

Software Requirements Specification for Binary Star System Simulator

Xinlu Yan

June 30, 2026

Contents

1	Reference Material	3
1.1	Table of Units	3
1.2	Table of Symbols	3
1.3	Abbreviations and Acronyms	5
2	Introduction	5
2.1	Purpose of Document	6
2.2	Scope of Requirements	6
2.3	Characteristics of Intended Reader	6
2.4	Organization of Document	6
3	General System Description	6
3.1	System Context	7
3.2	User Characteristics	8
3.3	System Constraints	8
4	Specific System Description	8
4.1	Problem Description	8
4.1.1	Terminology and Definitions	8
4.1.2	Physical System Description	9
4.1.3	Goal Statements	10
4.2	Solution Characteristics Specification	10
4.2.1	Assumptions	10
4.2.2	Theoretical Models	11
4.2.3	General Definitions	16
4.2.4	Data Definitions	16
4.2.5	Instance Models	16
4.2.6	Data Constraints	21
4.2.7	Properties of a Correct Solution	25

5	Requirements	25
5.1	Functional Requirements	25
5.2	Non-Functional Requirements	26
6	Likely Changes	26
7	Unlikely Changes	27
8	Traceability Matrices and Graphs	27
9	Values of Auxiliary Constants	31
10	References	31

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 of Units](#) lists the symbol, a description, and the SI name.

Table 1: Table of Units

Symbol	Description	SI Name
J	energy	joule
kg	mass	kilogram
m	length	metre
N	force	newton
s	time	second

1.2 Table of Symbols

The symbols used in this document are summarized in the [Table of Symbols](#) along with their units. Throughout the document, symbols in bold will represent vectors, and scalars otherwise. The symbols are listed in alphabetical order. For vector quantities, the units shown are for each component of the vector.

Table 2: Table of Symbols

Symbol	Description	Units
a_{x1}	X-acceleration of the first star	$\frac{m}{s^2}$
a_{x2}	X-acceleration of the second star	$\frac{m}{s^2}$
a_{y1}	Y-acceleration of the first star	$\frac{m}{s^2}$
a_{y2}	Y-acceleration of the second star	$\frac{m}{s^2}$
$\mathbf{a}(t)$	Acceleration	$\frac{m}{s^2}$
E	Energy	J
\mathbf{F}	Force	N
\mathbf{F}_g	Force of gravity	N
G	Gravitational constant	$\frac{m^3}{kgs^2}$

Continued on next page

Table 2: Table of Symbols (Continued)

Symbol	Description	Units
i	Index	–
m	Mass	kg
m_1	Mass of the first star	kg
m_2	Mass of the second star	kg
m_{\max}	Maximum stellar mass	kg
m_{\min}	Minimum stellar mass	kg
n	Number of bodies	–
$\mathbf{p}(t)$	Position	m
r_{12}	Separation distance	m
r_{\max}	Maximum initial distance from origin	m
t	Time	s
t_{final}	Final time	s
t_{\max}	Maximum simulation time	s
v_{\max}	Maximum initial speed	$\frac{\text{m}}{\text{s}}$
v_{x1}	X-velocity of the first star	$\frac{\text{m}}{\text{s}}$
v_{x1}^0	Initial x-velocity of the first star	$\frac{\text{m}}{\text{s}}$
v_{x2}	X-velocity of the second star	$\frac{\text{m}}{\text{s}}$
v_{x2}^0	Initial x-velocity of the second star	$\frac{\text{m}}{\text{s}}$
v_{y1}	Y-velocity of the first star	$\frac{\text{m}}{\text{s}}$
v_{y1}^0	Initial y-velocity of the first star	$\frac{\text{m}}{\text{s}}$
v_{y2}	Y-velocity of the second star	$\frac{\text{m}}{\text{s}}$
v_{y2}^0	Initial y-velocity of the second star	$\frac{\text{m}}{\text{s}}$
$\mathbf{v}(t)$	Velocity	$\frac{\text{m}}{\text{s}}$
x_1	X-position of the first star	m
x_1^0	Initial x-position of the first star	m
x_2	X-position of the second star	m
x_2^0	Initial x-position of the second star	m
y_1	Y-position of the first star	m
y_1^0	Initial y-position of the first star	m
y_2	Y-position of the second star	m

Continued on next page

Table 2: Table of Symbols (Continued)

Symbol	Description	Units
y_2^0	Initial y-position of the second star	m

1.3 Abbreviations and Acronyms

Table 3: Abbreviations and Acronyms

Abbreviation	Full Form
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
ODE	Ordinary Differential Equation
PS	Physical System Description
R	Requirement
RefBy	Referenced by
Refname	Reference Name
SRS	Software Requirements Specification
TM	Theoretical Model
UC	Unlikely Change
Uncert.	Typical Uncertainty

2 Introduction

Binary star systems are common in astronomy. Two stars orbit because of gravity. This software simulates how a binary star system evolves over time.

The following section provides an overview of the software requirements specification (SRS) for Binary Star System Simulator. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The primary purpose of this document is to record the requirements of BSS. Goals, assumptions, theoretical models, definitions, and other model derivation information are specified, allowing the reader to fully understand and verify the purpose and scientific basis of BSS. With the exception of **system constraints**, this SRS will remain abstract, describing what problem is being solved, but not how to solve it.

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 [5], 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 two-dimensional, rigid-body mechanics where the only force is gravitational attraction.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have an understanding of undergraduate level 1 physics (Newtonian mechanics), undergraduate level 1 calculus, and ordinary differential equations. The users of BSS can have a lower level of expertise, as explained in **Sec:User Characteristics**.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [4], [7], [8], and [6]. 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** 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. It identifies the interfaces between the system and its environment, describes the user characteristics, and lists the

system constraints.

3.1 System Context

Fig:sysCtxDiag shows the system context. A circle represents an entity external to the software, the user in this case. A rectangle represents the software system itself (BSS). Arrows are used to show the data flow between the system and its environment.

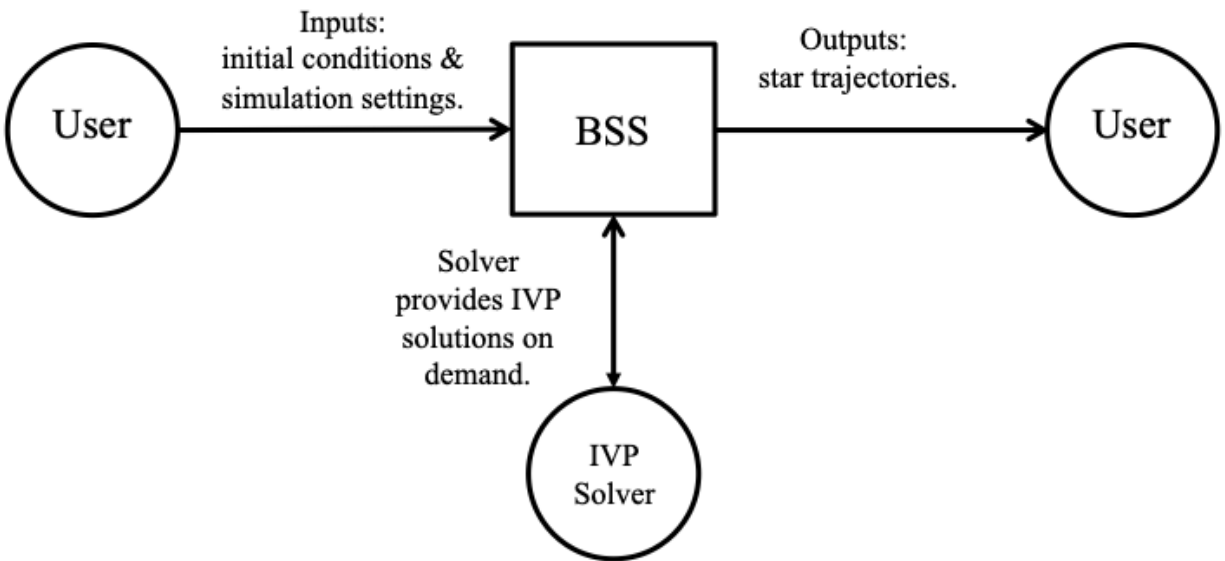


Figure 1: System Context

The interaction between the product and the user is through an application programming interface. The responsibilities of the user and the system are as follows:

- User Responsibilities
 - Provide the physical parameters of the binary star system, including stellar masses, initial positions, and initial velocities, ensuring no errors in the data entry.

- Ensure that consistent units are used for inputs.
- Ensure required **software assumptions** are appropriate for any particular problem input to the software.
- BSS Responsibilities
 - Detect data type mismatch, such as a string of characters input instead of a floating point number.
 - Determine if the inputs satisfy the required physical and software constraints.
 - Calculate the required outputs.

3.2 User Characteristics

The end user of BSS should have an understanding of undergraduate level 1 physics (Newtonian mechanics), undergraduate level 1 calculus and ordinary differential equations.

3.3 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.

4.1 Problem Description

A system is needed to simulate the motion of a binary star system under mutual gravitational interaction.

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.

- Binary star system: A system consisting of two stars that orbit around their common center of mass due to gravitational interaction.
- Star: A massive astronomical object that is treated as a point mass in this context.

- Gravitational interaction: The mutual attractive force between two masses as described by Newtonian gravity.
- Newton’s law of universal gravitation: The law stating that every mass attracts every other mass with a force proportional to the product of their masses and inversely proportional to the square of the distance between them.
- Initial conditions: The positions and velocities of the stars at the start of the simulation.
- Trajectory: The path traced by a star in space as a function of time.
- Center of mass: The point representing the average position of the mass distribution of the system.
- Inertial reference frame: A reference frame in which Newton’s laws of motion are valid without the introduction of fictitious forces.
- Simulation time span: The duration over which the evolution of the system is computed.

4.1.2 Physical System Description

The physical system of BSS, as shown in [Fig:bssPhysSys](#), includes the following elements:

PS1: The first star with mass m_1 .

PS2: The second star with mass m_2 .

PS3: The gravitational interaction between the two stars.

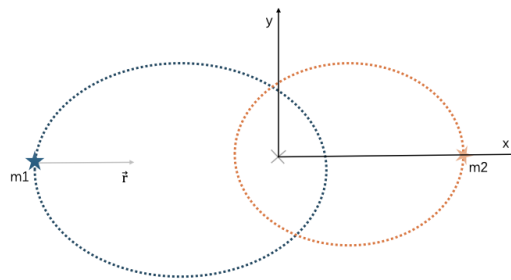


Figure 2: The physical system

4.1.3 Goal Statements

Given the masses (m_1 and m_2), the initial positions ($x_1^0, y_1^0, x_2^0, y_2^0$), the initial velocities ($v_{x1}^0, v_{y1}^0, v_{x2}^0, v_{y2}^0$), and the simulation time (t_{final}), the goal statement is:

positionsOverTime: Determine the positions of both stars as functions of time over the specified simulation interval.

4.2 Solution Characteristics Specification

The instance models that govern BSS are presented in the [Instance Model Section](#). 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 models by filling in the missing information for the physical system. The assumptions refine the scope by providing more detail.

twoBody: The system consists of exactly two stars with masses m_1 and m_2 . Perturbations from third-body gravity are ignored. (RefBy: [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

isolated: Non-gravitational forces (e.g., drag, thrust, radiation pressure) are neglected; the only interaction modeled is mutual gravity between the two stars (from [A:newtonian-Gravity](#)). (RefBy: [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

newtonianGravity: The gravitational interaction between the stars is modeled using Newton's law of universal gravitation, with the gravitational constant G provided in [Values of Auxiliary Constants](#). (RefBy: [TM:UniversalGravLaw](#), [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), [UC:Newtonian-Gravity-Model](#), [A:nonzeroSeparation](#), [LC:Derived-Output-Quantities](#), and [A:isolated](#).)

nonRelativistic: The motion is modeled using classical (non-relativistic) mechanics; relativistic effects are neglected. (RefBy: [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

pointMass: Each star (m_1, m_2) is modeled as a point mass, and effects due to stellar size, deformation, or rotation are neglected. (RefBy: [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), [A:nonzeroSeparation](#), and [LC:Derived-Output-Quantities](#).)

constantMass: The masses m_1 and m_2 remain constant over time. (RefBy: [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

inertialFrame: The simulation is performed in an inertial reference frame. (RefBy: [TM:centerOfMass](#), [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

planar: The motion of the binary star system is confined to a two-dimensional plane. (RefBy: [TM:relPosAndSep](#), [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

nonzeroSeparation: Collisions are out of scope: the separation distance r_{12} is positive for all simulated times, so the gravitational force model remains well-defined (from [A:newtonianGravity](#) and [A:pointMass](#)). (RefBy: [TM:UniversalGravLaw](#), [IM:accelY2](#), [IM:accelY1](#), [IM:accelX2](#), [IM:accelX1](#), and [LC:Derived-Output-Quantities](#).)

4.2.2 Theoretical Models

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

Refname	TM:centerOfMass
Label	Center-of-mass constraint
Equation	$\sum_{i=1}^n m_i \mathbf{p}(t)_i = 0$ $m_1 x_1^0 + m_2 x_2^0 = 0$ $m_1 y_1^0 + m_2 y_2^0 = 0$
Description	<p> m is the mass (kg) i is the index (Unitless) $\mathbf{p}(t)$ is the position (m) m_1 is the mass of the first star (kg) x_1^0 is the initial x-position of the first star (m) m_2 is the mass of the second star (kg) x_2^0 is the initial x-position of the second star (m) m_1 is the mass of the first star (kg) y_1^0 is the initial y-position of the first star (m) m_2 is the mass of the second star (kg) y_2^0 is the initial y-position of the second star (m) </p>
Notes	<p>The first equation gives the general center-of-mass constraint for a system of n point masses, where i indexes each body. In the center-of-mass reference frame (A:inertialFrame), the origin is chosen so that the mass-weighted sum of positions vanishes. The remaining equations specialize to the binary case ($n = 2$), decomposing the vector constraint into x-direction and y-direction components of the initial positions.</p>
Source	–
RefBy	IM:accelY2 , IM:accelY1 , IM:accelX2 , and IM:accelX1

Refname	TM:NewtonSecLawMot
Label	Newton's second law of motion
Equation	$\mathbf{F} = m \mathbf{a}(t)$
Description	<p>\mathbf{F} is the force (N) m is the mass (kg) $\mathbf{a}(t)$ is the acceleration ($\frac{\text{m}}{\text{s}^2}$)</p>
Notes	The net force \mathbf{F} on a body is proportional to the acceleration $\mathbf{a}(t)$ of the body, where m denotes the mass of the body as the constant of proportionality.
Source	[3]
RefBy	IM:accelY2, IM:accelY1, IM:accelX2, and IM:accelX1
Refname	TM:velocity
Label	Velocity
Equation	$\mathbf{v}(t) = \frac{d\mathbf{p}(t)}{dt}$
Description	<p>$\mathbf{v}(t)$ is the velocity ($\frac{\text{m}}{\text{s}}$) t is the time (s) $\mathbf{p}(t)$ is the position (m)</p>
Source	[2]
RefBy	IM:accelY2, IM:accelY1, IM:accelX2, and IM:accelX1

Refname	TM:acceleration
Label	Acceleration
Equation	$\mathbf{a}(t) = \frac{d\mathbf{v}(t)}{dt}$
Description	<p>$\mathbf{a}(t)$ is the acceleration ($\frac{\text{m}}{\text{s}^2}$) t is the time (s) $\mathbf{v}(t)$ is the velocity ($\frac{\text{m}}{\text{s}}$)</p>
Source	[1]
RefBy	IM:accelY2, IM:accelY1, IM:accelX2, and IM:accelX1

Refname	TM:UniversalGravLaw
Label	Newton's law of universal gravitation
Equation	$\mathbf{F}_g = G \frac{m_1 m_2}{r_{12}^2}$
Description	<p>\mathbf{F}_g is the force of gravity (N) G is the gravitational constant ($\frac{\text{m}^3}{\text{kg}\text{s}^2}$) m_1 is the mass of the first star (kg) m_2 is the mass of the second star (kg) r_{12} is the separation distance (m)</p>
Notes	<p>\mathbf{F}_g is the magnitude of the gravitational force between two bodies; it is proportional to the product of their masses and inversely proportional to the square of the distance between them. The force acts along the relative-position vector (TM:relPosAndSep), directed from one body toward the other. This assumes Newtonian gravitation (A:newtonianGravity) and requires the separation distance to be positive (A:nonzeroSeparation).</p>
Source	–
RefBy	IM:accelY2 , IM:accelY1 , IM:accelX2 , and IM:accelX1

Refname	TM:relPosAndSep
Label	Relative position and separation
Equation	$r_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
Description	<p> r_{12} is the separation distance (m) x_1 is the x-position of the first star (m) x_2 is the x-position of the second star (m) y_1 is the y-position of the first star (m) y_2 is the y-position of the second star (m) </p>
Notes	<p>The relative position vector is defined as the difference of the two star positions (i.e., $r_{12} = r_1 - r_2$). r_{12} is the corresponding separation distance (the magnitude of the relative position vector), computed from the position coordinates x_1, y_1, x_2, y_2. The motion is confined to a 2D plane (A:planar).</p>
Source	–
RefBy	TM:UniversalGravLaw , IM:accelY2 , IM:accelY1 , IM:accelX2 , and IM:accelX1

4.2.3 General Definitions

There are no general definitions.

4.2.4 Data Definitions

There are no data definitions.

4.2.5 Instance Models

This section transforms the problem defined in the **problem description** into one which is expressed in mathematical terms. It uses concrete symbols defined in the **data definitions** to replace the abstract symbols in the models identified in **theoretical models** and **general definitions**.

Refname	IM:accelX1	
Label	X-acceleration of the first star	
Input	$m_1, m_2, x_1, y_1, x_2, y_2$	
Output	a_{x1}	
Input Constraints	$m_1 > 0$ $m_2 > 0$	
Output Constraints		
Equation	$a_{x1}(x_1, y_1, x_2, y_2) = \frac{-G m_2 (x_1 - x_2)}{r_{12}^3}$	
Description	<p> a_{x1} is the x-acceleration of the first star ($\frac{m}{s^2}$) x_1 is the x-position of the first star (m) y_1 is the y-position of the first star (m) x_2 is the x-position of the second star (m) y_2 is the y-position of the second star (m) G is the gravitational constant ($\frac{m^3}{kgs^2}$) m_2 is the mass of the second star (kg) r_{12} is the separation distance (m) </p>	
Notes	<p> a_{x1} is calculated by solving the ODE together with the initial conditions . This model is derived from Newton's second law (TM:NewtonSecLawMot), universal gravitation (TM:UniversalGravLaw), and the definitions of acceleration (TM:acceleration), velocity (TM:velocity) and relative position (TM:relPosAndSep), under the center-of-mass constraint (TM:centerOfMass) . The assumptions are: two-body system (A:twoBody), isolated system (A:isolated), Newtonian gravitation (A:newtonianGravity), non-relativistic mechanics (A:nonRelativistic), point masses (A:pointMass), constant masses (A:constantMass), inertial reference frame (A:inertialFrame), planar motion (A:planar) and non-zero separation (A:nonzeroSeparation). </p>	
Source	-	
RefBy	FR:Output-Values and FR:Calculate-Positions	

Refname	IM:accelY1	
Label	Y-acceleration of the first star	
Input	$m_1, m_2, x_1, y_1, x_2, y_2$	
Output	a_{y1}	
Input Constraints	$m_1 > 0$ $m_2 > 0$	
Output Constraints		
Equation	$a_{y1}(x_1, y_1, x_2, y_2) = \frac{-G m_2 (y_1 - y_2)}{r_{12}^3}$	
Description	<p> a_{y1} is the y-acceleration of the first star ($\frac{\text{m}}{\text{s}^2}$) x_1 is the x-position of the first star (m) y_1 is the y-position of the first star (m) x_2 is the x-position of the second star (m) y_2 is the y-position of the second star (m) G is the gravitational constant ($\frac{\text{m}^3}{\text{kg s}^2}$) m_2 is the mass of the second star (kg) r_{12} is the separation distance (m) </p>	
Notes	<p> a_{y1} is calculated by solving the ODE together with the initial conditions . This model is derived from Newton's second law (TM:NewtonSecLawMot), universal gravitation (TM:UniversalGravLaw), and the definitions of acceleration (TM:acceleration), velocity (TM:velocity) and relative position (TM:relPosAndSep), under the center-of-mass constraint (TM:centerOfMass) . The assumptions are: two-body system (A:twoBody), isolated system (A:isolated), Newtonian gravitation (A:newtonianGravity), non-relativistic mechanics (A:nonRelativistic), point masses (A:pointMass), constant masses (A:constantMass), inertial reference frame (A:inertialFrame), planar motion (A:planar) and non-zero separation (A:nonzeroSeparation). </p>	
Source	-	
RefBy	FR:Output-Values and FR:Calculate-Positions	

Refname	IM:accelX2	
Label	X-acceleration of the second star	
Input	$m_1, m_2, x_1, y_1, x_2, y_2$	
Output	a_{x2}	
Input Constraints	$m_1 > 0$ $m_2 > 0$	
Output Constraints		
Equation	$a_{x2}(x_1, y_1, x_2, y_2) = \frac{G m_1 (x_1 - x_2)}{r_{12}^3}$	
Description	<p> a_{x2} is the x-acceleration of the second star ($\frac{m}{s^2}$) x_1 is the x-position of the first star (m) y_1 is the y-position of the first star (m) x_2 is the x-position of the second star (m) y_2 is the y-position of the second star (m) G is the gravitational constant ($\frac{m^3}{kgs^2}$) m_1 is the mass of the first star (kg) r_{12} is the separation distance (m) </p>	
Notes	<p> a_{x2} is calculated by solving the ODE together with the initial conditions . This model is derived from Newton's second law (TM:NewtonSecLawMot), universal gravitation (TM:UniversalGravLaw), and the definitions of acceleration (TM:acceleration), velocity (TM:velocity) and relative position (TM:relPosAndSep), under the center-of-mass constraint (TM:centerOfMass) . The assumptions are: two-body system (A:twoBody), isolated system (A:isolated), Newtonian gravitation (A:newtonianGravity), non-relativistic mechanics (A:nonRelativistic), point masses (A:pointMass), constant masses (A:constantMass), inertial reference frame (A:inertialFrame), planar motion (A:planar) and non-zero separation (A:nonzeroSeparation). </p>	
Source	-	
RefBy	FR:Output-Values and FR:Calculate-Positions	

Refname	IM:accelY2	
Label	Y-acceleration of the second star	
Input	$m_1, m_2, x_1, y_1, x_2, y_2$	
Output	a_{y2}	
Input Constraints	$m_1 > 0$ $m_2 > 0$	
Output Constraints		
Equation	$a_{y2}(x_1, y_1, x_2, y_2) = \frac{G m_1 (y_1 - y_2)}{r_{12}^3}$	
Description	<p> a_{y2} is the y-acceleration of the second star ($\frac{\text{m}}{\text{s}^2}$) x_1 is the x-position of the first star (m) y_1 is the y-position of the first star (m) x_2 is the x-position of the second star (m) y_2 is the y-position of the second star (m) G is the gravitational constant ($\frac{\text{m}^3}{\text{kg s}^2}$) m_1 is the mass of the first star (kg) r_{12} is the separation distance (m) </p>	
Notes	<p> a_{y2} is calculated by solving the ODE together with the initial conditions . This model is derived from Newton's second law (TM:NewtonSecLawMot), universal gravitation (TM:UniversalGravLaw), and the definitions of acceleration (TM:acceleration), velocity (TM:velocity) and relative position (TM:relPosAndSep), under the center-of-mass constraint (TM:centerOfMass) . The assumptions are: two-body system (A:twoBody), isolated system (A:isolated), Newtonian gravitation (A:newtonianGravity), non-relativistic mechanics (A:nonRelativistic), point masses (A:pointMass), constant masses (A:constantMass), inertial reference frame (A:inertialFrame), planar motion (A:planar) and non-zero separation (A:nonzeroSeparation). </p>	
Source	-	
RefBy	FR:Output-Values and FR:Calculate-Positions	

4.2.6 Data Constraints

The **Input Data Constraints Table** shows the data constraints on the input variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. 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. 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.

Table 4: Input Data Constraints

Var	Physical Constraints	Software Constraints	Typical Value	RaUncert. tio- nale
m_1	$m_1 > 0$	$m_{\min} \leq m_1 \leq m_{\max}$	$2.0 \cdot 10^{30}$ kg	ap-5.0% prox- i- mately 1 so- lar mass
m_2	$m_2 > 0$	$m_{\min} \leq m_2 \leq m_{\max}$	$1.6 \cdot 10^{30}$ kg	ap-5.0% prox- i- mately 0.8 so- lar masses
t_{final}	$t_{\text{final}} \geq 0$	$t_{\text{final}} \leq t_{\text{max}}$	$100.0 \cdot 10^3$ s	shot0% in- te- gra- tion win- dow

Continued on next page

Table 4: Input Data Constraints (Continued)

Var	Physical Constraints	Software Constraints	Typical Value	RaUncert. tio- nale
v_{x1}^0	–	$-v_{\max} \leq v_{x1}^0 \leq v_{\max}$	$2.0 \cdot 10^3 \frac{\text{m}}{\text{s}}$	non-5.0% zero x- velocity pro- duces a gen- eral el- lip- ti- cal or- bit
v_{x2}^0	–	$-v_{\max} \leq v_{x2}^0 \leq v_{\max}$	$-2.0 \cdot 10^3 \frac{\text{m}}{\text{s}}$	de-5.0% rived from COM con- straint

Continued on next page

Table 4: Input Data Constraints (Continued)

Var	Physical Constraints	Software Constraints	Typical Value	RaUncert. tio- nale
v_{y1}^0	–	$-v_{\max} \leq v_{y1}^0 \leq v_{\max}$	$9.0 \cdot 10^3 \frac{\text{m}}{\text{s}}$	yield 5.0% a bound or- bit for the typ- i- cal masses and sep- a- ra- tion
v_{y2}^0	–	$-v_{\max} \leq v_{y2}^0 \leq v_{\max}$	$-11.0 \cdot 10^3 \frac{\text{m}}{\text{s}}$	de-5.0% rived from COM con- straint

Continued on next page

Table 4: Input Data Constraints (Continued)

Var	Physical Constraints	Software Constraints	Typical Value	RaUncert. tio- nale
x_1^0	–	$-r_{\max} \leq x_1^0 \leq r_{\max}$	$75.0 \cdot 10^9$ m	ap-1.0% prox- i- mately 0.5 AU; sat- is- fies COM con- straint with star 2
x_2^0	–	$-r_{\max} \leq x_2^0 \leq r_{\max}$	$-90.0 \cdot 10^9$ m	de-1.0% rived from COM con- straint
y_1^0	–	$-r_{\max} \leq y_1^0 \leq r_{\max}$	0.0 m	star-1.0% ini- tially on the x- axis
y_2^0	–	$-r_{\max} \leq y_2^0 \leq r_{\max}$	0.0 m	star-1.0% ini- tially on the x- axis

4.2.7 Properties of a Correct Solution

The total mechanical energy of the system must remain constant over time (up to numerical tolerance), since the system is isolated (**A:isolated**) and masses are constant (**A:constant-Mass**).

$$E = \frac{m_1(v_{x1}^2 + v_{y1}^2)}{2} + \frac{m_2(v_{x2}^2 + v_{y2}^2)}{2} - \frac{G m_1 m_2}{r_{12}}$$

5 Requirements

This section provides the functional requirements, the tasks and behaviours 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

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete.

Input-Values: Input the values from **Tab:ReqInputs**.

Verify-Input-Values: Check the entered input values to ensure that they do not exceed the **data constraints**. If any of the input values are out of bounds, an error message is displayed and the computation stops.

Calculate-Positions: Calculate the positions of both stars over the simulation interval by solving **IM:accelX1**, **IM:accelY1**, **IM:accelX2** and **IM:accelY2**.

Verify-Output: Verify that the computed results satisfy the conservation of total mechanical energy within a specified numerical tolerance.

Output-Values: Output x_1 , y_1 , x_2 and y_2 from **IM:accelX1**, **IM:accelY1**, **IM:accelX2** and **IM:accelY2**.

Table 5: Required Inputs

Symbol	Description	Units
m_1	Mass of the first star	kg
m_2	Mass of the second star	kg
t_{final}	Final time	s
v_{x1}^0	Initial x-velocity of the first star	$\frac{\text{m}}{\text{s}}$

Continued on next page

Table 5: Required Inputs (Continued)

Symbol	Description	Units
v_{x2}^0	Initial x-velocity of the second star	$\frac{m}{s}$
v_{y1}^0	Initial y-velocity of the first star	$\frac{m}{s}$
v_{y2}^0	Initial y-velocity of the second star	$\frac{m}{s}$
x_1^0	Initial x-position of the first star	m
x_2^0	Initial x-position of the second star	m
y_1^0	Initial y-position of the first star	m
y_2^0	Initial y-position of the second star	m

5.2 Non-Functional Requirements

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

Correctness: The outputs of the code have the [properties of a correct solution](#).

Portability: The code shall be portable to multiple environments, particularly Linux, Mac OSX, and Windows.

Maintainability: If a likely change is made to the finished software, it will take at most 10% of the original development time, assuming the same development resources are available.

Usability: The outputs should be easy to inspect and reuse. The software should export results in a consistent format so that users can post-process and visualize trajectories with external tools.

6 Likely Changes

This section lists the likely changes (LC) to be made to the software.

Derived-Output-Quantities: Additional derived outputs, such as total energy or angular momentum, may be added to support validation and analysis of the simulation results. This change could affect results derived under all current assumptions: [A:two-Body](#), [A:isolated](#), [A:newtonianGravity](#), [A:nonRelativistic](#), [A:pointMass](#), [A:constant-Mass](#), [A:inertialFrame](#), [A:planar](#) and [A:nonzeroSeparation](#).

7 Unlikely Changes

This section lists the unlikely changes (UC) to be made to the software.

Newtonian-Gravity-Model: The gravitational interaction model ([A:newtonianGravity](#)) is unlikely to change, since BSS is specifically designed for Newtonian two-body dynamics.

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. [Tab:TraceMatAvsA](#) shows the dependencies of the assumptions on each other. [Tab:TraceMatAvsAll](#) shows the dependencies of the data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. [Tab:TraceMatRefvsRef](#) shows the dependencies of the data definitions, theoretical models, general definitions, and instance models on each other. [Tab:TraceMatAllvsR](#) shows the dependencies of the requirements and goal statements on the data definitions, theoretical models, general definitions, and instance models.

Table 6: Traceability Matrix Showing the Connections Between

	A:twoBody	A:isolated	A:newtonianGravity	A:nonRelativistic	A:pointMass
A:twoBody					
A:isolated			X		
A:newtonianGravity					
A:nonRelativistic					
A:pointMass					
A:constantMass					
A:inertialFrame					
A:planar					
A:nonzeroSeparation			X		X

Table 7: Traceability Matrix Showing the Connections Between

	A:twoBody	A:isolated	A:newtonianGravity	A:nonRelativistic	A:pointMass
TM:centerOfMass					

Table 7: Traceability Matrix Showing the Connections I

	A:twoBody	A:isolated	A:newtonianGravity	A:nonRelativis
TM:NewtonSecLawMot				
TM:velocity				
TM:acceleration				
TM:UniversalGravLaw			X	
TM:relPosAndSep				
IM:accelX1	X	X	X	X
IM:accelY1	X	X	X	X
IM:accelX2	X	X	X	X
IM:accelY2	X	X	X	X
FR:Input-Values				
FR:Verify-Input-Values				
FR:Calculate-Positions				
FR:Verify-Output				
FR:Output-Values				
NFR:Correctness				
NFR:Portability				
NFR:Maintainability				
NFR:Usability				
LC:Derived-Output-Quantities	X	X	X	X
UC:Newtonian-Gravity-Model			X	

Table 8: Traceability Matrix Showing

	TM:centerOfMass	TM:NewtonSecLawMot	TM:velocity	TM:acce
TM:centerOfMass				
TM:NewtonSecLawMot				
TM:velocity				
TM:acceleration				
TM:UniversalGravLaw				
TM:relPosAndSep				

Table 8: Traceability Matrix Showing the

	TM:centerOfMass	TM:NewtonSecLawMot	TM:velocity	TM:acce
IM:accelX1	X	X	X	X
IM:accelY1	X	X	X	X
IM:accelX2	X	X	X	X
IM:accelY2	X	X	X	X

	TM:centerOfMass	TM:NewtonSecLawMot	TM:velocity	TM:acce
GS:positionsOverTime				
FR:Input-Values				
FR:Verify-Input-Values				
FR:Calculate-Positions				
FR:Verify-Output				
FR:Output-Values				
NFR:Correctness				
NFR:Portability				
NFR:Maintainability				
NFR:Usability				

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. [Fig:TraceGraphAvsA](#) shows the dependencies of assumptions on each other. [Fig:TraceGraphAvsAll](#) shows the dependencies of data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. [Fig:TraceGraphRefsRef](#) shows the dependencies of data definitions, theoretical models, general definitions, and instance models on each other. [Fig:TraceGraphAllvsR](#) shows the dependencies of requirements and goal statements on the data definitions, theoretical models, general definitions, and instance models. [Fig:TraceGraphAllvsAll](#) shows the dependencies of dependencies of assumptions, models, definitions, requirements, goals, and changes with each other.

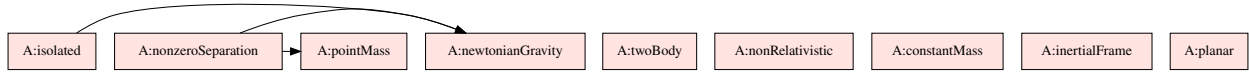


Figure 3: TraceGraphAvsA



Figure 4: TraceGraphAvsAll

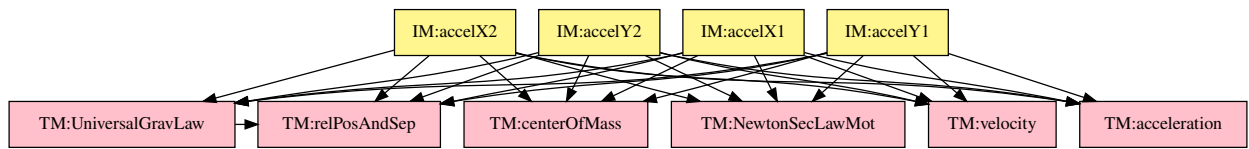


Figure 5: TraceGraphRefsRef



Figure 6: TraceGraphAllvsR

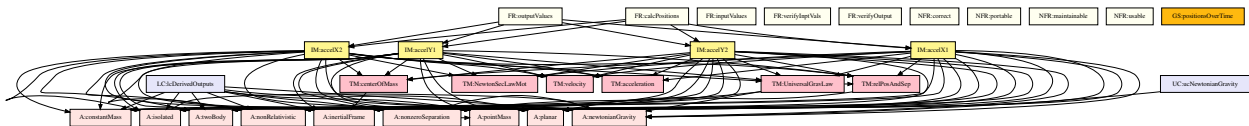


Figure 7: TraceGraphAllvsAll

For convenience, the following graphs can be found at the links below:

- [TraceGraphAvsA](#)
- [TraceGraphAvsAll](#)
- [TraceGraphRefsRef](#)
- [TraceGraphAllvsR](#)
- [TraceGraphAllvsAll](#)

9 Values of Auxiliary Constants

This section contains the standard values that are used for calculations in BSS.

Table 10: Auxiliary Constants

Symbol	Description	Value	Unit
G	gravitational constant	$66.743 \cdot 10^{-12}$	$\frac{\text{m}^3}{\text{kg}\text{s}^2}$
m_{max}	maximum stellar mass	$100.0 \cdot 10^{30}$	kg
m_{min}	minimum stellar mass	$100.0 \cdot 10^{27}$	kg
r_{max}	maximum initial distance from origin	$10.0 \cdot 10^{12}$	m
t_{max}	maximum simulation time	$10.0 \cdot 10^9$	s
v_{max}	maximum initial speed	$1.0 \cdot 10^6$	$\frac{\text{m}}{\text{s}}$

10 References

- [1] Wikipedia Contributors. *Acceleration*. <https://en.wikipedia.org/wiki/Acceleration>. June 2019.
- [2] Wikipedia Contributors. *Velocity*. <https://en.wikipedia.org/wiki/Velocity>. June 2019.
- [3] R. C. Hibbeler. *Engineering Mechanics: Dynamics*. Pearson Prentice Hall, 2004.
- [4] Nirmitha Koothoor. “A Document Driven Approach to Certifying Scientific Computing Software”. MA thesis. Hamilton, ON, Canada: McMaster University, 2013.
- [5] David L. Parnas and P. C. Clements. “A rational design process: How and why to fake it”. In: *IEEE Transactions on Software Engineering* 12.2 (Feb. 1986), pp. 251–257.
- [6] W. Spencer Smith and Nirmitha Koothoor. “A Document-Driven Method for Certifying Scientific Computing Software for Use in Nuclear Safety Analysis”. In: *Nuclear Engineering and Technology* 48.2 (Apr. 2016), pp. 404–418.
- [7] W. Spencer Smith and Lei Lai. “A new requirements template for scientific computing”. In: *Proceedings of the First International Workshop on Situational Requirements Engineering Processes - Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP’05*. Ed. by PJ Agerfalk, N. Kraiem, and J. Ralyte. In conjunction with 13th IEEE International Requirements Engineering Conference, Paris, France, 2005, pp. 107–121.
- [8] W. Spencer Smith, Lei Lai, and Ridha Khedri. “Requirements Analysis for Engineering Computation: A Systematic Approach for Improving Software Reliability”. In: *Reliable Computing, Special Issue on Reliable Engineering Computation* 13.1 (Feb. 2007), pp. 83–107.