Software Requirements Specification for Chipmunk2D

Alex Halliwushka and Luthfi Mawarid

May 3, 2018

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

The unit system used throughout is SI (Système International d'Unités). In addition to the basic units, several derived units are also used. For each unit, the table lists the symbol, a description and the SI name.

Symbol	Description
kg	mass (kilogram)
m	length (metre)
N	force (newton)
rad	angle (radian)
s	time (second)

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. Throughout the document, symbols in bold will represent vectors, and scalars otherwise. The symbols are listed in alphabetical order.

Symbol	Description	Units
$\mathbf{a}$	Acceleration	$\frac{\text{m}}{\text{s}^2}$
$\mathbf{a}_i$	The I-Th Body's Acceleration	$\frac{\text{m}}{\text{s}^2}$
$\mathbf{a}(t)$	Linear Acceleration	$\frac{\text{m}}{\text{s}^2}$
C_R	Coefficient of restitution	
$\mathbf{F}_1$	Force exerted by the first body (on another body)	N
$\mathbf{F}_2$	Force exerted by the second body (on another body)	N
$\mathbf{F}_i$	Force Applied to the I-Th Body at Time T	N
$\mathbf{F}$	Force	N
g	Gravitational acceleration	$\frac{\text{m}}{\text{s}^2}$
G	Gravitational constant	$\frac{\text{m}^3}{(\text{kgs}^2)}$
$\hat{\mathbf{i}}$	Horizontal Unit Vector	m
$\mathbf{I}_A$	Moment of Inertia Of Rigid Body a	kgm^2
$\mathbf{I}_B$	Moment of Inertia Of Rigid Body B	kgm^2

$\mathbf{I}_k$	Moment of Inertia Of the K-Th Rigid Body	kgm^2
$\mathbf{I}$	Moment of inertia	kgm^2
$\mathbf{J}$	Impulse (vector)	Ns
j	Impulse (scalar)	Ns
$\hat{\mathbf{j}}$	Vertical Unit Vector	m
j	Collision Impulse (scalar)	Ns
L	Length	m
m_1	Mass of the first body	kg
m_2	Mass of the second body	kg
m_A	Mass Of Rigid Body a	kg
m_B	Mass Of Rigid Body B	kg
m_i	Mass Of the I-Th Rigid Body	kg
m_j	Mass Of the J-Th Particle	kg
M	Total Mass of the Rigid Body	kg
m	Mass	kg
$\mathbf{n}$	Collision Normal Vector	m
n	Number of particles in a rigid body	
$\mathbf{p}_{CM}$	The mass-weighted average position of a rigid body's particles	m
$\mathbf{p}_j$	Position Vector of the J-Th Particle	m
P	Point of collision	m
$\mathbf{p}$	Position	m
r	Distance	m
$\mathbf{r}$	Displacement	m
$\hat{\mathbf{r}}$	Displacement unit vector	m
$\mathbf{r}_{AP}$	Displacement vector between the centre of mass of rigid body A and contact point P	m
$\mathbf{r}_{BP}$	Displacement vector between the centre of mass of rigid body B and contact point P	m
$\mathbf{r}_{kP}$	Displacement vector between the centre of mass of rigid body k and contact point P	m
$\mathbf{r}_{OB}$	Displacement vector between the origin and point B	m
$\mathbf{r}(t)$	Linear Displacement	m
t	Time	s
t_c	Denotes the time at collision	s
t	Point in time	s
t_0	Denotes the initial time	s
${\mathbf{v}_i}^{AB}$	Relative Velocity Between Rigid Bodies of a and B	$\frac{\text{m}}{\text{s}}$

$\mathbf{v}_A$	Velocity At Point a	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}_B$	Velocity At Point B	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}_i$	Velocity Of the I-Th Body's Velocity	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}_O$	Velocity At Point Origin	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}$	Velocity	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}(t)$	Linear Velocity	$\frac{\text{m}}{\text{s}}$
$\|\|\mathbf{n}\|\|$	Length of the Normal Vector	m
$\|\|\mathbf{r}_{AP} * \mathbf{n}\|\|$	Length of the Perpendicular Vector To the Contact Displacement Vector of Rigid Body a	m
$\|\|\mathbf{r}_{BP} * \mathbf{n}\|\|$	Length of the Perpendicular Vector To the Contact Displacement Vector of Rigid Body B	m
$\|\|\mathbf{r}\|\|$	Euclidean norm of the displacement	m
$\|\|\mathbf{r}\|\|^2$	Squared distance	m^2
α	Angular Acceleration	$\frac{\text{rad}}{\text{s}^2}$
θ	Angular Displacement	rad
π	Pi	
τ_i	Is the torque applied to the i-th body	Nm
τ	Torque	Nm
ϕ	Orientation	rad
ω_A	Is the a Body's Angular Velocity	$\frac{\text{rad}}{\text{s}}$
ω_B	Is the B Body's Angular Velocity	$\frac{\text{rad}}{\text{s}}$
ω	Angular Velocity	$\frac{\text{rad}}{\text{s}}$

1.3 Abbreviations and Acronyms

Symbol	Description
2D	Two-Dimensional
A	Assumption
CM	Centre of Mass
Chipmunk2D	Chipmunk2D game physics library
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model

LC	Likely Change
ODE	Ordinary Differential Equation
R	Requirement
SRS	Software Requirements Specification
T	Theoretical Model

2 Introduction

Due to the rising cost of developing video games, developers are looking for ways to save time and money for their projects. Using an open source physics library that is reliable and free will cut down development costs and lead to better quality products. The following section provides an overview of the Software Requirements Specification (SRS) for Chipmunk2D. This section explains the purpose of this document, the scope of the system, the organization of the document, and the characteristics of the intended reader.

2.1 Purpose of Document

This document descibes the modeling of an open source 2D rigid body physics library used for games. The theoretical models and goal statements used in Chipmunk2D are provided. This document is intended to be used as a reference to provide all necessary information to understand and verify the model. This document will be used as a starting point for subsequent development phases, including writing the design specification and the software verification and validation plan. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation. Although the SRS fits in a series of documents that follow the so-called waterfall model, the actual development process is not constrained in any way. Even when the waterfall model is not followed, as Parnas and Clements point out, the most logical way to present the documentation is still to "fake" a rational design process.

2.2 Scope of Requirements

The scope of the requirements includes the physical simulation of 2D rigid bodies acted on by forces. Given the appropriate inputs, the code for Chipmunk2D is intended to simulate how these rigid bodies interact with one another.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have a strong knowledge in rigid body dynamics. The reviewers should also have an understanding of high school calculus. The users of Chipmunk2D can have a lower level of expertise, as explained in Section **??**.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [1] and [2]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the instance models in Section **??** and trace back to find any additional information they require. The goal statements are refined to the theoretical models, and the theoretical models to the instance models.

3 General System Description

This section provides general information about the system, identifies the interfaces between the system and its environment, and describes the user characteristics and the system constraints.

3.1 User Characteristics

The end user of Chipmunk2D should have an understanding of first year programming concepts and an understanding of high school physics.

3.2 System Constraints

There are no system constraints.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, and definitions that are used for the physics library.

4.1 Problem Description

Creating a gaming physics library is a difficult task. Games need physics libraries that simulate objects acting under various physical conditions, while simultaneously being fast and efficient enough to work in soft real-time during the game. Developing a physics library from scratch takes a long period of time and is very costly, presenting barriers of entry which make it difficult for game developers to include physics in their products. There are a few free, open source and high quality physics libraries available to be used for consumer products (Section **??**). By creating a simple, lightweight, fast and portable 2D rigid body physics library, game development will be more accessible to the masses and higher quality products will be produced.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements

1. Rigid body: A solid body in which deformation is neglected.
2. Elasticity: Ratio of the relative velocities of two colliding objects after and before a collision.
3. Centre of mass: The mean location of the distribution of mass of the object.
4. Cartesian coordinate system: A coordinate system that specifies each point uniquely in a plane by a pair of numerical coordinates.
5. Right-handed coordinate system: A coordinate system where the positive z-axis comes out of the screen.

4.1.2 Goal Statements

Given the inputs, the goal statements are:

GS1: Given the physical properties, initial positions and velocities, and forces applied on a set of rigid bodies, determine their new positions and velocities over a period of time.

GS2: Given the physical properties, initial orientations and angular velocities, and forces applied on a set of rigid bodies, determine their new orientations and angular velocities over a period of time.

GS3: Given the initial positions and velocities of a set of rigid bodies, determine if any of them will collide with one another over a period of time.

GS4: Given the physical properties, initial positions, orientations, linear velocities, and angular velocities of a set of rigid bodies, determine the new positions, orientations, linear velocities, and angular velocities over a period of time of the rigid bodies that have undergone a collision.

4.2 Solution Characteristics Specification

The instance models that govern Chipmunk2D are presented in FIXME REF to IModSection. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the Theoretical Models, General Definitions, Data Definitions, Instance Models, or Likely Changes, in which the respective assumption is used.

A1: All objects are rigid bodies.

A2: All objects are 2D.

A3: The library uses a Cartesian coordinate system.

A4: The axes are defined using right-handed coordinate system.

A5: All rigid bodies collisions are vertex-to-edge collisions.

A6: There is no damping involved throughout the simulation.

A7: There are no constraints and joints involved throughout the simulation.

4.2.2 Theoretical Models

This section focuses on the general equations and laws that Chipmunk2D is based on.

Refname	**T:t1NewtonSL**
Label	Newton's Second Law of Motion
Equation	$\mathbf{F} = m\mathbf{a}$
Description	The net force $\mathbf{F}$ (N) on a rigid body is proportional to the acceleration $\mathbf{a}$ ($\frac{\text{m}}{\text{s}^2}$) of the rigid body, where m (kg) denotes the mass of the rigid body as the constant of proportionality.

Refname	**T:t2NewtonTL**
Label	Newton's Third Law of Motion
Equation	$\mathbf{F}_1 = -\mathbf{F}_2$
Description	Every action has an equal and opposite reaction. In other words, the force $\mathbf{F}_1$ (N) exerted on the second rigid body by the first is equal in magnitude and in the opposite direction to the force $\mathbf{F}_2$ (N) exerted on the first rigid body by the second.

Refname	**T:t3NewtonLUG**
Label	Newton's Law of Universal Gravitation
Equation	$\mathbf{F} = G\frac{m_1 m_2}{\lvert\lvert\mathbf{r}\rvert\rvert^2}\hat{\mathbf{r}} = G\frac{m_1 m_2}{\lvert\lvert\mathbf{r}\rvert\rvert^2}\frac{\mathbf{r}}{\lvert\lvert\mathbf{r}\rvert\rvert}$
Description	Two rigid bodies in the universe attract each other with a force $\mathbf{F}$ (N) that is directly proportional to the product of their masses, m_1 and m_2 (kg), and inversely proportional to the squared distance $\lvert\lvert\mathbf{r}\rvert\rvert^2$ (m^2) between them. The vector $\mathbf{r}$ (m) is the displacement between the centres of the rigid bodies and $\lvert\lvert\mathbf{r}\rvert\rvert$ (m) represents the Euclidean norm of the displacement, or absolute distance between the two. $\hat{\mathbf{r}}$ denotes the displacement unit vector, equivalent to the displacement divided by the Euclidean norm of the displacement, as shown above. Finally, G is the gravitational constant (6.673 * 10E-11) ($\frac{\text{m}^3}{\text{(kgs}^2\text{)}}$).

Refname	**T:t4ChaslesThm**
Label	Chasles' Theorem
Equation	$\mathbf{v}_B = \mathbf{v}_O + \omega \times \mathbf{r}_{OB}$
Description	The linear velocity $\mathbf{v}_B$ ($\frac{\text{m}}{\text{s}}$) of any point B in a rigid body is the sum of the linear velocity $\mathbf{v}_O$ ($\frac{\text{m}}{\text{s}}$) of the rigid body at the origin (axis of rotation) and the resultant vector from the cross product of the rigid body's angular velocity ω ($\frac{\text{rad}}{\text{s}}$) and the displacement vector between the origin and point B, $\mathbf{r}_{OB}$ (m).

Refname	**T:t5NewtonSLR**
Label	Newton's Second Law for Rotational Motion
Equation	$\tau = \mathbf{I}\alpha$
Description	The net torque τ (Nm) on a rigid body is proportional to its angular acceleration α ($\frac{\text{rad}}{\text{s}^2}$). Here, $\mathbf{I}$ (kgm^2) denotes the moment of inertia of the rigid body. We also assume that all rigid bodies involved are two-dimensional (A2).

4.2.3 General Definitions

There are no general definitions.

4.2.4 Instance Models

This section transforms the problem defined in FIXME REF into one which is expressed in mathematical terms. It uses concrete symbols defined in FIXME REF to replace the abstract symbols in the models identified in FIXME REF and FIXME REF.

Refname	**T:im1**
Label	Force on the Translational Motion of a Set of 2d Rigid Bodies
Equation	$\mathbf{a}_i = \frac{d\mathbf{v}_i(t)}{dt} = g + \frac{\mathbf{F}_i(t)}{m_j}$
Description	The above equation expresses the total acceleration of the rigid body (A1, A2) i as the sum of gravitational acceleration (GD3) and acceleration due to applied force Fi(t) (T1). The resultant outputs are then obtained from this equation using DD2, DD3 and DD4. It is currently assumed that there is no damping (A6) or constraints (A7) involved. m_i is the mass of the i-th rigid body (kg). g is the gravitational acceleration ($\frac{\text{m}}{\text{s}^2}$). t is the point in time (s). t_0 is the denotes the initial time (s). $\mathbf{p}_{CM}$ is the the mass-weighted average position of a rigid body's particles (m). $\mathbf{a}$ is the acceleration ($\frac{\text{m}}{\text{s}^2}$). $\mathbf{v}$ is the velocity ($\frac{\text{m}}{\text{s}}$). $\mathbf{F}_i$ is the force applied to the i-th body at time t (N).

Refname	**T:im2**
Label	Force on the Rotational Motion of a Set of 2D Rigid Body
Equation	$\alpha = \frac{d\,\omega(t)}{d\,t} = \frac{\tau_i(t)}{\mathbf{I}}$
Description	The above equation for the total angular acceleration of the rigid body (A1, A2) i is derived from T5, and the resultant outputs are then obtained from this equation using DD5, DD6 and DD7. It is currently assumed that there is no damping (A6) or constraints (A7) involved. m_i is the mass of the i-th rigid body (kg). g is the gravitational acceleration ($\frac{\text{m}}{\text{s}^2}$). t is the point in time (s). t_0 is the denotes the initial time (s). ϕ is the orientation (rad). ω is the angular velocity ($\frac{\text{rad}}{\text{s}}$). α is the angular acceleration ($\frac{\text{rad}}{\text{s}^2}$). τ_i is the is the torque applied to the i-th body (Nm). $\mathbf{I}_k$ is the moment of inertia of the k-th rigid body (kgm^2).

Refname	**T:im3**
Label	Collisions on 2D Rigid Bodies
Equation	$\mathbf{v}_A\left(t_c\right) = \mathbf{v}_A\left(t\right) + \frac{j}{m_A}\mathbf{n}$
Description	This instance model is based on our assumptions regarding rigid body (A1, A2) collisions (A5). Again, this does not take damping (A6) or constraints (A7) into account. m_i is the mass of the i-th rigid body (kg). $\mathbf{I}_k$ is the moment of inertia of the k-th rigid body (kgm^2). t is the point in time (s). t_0 is the denotes the initial time (s). t_c is the denotes the time at collision (s). $\mathbf{p}_{CM}$ is the the mass-weighted average position of a rigid body's particles (m). $\mathbf{v}$ is the velocity ($\frac{\text{m}}{\text{s}}$). ϕ is the orientation (rad). ω is the angular velocity ($\frac{\text{rad}}{\text{s}}$). $\mathbf{n}$ is the collision normal vector (m). j is the collision impulse (scalar) (Ns). P is the point of collision (m). $\mathbf{r}_{kP}$ is the displacement vector between the centre of mass of rigid body k and contact point P (m).

4.2.5 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Refname	**DD:p.CM**
Label	The Mass-Weighted Average Position of a Rigid Body's Particles
Units	m
Equation	$\mathbf{p}_{CM} = \frac{\sum m_j \mathbf{p}_j}{M}$
Description	$\mathbf{p}_{CM}$ is the the mass-weighted average position of a rigid body's particles (m) m_j is the mass of the j-th particle (kg) $\mathbf{p}_j$ is the position vector of the j-th particle (m) M is the total mass of the rigid body (kg)

Refname	**DD:linearDisplacement**
Label	Linear Displacement
Units	m
Equation	$\mathbf{r}(t) = \frac{d\,\mathbf{p}(t)}{d\,t}$
Description	$\mathbf{r}(t)$ is the linear displacement (m) $\mathbf{p}$ is the position (m) t is the time (s)

Refname	**DD:linearVelocity**
Label	Linear Velocity
Units	$\frac{\text{m}}{\text{s}}$
Equation	$\mathbf{v}(t) = \frac{d\,\mathbf{r}(t)}{d\,t}$
Description	$\mathbf{v}(t)$ is the linear velocity ($\frac{\text{m}}{\text{s}}$) $\mathbf{r}$ is the displacement (m) t is the time (s)

Refname	**DD:linearAcceleration**
Label	Linear Acceleration
Units	$\frac{\text{m}}{\text{s}^2}$
Equation	$\mathbf{a}(t) = \frac{d\,\mathbf{v}(t)}{d\,t}$
Description	$\mathbf{a}(t)$ is the linear acceleration ($\frac{\text{m}}{\text{s}^2}$) $\mathbf{v}$ is the velocity ($\frac{\text{m}}{\text{s}}$) t is the time (s)

Refname	**DD:angularDisplacement**
Label	Angular Displacement
Units	rad
Equation	$\theta = \frac{d\,\phi(t)}{d\,t}$
Description	θ is the angular displacement (rad) ϕ is the orientation (rad) t is the time (s)

Refname	**DD:angularVelocity**
Label	Angular Velocity
Units	$\frac{\text{rad}}{\text{s}}$
Equation	$\omega=\frac{d\,\theta(t)}{d\,t}$
Description	ω is the angular velocity ($\frac{\text{rad}}{\text{s}}$) θ is the angular displacement (rad) t is the time (s)

Refname	**DD:angularAcceleration**
Label	Angular Acceleration
Units	$\frac{\text{rad}}{\text{s}^2}$
Equation	$\alpha=\frac{d\,\omega(t)}{d\,t}$
Description	α is the angular acceleration ($\frac{\text{rad}}{\text{s}^2}$) ω is the angular velocity ($\frac{\text{rad}}{\text{s}}$) t is the time (s)

Refname	**DD:impulseS**
Label	Impulse (scalar)
Units	Ns
Equation	$j = \frac{-(1+C_R)\mathbf{v}_i{}^{AB}\cdot\mathbf{n}}{\left(\frac{1}{m_A}+\frac{1}{m_B}\right)\|\mathbf{n}\|^2+\frac{\|\mathbf{r}_{AP}*\mathbf{n}\|^2}{\mathbf{I}_A}+\frac{\|\mathbf{r}_{BP}*\mathbf{n}\|^2}{\mathbf{I}_B}}$
Description	j is the impulse (scalar) (Ns) C_R is the coefficient of restitution $\mathbf{v}_i{}^{AB}$ is the relative velocity between rigid bodies of A and B ($\frac{\text{m}}{\text{s}}$) $\mathbf{n}$ is the collision normal vector (m) m_A is the mass of rigid body A (kg) m_B is the mass of rigid body B (kg) $\|\mathbf{n}\|$ is the length of the normal vector (m) $\|\mathbf{r}_{AP}*\mathbf{n}\|$ is the length of the perpendicular vector to the contact displacement vector of rigid body A (m) $\mathbf{I}_A$ is the moment of inertia of rigid body A (kgm^2) $\|\mathbf{r}_{BP}*\mathbf{n}\|$ is the length of the perpendicular vector to the contact displacement vector of rigid body B (m) $\mathbf{I}_B$ is the moment of inertia of rigid body B (kgm^2)

4.2.6 Data Constraints

the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise. FIXME

Var	Physical Constraints	Typical Value	TU
L	$L \geq 0$	44.2 m	0.1
m	$m \geq 0$	56.2 kg	0.1
$\mathbf{I}$	$\mathbf{I} \geq 0$	74.5 kgm^2	0.1
G	None	9.8 $\frac{\text{m}^3}{(\text{kgs}^2)}$	0.1
$\mathbf{p}$	None	0.412 m	0.1
ϕ	None	$\frac{\pi}{2}$ rad	0.1

$\mathbf{v}$	None	2.51 $\frac{\text{m}}{\text{s}}$	0.1
ω	None	2.1 $\frac{\text{rad}}{\text{s}}$	0.1
$\mathbf{F}$	None	98.1 N	0.1
τ	None	200.0 Nm	0.1
C_R	$0 \leq C_R \leq 1$	0.8	0.1

Table 2: Input Data Constraints

Var
$\mathbf{p}$
$\mathbf{v}$
ϕ
ω

Table 3: Output Data Constraints

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the non-functional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: Create a space for all of the rigid bodies in the physical simulation to interact in.

R2: Input the initial masses, velocities, orientations, angular velocities of, and forces applied on rigid bodies.

R3: Input the surface properties of the bodies such as friction or elasticity.

R4: Verify that the inputs satisfy the required physical constraints from Section **??**.

R5: Determine the positions and velocities over a period of time of the 2D rigid bodies acted upon by a force.

R6: Determine the orientations and angular velocities over a period of time of the 2D rigid bodies.

R7: Determine if any of the rigid bodies in the space have collided.

R8: Determine the positions and velocities over a period of time of the 2D rigid bodies that have undergone a collision.

5.2 Non-Functional Requirements

Games are resource intensive, so performance is a high priority. Other non-functional requirements that are a priority are: correctness, understandability, portability, reliability, and maintainability.

6 Likely Changes

This section lists the likely changes to be made to the physics game library.

LC1: The internal ODE-solving algorithm used by the library may be changed in the future.

LC2: The library may be expanded to deal with edge-to-edge and vertex-to-vertex collisions.

LC3: The library may be expanded to include motion with damping.

LC4: The library may be expanded to include joints and constraints.

7 Off-The-Shelf Solutions

As mentioned in Section **??**, there already exist free open source game physics libraries. Similar 2D physics libraries are:

1. Box2D: http://box2d.org/
2. Nape Physics Engine: http://napephys.com/

Free open source 3D game physics libraries include:

1. Bullet: http://bulletphysics.org/
2. Open Dynamics Engine: http://www.ode.org/
3. Newton Game Dynamics: http://newtondynamics.com/

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" should be modified as well. Table **??** shows the dependencies of goal statements, requirements, instance models, and data constraints with each other. Table **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, and on the assumptions. Table **??** shows the dependencies of theoretical models, general definitions, data definitions, and instance models on each other.

	IM1 (Definition T:im1)	IM2 (Definition T:im2)	IM3 (Definition T:im3)	R1 (Sect
GS1 (Section **??**)	X			
GS2 (Section **??**)		X		
GS3 (Section **??**)			X	
GS4 (Section **??**)			X	
R1 (Section **??**)				
R2 (Section **??**)	X	X		
R3 (Section **??**)			X	
R4 (Section **??**)				
R5 (Section **??**)	X			
R6 (Section **??**)		X		
R7 (Section **??**)				X
R8 (Section **??**)			X	

Table 4: Traceability Matrix Showing the Connections Betw
Goal Statements (Section **??**) and Other Items

	A1 (Section **??**)	A2 (Section **??**)	A3 (Section **??**)	A4 (
T1 (Definition T:t1NewtonSL)				
T2 (Definition T:t2NewtonTL)				
T3 (Definition T:t3NewtonLUG)				
T4 (Definition T:t4ChaslesThm)	X			
T5 (Definition T:t5NewtonSLR)				
GD1 (Section **??**)				
GD2 (Section **??**)				
GD3 (Section **??**)		X	X	
GD4 (Section **??**)				
GD5 (Section **??**)				
GD6 (Section **??**)				
GD7 (Section **??**)				
DD1 (Definition DD:p.CM)	X	X		
DD2 (Definition DD:linearDisplacement)	X	X		
DD3 (Definition DD:linearVelocity)	X	X		
DD4 (Definition DD:linearAcceleration)	X	X		
DD5 (Definition DD:angularDisplacement)	X	X		
DD6 (Definition DD:angularVelocity)	X	X		
DD7 (Definition DD:angularAcceleration)	X	X		
DD8 (Definition DD:impulseS)	X	X		X
IM1 (Definition T:im1)	X	X		
IM2 (Definition T:im2)	X	X		X

IM3 (Definition T:im3)	X	X
LC1 (Section ??)		
LC2 (Section ??)		
LC3 (Section ??)		
LC4 (Section ??)		

Table 5: Traceability Matrix Showing the Connections Between and Other Items

	T1 (Definition T:t1NewtonSL)	T2 (Definition T:t2Newtor
T1 (Definition T:t1NewtonSL)		
T2 (Definition T:t2NewtonTL)		
T3 (Definition T:t3NewtonLUG)		
T4 (Definition T:t4ChaslesThm)		
T5 (Definition T:t5NewtonSLR)		
GD1 (Section ??)	X	
GD2 (Section ??)		X
GD3 (Section ??)	X	
GD4 (Section ??)		
GD5 (Section ??)		
GD6 (Section ??)		
GD7 (Section ??)		
DD1 (Definition DD:p.CM)		
DD2 (Definition DD:linearDisplacement)		
DD3 (Definition DD:linearVelocity)		
DD4 (Definition DD:linearAcceleration)		
DD5 (Definition DD:angularDisplacement)		
DD6 (Definition DD:angularVelocity)		
DD7 (Definition DD:angularAcceleration)		
DD8 (Definition DD:impulseS)		
IM1 (Definition T:im1)	X	
IM2 (Definition T:im2)		
IM3 (Definition T:im3)		

9 Values of Auxiliary Constants

There are no auxiliary constants.

10 References