

Tudat mathematical model definition

Dominic Dirkx

January 24, 2022

Preface

This document describes the mathematical models that are implemented and used in the Tudat software, developed at TU Delft. The code is hosted at <https://github.com/tudat-team/tudat-bundle>, with the software documentation at <https://tudat.tudelft.nl> (for Tudat, partly outdated) and <https://tudat-space.readthedocs.io/en/latest/> (for TudatPy, up to date).

This is meant to be a 'living document', which means that we will add details and models as time goes on, and it is (at least at present) *not* meant to be a comprehensive overview of Tudat's underlying mathematical models. In case there are requests for (more) details on specific aspects of the Tudat software, please contact the Tudat team. Since this is a living document, some chapters and sections will be incomplete, in the sense that they do not fully describe all relevant models. We have chosen to include new material at the subsection level when we feel that this material is sufficiently presented to be of use to the reader. For those chapters that are still at a very low stage of development, we have added the note "**NOTE: This chapter is still very incomplete**" at the beginning of the chapter.

In addition to serving as a general mathematical model definition document, this document will be used as lecture notes for the AE4868 (Numerical Astrodynamics) and AE4866 (Propagation and Optimization in Astrodynamics) courses in the M.Sc. curriculum of TU Delft. Consequently, in addition to presenting the mathematical models, the manner in which these models will be used is also presented in a number of cases.

The current version of the document derives from numerous sources, for which the reader is referred to the bibliography. We have not included bibliographic references for all equations, statements, *etc.* in this document.

Many thanks go to Erwin Mooij for comments and discussion on the contents and notation of this document.

Contents

1	Basic Definitions and Notation	1
1.1	Reference Frames - Definitions and Conventions	1
1.2	States - Notation	2
1.2.1	Translational States	2
1.2.2	Accelerations	3
1.2.3	Rotations	3
2	Environment Models	5
2.1	Rotation Models	5
2.1.1	Simple Rotation Model	5
2.2	Gravity Fields	6
2.2.1	Point-mass gravity field	7
2.2.2	Spherical Harmonic Gravity Field	7
2.3	Inertia Tensor	10
2.4	Gravity Field Variation Models	11
2.4.1	Tidal gravity field variations	11
2.4.2	Tabulated gravity field variations	12
2.5	Atmosphere Models	12
2.5.1	Exponential Atmosphere	12
2.6	Ephemeris Models	13
2.6.1	Kepler Ephemeris	13
2.6.2	Spice Ephemeris	14
2.6.3	Tabulated Ephemeris	14
2.7	Shape Models	15
2.7.1	Spherical Shape	15
2.7.2	Oblate Spheroidal Shape	15
3	System Models	17
3.1	Aerodynamic Coefficients	17
4	Acceleration Models	19
4.1	Gravitational Acceleration Models	19
4.1.1	Point-mass gravity	20
4.1.2	Spherical-harmonic gravity (body exerting acceleration)	20

4.1.3	Spherical-harmonic gravity (body undergoing acceleration)	21
4.1.4	Non-inertial origins	22
4.1.5	Example - Dynamics of Planetary Satellite System	24
4.2	Aerodynamic acceleration	25
4.2.1	Aerodynamic Guidance	25
4.3	Radiation pressure acceleration	26
4.3.1	Cannon-ball radiation pressure acceleration	26
4.3.2	Panelled radiation pressure acceleration	26
4.4	Thrust acceleration	27
4.5	Relativistic acceleration	27
4.5.1	Schwarzschild acceleration	28
4.5.2	Lense-Thirring acceleration	28
4.5.3	De Sitter acceleration	28
4.6	Empirical acceleration	28
5	Torque Models	31
5.1	Extended-body Gravitational Torque	31
5.2	Degree-two Gravitational Torque	31
6	Propagation Models	33
6.1	Translational Dynamics	33
6.1.1	Cowell	33
6.1.2	Encke	34
6.2	Rotational Dynamics	35
6.2.1	Angular Velocity Vector	35
6.3	Mass Dynamics	36
7	Variational Equations	37
7.1	Translational dynamics	38
7.2	Application to differential correction	39
8	Numerical Integrators	41
8.1	Integration Errors	42
8.1.1	Truncation Error	42
8.1.2	Error Sources - Comparison	45
8.2	Multi-stage Methods	45
8.2.1	Fixed-step Method - Runge-Kutta 4	46
8.2.2	Variable step-size methods	46
8.3	Multi-step Methods	47
8.3.1	Step-size and order control	48
8.4	Extrapolation Methods	48
9	Numerical Interpolators	49
9.1	Piecewise constant interpolator	49
9.2	Linear interpolator	50
9.3	Lagrange interpolator	50

CONTENTS

v

A Summary of Notation

53

Chapter 1

Basic Definitions and Notation

1.1 Reference Frames - Definitions and Conventions

Here, we introduce a consistent set of notations and definitions concerning reference frames, which we will use throughout this document. Details and definitions of specific reference frames can be found in numerous sources (*e.g.* Vallado and McClain, 2001)

We define a reference frame by two properties: its *origin*, and its *orientation*. With this distinction, the term 'inertial' reference takes on a double meaning. We must distinguish between:

- An inertial origin: the origin of the frame is fixed in space (or moving with a constant velocity). A typical inertial origin is the solar system barycenter (SSB).
- An inertial orientation: the frame is not *rotating*, in the sense that its axes are fixed w.r.t. distant reference points (*e.g.* quasars),

Both definitions are in practice impossible to fully realize. Nevertheless, in practice we use a slight abuse of terminology, by using the term 'inertial' (instead of the more precise pseudo-inertial) to represent our approximations to these concepts.

In Tudat, all states which are retrieved directly from the environment (*e.g.* from a body object) are expressed in the same frame. This frame is defined by the user at the start of a simulation, and must remain fixed thereafter. We refer to this frame as the *global frame*, defined by the global origin and global orientation. The origin may be defined as inertial (I , represents the SSB), or it may be fixed to the center of mass of any of the bodies in the simulation. The

global orientation must be inertial, and current options are limited to Spice-defined J2000 and ECLIPJ2000 frames. In the following, the selected inertial orientation is denoted by I .

For frames with non-inertial orientations, we use the same index for the body itself, as for its body-fixed frame. That is, we use a superscript (C) to refer to the frame whose orientation is fixed to a body C , and use C for a frame with origin at body C .

1.2 States - Notation

In this section, we define our basic notation for translational states, accelerations, rotations, *etc.* Throughout this document, we write vectors in bold, with a lower-case symbol \mathbf{a} , and matrices/tensors in bold, with an upper-case symbol \mathbf{A} .

1.2.1 Translational States

We use \mathbf{r} and \mathbf{v} for Cartesian position and velocity, and $\mathbf{x} = [\mathbf{r}; \mathbf{v}]^T$ for the Cartesian translational state vector. We represent Cartesian positions of body A w.r.t. body B (*e.g.* with the origin at body B) as \mathbf{r}_{BA} (and similarly for velocity and full state). Whenever deemed useful, we will separate subscripts by a comma, so \mathbf{r}_{BA} and $\mathbf{r}_{B,A}$ are simply different notations for the same quantity.

We use a superscript to represent the orientation of the reference frame in which a vector is expressed, so that $\mathbf{r}_{BA}^{(C)}$ represents the position of A , w.r.t. the center of mass of B , expressed in frame C .

To simplify the notation when possible, we omit the superscript when the orientation of the frame is inertial:

$$\mathbf{r}_{BA} = \mathbf{r}_{BA}^{(I)} \quad (1.1)$$

Also, we omit one of the subscripts when the origin w.r.t. which the vector is expressed is the inertial origin, so:

$$\mathbf{r}_A^{(C)} = \mathbf{r}_{IA}^{(C)} \quad (1.2)$$

$$\mathbf{r}_A = \mathbf{r}_{IA}^{(I)} \quad (1.3)$$

and therefore:

$$\mathbf{r}_{BA} = \mathbf{r}_A - \mathbf{r}_B \quad (1.4)$$

Whenever possible, we will use the common notation of Eqs. (1.1) and (1.3).

There are numerous possible representations for a translational state. We use \mathbf{x}_{BA} to denote the Cartesian state of body A w.r.t. body B . To represent a Keplerian state, we will use the symbol χ_{BA} to (*e.g.* \mathbf{x}_{BA} in Keplerian elements w.r.t. body B). For a general, non-Cartesian, representation of a translational state, we will use $\tilde{\mathbf{x}}_{BA}$. Such states need not be of size 6, and can consist of Modified Equinoctial elements, Kepler elements, *etc.*

1.2.2 Accelerations

We denote an acceleration as \mathbf{a} , where the acceleration exerted by body B on body A is denoted by \mathbf{a}_{BA} . The total acceleration acting on a body A is denoted as \mathbf{a}_A , so that:

$$\mathbf{a}_A = \sum_B \mathbf{a}_{BA} \quad (1.5)$$

Here, we have used the same convention as in Eq. (1.1), implicitly assuming that the acceleration is denoted in a frame with inertial acceleration, using the default notation:

$$\mathbf{a}_{BA} = \mathbf{a}_{BA}^{(I)} \quad (1.6)$$

Thus far, we have assumed that the accelerations are computed in a frame with an inertial *origin*. For most types of acceleration, changes in this origin do not influence their formulation. For some situations, the frame origin C will, however, result in a modification of the acceleration. This is discussed in more detail in Section 4.1.4, where the third-body gravitational perturbations are introduced. To explicitly denote such a situation, we use the notation $(\mathbf{a}_{BA})_C$ to denote the acceleration exerted by B on A , in a frame centered on C . If frame C is an inertial origin, we omit the subscript, so:

$$\mathbf{a}_{BA} = (\mathbf{a}_{BA})_I \quad (1.7)$$

Also, in cases where the acceleration \mathbf{a} does not depend on the frame origin C , we omit the subscript from $(\mathbf{a}_{BA})_C$, as in such cases we will have $(\mathbf{a}_{BA})_C = (\mathbf{a}_{BA})_I$, and we revert to the notation of Eq. (1.7).

1.2.3 Rotations

A rotation between two frames can be represented in many equivalent forms. A typical formulation is the directional cosine matrix (DCM; also termed rotation matrix here), which may be formulated in terms of Euler angles. Alternatively, formulations such as quaternions or modified Rodrigues parameters may be used. In this document, rotations shall be represented as rotation matrices, unless explicitly indicated otherwise.

Rotation matrices representing an elementary rotation of an angle θ about the x , y and z axes are denoted as $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\theta)$ and $\mathbf{R}_z(\theta)$, respectively. The rotation matrix from frame A to frame B is denoted as $\mathbf{R}^{(B/A)}$.

Transforming a translational state vector \mathbf{x} from a frame C to a frame D is done by:

$$\mathbf{x}^{(D)} = \begin{pmatrix} \mathbf{R}^{(D/C)} & 0_{3 \times 3} \\ \dot{\mathbf{R}}^{(D/C)} & \mathbf{R}^{(D/C)} \end{pmatrix} \mathbf{x}^{(C)} \quad (1.8)$$

where $\dot{\mathbf{R}}^{(D/C)}$ is the time derivative of the rotation matrix $\mathbf{R}^{(D/C)}$.

Due to their orthonormality, rotation matrices obey the following properties:

$$\mathbf{R}^{(B/I)} = \left(\mathbf{R}^{(I/B)} \right)^T \quad (1.9)$$

$$= \left(\mathbf{R}^{(I/B)} \right)^{-1} \quad (1.10)$$

$$\dot{\mathbf{R}}^{(B/I)} = \left(\dot{\mathbf{R}}^{(I/B)} \right)^T \quad (1.11)$$

The angular velocity vector of frame A w.r.t. frame B , expressed in frame C , is denoted as $\boldsymbol{\omega}_{BA}^{(C)}$. Typically, this vector is expressed in the frame A , so that we often use the default notation:

$$\boldsymbol{\omega}_{BA} = \boldsymbol{\omega}_{BA}^{(A)} \quad (1.12)$$

Note that this default notation, in which the frame orientation superscript is omitted, is *different* from that for translational state and acceleration vectors in Eqs. (1.1) and (1.6).

Moreover, when the frame w.r.t. which the angular velocity is expressed is the inertial one, we write:

$$\boldsymbol{\omega}_A^{(C)} = \boldsymbol{\omega}_{IA}^{(C)} \quad (1.13)$$

$$\boldsymbol{\omega}_A = \boldsymbol{\omega}_{IA}^{(A)} \quad (1.14)$$

From the rotation matrix and its time-derivative, the angular velocity vector can be computed:

$$\left[\boldsymbol{\omega}_{A,B}^{(A)} \right]_{\times} = \dot{\mathbf{R}}^{(A/B)} \mathbf{R}^{(B/A)} \quad (1.15)$$

where the $[*]_{\times}$ notation is defined by the cross product:

$$[\mathbf{a}]_{\times} \mathbf{b} = \mathbf{a} \times \mathbf{b} \quad (1.16)$$

Transformation of the angular velocity vector is achieved by:

$$\boldsymbol{\omega}_{BA}^{(C)} = -\boldsymbol{\omega}_{AB}^{(C)} \quad (1.17)$$

$$\boldsymbol{\omega}_{BA}^{(D)} = \mathbf{R}^{(D/C)} \boldsymbol{\omega}_{BA}^{(C)} \quad (1.18)$$

similarly to position and velocity vectors.

Chapter 2

Environment Models

2.1 Rotation Models

The rotation model of a body B is responsible for computing the rotation matrix $\mathbf{R}^{(B/I)}$ from the inertial frame I to a frame fixed to body B , and its time derivatives $\dot{\mathbf{R}}^{(B/I)}$. From these matrices, the angular velocity vector $\boldsymbol{\omega}_B$ can also be computed as per Eq. (1.15).

2.1.1 Simple Rotation Model

This basic rotation model assumes that the body has a rotation axis that is in a frame with inertial orientation, and that the body has a constant rotation rate about this fixed axis. The direction of the pole in the inertial frame is defined by the right ascension α and declination δ of the pole (defined in the global frame). The first intermediate rotation to frame B'' is then defined by:

$$\mathbf{R}^{(B''/I)} = \mathbf{R}_x(\pi/2 - \delta)\mathbf{R}_z(\alpha + \pi/2) \quad (2.1)$$

The direction of the body's prime meridian at a reference epoch t_0 is parameterized by the angle W_0 , leading to the second intermediate rotation:

$$\mathbf{R}^{(B'/I)} = \mathbf{R}_z(W_0)\mathbf{R}^{(B''/I)} \quad (2.2)$$

$$\mathbf{R}^{(B/I)}(t_0) = \mathbf{R}^{(B'/I)} \quad (2.3)$$

Now, given a rotation rate ω , we have:

$$\mathbf{R}^{(B/I)}(t) = \mathbf{R}_z(\omega(t - t_0))\mathbf{R}^{(B'/I)} \quad (2.4)$$

where the only time dependency lies in the first \mathbf{R}_z matrix.

The time derivative of this rotation matrix is easily obtained from:

$$\dot{\mathbf{R}}^{(B/I)}(t) = \omega\dot{\mathbf{R}}_z(\omega(t - t_0))\mathbf{R}^{(B'/I)} \quad (2.5)$$

Model inputs

- Constant pole right-ascension and declination α and δ .
- Prime meridian location at initial epoch W_0
- Initial epoch t_0
- Angular rotation rate ω of the body

2.2 Gravity Fields

Gravity fields formulations are defined by the manner in which they express the gravitational potential U . Derived computations, such as potential gradients (accelerations), and potential Hessians (acceleration partials) are discussed in Section 4.1 on accelerations and Section ?? on partial derivatives of accelerations, respectively. We denote the gravitational potential of body B , evaluated at the location of body A as $U_B(\mathbf{r}_{BA})$.

The potential of a point mass is given by the following well-known equation:

$$U_B(\mathbf{r}_{BA}) = \frac{\mu_B}{\|\mathbf{r}_{BA}\|} \quad (2.6)$$

The total potential U of many point masses B_i can be computed from:

$$U(\mathbf{r}_A) = \sum_i U_{B_i}(\mathbf{r}_{B_i A}) \quad (2.7)$$

These equations can be used to define the gravitational potential of a full body, by integrating it over the body's volume V_B , see Fig. (2.1):

$$U_B(\mathbf{r}_{BA}) = G \int_B \frac{\rho(\mathbf{s}) dV_B}{\|\mathbf{r}_A - \mathbf{s}\|} \quad (2.8)$$

where \mathbf{s} denotes the point inside the body at the volume element dV_B , and $\rho(\mathbf{s})$ defines the body's internal mass distribution. Note that, since only scalar quantities are required in the above formulation, it is invariant under frame origin/orientation changes of the vector $\mathbf{r}_A - \mathbf{s}$.

For some applications, in particular those involving gravitationally interacting natural bodies, it will be useful to decompose the potential into a point mass contribution $U_{\bar{B}}$, and an extended-body contribution $U_{\hat{B}}$, so that:

$$U_B = \underbrace{U_{\bar{B}}}_{\text{point mass}} + \underbrace{U_{\hat{B}}}_{\text{extended body}} \quad (2.9)$$

where the $U_{\bar{B}}$ can always be written as Eq. (2.6), and any *deviations* from a point-mass gravity field are defined by $U_{\hat{B}}$ (see, for instance, Section 2.2.2 for the spherical harmonic representation). These deviations are sometimes termed the 'extended-body' potential.

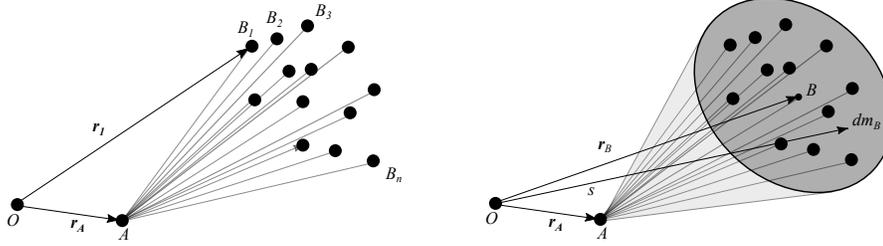


Figure 2.1: Graphical representation of potential due to many point masses (left), as per Eq. (2.7), and due to a single extended body (right), as per Eq. (2.8).

2.2.1 Point-mass gravity field

For a point-mass, the potential $U_B(\mathbf{r}_{BA})$ is expressed directly by Eq. (2.6). We note that this expression is also valid for bodies for which the internal mass distribution $\rho(\mathbf{s})$ is spherically symmetric, as long as the potential is evaluated *outside* the body B .

For this gravity field model we have, in the decomposition of Eq. (2.9):

$$U_{\hat{B}} = U_B \quad (2.10)$$

$$U_{\hat{B}} = 0 \quad (2.11)$$

Model inputs

- Gravitational parameter of body μ_B

2.2.2 Spherical Harmonic Gravity Field

There are numerous ways to represent deviations from spherical symmetry (see Section 2.2.1) when expressing a body's gravitational potential. A typical formulation in astrodynamics is the spherical harmonic expansion, for which:

$$U_B(\mathbf{r}_{BA}) = \sum_{l=0}^{\infty} \sum_{m=0}^l U_{B,lm} \quad (2.12)$$

$$U_{B,lm} = \mu_B \left(\frac{R^l}{r^{l+1}} \right) \bar{P}_{lm}(\sin \phi) (\bar{C}_{lm} \cos m\theta + \bar{S}_{lm} \sin m\theta) \quad (2.13)$$

Here, \bar{C}_{lm} and \bar{S}_{lm} represent fully-normalized¹ spherical harmonic coefficients (at degree l and order m) of body B . We typically do not add an explicit index for the body to which \bar{C}_{lm} and \bar{S}_{lm} belong, this being implicit due to the B subscript in the lefthand side of Eq. (2.13). In specific cases where we

¹A number of normalizations are used in different disciplines for spherical harmonics, we will only use the 'fully normalized' formulation, which we will refer to as 'normalized' from hereon.

want to explicitly specify that coefficients belong to a body B , we write $\bar{C}_{B,lm}$ and $\bar{S}_{B,lm}$. The symbol R denotes the gravity field's reference radius (typically the equatorial radius), r represents the distance between the centers of mass of bodies A and B , and ϕ and θ represent the latitude and longitude of body A , expressed in a frame with its origin at the center of mass of body B , and fixed to body B . That is, we have:

$$\mathbf{r}_{BA}^{(B)} = \mathbf{R}^{(B/I)} \mathbf{r}_{BA} \quad (2.14)$$

$$\theta = \theta \left(\mathbf{r}_{BA}^{(B)} \right) \quad (2.15)$$

$$\phi = \phi \left(\mathbf{r}_{BA}^{(B)} \right) \quad (2.16)$$

so that ϕ and θ depend on the inertial positions of bodies A and B , as well as the orientation of body B . We do not add the sub/superscripts to the angles to keep the formulation of the spherical harmonics tractable.

In Eq. (2.13), the coefficients \bar{C}_{lm} and \bar{S}_{lm} represent the deviations from spherical symmetry. The degree/order zero ($l = m = 0$) represents the point-mass contribution of body B . Therefore, we must have $U_{B,00}$ equal to Eq. (2.6), leading to the fact that $C_{00} = 1$. Additionally, since terms $U_{B,l0}$ (*e.g.*, the order m is zero) are invariant under changes in \bar{S}_{lm} , see Eq. (2.13), $\bar{S}_{l0} = 0$ is undefined.

The series expansion in Eq. (2.13) converges only for $r > R$, and is only a valid representation for the potential outside of the body's circumscribing sphere. That is, outside the the smallest sphere which circumscribes all mass of the body, with the sphere having its center at the origin (typically the center of mass). This sphere is termed the Brillouin sphere. For typical spacecraft situations this condition is automatically satisfied, but it may not be for close orbits of highly irregular bodies, or ascent/descent trajectories at low altitudes.

The expansion in Eq. (2.12) contains an infinite number of terms. For practical applications, the series is truncated at a maximum degree l_{\max} , so that the first summation (over l) is done up to $l = l_{\max}$. For some cases, it is also desirable to define a maximum order m_{\max} , leading to the general:

$$U_B(\mathbf{r}_{BA}) = \sum_{l=0}^{l_{\max}} \sum_{m=0}^{\min(l, m_{\max})} U_{B,lm} \quad (2.17)$$

Alternatively, one may define a set of degree/order combinations $\{(l, m)\}$ over which to sum the potential (*e.g.* only $(l, m)=(2,0)$, $(2,2)$ and $(3,0)$).

In contrast to the normalized formulation given in Eq. (2.13), the gravitational potential may also be represented in unnormalized coefficients and Legendre polynomials, for which:

$$U_{B,lm} = \mu_B \left(\frac{R^l}{r^{l+1}} \right) P_{lm}(\sin \phi) (C_{lm} \cos m\theta + S_{lm} \sin m\theta) \quad (2.18)$$

where the transformation from unnormalized to normalized coefficients follows from:

$$\bar{N}_{lm} = \sqrt{\frac{(2 - \delta_{0m})(2l + 1)(l - m)!}{(l + m)!}} \quad (2.19)$$

$$\bar{P}_{lm} = \bar{N}_{lm} P_{lm} \quad (2.20)$$

$$(C, S)_{l,m} = \bar{N}_{lm} (\bar{C}, \bar{S})_{l,m} \quad (2.21)$$

The degree-two unnormalized coefficients can be used directly as input to compute a body's inertia tensor (see Section 2.3). The degree-one unnormalized coefficients are directly related to a body's center-of-mass (denoted $\bar{\mathbf{r}}$) as:

$$\bar{z} = C_{10}R, \quad \bar{x} = C_{11}R, \quad \bar{y} = S_{11}R \quad (2.22)$$

Typically, these coefficients are zero, implicitly assuming that the center of mass coincides with the origin of the frame in which the gravity field coefficients are defined.

The zonal (order 0) unnormalized coefficients of degree l are sometimes distributed as coefficients J_l , for which:

$$J_l = -C_{l,0} = \bar{N}_{lm} \bar{C}_{l,0} \quad (2.23)$$

We stress that the physical information contained in the normalized and unnormalized coefficients is identical, and they merely represent different conventions for disseminating this information.

In the above, we have written the spherical harmonic coefficients as constant. In practice, they may be time-dependent quantities. Model for time-variations of these coefficients are discussed in Section 2.4.2.

Finally, we note that for the extended body contribution to the gravity field, as per the decomposition of Eq. (2.9), we then have:

$$U_{\hat{B}}(\mathbf{r}_{BA}) = \sum_{l=1}^{\infty} \sum_{m=0}^l U_{B,lm} \quad (2.24)$$

Model inputs

- Gravitational parameter of body μ_B
- Spherical harmonic coefficients C_{lm} and S_{lm} **or** normalized spherical harmonic coefficients \bar{C}_{lm} and \bar{S}_{lm}
- Maximum degree l_{max} and maximum order m_{max} **or** a set of degrees/orders (l, m)
- Gravity field reference radius R

2.3 Inertia Tensor

The inertia tensor \mathbf{I} of a body B is defined by its interior mass distribution from:

$$\mathbf{I}_B^{(B)} = \int_B \rho(\mathbf{s}^{(B)}) \cdot \left(\|\mathbf{s}\|^2 \mathbf{1}_{3 \times 3} - \mathbf{s}^{(B)} \cdot \left(\mathbf{s}^{(B)} \right)^T \right) dV_B \quad (2.25)$$

where the quantities in the integral are the same as in Eq. (2.8), with the exception that the frame orientation of the vector \mathbf{s} is now explicitly denoted. We use the following default notation for the inertia tensor:

$$\mathbf{I}_B = \mathbf{I}_B^{(B)} \quad (2.26)$$

That is, in contrast to most other quantities (positions, accelerations, *etc.*), we omit the superscript for the frame in which the tensor is defined for the case where it is written in the frame fixed to the body itself. If needed, the inertia tensor is transformed to a different frame by:

$$\mathbf{I}_B^{(A)} = \mathbf{R}^{(A/B)} \mathbf{I}_B \mathbf{R}^{(B/A)} \quad (2.27)$$

which represents the inertia tensor of body B , expressed in frame A .

The separate components of the tensor are typically referred to explicitly as:

$$\mathbf{I}_B = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \quad (2.28)$$

where the diagonal elements are termed the moments of inertia, and the off-diagonal terms the products of inertia. As a matter of convention, the products of inertia are in some cases preceded by a minus sign in Eq. (2.28). We do *not* follow this convention here.

There exists a simple relation between the inertia tensor, and the degree 2 unnormalized spherical harmonic gravity field coefficients (see Section 2.2.2):

$$\mathbf{I}_B = MR^2 \left(\begin{pmatrix} \frac{C_{20}}{3} - 2C_{22} & -2S_{22} & -C_{21} \\ -2S_{22} & \frac{C_{20}}{3} + 2C_{22} & -S_{21} \\ -C_{21} & -S_{21} & -\frac{2C_{20}}{3} \end{pmatrix} + \bar{I} \mathbf{1}_{3 \times 3} \right) \quad (2.29)$$

where all quantities (mass, reference radius, spherical harmonic coefficients) are properties of the gravity field of body B . The only exception is \bar{I} , which represents the normalized mean moment of inertia, which is equal to the average value of the diagonal of \mathbf{I} . Therefore, when a body is endowed with a spherical harmonic gravity field, one only needs to define \bar{I} to uniquely set its inertia tensor. Note that, as with the gravity field coefficients, the entries of the inertia tensor may be time-dependent (Section 2.4.2).

Model inputs

- Tensor components I_{xx} , I_{yy} , I_{zz} , I_{xy} , I_{xz} and I_{yz} , **or** \bar{I} and degree-two spherical harmonic gravity field.

2.4 Gravity Field Variation Models

The formulations for the potential in Section 2.2 was written as $U = U(\mathbf{r})$. This makes the implicit assumption that the interior mass distribution $\rho(\mathbf{s})$ of a body, as used in Eq. (2.8) is time-invariant. For detailed models, however, it may be necessary to include time-variations in the gravity field, so that $U = U(\mathbf{r}, t)$. These changes can be due to tidal deformation, mass transport, mass loss, *etc.*. Such variations in the gravity field are typically (but not necessarily) formulated and disseminated as variations in the spherical harmonic coefficients \bar{C}_{lm} and \bar{S}_{lm} , as used in Eq. (2.13).

2.4.1 Tidal gravity field variations

This model for variations in the gravity field of a solid body B due to any number of tide-raising point masses (perturbers) can be written as follows for coefficients at a single degree and order l and m :

$$\Delta\bar{C}_{lm} - i\Delta\bar{S}_{lm} = \frac{1}{2l+1} \sum_j k_{lm}^{(j)} \frac{\mu_j}{\mu} \left(\frac{R}{r_j}\right)^{l+1} \bar{P}_{lm}(\sin\phi_j) (\cos m\theta_j - i \sin m\theta_j) \quad (2.30)$$

where quantities without subscripts represent properties of the body B for which the gravity field's time variations are under consideration. The summation runs over all tide-raising bodies j , where μ_j is the gravitational parameter of the tide-raising body j , and r_j , ϕ_j and θ_j represent the spherical position of body j in a frame fixed to body B .

The Love number $k_{lm}^{(j)}$ is the tidal Love number at degree l , and order m , at the forcing frequency of body j . In case only a single deforming body is considered (or the frequency-dependence of the Love numbers is omitted), the symbol k_{lm} can be used instead². In addition, if the Love number is assumed independent of the order m , we only consider the Love number k_l .

The Love number can be either a real or complex number. In the case that the Love numbers are complex, the imaginary part represents the rate of tidal dissipation through:

$$\Im(k_l) = -\frac{k_l}{Q_l} \quad (2.31)$$

where Q represents the quality factor of the tidal dissipation at degree l (omitting the m subscript and (j) superscript). Eq. (2.30) is only valid for perturbers j in a near-spherical and near-equatorial orbit around body A . In essence, adding an imaginary part to k_l introduces a constant longitudinal lag angle $\Delta\theta_j$

²Note that in this formulation, a single forcing frequency due to a body j is assumed. Higher-order harmonics representing tidal deformation at frequencies due to the combined forcing of multiple bodies, or the situation with a tide-raising body in a non-spherical orbit, are ignored

into the location of the tidal bulge.

Model inputs

- A list of tide-raising bodies j
- A list of considered degrees l and orders m
- A set of Love numbers $k_{lm}^{(j)}$ for gravity field variations at degree l and order m , due to body j **or** k_{lm} , without dependency on forcing body j **or** k_l without dependency on degree l and forcing body j

2.4.2 Tabulated gravity field variations

This general gravity field variation model takes two time series $\Delta C_{lm}(t_i)$, $\Delta S_{lm}(t_i)$, defined at a discrete set of times t_i , for any degrees/orders l, m , and uses these as input to an interpolator to compute continuous time series $\Delta C_{lm}(t)$, $\Delta S_{lm}(t)$.

Model inputs

- Gravity field variations time-series $\Delta C_{lm}(t_i)$ and $\Delta S_{lm}(t_i)$, at times t_i
- Interpolation settings

2.5 Atmosphere Models

In dynamical modelling, atmosphere models are primarily used to generate values for the density ρ at a given point in space. We distinguish between standard atmospheres, which only depend on the altitude over the body (not on latitude, longitude or time) and reference atmospheres such as NRLMSISE-00 and EMCD, which can have a much more complicated position- and time-dependent formulation.

In addition to the density, atmosphere models produce values for the local temperature T , pressure p , as well as the atmospheric composition. In case the ideal gas law is assumed to hold, we have the following relation:

$$p = \rho RT \quad (2.32)$$

Properties of the atmospheric composition that may be required are the mean molar mass \mathcal{M} , the specific gas constant R and the ratio of specific heats γ .

2.5.1 Exponential Atmosphere

An exponential atmosphere is a highly simplified model, in which the atmospheric properties are assumed to only depend on the altitude h . It uses the following model for the density:

$$\rho = \rho_0 e^{-h/H} \quad (2.33)$$

where ρ_0 is the density at $h = 0$, and H is a scale height. This scale height is related to the atmospheric properties as:

$$H = \frac{RT}{\mathcal{M}g} \quad (2.34)$$

Since the gravitational acceleration g is approximately constant over the altitude ranges where exponential atmospheres are typically applied, the model assumptions are roughly met by assuming constant temperature T and composition (through the molar mass \mathcal{M}) of the atmosphere. Atmospheric pressure is typically computed from the ideal gas law when using this model.

The exponential atmosphere model can be used in a 'piecewise' manner, where different values of H and ρ_0 are used over different altitude ranges. Such an option is not included in Tudat, where these values are constant over the full altitude range.

Model inputs

- Scale height H
- Density at zero altitude ρ_0

2.6 Ephemeris Models

Ephemerides provide the state \mathbf{x}_A of a celestial body A as a function of time. In many cases, *a priori* ephemerides, for instance disseminated in the form of Spice kernels, are used to retrieve the states of natural bodies. In addition to such pre-defined ephemerides, the state of a (celestial) body may be represented by a number of other models.

Each ephemeris of a body A is defined w.r.t. some central point B , which can be but need not be an inertial origin. Ephemerides are generally defined in a frame with an inertial orientation. In Tudat, the ephemeris orientation *must* be set to the global frame orientation.

2.6.1 Kepler Ephemeris

This ephemeris model assumes that the body A is in an unperturbed Kepler orbit w.r.t. body B . Given an initial Keplerian state χ_0 , an initial time t_0 , and an effective gravitational parameter $\bar{\mu}_B$, this ephemeris model propagates the orbit to a time t by:

$$M = M_0 + n(t - t_0) \quad (2.35)$$

with M and n the mean anomaly, and the constant mean motion, respectively. Since all other Keplerian elements remain constant for this unperturbed model, the Cartesian state \mathbf{x}_{BA} is obtained by a simple Keplerian-to-Cartesian coordinate conversion.

Finally, note that the effective gravitational parameter $\bar{\mu}_B$ is only equal to the actual gravitational parameter μ_B of body B , if body A is assumed to be massless (so if $\mu_A/\mu_B \rightarrow 0$). If body A is massive, we have:

$$\bar{\mu}_B = \mu_B + \mu_A \quad (2.36)$$

as deduced from Eq. (4.14).

Model inputs

- Initial Kepler elements χ_0
- Initial time t_0
- Central body effective gravitational parameter $\bar{\mu}_B$

2.6.2 Spice Ephemeris

The Spice toolbox underlies much of the solr system ephemerides used in Tundat. Spice uses so-called 'kernels' to disseminate ephemerides, which are large (binary) files containing the required data. When using Spice to extract an ephemeris, the position $\mathbf{r}_{BA}(t)$ is extracted from these kernel files.

Model inputs

- Name (or other identifier) of body A and B

2.6.3 Tabulated Ephemeris

This general ephemeris model takes a time series for the state $\mathbf{r}_{BA}(t_i)$, defined at a discrete set of times t_i , and uses these as input to an interpolator to compute continuous time series $\mathbf{r}_{BA}(t)$. Such a model is often used to create a continuous ephemeris from the result of a numerical integration.

Note that a tabulated may be created from any other type of ephemeris, by evaluating this ephemeris at a discrete set of times, and then creating a tabulated ephemeris from the resulting data. A typical example is the Spice ephemeris. Evaluating the Spice ephemeris is a relatively slow computational process, and it may be more efficient to create a tabulated ephemeris from a Spice ephemeris before performing the numerical propagation.

Model inputs

- State time-series $\mathbf{r}_{BA}(t_i)$, at times t_i
- Interpolation settings

2.7 Shape Models

In dynamical modelling, a shape model is primarily used to convert a relative state w.r.t. a body's center of mass, to quantities such as the local altitude, longitude and latitude w.r.t. that body. In applications on exo-atmospheric flight, the influence of the shape model is relatively minor. For atmospheric, in particular low-altitude, dynamics the influence of the body shape becomes more influential. We stress here that shape and gravity field models are in principle decoupled entities. Although there is typically a strong correlation between overall shape and topography of a body and its gravity fields, this link is neither direct nor unique: for a given body shape, an infinite set of gravity fields is possible, depending on the mass distribution inside the body. Therefore, changing a body's shape model does *not* automatically change its gravity field.

2.7.1 Spherical Shape

The spherical shape model has only a single free parameter: its radius R . Due to its symmetry, the computation of the local altitude h is independent of the frame orientation in which an orbiting body's position is expressed and, for any frame C , we have:

$$h = \|\mathbf{r}_{BA}^C\| - R \quad (2.37)$$

Model inputs

- Body radius R

2.7.2 Oblate Spheroidal Shape

This model represents a sphere that is symmetrically flattened at its poles, to obtain an oblate spheroid. The shape is longitudinally, but not latitudinally, symmetric. The altitude h , body-fixed longitude θ , and geodetic latitude ϕ' are related to the body-fixed position $\mathbf{r}_{BA}^{(B)}$ by the following implicit equation:

$$\mathbf{r}_{BA}^{(B)} = \begin{pmatrix} (N + h) \cos \phi' \cos \theta \\ (N + h) \cos \phi' \sin \theta \\ ((1 - f^2)N + h) \cos \phi' \end{pmatrix} \quad (2.38)$$

$$N = \frac{R_e}{\sqrt{1 - f(2 - f) \sin^2 \phi'}} \quad (2.39)$$

which must be solved implicitly for the geodetic position. In the above equations, where the flattening f of the body is defined from:

$$f = \frac{R_e - R_p}{R_e} \quad (2.40)$$

with R_p the body's polar radius.

Model inputs

- Two out of the following three: Body polar radius R_p , body equatorial radius R_e , body flattening f

Chapter 3

System Models

NOTE: This chapter is still very incomplete

3.1 Aerodynamic Coefficients

Aerodynamic force coefficients are a normalized representation of the aerodynamic force \mathbf{f}_a . Two typical representations can be found in literature, the force coefficients in the aerodynamic frame (C_D, C_S, C_L), and the force coefficients in the body-fixed frame, of the vehicle A (C_x, C_y, C_z):

$$S_{ref} \begin{pmatrix} C_D \\ C_S \\ C_L \end{pmatrix} = -\frac{\mathbf{f}_a^{(\text{Aero})}}{\frac{1}{2}\rho v_{\text{air}}^2} \quad (3.1)$$

$$S_{ref} \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} = \frac{\mathbf{f}_a^{(A)}}{\frac{1}{2}\rho v_{\text{air}}^2} \quad (3.2)$$

where S_{ref} denotes the reference area associated with the coefficients, ρ denotes the local freestream atmospheric density and v_{air} denotes the vehicle's airspeed. The difference in sign between the two cases is a matter of convention (in the definition of the two sets of coefficients). The airspeed $v_{\text{air}} (= ||\mathbf{v}_{\text{air}}||)$ defines the relative speed of the vehicle w.r.t. the freestream flow \mathbf{v}_{∞} of the atmosphere, so:

$$\mathbf{v}_{\text{air}} = \mathbf{v}_{BA} - \mathbf{v}_{\infty} \quad (3.3)$$

A typical case is one where the atmosphere is assumed to corotate with the central body. In this case, we have:

$$\mathbf{v}_{\infty} = \dot{\mathbf{R}}^{(I/B)} \mathbf{r}_{BA}^{(B)} \quad (3.4)$$

which gives the freestream velocity \mathbf{v}_{∞} in the frame with inertial orientation. Typically, any deviations of freestream velocity from this corotating model are defined by a wind model, providing wind velocities \mathbf{v}_w .

Depending on the user's definition, aerodynamic coefficients may be a function of any number of properties of the vehicle and/or environment. Typical parameters are freestream Mach number M_∞ and vehicle angle-of-attack α and sideslip angle β , as well as control surface deflections δ_i (for a control surface i).

Chapter 4

Acceleration Models

4.1 Gravitational Acceleration Models

The gravitational acceleration of a body B , acting on a point mass body A , as expressed in a frame with inertial origin can be written as:

$$\mathbf{a}_{BA} = \nabla U_B(\mathbf{r}_{BA}) \quad (4.1)$$

Here, U_B represents the gravitational potential of body B , and $\nabla = \partial/\partial\mathbf{r}$ (*e.g.* the gradient w.r.t. inertial position). Eq. (4.1) indicates that the acceleration acting on A is equal to the gradient of the potential of body B , evaluated at the location of body A .

We can decompose the total acceleration between two arbitrary masses as follows, splitting the contributions of point masses and extended bodies, as in Eq. (2.9):

$$\mathbf{a}_{BA} = \mathbf{a}_{B\bar{A}} + \mathbf{a}_{\hat{B}\bar{A}} + \mathbf{a}_{B\hat{A}} + \mathbf{a}_{\hat{B}\hat{A}} \quad (4.2)$$

this decomposition can be very useful for natural body dynamics, but is typically *not* used for spacecraft dynamics. The terms on the right-hand side of this equation are shown graphically in Fig. 4.1, and represent the accelerations:

- $\mathbf{a}_{B\bar{A}}$: Exerted by a point mass B on a point mass A (see Section 4.1.1).
- $\mathbf{a}_{\hat{B}\bar{A}}$: Exerted by extended body¹ B on a point mass A (see Section 4.1.2).
- $\mathbf{a}_{B\hat{A}}$: Exerted by point mass B on an extended body A (see Section 4.1.3). This contribution can be neglected if $\mu_A/\mu_B \rightarrow 0$, such as for spacecraft dynamics.
- $\mathbf{a}_{\hat{B}\hat{A}}$: Exerted by extended body B on an extended body. This contribution can be neglected if $\mu_A/\mu_B \rightarrow 0$, such as for spacecraft dynamics. For most

¹Note that in this context the term 'extended body' refers to the gravity field *without* the point mass contribution.

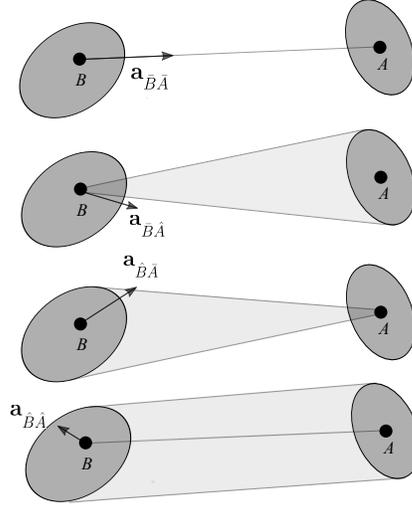


Figure 4.1: Graphical representation of the terms in Eq. (4.2), indicating point mass and extended body attractions. Note that, with the exception of $\mathbf{a}_{B\hat{A}}$, the directions of the accelerations depend on the specific extended gravity fields of the bodies.

cases, it can also be neglected for μ_A/μ_B not approaching zero, and we therefore do not treat it further here.

All acceleration components in Eq. (4.2) are expressed w.r.t. an inertial origin. We discuss the influence of a non-inertial frame origin on the formulation of the gravitational accelerations, leading to the formulation of the third-body acceleration, in Section 4.1.4.

4.1.1 Point-mass gravity

For a point-mass, the potential is expressed simply by Eq. (2.6), which results from Eq. (4.1) in the well-known inverse-square law for the acceleration on a point mass, exerted by a point mass:

$$\mathbf{a}_{B\hat{A}} = -\frac{\mu_B}{\|\mathbf{r}_{BA}\|^2} \hat{\mathbf{r}}_{BA} \quad (4.3)$$

4.1.2 Spherical-harmonic gravity (body exerting acceleration)

We use Eq. (4.1) to compute the gravitational acceleration, using the spherical harmonic potential in Eq. (2.12). Since the terms $U_{B,lm}$ are a function of the body-fixed position of body A, see Eqs. (2.14)-(2.16), the potential gradient is more easily calculated in a frame fixed to body B (denoted $\nabla^{(B)} = \partial/\partial\mathbf{r}^{(B)}$),

and we have for the acceleration in a frame with inertial orientation:

$$\mathbf{a}_{BA} = \nabla \left(\sum_{l=0}^{\infty} \sum_{m=0}^l U_{B,lm}(\mathbf{r}_{BA}) \right) \quad (4.4)$$

$$= \mathbf{R}^{(I/B)} \nabla^{(B)} \left(\sum_{l=0}^{\infty} \sum_{m=0}^l U_{B,lm}(\mathbf{r}_{BA}) \right) \quad (4.5)$$

$$= \mathbf{R}^{(I/B)} \left(\sum_{l=0}^{l_{\max}} \sum_{m=0}^l \left(\nabla^{(B)} U_{B,lm}(\mathbf{r}_{BA}) \right) \right) \quad (4.6)$$

The complete formulation for $\nabla^{(B)} U_{B,lm}$ can be found in *e.g.* Montenbruck and Gill (2000), and shall not be written out here.

For completeness, we note that Eq. (4.6) includes both the point mass ($l = 0$) and extended body contributions of body B . In the notation of Eq. (4.2), we have:

$$\mathbf{a}_{\hat{B}\hat{A}} = \mathbf{R}^{(I/B)} \left(\sum_{l=1}^{l_{\max}} \sum_{m=0}^l \nabla^{(B)} U_{B,lm}(\mathbf{r}_{BA}) \right) \quad (4.7)$$

4.1.3 Spherical-harmonic gravity (body undergoing acceleration)

The previous section described the typical case where the body B , which is *exerting* the acceleration, possesses a spherical harmonic gravity field. In cases where the body A that is *undergoing* the acceleration is itself endowed with a spherical harmonic gravity field, this will directly influence body A 's own dynamics. This effect is described in this section. For the dynamics of spacecraft, which are typically assumed to be massless, this term can be neglected.

The acceleration under consideration can be understood as a direct result of Newton's third law. Consider a body A with a spherical harmonic gravity field, exerting an acceleration on a point mass B (swapping the roles of A and B compared to previous sections). Eq. (4.7) then provides a formulation for $\mathbf{a}_{\hat{A},\hat{B}}$. By Newton's third law, we have:

$$\mathbf{a}_{\hat{B}\hat{A}} = -\frac{\mu_B}{\mu_A} \mathbf{a}_{\hat{A}\hat{B}} \quad (4.8)$$

$$= -\frac{\mu_B}{\mu_A} \nabla U_{\hat{A}}(\mathbf{r}_{AB}) \quad (4.9)$$

which is exactly the effect which we require in this section.

The total gravitational acceleration between two extended bodies in Eq.

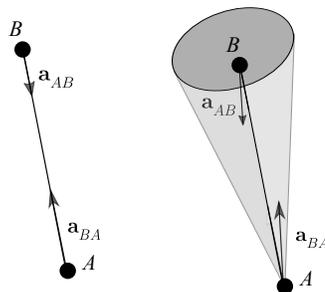


Figure 4.2: Graphical representation of reaction force between two point masses A and B , and a point mass A and extended body B .

(4.2), omitting the contribution of $\mathbf{a}_{\hat{B}\hat{A}}$, is then written as:

$$\mathbf{a}_{BA} = -\frac{\mu_B}{\|\mathbf{r}_{BA}\|^2} \hat{\mathbf{r}}_{BA} + \nabla U_{\hat{B}}(\mathbf{r}_{BA}) - \frac{\mu_B}{\mu_A} \nabla U_{\hat{A}}(\mathbf{r}_{AB}) \quad (4.10)$$

$$= \underbrace{-\frac{\mu_B}{\|\mathbf{r}_{BA}\|^2} \hat{\mathbf{r}}_{BA}}_{\mathbf{a}_{B\hat{A}}} + \underbrace{\mathbf{R}^{(I/B)} \nabla^{(B)} U_{\hat{B}}(\mathbf{r}_{BA})}_{\mathbf{a}_{\hat{B}\hat{A}}} - \underbrace{\frac{\mu_B}{\mu_A} \mathbf{R}^{(I/A)} \nabla^{(A)} U_{\hat{A}}(\mathbf{r}_{AB})}_{\mathbf{a}_{B\hat{A}}} \quad (4.11)$$

Consequently, the total acceleration on body A depends on the orientation and gravity field of body A *itself* through $\mathbf{R}^{(I/A)}$ and $U_{\hat{A}}$.

4.1.4 Non-inertial origins

The models in Sections 4.1.1-4.1.3 hold for propagation w.r.t. an inertial origin (*e.g.* the barycenter). In typical situations, propagations are not done w.r.t. the barycenter, but rather w.r.t. some central body. For instance an Earth-orbiting spacecraft is propagated w.r.t. the center of mass of the Earth, not w.r.t. the barycenter.

Here, we distinguish two different cases:

- i The computation of $(\mathbf{a}_{BA})_B$: acceleration due to body B , acting on body A , with the frame origin at body B 's center of mass: the acceleration due to the central body itself.
- ii The computation of $(\mathbf{a}_{BA})_C$: acceleration due to body B , acting on body A , with the frame origin at some other body C : the so-called third-body perturbation

For the first case, we have:

$$(\mathbf{a}_{BA})_B = \mathbf{a}_{BA} - \mathbf{a}_{AB} \quad (4.12)$$

This second term on the r.h.s. of expression takes into account the fact that the body A may also exert a gravitational acceleration on body B . This term

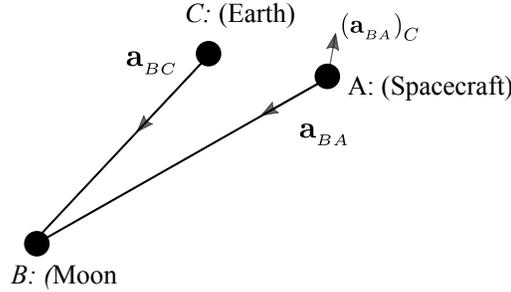


Figure 4.3: Graphical representation of third body perturbation $(\mathbf{a}_{BA})_C$, for point masses A , B and C

is negligible for the cases where $\mu_A/\mu_B \rightarrow 0$, for instance when body A is a spacecraft and body B is the Earth. In this case, we have:

$$(\mathbf{a}_{BA})_B \rightarrow \mathbf{a}_{BA} \quad (4.13)$$

However, it is important to take the second term of Eq. (4.12) into account for natural body dynamics, for instance when computing the Moon's dynamics w.r.t. the Earth. For example, for the case of bodies A and B both having a point-mass gravity field, we get the following by using Eqs. (4.3) and (4.12):

$$(\mathbf{a}_{BA})_B = -\frac{\mu_A + \mu_B}{\|\mathbf{r}_{BA}\|^2} \hat{\mathbf{r}}_{BA} \quad (4.14)$$

This expression also shows that, when computing Kepler elements of one massive body w.r.t. another massive body, the *sum* of their gravitational parameters should be used, not only the gravitational parameter of the central body.

For case (ii), the third-body perturbation, we have:

$$(\mathbf{a}_{BA})_C = \mathbf{a}_{BA} - \mathbf{a}_{BC} \quad (4.15)$$

which, for interacting point-masses, reduces to the following by applying Eq. (4.3):

$$(\mathbf{a}_{BA})_C = \mu_B \left(-\frac{\hat{\mathbf{r}}_{BA}}{\|\mathbf{r}_{BA}\|^2} + \frac{\hat{\mathbf{r}}_{BC}}{\|\mathbf{r}_{BC}\|^2} \right) \quad (4.16)$$

A graphical representation of the third body perturbation on a spacecraft (body A) exerted by the Moon (body B), with propagation origin at the Earth (body C) is given in Fig. 4.3

Although less familiar in typical applications, Eqs. (4.12) and (4.15) apply equally to point mass and extended-body accelerations. For example, it can be used directly to compute a 'third-body spherical-harmonic' acceleration. Such an acceleration may be relevant for spacecraft dynamics. Consider, for instance,

the influence of the Moon's spherical harmonic gravity field on a high-altitude Earth orbiter. For such a situation, we would have:

$$(\mathbf{a}_{BA})_C = \nabla U_B(\mathbf{r}_{BA}) - \nabla U_B(\mathbf{r}_{BC}) \quad (4.17)$$

$$(\mathbf{a}_{BA})_C = \nabla \left(\sum_{l=0}^{l_{max}} \sum_{m=0}^l U_{B,lm}(\mathbf{r}_{BA}) \right) - \nabla \left(\sum_{l=0}^{l_{max}} \sum_{m=0}^l U_{B,lm}(\mathbf{r}_{BC}) \right) \quad (4.18)$$

with A , B and C representing the spacecraft, the Moon and the Earth, respectively.²

A worked out case for the influence of the models discussed in this section on the dynamics of a system of massive bodies is given in Eq. (4.21) for the dynamics of the Galilean moons, for which the mutual spherical harmonic accelerations between the moons, and between Jupiter and the moons, become relevant.

4.1.5 Example - Dynamics of Planetary Satellite System

As a synthesis of the above sections, and to provide a more concrete examples of how all equations are combined in practice, we provide here the explicit formulation of the dynamics of a planetary satellite system (*e.g.* Galilean moons) w.r.t. a central body (*e.g.* Jupiter). We consider the dynamics of N extended bodies, denoted with $i = 1 \dots N$, w.r.t. a central body, which is denoted by index 0.

The total acceleration is then written as:

$$(\mathbf{a}_i)_0 = (\mathbf{a}_{0,i})_0 + \sum_{j=1}^{N(i \neq j)} (\mathbf{a}_{j,i})_0 \quad (4.19)$$

First, we compute $(\mathbf{a}_{0,i})_0$ from Eq. (4.12), using Eq. (4.10) for both $\mathbf{a}_{0,i}$ and $\mathbf{a}_{i,0}$, to get:

$$(\mathbf{a}_{0,i})_0 = -(\mu_0 + \mu_i) \overbrace{\left(\frac{\hat{\mathbf{r}}_{0i}}{\|\mathbf{r}_{0i}\|^2} - \frac{\nabla U_{\hat{0}}(\mathbf{r}_{0i})}{\mu_0} + \frac{\nabla U_{\hat{i}}(\mathbf{r}_{i0})}{\mu_i} \right)}^{\text{central body - satellite}} \quad (4.20)$$

For the third-body effects $(\mathbf{a}_{j,i})_0$, we use Eq. (4.15), again using Eq. (4.10) for the constituents $\mathbf{a}_{j,i}$ and $\mathbf{a}_{h,0}$:

$$(\mathbf{a}_{j,i})_0 = -\mu_j \left(\underbrace{\left(\frac{\hat{\mathbf{r}}_{ji}}{\|\mathbf{r}_{ji}\|^2} - \frac{\nabla U_{\hat{j}}(\mathbf{r}_{ji})}{\mu_j} + \frac{\nabla U_{\hat{i}}(\mathbf{r}_{ij})}{\mu_i} \right)}_{\text{third body - satellite}} - \underbrace{\left(\frac{\hat{\mathbf{r}}_{j0}}{\|\mathbf{r}_{j0}\|^2} - \frac{\nabla U_{\hat{j}}(\mathbf{r}_{j0})}{\mu_j} + \frac{\nabla U_{\hat{0}}(\mathbf{r}_{0j})}{\mu_0} \right)}_{\text{third body - central body}} \right) \quad (4.21)$$

²In this model, we have neglected the contribution of $\mathbf{a}_{\hat{B},\hat{C}}$, involving the possible extended body potential $U_{\hat{C}}$ of the central body C .

Note that this expression is independent of the specific formulation of the extended gravity fields of any of the bodies, or any specific rotation models of these bodies. The physical limitation of this expression is the omission of the terms $\mathbf{a}_{i,\hat{j}}$.

4.2 Aerodynamic acceleration

The aerodynamic acceleration can be computed from one of the following formulations:

$$\mathbf{a}_{BA} = -\frac{1}{m}\mathbf{R}^{(I/B)} \cdot \mathbf{R}^{(B/Aero)} \left(\frac{1}{2}\rho v_{\text{air}}^2 S_{ref} \begin{pmatrix} C_D \\ C_S \\ C_L \end{pmatrix} \right) \quad (4.22)$$

$$\mathbf{a}_{BA} = \frac{1}{m}\mathbf{R}^{(I/B)} \cdot \mathbf{R}^{(B/A)} \left(\frac{1}{2}\rho v_{\text{air}}^2 S_{ref} \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} \right) \quad (4.23)$$

depending on the aerodynamic force coefficient set that is used (see Section 3.1).

A special case for the formulation of the aerodynamic acceleration is the cannon-ball model. In this model, we assume the aerodynamic force to consist of only drag ($C_L = C_S = 0$), in which case we can dispense with the various frame transformations, and simply evaluate:

$$\mathbf{a}_{BA} = -\left(\frac{\rho v_{\text{air}}^2 C_D S_{ref}}{2m} \right) \hat{\mathbf{v}}_{\text{air}} \quad (4.24)$$

The great advantage of this formulation is that it forgoes the need to determine any of the rotation matrices required for Eqs. (4.22) and (4.23).

4.2.1 Aerodynamic Guidance

Without changing the orientation of the vehicle w.r.t. the oncoming flow, the bank angle σ may be modified to influence the rotation $\mathbf{R}^{(B/Aero)}$, so that:

$$\mathbf{R}^{(B/Aero)} = \mathbf{R}^{(B/V)}(\delta, \tau) \mathbf{R}^{(V/T)}(\gamma, \chi) \mathbf{R}^{(T/Aero)}(\sigma) \quad (4.25)$$

where the frames B , V , T and Aero represent the body-fixed frame of body B , and the vertical, trajectory and aerodynamic frames of body A w.r.t. body B , respectively. These frames, and their frame transformations, are discussed in more detail by Mooij (1994).

In addition, the orientation of the vehicle w.r.t. the oncoming flow, parameterized by the angle of attack α and the sideslip angle β , can be used to modulate the aerodynamic coefficients, since generally:

$$C_k = C_k(\alpha, \beta, \dots) \quad (4.26)$$

where $k = D, S, L$. Here, we stress that this dependency will in general exist, but is not taken into account by Tudat unless such a dependency is explicitly defined

(*e.g.* coefficients are defined as function of α and/or β). The ... in the above equation is given to indicate the dependency on numerous other quantities, which may or may not be directly influenced by the vehicle's state.

4.3 Radiation pressure acceleration

4.3.1 Cannon-ball radiation pressure acceleration

Under the assumptions of the cannon-ball model (radiation from source at infinite distance illuminating a perfectly spherically symmetric object), we have:

$$\mathbf{a}_{BA} = p_B(\mathbf{r}_A) \left(\frac{C_r S}{m} \right) \hat{\mathbf{r}}_{BA} \quad (4.27)$$

where $p_B(\mathbf{r}_A)$ is the radiation pressure due to body B (in units of force per area) at the location of body A . For typical applications, body B will be the Sun.

The radiation pressure can be computed from:

$$p = \left(\frac{P}{c} \right) \frac{1}{4\pi \|\mathbf{r}_{BA}\|^2} \quad (4.28)$$

$$\mathbf{a}_{BA} = \left(\frac{P}{4\pi c} \right) \left(\frac{C_r S_{ref}}{m} \right) \frac{\hat{\mathbf{r}}_{BA}}{\|\mathbf{r}_{BA}\|^2} \quad (4.29)$$

where P is the total power emitted by the Sun (in units energy per time), and c is the speed of light. Comparing this expression with Eq. (4.3), we can see that the formulation is identical to that of a point mass gravity, under the transformation $\mu_B \rightarrow -\left(\frac{P}{4\pi c}\right) \left(\frac{C_r S_{ref}}{m}\right)$

4.3.2 Panelled radiation pressure acceleration

This radiation pressure model is much more detailed than the simplified cannon-ball model. It uses a so-called 'macro-model', in which the exterior of the vehicle is divided into a number of panels. The radiation pressure force on each of these panels i is then computed from:

$$(\mathbf{f}_{BA})_i = \begin{cases} p_B(\mathbf{r}_A) \cos \theta_i A_i \left((1 - \epsilon_i) \hat{\mathbf{r}}_{AB} + 2 \left(\epsilon_i \cos \theta_i + \frac{\rho_i}{3} \right) \hat{\mathbf{n}}_i \right) & \cos \theta_i > 0 \\ 0 & \cos \theta_i \leq 0 \end{cases} \quad (4.30)$$

where ϵ_i , ρ_i and A_i are the emissivity, diffuse reflectivity and area of panel i , respectively, and θ_i is the angle between the vector $\hat{\mathbf{r}}_{AB}$ (from the body A to the source of the radiation B) and the panel outward surface normal vector $\hat{\mathbf{n}}_i$:

$$\hat{\mathbf{n}}_i \cdot \hat{\mathbf{r}}_{AB} = \cos \theta_i \quad (4.31)$$

The total radiation pressure acceleration is then computed from:

$$\mathbf{a}_{BA} = \sum_i (\mathbf{f}_{BA})_i \quad (4.32)$$

Although much more accurate than the cannon-ball model in Section 4.3.1, this panelled model still makes a number of simplifying assumptions:

- Radiation is either absorbed, specularly reflected, or diffusely reflected in a Lambertian pattern
- All absorbed radiation is instantaneously re-radiated in a Lambertian pattern
- There is no 'shadowing' of one panel by another panel
- Each photon interacts with the body only once (no multi-path effects)

4.4 Thrust acceleration

Tudat uses a single unified thrust model, which is applied in an identical way for high-thrust (launchers, spacecraft, ...), as well as low-thrust propulsion systems. Also, a model is provided which allows an 'impulsive' thrust to be incorporated into the model by spreading the impulse out over a short time interval (termed pseudo-impulsive thrust), allowing its direct inclusion into state propagation.

In a general formulation, thrust is defined by two separate aspects, the magnitude T of the thrust force, and the thrust direction vector $\hat{\mathbf{f}}_T$. The thrust acceleration is then computed from:

$$\mathbf{a}_{AA} = \frac{T \cdot \hat{\mathbf{f}}_T}{m_A} \quad (4.33)$$

where we have a double subscript A (body exerting/undergoing thrust), as (the body exerts a force 'on itself'.

Numerous options are available in Tudat to compute and set the thrust magnitude and direction. Among these are options for user-defined custom functions that compute these quantities, as a function of time, or other variables (*e.g.* altitude, Mach number, *etc.*).

4.5 Relativistic acceleration

The gravitational accelerations described in Section (4.1) are all based on a Newtonian model of gravity. A physically complete description of gravitational acceleration should be formulated in terms of general relativity, however. Since solar-system scale astrodynamics problems take place in the so-called 'weak-field slow-motion' regime, Newtonian gravity is a very good approximation for its relativistic counterpart. Therefore, relativistic effects are typically included as correction accelerations to the Newtonian acceleration models.

4.5.1 Schwarzschild acceleration

The Schwarzschild acceleration provides the relativistic correction to Eq. (4.3), the gravitational acceleration due to a point mass. The correction acceleration \mathbf{a}_{BA} on a body A due to a point mass B is given by:

$$\mathbf{a}_{BA} = \frac{\mu_B}{c^2 r^3} \left(\left(2(\beta + \gamma) \frac{\mu_B}{r} - \gamma(\mathbf{v} \cdot \mathbf{v}) \right) \mathbf{r} + 2(1 + \gamma)(\mathbf{r} \cdot \mathbf{v})\mathbf{v} \right) \quad (4.34)$$

where we have omitted the BA subscripts on r , \mathbf{r} and \mathbf{v} for the sake of clarity. The parameters γ and β are so-called Parametric Post-Newtonian (PPN) parameters, which represent possible deviations from general relativity. In general relativity, they both have a value of 1.

4.5.2 Lense-Thirring acceleration

The Lense-Thirring effect knows no counterpart in Newtonian gravity. It is a result of the fact that not only mass, but the motion of mass, causes space-time curvature, and an associated acceleration. The Lense-Thirring effect is a result of the space-time curvature, induced by a central body's angular momentum \mathbf{h} . This angular momentum of a body B (for instance of the Earth) induces the following acceleration on a body A :

$$\mathbf{a}_{BA} = (1 + \gamma) \frac{\mu_B}{c^2 r^3} \left(\frac{3}{r^2} (\mathbf{r} \times \mathbf{v}) (\mathbf{r} \cdot \mathbf{h}) + (\mathbf{v} \times \mathbf{h}_B) \right) \quad (4.35)$$

where we have again omitted the BA subscripts on r , \mathbf{r} and \mathbf{v} for the sake of clarity.

4.5.3 De Sitter acceleration

The de Sitter acceleration is a correction to the third-body point-mass acceleration $(\mathbf{a}_{BA})_C$, given by Eq. (4.16). It is relevant, for instance, when computing high-accuracy dynamics of a spacecraft A about the Earth (C), in which case the de Sitter effect due to the Sun (B) may need to be taken into account. It can be formulated as:

$$(\mathbf{a}_{BA})_C = -(1 + 2\gamma) \left(\mathbf{r}_{BC} \times \left(\frac{\mu_B \mathbf{r}_{BC}}{c^2 r_{BC}^3} \right) \right) \times \mathbf{v}_{CA} \quad (4.36)$$

4.6 Empirical acceleration

The empirical acceleration is a model that is often applied in spacecraft state estimation, and is used to 'absorb' mismodelling in the spacecraft's dynamical model. Any arbitrary parameterized formulation could be used for this acceleration, but we limit ourselves to the 'typical' implementation of the empirical acceleration.

$$\mathbf{a}_{BA} = R^{(I/B_{RSW})} (\mathbf{a}_1 \sin \theta + \mathbf{a}_2 \cos \theta + \mathbf{a}_3) \quad (4.37)$$

which defines three vectors \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{a}_3 , defining an acceleration in the *RSW* frame that is modulate by the sin of the true anomaly, the cosine of the true anomaly, and one constant acceleration.

Chapter 5

Torque Models

NOTE: This chapter is still very incomplete

5.1 Extended-body Gravitational Torque

For the general case where body A has an arbitrary mass distribution, we can write the gravitational torque exerted by a point mass B on A from:

$$\mathbf{m}_{BA}^{(A)} = -m_A \mathbf{r}_{BA}^{(A)} \times \left(\nabla^{(A)} U_A(\mathbf{r}_{AB}) \right) \quad (5.1)$$

where the potential of body A may be expanded to arbitrary degree and order. In the above formulation, the contribution of the degree zero terms in U_A (the point mass contributions to the torque) naturally cancel out.

5.2 Degree-two Gravitational Torque

For the case where we only consider the degree-two gravity field of body A , we can rewrite the gravitational torque as:

$$\mathbf{m}_{BA}^{(A)} = \frac{3\mu_B}{\|\mathbf{r}_{BA}\|^5} \mathbf{r}_{BA}^{(A)} \times \left(\mathbf{I}_A \mathbf{r}_{BA}^{(A)} \right) \quad (5.2)$$

which is written in terms of the inertia tensor. In this expression, we note that the contribution of \bar{I}_B to the torque is always zero, and the result is identical to the torque given by Eq. (5.1), with U_B truncated at degree two (and degree-one terms equal to zero).

Chapter 6

Propagation Models

NOTE: This chapter is still very incomplete

In this chapter, we will discuss the state derivative models for different types of states. We use the general notation \mathbf{y} for an arbitrary state vector, which may be any combination of any type of state for any number of bodies. In the following sections, we discuss the specific types of state \mathbf{y} and their propagation.

6.1 Translational Dynamics

Here, we treat the different formulations for the state derivative of the translational dynamics of a single body A , w.r.t. a body B . We denote the Cartesian formulation of the propagated state as \mathbf{x}_{BA} , while denoting the specific formulations (*e.g.* Cowell, Encke, *etc.*) for the dynamics as $\tilde{\mathbf{x}}_{BA}$. Considering the propagation of a single body, we have:

$$\mathbf{y} = \tilde{\mathbf{x}}_{BA} \tag{6.1}$$

for our generic state vector \mathbf{y} .

6.1.1 Cowell

The Cowell propagator is the most straightforward case, for which:

$$\tilde{\mathbf{x}}_{BA} = \begin{pmatrix} \mathbf{r}_{BA} \\ \mathbf{v}_{BA} \end{pmatrix} \tag{6.2}$$

$$= \mathbf{x}_{BA} \tag{6.3}$$

This leads to the simple formulation for the state derivative:

$$\dot{\tilde{\mathbf{x}}}_{BA} = \begin{pmatrix} \mathbf{v}_{BA} \\ (\mathbf{a}_A)_B \end{pmatrix} \tag{6.4}$$

6.1.2 Encke

In the Encke formulation, we use a reference Keplerian orbit w.r.t. the propagation origin B , and we propagate the deviations from this reference orbit. Typically, this reference orbit is defined from the initial conditions.

We denote the Cartesian position of the reference orbit as $\boldsymbol{\rho}_{BA}(t; \mu_B)$, where μ_B is the gravitational parameter of the central body w.r.t. which the reference Kepler orbit is defined. Crucially, this means that the origin B must be endowed with a gravitational parameter (which *e.g.* the SSB is not).

In the rest of this section, we will omit the BA subscripts for position, velocity and acceleration vectors, for the sake of readability and brevity, so that $\mathbf{r}_{BA} \rightarrow \mathbf{r}$, $\mathbf{v}_{BA} \rightarrow \mathbf{v}$, $\boldsymbol{\rho}_{BA} \rightarrow \boldsymbol{\rho}$. For the *total* accelerations, we use $(\mathbf{a}_A)_B \rightarrow \mathbf{a}$, and we finally we use $\mu \rightarrow \mu_B$.

The propagated Encke state is then defined by:

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\rho}(t; \mu) \\ \dot{\boldsymbol{\rho}}(t; \mu) \end{pmatrix} \quad (6.5)$$

$$= \begin{pmatrix} \Delta \mathbf{r} \\ \Delta \mathbf{v} \end{pmatrix} \quad (6.6)$$

so that, in this section, $\Delta \mathbf{r}$ and $\Delta \mathbf{v}$ denote the deviation of the true orbit from the Keplerian reference orbit $\boldsymbol{\rho}$.

The Encke state derivative requires the perturbing accelerations acting on body A , which is defined from:

$$(\mathbf{a})_{\text{pert.}} = \mathbf{a} - (\mathbf{a})_{\text{cent.}} \quad (6.7)$$

where $(\mathbf{a})_{\text{cent.}}$ denotes *only* the point-mass acceleration of body B acting on body A (with B as the propagation origin), and $(\mathbf{a})_{\text{pert.}}$ denotes the remainder of the total acceleration. This central acceleration is computed directly from Eq. (4.14), which reduces to Eq. (4.3) for the case where we assume body A to be massless.

The dynamics of $\boldsymbol{\rho}$ and \mathbf{r} obey:

$$\ddot{\boldsymbol{\rho}} = -\mu \frac{\hat{\boldsymbol{\rho}}}{\|\boldsymbol{\rho}\|^2} \quad (6.8)$$

$$\ddot{\mathbf{r}} = -\bar{\mu} \frac{\hat{\mathbf{r}}}{\|\mathbf{r}\|^2} + (\mathbf{a})_{\text{pert.}} \quad (6.9)$$

where $\bar{\mu} = \mu_A + \mu_B$. The 'Encke acceleration' $\tilde{\mathbf{a}}$ is then computed from:

$$q = \frac{1}{2} \left(\frac{\|\mathbf{r}\|^2}{\|\boldsymbol{\rho}\|^2} - 1 \right) \quad (6.10)$$

$$\mathcal{F}(q) = \frac{2q}{1+2q} \left(1 + \frac{1}{1+2q+\sqrt{1+2q}} \right) \quad (6.11)$$

$$\tilde{\mathbf{a}} = \frac{\mu}{\rho^3} (\mathbf{r}\mathcal{F}(q) - \Delta \mathbf{r}) + (\mathbf{a})_{\text{pert.}} \quad (6.12)$$

which leads to the following state derivative (now reintroducing all sub/superscripts):

$$\dot{\mathbf{x}}_{BA} = \begin{pmatrix} \dot{\rho}_{BA} \\ (\tilde{\mathbf{a}}_A)_B \end{pmatrix} \quad (6.13)$$

6.2 Rotational Dynamics

The rotational state of a single body is defined by the full state vector:

$$\mathbf{y} = \begin{pmatrix} \mathbf{b}_B \\ \boldsymbol{\omega}_B \end{pmatrix} \quad (6.14)$$

where \mathbf{b}_B is a representation of the body's orientation w.r.t. inertial space, and $\boldsymbol{\omega}_B$ is the body's angular velocity vector w.r.t. inertial space, expressed in B 's body-fixed frame, as per Eq. (1.14). We note that there exist other representations, that do not explicitly use the angular velocity vector, but we will not cover those here.

6.2.1 Angular Velocity Vector

Our representations for the rotational state all incorporate the angular velocity vector $\boldsymbol{\omega}_B$ into the state vector, requiring a formulation for $\dot{\boldsymbol{\omega}}_B$. We define the angular momentum vector \mathbf{h}_B as:

$$\mathbf{h}_B = \mathbf{I}_B \boldsymbol{\omega}_B \quad (6.15)$$

where we have implicitly used \mathbf{h}_B for the angular momentum vector in the frame B (fully: $\mathbf{h}_B^{(B)}$). By Euler's equations, we have :

$$\frac{d\mathbf{h}_B}{dt} + \boldsymbol{\omega}_B \times \mathbf{h}_B = \mathbf{m}_B \quad (6.16)$$

where \mathbf{m}_B denotes the total torque acting on body B , expressed in frame B (as a simplified notation for $\mathbf{m}_B^{(B)}$). Note that for bodies with a variable mass, in which is flowing into or out of the body, this will affect the total torque \mathbf{b} acting on the body.

The above leads to the following differential equation governing the evolution of $\boldsymbol{\omega}_B$:

$$\dot{\boldsymbol{\omega}}_B = (\mathbf{I}_B)^{-1} \left(\mathbf{m}_B - \boldsymbol{\omega}_B \times (\mathbf{I}_B \boldsymbol{\omega}_B) - \dot{\mathbf{I}}_B \boldsymbol{\omega}_B \right) \quad (6.17)$$

In many cases, we have an (almost) constant internal mass distribution of our body. In such cases, we can neglect $\dot{\mathbf{I}}_B$, leading to:

$$\dot{\boldsymbol{\omega}}_B = (\mathbf{I}_B)^{-1} (\mathbf{m}_B - \boldsymbol{\omega}_B \times (\mathbf{I}_B \boldsymbol{\omega}_B)) \quad (6.18)$$

For cases where $\dot{\mathbf{I}}_B$ is very small, but non-zero, the above may still be a valid approximation, but it can often still be important to update \mathbf{I}_B for every evaluation of Eq. (6.18).

6.3 Mass Dynamics

The dynamics of the mass is computed in a straightforward manner, for which we only use a single formulation. For the mass state of body B , we have:

$$\mathbf{y} = m_B \quad (6.19)$$

The state derivative simply becomes:

$$\dot{\mathbf{y}} = \dot{m}_B \quad (6.20)$$

where the total mass rate \dot{m}_B of the body may be composed of multiple contributions i :

$$\dot{m}_B = \sum_i \dot{m}_{B,i} \quad (6.21)$$

In the case where we are computing mass flow of a (rocket) motor that is exerting a thrust force \mathbf{f}_T , we have:

$$\dot{m}_{B,i} = \sum_i \frac{|\mathbf{f}_T|}{I_{sp}g_0} \quad (6.22)$$

with I_{sp} and g_0 the specific impulse of the motor, and a reference acceleration ($=9.80665 \text{ m/s}^2$), which are related to the propellant expulsion speed u (w.r.t. the body B) as:

$$u = I_{sp}g_0 \quad (6.23)$$

Chapter 7

Variational Equations

We now consider the general case where we have a differential equation governing the dynamics of a state vector \mathbf{y} as a function of time t of the form:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t; \mathbf{p}) \quad (7.1)$$

$$\mathbf{y}(t_0) = \mathbf{y}_0 \quad (7.2)$$

where the vector \mathbf{p} is a vector of parameters that influence the dynamics, and t_0 is the initial time (with \mathbf{y}_0 the associated initial condition).

For various applications, it is useful to find a formulation for the influence of a change in \mathbf{y}_0 and/or \mathbf{p} on the resulting state history $\mathbf{y}(t)$. A typical approach to this is to make a linear approximation, so that we can write:

$$\Delta\mathbf{y}(t) \approx \frac{\partial\mathbf{y}(t)}{\partial\mathbf{y}_0} \Delta\mathbf{y}_0 + \frac{\partial\mathbf{y}(t)}{\partial\mathbf{p}} \Delta\mathbf{p} \quad (7.3)$$

where $\Delta\mathbf{y}_0$ and $\Delta\mathbf{p}$ represent the change in initial state and parameter vector, and $\Delta\mathbf{y}$ represents the resulting change in state history. The partial derivatives are typically abbreviated as:

$$\Phi(t, t_0) = \frac{\partial\mathbf{y}(t)}{\partial\mathbf{y}_0} \quad (7.4)$$

$$\mathbf{S}(t) = \frac{\partial\mathbf{y}(t)}{\partial\mathbf{p}} \quad (7.5)$$

with Φ and \mathbf{S} termed the state transition and sensitivity matrices. Before discussing further details of these matrices, we note that a generalization of Eq (7.3), with second- and higher-order derivatives (termed state transition tensors) such as $\frac{\partial^2\mathbf{y}(t)}{\partial\mathbf{y}_0^2}$, can be set up. However, this is beyond the scope of this document.

The state transition and sensitivity matrices can be obtained by numerical integration of their governing differential equations, which are termed the

variational equations. They are defined by:

$$\frac{d\Phi(t, t_0)}{dt} = \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} \quad (7.6)$$

$$= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right) \Phi(t, t_0) \quad (7.7)$$

$$\frac{d\mathbf{S}(t)}{dt} = \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{p}} + \frac{\partial \dot{\mathbf{y}}}{\partial \mathbf{p}} \quad (7.8)$$

$$= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}} \right) \mathbf{S}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \quad (7.9)$$

$$(7.10)$$

where \mathbf{y} , $\dot{\mathbf{y}}$ and \mathbf{f} are all evaluated at time t . These equations are typically, but not necessarily, numerically integrated concurrently with the state \mathbf{y} , with initial conditions:

$$\frac{d\Phi(t_0, t_0)}{dt} = \mathbf{1}_{n_s \times n_s} \quad (7.11)$$

$$\mathbf{S}(t_0) = \mathbf{0}_{n_s \times n_p} \quad (7.12)$$

$$(7.13)$$

where n_s and n_p denote the size of the state vector \mathbf{y} and the parameter vector \mathbf{p} , respectively. The state transition matrix has a number of useful properties:

$$\Phi(t_2, t_0) = \Phi(t_2, t_1) \Phi(t_1, t_0) \quad (7.14)$$

$$(\Phi(t_1, t_0))^{-1} = \Phi(t_0, t_1) \quad (7.15)$$

7.1 Translational dynamics

Many typical applications of the variational equations involve the situation where the vector \mathbf{y} represents the translational state of one or more bodies. For instance, for the case of a single body's translational dynamics, we have:

$$\mathbf{y} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} \quad (7.16)$$

$$\mathbf{f} = \begin{pmatrix} \mathbf{v} \\ \mathbf{a} \end{pmatrix} \quad (7.17)$$

and, therefore, we get the partial derivatives in the variational equations as:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} \\ \frac{\partial \mathbf{a}}{\partial \mathbf{r}} & \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \end{pmatrix} \quad (7.18)$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \begin{pmatrix} \mathbf{0}_{3 \times n_p} \\ \frac{\partial \mathbf{a}}{\partial \mathbf{p}} \end{pmatrix} \quad (7.19)$$

where we have omitted the sub- and supercripts indicating frames, and the time argument t , for the sake of clarity. The partial derivatives of the accelerations \mathbf{a} can in most cases be evaluated analytically. Their formulation is given by *e.g.*, (Moyer, 1971; Montenbruck and Gill, 2000; ?; Dirx et al., 2019) for numerous accelerations and parameters.

The same tools can also be used when \mathbf{y} represents the state of any number of bodies. For the case of translational dynamics of N bodies, we have:

$$\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \quad (7.20)$$

The resulting state transition matrix will then be built up as:

$$\Phi(t, t_0) = \begin{pmatrix} \Phi_{1,1}(t, t_0) & \Phi_{1,2}(t, t_0) & \dots & \Phi_{1,N}(t, t_0) \\ \Phi_{2,1}(t, t_0) & \Phi_{2,2}(t, t_0) & \dots & \Phi_{2,N}(t, t_0) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{N,1}(t, t_0) & \Phi_{N,2}(t, t_0) & \dots & \Phi_{N,N}(t, t_0) \end{pmatrix} \quad (7.21)$$

with

$$\Phi_{i,j}(t, t_0) = \frac{\partial \mathbf{x}_i(t)}{\partial \mathbf{x}_j(t_0)} \quad (7.22)$$

For the case where $i = j$, the associated partial derivatives of the \mathbf{f}_i are as in Eqs. (7.18) and (7.19), with \mathbf{r} , \mathbf{v} , \mathbf{a} and \mathbf{f} denoting properties of body i . For $i \neq j$, we have:

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{\partial \mathbf{a}_i}{\partial \mathbf{r}_j} & \frac{\partial \mathbf{a}_i}{\partial \mathbf{v}_j} \end{pmatrix} \quad (7.23)$$

7.2 Application to differential correction

An important application of the state transition and sensitivity matrices is in computing differential corrections to a trajectory. That is, we have a solution $\mathbf{y}(t)$ to some dynamical model, with initial state $\mathbf{y}(t_0)$. Now, assume we have a boundary condition $\bar{\mathbf{y}}(t_1)$ that we wish to meet at some time t_1 . In the case where the entries of the mismatch $\bar{\mathbf{y}}(t_1) - \mathbf{y}(t_1)$ are 'small', we can efficiently use the state transition matrix to compute the changes to the initial state $\mathbf{y}(t_0)$ that will result in meeting the required final state. That is, we want to find the correction to the initial state $\Delta \mathbf{y}(t_0)$, such that:

$$\mathbf{y}(t_1) = \bar{\mathbf{y}}(t_1) \quad (7.24)$$

Defining the state correction $\Delta \mathbf{y}(t_1)$ that we need to achieve at t_1 as:

$$\Delta \mathbf{y}(t_1) = \bar{\mathbf{y}}(t_1) - \mathbf{y}(t_1) \quad (7.25)$$

We have:

$$\Delta \mathbf{y}(t_0) = \Phi(t_0, t_1) \Delta \mathbf{y}(t_1) \quad (7.26)$$

Now, consider the more specific case of a spacecraft orbit, for which we have a numerical solution $\mathbf{x}(t) = [\mathbf{r}(t), \mathbf{v}(t)]^T$, and we wish to compute a transfer trajectory from $\bar{\mathbf{r}}(t_0)$ to $\bar{\mathbf{r}}(t_1)$: we have boundary conditions on the position at both the initial and final time. We leave the velocities at both boundaries free, as any mismatch w.r.t. some desired solution may be corrected by (small) maneuvers.

To find the required solution that meets both boundary conditions, we can start by setting the initial position equal to $\bar{\mathbf{r}}(t_0)$, and propagating the dynamics until t_1 , which will give us a final position $\mathbf{r}(t_1)$. To compute the required change in state at t_0 and t_1 , which we denote $\Delta \mathbf{x}(t_0)$ and $\Delta \mathbf{x}(t_1)$, respectively, we have the following two conditions:

$$\Delta \mathbf{r}(t_0) = \mathbf{0} \quad (7.27)$$

$$\Delta \mathbf{r}(t_1) = \bar{\mathbf{r}}(t_1) - \mathbf{r}(t_1) \quad (7.28)$$

By generating a solution for $\Phi(t, t_0)$, and using Eq. (7.26), these conditions result in a uniquely defined (assuming $\Phi(t_1, t_0)$ to be invertible) impulsive maneuver $\Delta \mathbf{v}(t_0)$ that will satisfy Eqs. (7.27) and (7.28).

$$\begin{pmatrix} \Delta \mathbf{r}(t_0) \\ \Delta \mathbf{v}(t_0) \end{pmatrix} = \Phi(t_0, t_1) \begin{pmatrix} \Delta \mathbf{r}(t_1) \\ \Delta \mathbf{v}(t_1) \end{pmatrix} \quad (7.29)$$

Since our initial position $\mathbf{r}(t_0)$ is assumed to meet our boundary condition, we set $\Delta \mathbf{r}(t_0) = \mathbf{0}$. Now, to compute $\Delta \mathbf{v}(t_0)$, we have an infinite number of remaining solutions, as we are free to choose our value of $\Delta \mathbf{v}(t_1)$. Typical solutions include simply setting $\Delta \mathbf{v}(t_1) = \mathbf{0}$ or generating some condition $\bar{\mathbf{v}}(t_1)$ that the state is to meet at the final time.

Chapter 8

Numerical Integrators

Numerical integrators are tools to obtain an approximate solution to an ordinary differential equation (ODE) of the form:

$$\frac{d\mathbf{y}}{ds} = \mathbf{f}(\mathbf{y}, s) \quad (8.1)$$

subject to the initial condition:

$$\mathbf{y}(s_0) = \mathbf{y}_0 \quad (8.2)$$

where s represents the independent variable (typically, but not necessarily, time), and \mathbf{y} represents the state that is to be propagated. We will limit ourselves to first-order ODEs.

In the remainder of this section, we will use a slight abuse of notation, and denote the independent variable s by t (and consequently $d\mathbf{y}/ds \rightarrow \dot{\mathbf{y}}$). Moreover, in this section we will write all formulations for the case that the state \mathbf{y} is a vector (unless indicated otherwise). However, all equations work equally well for the case where the state is a scalar y or a matrix \mathbf{Y} .

The output of the numerical integrators is an approximation $\bar{\mathbf{y}}$ to the true state \mathbf{y} , at a discrete set of epochs t_i

$$\bar{\mathbf{y}}(t_i) \approx \mathbf{y}(t_i), \quad i = 0 \dots N \quad (8.3)$$

where we will occasionally use the notation:

$$\bar{\mathbf{y}}_i = \bar{\mathbf{y}}(t_i) \quad (8.4)$$

The different integrators differ in their approach in generating these approximations. Here, we treat three classes of integrators, multi-stage (Section 8.2), multi-step (Section 8.3) and extrapolation methods (Section 8.4). Before discussing specifics of integration methods, we discuss error properties of the numerical solution $\bar{\mathbf{y}}_i$ in Section 8.1

8.1 Integration Errors

The error of the numerical solution produced by the integrator at time i can be written as:

$$\epsilon(t_i) = \bar{\mathbf{y}}(t_i) - \mathbf{y}(t_i) \quad (8.5)$$

Where, even though we do not *know* what the true solution $\mathbf{y}(t_i)$ is in typical cases ¹, we can postulate its existence, and from there derive useful error properties that will aid the comparison and analysis of different integrators.

There are two distinct aspects of integration error, which combine to form the *full* integration error. These two aspects are:

- Truncation error
- Numerical (or rounding) error

Here, we ignore any model error that exists, we are here interested in the *mathematical* error properties of the ODE defined by Eqs. (8.1) and (8.2), and do not consider the fact that these equations may not be *physically* exact. Issues related to model errors are discussed in Section ??.

The truncation error represents the inherent mathematical limitation of the particular integration method. To obtain an approximate solution to Eq. (8.1), the integrator must discard part of the behaviour of the true solution. The error that is incurred as a result of this process is termed the truncation error. For example, in the case of the Euler integrator, we rely on the underlying assumption that the state derivative $\dot{\mathbf{y}}$ remains constant over the interval $[t_i, t_{i+1})$. Truncation error is discussed in more detail in Section 8.1.1, and is a property of the specific integration method that is used, and its settings.

Numerical error is a source of the computational implementation of an integrator, as a direct result of the fact that computers use a limited number of digits (*e.g.* 16 digits for a `double` precision variable). Consequently, there is an inherent randomness in the 'final digit' (colloquially speaking) when performing a numerical computation. Numerical error is discussed in more detail in Section ??. Numerical error is not a property of a specific integration method, but is instead incurred because we implement the method on a digital computing system.

8.1.1 Truncation Error

To better understand the truncation error that is incurred during a single step of a numerical integrator, we expand the *true* solution as follows:

$$\mathbf{y}(t + \Delta t) = \mathbf{y}(t) + \int_t^{t+\Delta t} f(\mathbf{y}, \tilde{t}) d\tilde{t} \quad (8.6)$$

¹If we did, there would be no point in doing the integration

into a power series as follows:

$$\mathbf{y}(t + \Delta t) = \sum_{i=0}^{\infty} \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} (\mathbf{y}(t)) \quad (8.7)$$

Now, to analyze the emergence of the truncation error, consider the situation where we know the true solution at some epoch t , so that $\bar{\mathbf{y}}(t) = \mathbf{y}(t)$. In this manner we can analyze the error that accumulates when making the time step $t \rightarrow t + \Delta t$.

To illustrate the truncation error, we will start with an example, before moving to the more general situation. Consider the basic Euler integrator, for which we have:

$$\bar{\mathbf{y}}(t + \Delta t) = \mathbf{y}(t) + \Delta t \mathbf{f}(\mathbf{y}, t) \quad (8.8)$$

$$= \sum_{i=0}^1 \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} \mathbf{y}(t) \quad (8.9)$$

since \mathbf{f} is just the first time-derivative of \mathbf{y} . We then have, by combining Eqs. (8.5), (8.7) and (8.8)

$$\boldsymbol{\epsilon}(t + \Delta t) = - \sum_{i=2}^{\infty} \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} \mathbf{y}(t) \quad (8.10)$$

$$= O(\Delta t^2) \quad (8.11)$$

where notation $O(\Delta t^2)$ indicates that, for a given integration step, the error will increase approximately proportional to Δt^2 . Using a similar approach, it can be shown that the error incurred during a single step of the Runge-Kutta 4 method is $O(\Delta t^5)$.

We now make a small 'leap of faith', and assume that the numerical approximation of *any* integration method can be written as:

$$\bar{\mathbf{y}}(t + \Delta t) = \sum_{i=0}^q \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} (\mathbf{y}(t)) \quad (8.12)$$

$$\bar{\mathbf{y}}(t + \Delta t) = \sum_{i=0}^{p-1} \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} (\mathbf{y}(t)) \quad (8.13)$$

$$\bar{\boldsymbol{\epsilon}}(t + \Delta t) = \sum_{i=p}^{\infty} \frac{(\Delta t)^i}{i!} \frac{d^i}{dt^i} (\mathbf{y}(t)) \quad (8.14)$$

$$\bar{\boldsymbol{\epsilon}}(t + \Delta t) = \sum_{i=p}^{\infty} \mathbf{K}_i (\Delta t)^i \quad (8.15)$$

$$\approx \mathbf{K}_p (\Delta t)^p \quad (8.16)$$

That is, we assume that a given integrator correctly reproduces the first $q + 1$ terms in the Taylor series expansion Eq. (8.7) of the true solution, but omits the others.

In this general case, the error incurred during a single step is $O(\Delta t^q)$. This error incurred during a single step is termed the local truncation error (LTE). The error incurred over the full integration time is termed the global truncation error (GTE). For N time steps, the GTE is obtained by combining N LTEs of $O(\Delta t^2)$. If N increases (more steps), the time step Δt will get smaller, so that:

$$\text{GTE} \sim N \cdot \text{LTE} \quad (8.17)$$

$$N \sim \frac{1}{\Delta t} \quad (8.18)$$

$$\text{GTE} \sim \text{LTE}/\Delta t \quad (8.19)$$

As a result, the GTE of the Euler integrator is $O(\Delta t)$, making the Euler integrator a so-called first-order integrator (similarly, the RK4 method is a fourth-order method).

This $O(\Delta t^p)$ notation for errors is termed the 'big O' notation. Stating that an error is of $O(\Delta t^p)$ implies:

$$\epsilon(t_i) = \sum_{j=p}^{\infty} \mathbf{K}_j(t_i) \Delta t^j \quad (8.20)$$

This formulation for the truncation error is used in various aspects of numerical integration (error analysis, step-size control, *etc.*), and it is therefore important to discuss the *meaning* of this formulation in a bit more detail.

The first crucial point to realize is that, although we know the value of Δt , the values of \mathbf{K}_j are not known, we merely know that values \mathbf{K}_j exist such that Eq. (8.20) will hold. Moreover, we know that if we change only Δt , the values of \mathbf{K}_j will not change. If we change the integration method, state derivative function, or the time t at which our step starts, however, the coefficients \mathbf{K}_j will take on different values.

A typical, though not necessarily correct, interpretation of an $O(\Delta t^p)$ method, is that:

$$\epsilon(t_i) \approx \mathbf{K}_p(t_i) \Delta t^p \quad (8.21)$$

which implicitly assumes that $\mathbf{K}_p \Delta t^p$ is larger than all remaining terms (starting at $p + 1$) in Eq. (8.20) combined. By making this assumption, the order p of a method is used to analyze how strongly the error decreases due to a decrease in the step size Δt . Typically, it is assumed that (using a scalar notation for the state and error):

$$\frac{\epsilon(t_i; \Delta t/2)}{\epsilon(t_i; \Delta t)} \approx K_p \left(\frac{1}{2} \right)^p \quad (8.22)$$

For typical differential equations, this approximation will hold fairly well. It breaks down, however, for so-called *stiff* differential equations. A very loose

definition of a stiff differential equation is one where there are sudden large (but not discontinuous) changes in the true solution, which can lead to the need for a sudden radical decrease of step size to properly capture the behaviour of the system. We will not discuss such equations in detail here. However, when analyzing the behaviour of a numerical solution, such as when numerical results seem 'curious', the root cause may be that the equation becomes stiff.

8.1.2 Error Sources - Comparison

There is a strong distinction between these numerical and truncation errors, both from a fundamental and a practical point of view. In the purely abstract mathematical sense, numerical error does not exist. When writing down a numerical integration scheme, we cannot derive any properties of the numerical error. It is only when we implement it numerically, and have chosen a floating point representation of our numbers, that the numerical error emerges. It is essentially a practical problem, as we could in theory remove this error source by selecting a floating point representation with n significant digits, and $n \rightarrow \infty$. The truncation error, on the other hand, is an inherent property of the mathematical formulation of the numerical method. We can derive properties of the truncation error, independent of any knowledge of the numerical implementation.

As was discussed in Sections 8.1.1 and ??, the behaviour of the two error sources is also fundamentally different. Truncation error typically behaves in a predictable manner when changing the time step/tolerance of an integrator, while the rounding error will behave in an inherently random manner. Moreover, rounding error will (on average) increase with increasing number of integration steps, while truncation error will generally decrease if we take shorter integration steps. As a result, there is an approximate optimal time step, at which the sum of rounding and truncation error are at a minimum. When reducing the time step below this value, we will incur a *larger* error at the cost of a *higher* error. In typical situations, where ultimate accuracy is not required, truncation error will be the dominant source of error.

8.2 Multi-stage Methods

The multi-stage method relies on multiple function evaluations of $\dot{\mathbf{y}}$ in the interval $[t_i, t_{i+1}]$ — to perform a single step from t_i to t_{i+1} . That is, multiple 'stages' \mathbf{k}_j are computed to perform a single step as:

$$\bar{\mathbf{y}}_{i+1} = \bar{\mathbf{y}}_i + \sum_{j=1}^s b_j \mathbf{k}_j \quad (8.23)$$

where s is the number of stages used to make a single time-step. The stages \mathbf{k}_j are computed from:

$$\mathbf{k}_1 = \mathbf{f}(\bar{\mathbf{y}}_i, t_i + c_1 \Delta t) \quad (8.24)$$

$$\mathbf{k}_j = \mathbf{f} \left(\left(\bar{\mathbf{y}}_i + \Delta t \sum_{k=1}^{j-1} (a_{jk} \mathbf{k}_k) \right), t_i + c_j \Delta t \right) \quad (8.25)$$

where the coefficients b_j determine the contribution of each stage to the final time step, coefficients c_i determine the times at which each stage are evaluated, and a_{jk} determine the contribution of stage k to the value of $\bar{\mathbf{y}}$ used for the computation of stage j .

8.2.1 Fixed-step Method - Runge-Kutta 4

In this section, we give the explicit equations for the Runge-Kutta 4 integrator, a popular multi-stage integrator with $s = 4$.

8.2.2 Variable step-size methods

The use of multi-step methods permits a scheme to perform automatic step-size control of the integrator. By performing step-size control, the integrator attempts to achieve a give error level per time step (with this error level defined by one or more free parameters in the methods). For astrodynamics applications, this allows the time step to be automatically made smaller during periapsis (for the case of a highly eccentric orbit), where the dynamics changes much faster, and the error incurred during a single step of given length will generally be larger than at apoapsis.

A variable step-size multi-stage method is set up, so that the s function evaluations, see Eq. (8.23), are used to generate two distinct values of \bar{y}_{i+1}

$$\bar{\mathbf{y}}_{i+1} = \bar{\mathbf{y}}_i + \sum_{j=1}^s b_j \mathbf{k}_j \quad (8.26)$$

$$\bar{\mathbf{y}}_{i+1}^* = \bar{\mathbf{y}}_i + \sum_{j=1}^s b_j^* \mathbf{k}_j \quad (8.27)$$

Where the coefficients b_j and b_j^* belong to two different multi-stage integrators, which use the function evaluations \mathbf{k}_j to generate two different approximations of y_{i+1} .

We note that some of the values \mathbf{k}_j may only be used by one of the two methods (so that $b_j = 0$ or $b_j^* = 0$ for some values of j). The strength of the method described above, however, lies in the fact that only a limited number of additional function evaluations needs to be performed to generated the approximation \bar{y}_{i+1} in addition to \bar{y}_{i+1}^* (or *vice versa*).

For both integration methods, we assume that we know the error properties, with the two methods having order p and $p - 1$, so that:

$$\epsilon_{i+1} = \bar{y}_{i+1} - y_{i+1} = O(\Delta t^p) \quad (8.28)$$

$$\epsilon_{i+1}^* = \bar{y}_{i+1}^* - y_{i+1} = O(\Delta t^{p-1}) \quad (8.29)$$

Now, for the case that truncation error is the dominant error source, we make the assumption that the higher-order method has a much *lower* error than the lower-order methods, so that:

$$\epsilon_{i+1} \ll \epsilon_{i+1}^* \quad (8.30)$$

and we obtain an error estimate for the lower-order methods:

$$\epsilon_{i+1}^* \approx \bar{y}_{i+1}^* - \bar{y}_{i+1} \quad (8.31)$$

This estimate for ϵ_{i+1}^* allows us to modify the time step that is taken, in an attempt to control the error incurred during a single step. Assume we want to incur an error $\epsilon_{i+1,\text{req}}^*$, we can adjust the time step accordingly, since we know the relation between ϵ_{i+1}^* and Δt through Eq. (8.29). Consequently, we can compute a modified time step as:

$$\Delta t_{\text{req}} = \Delta t \left(\frac{\epsilon_{i+1,\text{req}}^*}{\epsilon_{i+1}^*} \right)^{\frac{1}{p-1}} \quad (8.32)$$

8.3 Multi-step Methods

Multi-step methods are fundamentally different from multi-state methods, in that they do not rely on multiple function evaluations of $\dot{\mathbf{y}}$ to perform a single step. Instead, they re-use previous steps $\bar{\mathbf{y}}_{i-1}$, $\bar{\mathbf{y}}_{i-2}$, ... to compute $\bar{\mathbf{y}}_{i+1}$.

The explicit Adams-Bashforth (AB) method uses:

$$\bar{\mathbf{y}}_{i+1} = \bar{\mathbf{y}}_i + \Delta t \sum_{j=0}^s b_j \mathbf{f}(t_{i-j}, \bar{\mathbf{y}}_{i-j}) \quad (8.33)$$

where the times t_{n-j} are assumed to be equispaced, so that:

$$t_{i-j} = t_i - j\Delta t \quad (8.34)$$

The coefficients b_j define the weight of each past function evaluation, and are specific for a given method of given s .

In addition, the *implicit* Adams-Moulton method uses:

$$\bar{\mathbf{y}}_{i+1} = \bar{\mathbf{y}}_i + \Delta t \sum_{j=-1}^s c_j \mathbf{f}(t_{i-j}, \bar{\mathbf{y}}_{i-j}) \quad (8.35)$$

$$= \bar{\mathbf{y}}_i + \Delta t \left(c_{-1} \mathbf{f}(t_{i+1}, \bar{\mathbf{y}}_{i+1}) + \sum_{j=0}^s c_j \mathbf{f}(t_{i-j}, \bar{\mathbf{y}}_{i-j}) \right) \quad (8.36)$$

The problem in using Eq. (8.35) is that it has $\bar{\mathbf{y}}_{i+1}$, the quantity that is to be the output of the integration step, on both sides of the equation. One way of dealing with this issue is to use the predictor-corrector Adams-Bashforth-Moulton (ABM) method. This method is based on feeding a predicted value $\bar{\mathbf{y}}_{i+1}^*$, produced by the AB method, into the right-hand side of the AM method, so that:

$$\bar{\mathbf{y}}_{i+1}^* = \bar{\mathbf{y}}_i + \Delta t \sum_{j=0}^s b_j \mathbf{f}(t_{i-j}, \bar{\mathbf{y}}_{i-j}) \quad (8.37)$$

$$\bar{\mathbf{y}}_{i+1} = \bar{\mathbf{y}}_i + \Delta t \left(c_{-1} \mathbf{f}(t_{i+1}, \bar{\mathbf{y}}_{i+1}^*) + \sum_{j=0}^s c_j \mathbf{f}(t_{i-j}, \bar{\mathbf{y}}_{i-j}) \right) \quad (8.38)$$

which produces the required approximate state $\bar{\mathbf{y}}_{i+1}$.

8.3.1 Step-size and order control

8.4 Extrapolation Methods

Chapter 9

Numerical Interpolators

An interpolator is a tool that is used to generate a continuous time series $\mathbf{f}(t)$ from a discrete set of N data points $\mathbf{f}(t_i)$ at times t_i , with $i = 0 \dots N - 1$. We will assume in what follows that the list t_i is sorted in ascending order, so that $t_i < t_{i+1}$. Moreover, we will provide the governing equations for a vector function \mathbf{f} but, unless otherwise indicated, the methods are equally applicable to scalar and matrix functions.

An interpolator defines a method by which to compute a value of \mathbf{f} at a given time t . Typically, the first step in the algorithm is the computation of the index j for which:

$$t \in [t_j, t_{j+1}) \quad (9.1)$$

where we will denote the interval $[t_j, t_{j+1})$ as interval j . There are several algorithms to determine the interval j , which we will not discuss further here. In addition, we may have the situation where:

$$(t < t_0) \wedge (t > t_{N-1}) \quad (9.2)$$

in which case we must extrapolate (not interpolate) from the available data $\mathbf{f}(t_i)$.

9.1 Piecewise constant interpolator

This basic interpolator sets, after determining the interval j of the input time t as per Eq. (9.1), the interpolated value as:

$$\mathbf{f}(t) = \mathbf{f}(t_j) \quad (9.3)$$

As a result, the interpolated value of \mathbf{f} is constant over each interval j , and discontinuous on the boundary between two intervals.

9.2 Linear interpolator

This interpolator uses a linear function to interpolate using only the data at the limits of interval j , as follows:

$$\mathbf{f}(t) = \mathbf{f}(t_j) + \frac{\mathbf{f}(t_{j+1}) - \mathbf{f}(t_j)}{t_{j+1} - t_j}(t - t_j) \quad (9.4)$$

As a result, although the interpolated value of \mathbf{f} is continuous on the boundary between two intervals, its first derivative is constant over an interval, and discontinuous at the interval boundaries.

9.3 Lagrange interpolator

The Lagrange interpolator uses a degree- k polynomial to interpolate a data set. Given $k + 1$ data points $\mathbf{f}(t_0) \dots \mathbf{f}(t_k)$, it interpolates using the following algorithm:

$$\mathbf{f}(t) = \sum_{n=0}^k l_n(t) \mathbf{f}(t_n) \quad (9.5)$$

$$l_n(t) = \prod_{m=0, m \neq n}^k \frac{t - t_m}{t_n - t_m} \quad (9.6)$$

with $l_n(t)$ the Lagrange polynomial basis functions.

This definition of an interpolator will only work if exactly $k + 1$ data points are given as input, which is insufficiently general for our purposes. What we require is a definition for a degree- k interpolating function, using an arbitrary input data set of size M , with $M \geq k + 1$.

For a given data set of size M , an interpolation time t , and an associated interval j , we apply the following algorithm to generate the interpolating polynomial of degree k , where we will assume that $k + 1$ (the number of data points used to generate the polynomial) is an even number. A degree k polynomial spans data at k intervals. To interpolate data at the middle interval, we use $(k - 1)/2$ intervals before and after this middle- interval, so data at the set of times:

$$\left\{ t_{j - \frac{k-1}{2}} \dots t_{j + \frac{k}{2}} \right\} \quad (9.7)$$

The data at these times is used to generate an interpolating polynomial centered on interval j using Eq. (9.5). That is, for each interval j , we define a separate set of times as in (9.7), and a separate interpolating polynomial.

The reason that we use a set of times *centered* at interval j , is that higher-order interpolating polynomials suffer from 'Runge's phenomenon', which can cause strong oscillations between data points near the edges of the interpolating polynomial. As a result, the interpolating polynomial can only be 'trusted' in

the center interval. This does not pose any problems, except at the edges of the full data set. Particularly, for the first and last $(k - 1)/2$ intervals of the data set, the Lagrange polynomial is prone to large errors, and its results should be treated with extreme skepticism, and an alternative, lower-order, interpolating method should be used in these intervals.

Appendix A

Summary of Notation

In this document, we adhere to the following conventions:

- Vectors are written in bold, with a lower-case symbol: \mathbf{a}
- Matrices/tensors are written in bold, with an upper-case symbol: \mathbf{A}
- Cartesian position, velocity and acceleration vectors are denoted \mathbf{r} , \mathbf{v} and \mathbf{a} , respectively

For position and velocity vectors, we use the following:

- Position and velocity vectors of A , expressed in a frame with origin at B and a frame with non-inertial orientation C , are written as $\mathbf{r}_{BA}^{(C)}$ and $\mathbf{v}_{BA}^{(C)}$
- Position and velocity vectors of A , expressed in a frame with origin at B and an inertial orientation, are written as \mathbf{r}_{BA} and \mathbf{v}_{BA}

For accelerations, we use the following:

- Accelerations acting on A , exerted by B , w.r.t. an origin C , expressed in a frame with non-inertial orientation D are written as $(\mathbf{a}_{BA}^{(D)})_C$
- Accelerations acting on A , exerted by B , w.r.t. an origin C , expressed in a frame with an inertial orientation, are written as $(\mathbf{a}_{BA})_C$
- Accelerations acting on A , exerted by B , w.r.t. an inertial origin, expressed in a frame with an inertial orientation, are written as \mathbf{a}_{BA}

For rotation representations, we use the following:

- Rotation matrices from frame B to frame A are written as $\mathbf{R}^{(A/B)}$

Bibliography

- Archinal, B. A., A'Hearn, M. F., Bowell, E., Conrad, A., Consolmagno, G. J., Courtin, R., Fukushima, T., Hestroffer, D., Hilton, J. L., Krasinsky, G. A., Neumann, G., Oberst, J., Seidelmann, P. K., Stooke, P., Tholen, D. J., Thomas, P. C., and Williams, I. P. (2011). Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2009. *Celestial Mechanics and Dynamical Astronomy*, 109:101–135.
- Battin, R. H. (1999). *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics.
- Dirkx, D., Mooij, E., and Root, B. (2019). Propagation and estimation of the dynamical behaviour of gravitationally interacting rigid bodies. *Astrophysics and Space Science*, 364(2):37.
- Fienga, A., Laskar, J., Morley, T., Manche, H., Kuchynka, P., Le Poncin-Lafitte, C., Budnik, F., Gastineau, M., and Somenzi, L. (2009). INPOP08, a 4-D planetary ephemeris: from asteroid and time-scale computations to ESA Mars Express and Venus Express contributions. *Astronomy Astrophysics*, 507:1675–1686.
- Folkner, W. M., Williams, J. G., and Boggs, D. H. (2009). The Planetary and Lunar Ephemeris DE 421. *Interplanetary Network Progress Report*, 42(178):C1.
- Folkner, W. M., Williams, J. G., Boggs, D. H., Park, R. S., and Kuchynka, P. (2014). The Planetary and Lunar Ephemerides DE430 and DE431. *Interplanetary Network Progress Report*, 42(196):C1.
- Kopeikin, S., Efroimsky, M., and Kaplan, G. (2011). *Relativistic Celestial Mechanics of the Solar System*. Wiley VCH.
- Lainey, V., Duriez, L., and Vienne, A. (2004). New accurate ephemerides for the Galilean satellites of Jupiter. I. Numerical integration of elaborated equations of motion. *Astronomy and Astrophysics*, 420:1171–1183.
- Montenbruck, O. (1992). Numerical integration methods for orbital motion. *Celestial Mechanics and Dynamical Astronomy*, 53(1):59–69.

- Montenbruck, O. and Gill, E. (2000). *Satellite Orbits: Models, Methods, and Applications*. Physics and Astronomy Online Library. Springer Verlag.
- Mooij, E. (1994). *The Motion of a Vehicle in a Planetary Atmosphere*, volume LR-768. Delft University Press.
- Moyer, T. D. (1971). Mathematical formulation of the Double-Precision Orbit Determination Program (DPODP). Technical Report 32-1527, NASA, Jet Propulsion Laboratory.
- Petit, G., Luzum, B., and et al. (2010). IERS Conventions (2010). *IERS Technical Note*, No. 36.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). Numerical recipes in c++. *The art of scientific computing*, 2:1002.
- Rambaux, N., Castillo-Rogez, J. C., Le Maistre, S., and Rosenblatt, P. (2012). Rotational motion of Phobos. *Astronomy and Astrophysics*, 548:A14.
- Seeber, G. (2003). *Satellite Geodesy*. Walter de Gruyter, 2nd edition.
- Vallado, D. and McClain, W. (2001). *Fundamentals of Astrodynamics and Applications*. Springer.
- Vittaldev, V., Mooij, E., and Naeije, M. (2012). Unified state model theory and application in astrodynamics. *Celestial Mechanics and Dynamical Astronomy*, 112(3):253–282.
- Wakker, K. F. (2015). *Fundamentals of astrodynamics*. TU Delft Library.