 summarizes the allowable choices for components of $\mathbf{d}$ and the action GMAT will take depending upon the selection.

Table 10.1: 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, $\mathbf{p}$, is a flag that allows the user to tell GMAT whether or not a root is possible. If a component of $\mathbf{p}$ is zero, then GMAT will not attempt to try to find a root of the corresponding function in $\mathbf{f}$. 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 calculat-

Draft: Work in Progress

ing 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.

10.2 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 10.1 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.

$$\mathbf{F} = \mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) \tag{10.3}$$

We want to find all t such that

$$\mathbf{F}(t, \mathbf{x}(t), \mathbf{C}) = 0 \tag{10.4}$$

10.3 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 $\mathbf{p}$ and is discussed in section 10.1. If an element of $\mathbf{p}$ 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. $\mathbf{p}$ can be either zero or one, and can change value during propagation. If $\mathbf{p}$ 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.

Draft: Work in Progress

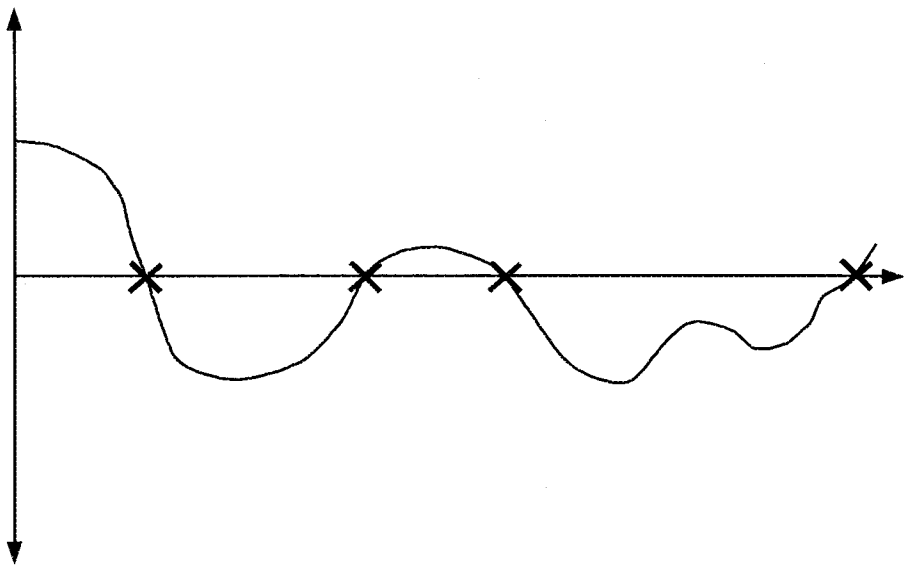

Figure 10.1: Sample Event Function Output

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

10.4 Algorithm for Event Functions

Table 10.2: 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.

Draft: Work in Progress

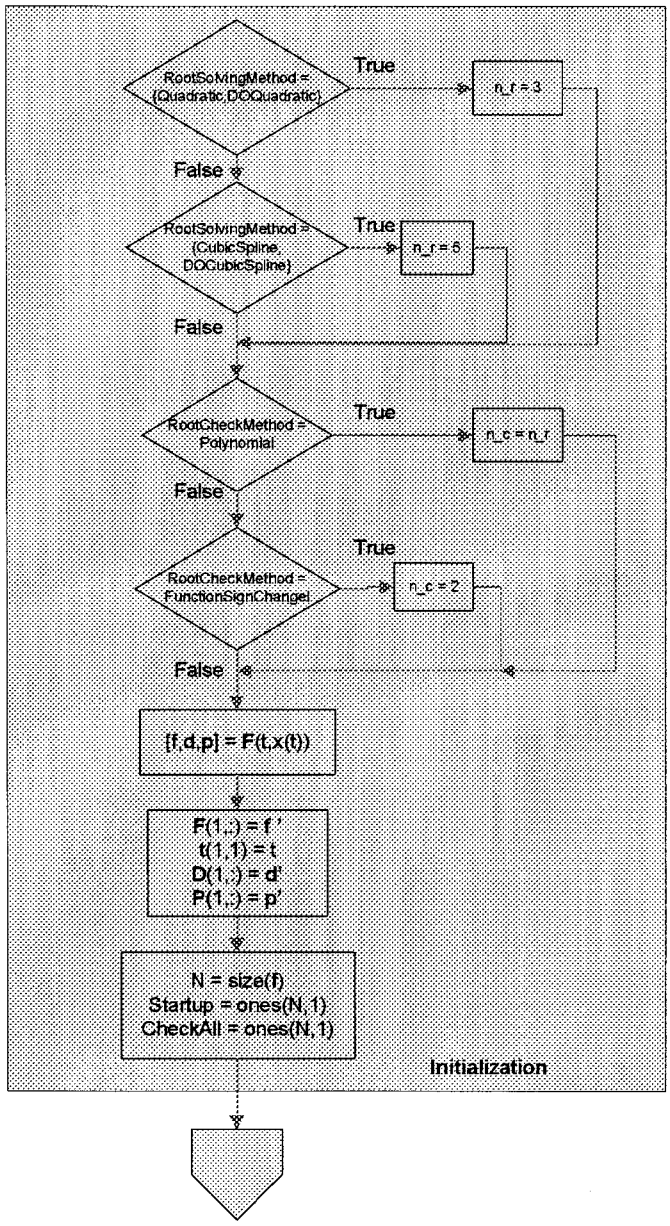

Figure 10.2: Initializations for the Event Location Algorithm

Draft: Work in Progress

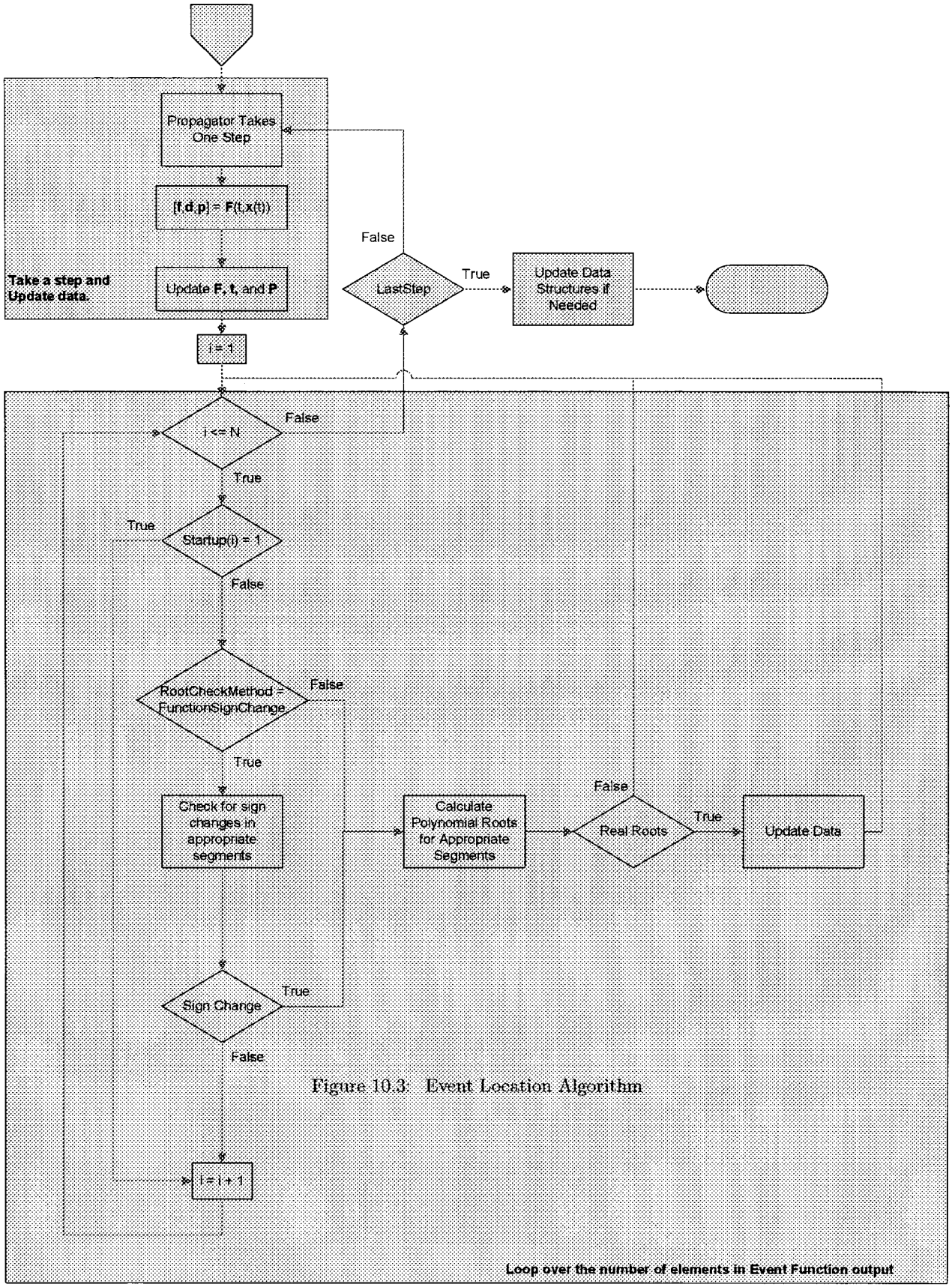

Figure 10.3: Event Location Algorithm

Draft: Work in Progress

10.5 Appendix 1: Root Finding Algorithms

10.6 Quadratic Polynomial

We are given three data points defined by a vector of independent variables

$$\mathbf{x} = [\ x_1\ x_2\ x_3\ ]^T \tag{10.5}$$

and a vector of corresponding dependent variables

$$\mathbf{y} = [\ y_1\ y_2\ y_3\ ]^T \tag{10.6}$$

we wish to find a quadratic polynomial that fits the data such that

$$y = Ax^2 + Bx + C \tag{10.7}$$

We begin by forming the system of linear equations

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \tag{10.8}$$

We can solve for the coefficients using

$$A = \frac{\begin{vmatrix} y_1 & x_1 & 1 \\ y_2 & x_2 & 1 \\ y_3 & x_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}} \tag{10.9}$$

$$B = \frac{\begin{vmatrix} x_1^2 & y_1 & 1 \\ x_2^2 & y_2 & 1 \\ x_3^2 & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{vmatrix}} \tag{10.10}$$

$$C = y_3 - Ax_3^2 - Bx_3 \tag{10.11}$$

10.7 Cubic Spline (Not-a-Knot)

We are given five data points defined by a vector of independent variables

$$\mathbf{x} = [\ x_1\ x_2\ x_3\ x_4\ x_5\ ]^T \tag{10.12}$$

and a vector of corresponding dependent variables

$$\mathbf{y} = [\ y_1\ y_2\ y_3\ y_4\ y_5\ ]^T \tag{10.13}$$

we wish to find the four cubic polynomials, $i = 1, 2, 3, 4$, such that

$$p_i = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{10.14}$$

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

$$h_i = x_{i+1} - x_i \tag{10.15}$$

$$\Delta_i = \frac{y_{i+1} - y_i}{h_i} \tag{10.16}$$

Next we solve the following system of linear equations

$$\mathbf{AS} = \mathbf{B} \tag{10.17}$$

where the components of $\mathbf{A}$ are given by

$$A_{11} = 2h_2 + h_1 \tag{10.18}$$

$$A_{12} = 2h_1 + h_2 \tag{10.19}$$

$$A_{13} = 0 \tag{10.20}$$

$$A_{21} = 0 \tag{10.21}$$

$$A_{22} = h_3 + 2h_4 \tag{10.22}$$

$$A_{23} = 2h_3 + h_4 \tag{10.23}$$

$$A_{31} = \frac{h_2^2}{h_1 + h_2} \tag{10.24}$$

$$A_{32} = \frac{h_1 h_2}{(h_1 + h_2)} + 2(h_2 + h_3) + \frac{h_3 h_4}{(h_3 + h_4)} \tag{10.25}$$

$$A_{33} = \frac{h_3^2}{h_3 + h_4} \tag{10.26}$$

the components of $\mathbf{B}$ are

$$B_{11} = 6(\Delta_2 - \Delta_1) \tag{10.27}$$

$$B_{21} = 6(\Delta_4 - \Delta_3) \tag{10.28}$$

$$B_{31} = 6(\Delta_3 - \Delta_2) \tag{10.29}$$

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

$$S_1 = \frac{\begin{vmatrix} B_{11} & A_{12} & A_{13} \\ B_{21} & A_{22} & A_{23} \\ B_{31} & A_{32} & A_{33} \end{vmatrix}}{|\mathbf{A}|} \tag{10.30}$$

$$S_3 = \frac{\begin{vmatrix} A_{11} & B_{11} & A_{13} \\ A_{21} & B_{21} & A_{23} \\ A_{31} & B_{31} & A_{33} \end{vmatrix}}{|\mathbf{A}|} \tag{10.31}$$

$$S_5 = \frac{B_{31} - A_{31}S_1 - A_{32}S_3}{A_{33}} \tag{10.32}$$

Now we can calculate S_2 and S_4 using

$$S_2 = \frac{h_2 S_1 + h_1 S_3}{h_1 + h_2}; \tag{10.33}$$

$$S_4 = \frac{h_4 S_3 + h_3 S_5}{h_3 + h_4}; \tag{10.34}$$

Draft: Work in Progress

Finally, the coefficients for the ith cubic polynomial are given by

$$a_i = \frac{S_{i+1} - S_i}{6h_i} \tag{10.35}$$

$$b_i = \frac{S_i}{2} \tag{10.36}$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \tag{10.37}$$

$$d_i = y_i \tag{10.38}$$

Draft: Work in Progress

Draft: Work in Progress

Chapter 11

Spacecraft Model

11.1 Orbit

11.2 Attitude

11.3 Ballistics and Mass

11.4 Actuators

11.4.1 Thrust and Impulse Models

GMAT uses polynomial expressions for the thrust and specific impulse imparted to the spacecraft by thrusters attached to the spacecraft. Both thrust and specific impulse are expressed as functions of pressure and temperature. The pressure and temperature are values obtained from fuel tanks containing the fuel. All measurements in GMAT are expressed in metric units. The thrust, in Newtons, applied by a spacecraft engine is given by

$$F_T(P,T) = \left\{C_1 + C_2P + C_3P^2 + C_4P^{C_5} + C_6P^{C_7} + C_8P^{C_9}\right\} + C_{10}C_{11}^{C_{12}P}\Big\}\left(\frac{T}{T_{ref}}\right)^{1+C_{13}+C_{14}P}$$

Pressures are expressed in kilopascals, and temperatures in degrees centigrade. The coefficients C1 - C14 are set by the user. Each coefficient is expressed in units commiserate with the final expression in Newtons; for example, C1 is expressed in Newtons, C2 in Newtons per kilopascal, and so forth.

Specific Impulse, measured in m/s (or, equivalently, Newton Seconds/kilogram) is expressed using a similar equation:

$$I_{sp}(P,T) = \left\{K_1 + K_2P + K_3P^2 + K_4P^{K_5} + K_6P^{K_7} + K_8P^{K_9} + K_{10}K_{11}^{K_{12}P}\right\}\left(\frac{T}{T_{ref}}\right)^{1+K_{13}+K_{14}P}$$

Draft: Work in Progress

Table 11.1 shows the default values for these coefficients.

11.5 Sensors

11.6 Tanks

Mass is depleted from the fuel tanks using equation

$$\frac{dm}{dt} = \frac{F_T}{I_{sp}g}, \tag{11.1}$$

where the thrust and specific impulse are given by ?? and 11.4.1.

This mass depletion is integrated along with the other parameters during propagation.

The tank model in GMAT manages the fuel mass and the input variables for the thrust and specific impulse polynomials. The tank can be run in either a blow-down or pressure regulated mode. In pressure regulated mode, the pressure in the polynomial is held at a fixed value. In blow-down mode, the pressure decreases as fuel is used, following the ideal gas law:

$$PV = nRT \tag{11.2}$$

In GMAT's blow-down model, the temperature T and the number of pressurant molecules n in the tank are held constant, so the right side of this equation is constant. The gas volume available in the tank grows as fuel is consumed, and the pressure decreases accordingly. The gas volume V_G in the tank is computed from the total tank volume, V_T, the mass of the fuel, M_F, and the density of the fuel, ρ:

$$V_G = V_T - \frac{M_F}{\rho} \tag{11.3}$$

Table 2 shown the default values for the tank parameters.

Table 11.2: Default Fuel Tank Parameters

Parameter	Default Value	Units
FuelMass	756	kg
Pressure	1500	kPa
Temperature	20	C
RefTemperature	20	C
Volume	0.75	m^3
FuelDensity	1260	kg/m^3
PressureRegulated	true	

Draft: Work in Progress

Table 11.1: Default Thrust and Specific Impulse Coefficients

Thrust Coefficient	Default	Units	Impulse Coefficient	Default	Units
C_1	500	N	K_1	2150	m/s
C_2	0	N/kPa	K_2	0	$\mathrm{m/(s \cdot kPa)}$
C_3	0	$\mathrm{N/kPa^2}$	K_3	0	$\mathrm{m/(s \cdot kPa^2)}$
C_4	0	$\mathrm{N/kPa^{C_5}}$	K_4	0	$\mathrm{m/(s \cdot kPa^{K_5})}$
C_5	0	none	K_5	0	none
C_6	0	$\mathrm{N/kPa^{C_7}}$	K_6	0	$\mathrm{m/(s \cdot kPa^{K_7})}$
C_7	0	none	K_7	0	none
C_8	0	$\mathrm{N/kPa^{C_9}}$	K_8	0	$\mathrm{m/(s \cdot kPa^{K_9})}$
C_9	0	none	K_9	0	none
C_{10}	0	N	K_{10}	0	m/s
C_{11}	1	none	K_{11}	1	none
C_{12}	0	1/kPa	K_{12}	0	1/kPa
C_{13}	0	none	K_{13}	0	none
C_{14}	0	1/kPa	K_{14}	0	1/kPa

Draft: Work in Progress

Draft: Work in Progress

Chapter 12

MathSpecAppendices

12.1 Vector Identities

$$\frac{\partial \mathbf{a}^T\mathbf{a}}{\partial x} = 2\mathbf{a}^T\frac{\partial \mathbf{a}}{\partial x} \tag{12.1}$$

$$\frac{\partial a}{\partial x} = \frac{\partial}{\partial x}\left(\mathbf{a}^T\mathbf{a}\right)^{1/2} = \frac{\mathbf{a}^T}{a}\frac{\partial \mathbf{a}}{\partial x} \tag{12.2}$$

$$\frac{\partial a^{-1}}{\partial x} = \frac{\partial}{\partial x}\left(\mathbf{a}^T\mathbf{a}\right)^{-1/2} = -\frac{\mathbf{a}^T}{a^3}\frac{\partial \mathbf{a}}{\partial x} \tag{12.3}$$

$$\frac{\partial}{\partial x}\left(\frac{\mathbf{a}}{a^3}\right) = \frac{1}{a^3}\frac{\partial \mathbf{a}}{\partial x} - 3\frac{\mathbf{a}\mathbf{a}^T}{a^5}\frac{\partial \mathbf{a}}{\partial x} \tag{12.4}$$

Draft: Work in Progress

Draft: Work in Progress

Bibliography

[1] Vallado, D. A., *Fundamentals of Astrodynamics and Applications, 2nd Ed.*, Microcosm Press, El Segundo, CA, 2001.

[2] P. Kenneth Seidelmann, E., *Explanatory Supplement to the Astronomical Almanac*, Univesity Science Books, Mill Valley, CA, 1992.

[3] P. K. Seidelmann, T. F., "Why New Time Scales?" *Astronomy and Astrophysics*, Vol. 265, 1992, pp. 833 – 838.

[4] 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.

[5] Long, A. and *et. al.*, J. C., *Goddard Trajectory Determination System Mathematical Theory, Revision 1*, Computer Sciences Corporation, Lanham-Seabrook, MD, 1989.

[6] Kizner, W., "A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories," *Jet Propulsion Laboratory External Publication*, , No. 674, 1959.

[7] Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods, Applications*, Springer, Berlin, Germany, 2005.

[8] Szebehely, V., *Theory of Orbits*, Academic Press Inc., New York, NewYork, 1967.

[9] Pines, S., "A Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, No. 11, 1973, pp. 1508 – 1511.

[10] 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.

Index

A1Gregorian, 41
A1ModJulian, 41
Altitude, 42
AOP, 42
Apoapsis, 42
Attitude Parameterization
    Axis/Angle to DCM, 66
    DCM to Axis/Angle, 65
    DCM to Quaternions, 65
    Quaternions to DCM, 65
AZI, 43

Barycenter, 56
BdotR, 43
BdotT, 43
BetaAngle, 44
BVectorAngle, 44
BVectorMag, 44

C3Energy, 44
Cartesian state
    to SphericalRADEC, 40
    definition, 33, 35
    from Equinotial, 37
    from Keplerian, 36
    from SphericalAZFPA, 39
    from SphericalRADEC , 40
    to Equinotial, 38
    to Keplerian, 33
    to SphericalAZFPA, 38
Celestial Ephemeris Pole, 18
Coordinate systems
    body equator, 21
    body fixed, 24
    body inertial, 25
    Equatorial, 23
    general transformations, 15
    mean J2000 ecliptic, 22
    mean of date equator, 23
    mean of epoch equator, 23, 24
    object referenced, 26
    transformation algorithm, 16, 18, 32
    true of date equator, 23, 24
    true of epoch equator, 22

DEC, 44
DECV, 45

EA, 45
ECC, 45
Energy, 45
Equinoctial elements
    definition, 38
    from cartesian, 38
    to cartesian, 37

FK5 Reduction
    1980 nutation theory, 20
    1996 nutation theory, 20
    Earth fixed system, 17
    long period motion, 19
    nutation, 19
    overview, 17
    polar motion, 19, 21
    precession, 19
    secular motion, 19
    sidereal time, 21
FPA, 45

HA, 46
HMAG, 45
HX, 46
HY, 46
HZ, 46

INC, 46

J2000 Epoch, 12

Keplerian elements, 34
    definition, 33, 35
    from cartesian, 33
    from modified Keplerian, 40
    special cases, 35
    to cartesian, 36
    to modified Keplerian, 40

Latitude, 46
Libration Points, 54
    Definition, 55, 56
    Location, 55
Longitude, 47
LST, 47
Luna
    pole locations, 31
    prime meridian locations, 31

MA, 48

Draft: Work in Progress

Math in script, 75
MM, 48
Modified Keplerian elements
    definition, 33, 41
    from Keplerian, 40
    to Keplerian, 40

Numerical integrators
    Adams-Bashforth-Moulton, 72
    Bulirsch-Stoer, 72
    Coefficients, 72
    PD45 coefficients, 73
    PD56 coefficients, 73
    RKF56 coefficients, 74
    Runge-Kutta, 71
    Stopping condition algorithm, 72

Orbit state representations, 33
OrbitPeriod, 48

Penumbra, 51
PercentShadow, 48
Planets
    pole locations, 30
    prime meridian locations, 30
Psuedo-Rotating Coordinate Systems, 17

RA, 49
RAAN, 50
RadApo, 50
RadPer, 50
RAV, 49
RMAG, 50

SemilatusRectum, 50
Shadow
    Penumbra, 52
    Umbra, 52
SMA, 50
Spacecraft properties
    C3Energy, 44
    Energy, 45
    mean motion, 48
    OrbitPeriod, 48
    RadApo, 50
    RadPer, 50
    VelApoapsis, 52
    VelPeriapsis, 53
Spherical elements, 36
    definition, 39
SphericalAZFPA
    from Cartesian state , 38
    to Cartesian state , 39
SphericalRADEC
    from Cartesian State , 40
Stopping condition algorithm, 72

TA, 51
TAIModJulian, 51
Time Formats, 12
Time formats
    Gregorian date, 13
    Julian date, 13
    modified Julian date, 13
Time systems, 11
    A.1, 11
    TAI, 11
    TCB, 12
    TDB, 12
    TT, 12
    UT1, 11
    UTC, 11
TTGregorian, 51
TTModJulian, 51

Umbra, 51
UTCModJulian, 52

VelApoapsis, 52
VelPeriapsis, 53
VMAG, 53

Draft: Work in Progress

- 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).[4] 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 3.5 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 3.1 and 3.2. From inspection of Fig. 3.5 we see that

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3^T(90^\circ + \alpha_o)\mathbf{R}_1^T(90^\circ - \delta_o) \tag{3.52}$$

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.

$$\dot{\mathbf{R}}_{J_{2k},i} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.53}$$

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 $\dot{\mathbf{R}}_{J_{2k},i}$.

$$\mathbf{R}_{J_{2k},i} = \mathbf{R}_3(\theta_1)^T\mathbf{R}_1(\theta_2)^T \tag{3.54}$$

$$\dot{\mathbf{R}}_{J_{2k},i} = \mathbf{R}_3(\theta_1)^T\dot{\mathbf{R}}_1^T(\theta_2) + \dot{\mathbf{R}}_3^T(\theta_1)\mathbf{R}_1^T(\theta_2) \tag{3.55}$$

where

$$\dot{\mathbf{R}}_1(\theta_2) = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & -\dot{\theta}_2\sin\theta_2 & \dot{\theta}_2\cos\theta_2 \\ 0.0 & -\dot{\theta}_2\cos\theta_2 & -\dot{\theta}_2\sin\theta_2 \end{pmatrix} \tag{3.56}$$

and

$$\dot{\mathbf{R}}_3(\theta_1) = \begin{pmatrix} -\dot{\theta}_1\sin\theta_1 & \dot{\theta}_1\cos\theta_1 & 0.0 \\ -\dot{\theta}_1\cos\theta_1 & -\dot{\theta}_1\sin\theta_1 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.57}$$

Finally, for the Earth, the Equator axis system a true of date equator system and is calculated using the algorithm described in Sec. 3.4.5.

3.4.2 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°:

$$\mathbf{R} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.91748206 & -0.397777156 \\ 0.0 & 0.39777716 & 0.9174820621 \end{pmatrix} \tag{3.58}$$

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

$$\dot{\mathbf{R}} = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{3.59}$$

3.4.3 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

$$\mathbf{R}_{Ii} = \mathbf{N}^T(t_o)\mathbf{P}^T(t_o) \tag{3.60}$$

where t_o is the epoch defined in the coordinate system description provided by the user in the epoch field. Hence,