Software Requirements Specification for a Library of ODE Solvers (LODES)

Paul Aoanan

September 23, 2017

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

There are no physical units used in this SRS.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
A_C	m^2	coil surface area
A_{in}	m^2	surface area over which heat is transferred in

[Use your problems actual symbols. The si package is a good idea to use for units. —SS]

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
LODES	Library of ODE Solvers
T	Theoretical Model

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

[Add any other abbreviations or acronyms that you add —SS]

2 Introduction

In physical sciences, mathematical models are derived from scientific models to represent a real world phenomenon through formal mathematical constructs.

Scientific models in the study of radioactivity, carbon decay, and Newton's Law of Cooling involve the use of ordinary differential equations (ODEs).

Known elementary techniques of solving ODEs in the discrete domain use the linear approximation method wherein the solution is based upon assuming or "approximating" the slope of the tangent line from one reference point to the next until the target point has been reached.

The following section provides an overview of the Software Requirements Specification (SRS) for a program family of ODE solvers. The developed program will be called Library of ODE Solvers (LODES). This section explains the purpose of this document, the scope of the system, and the characteristics of the intended readers and the organization of the document.

2.1 Purpose of Document

The main purpose of this document is to formally describe the known well-known methods of solving ODEs. The goals and mathematical models used in the LODES code are provided with an emphasis on explicitly identifying assumptions, constraints, and unambiguous definitions.

This document contains the required functionality of the LODES software library as well the non-functional requirements that the software has to meet. This document governs the subsequent software development activities, including writing the design specification, code, and the software verification and validation plan and execution.

2.2 Scope of Requirements

The scope of requirements is limited to the analysis of the library of ODE solvers. Given the appropriate inputs, each program in LODES is intended to find the solution to an ODE problem.

2.3 Characteristics of Intended Reader

Reviewers of this document should have an elementary understanding of ordinary differential equations and numerical methods, as typically covered in first and second year Calculus courses. The users of LODES can have a lower level expertise, as explained in Section 3.2.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by !!!INSERT CITATION HERE!!!. 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 instance models provide the methods to solve Ordinary Differential Equations (ODEs).

3 General System Description

This section identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 Potential System Contexts

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

Programs in LODES are used inside a wrapper program. The external interaction is through program calls. The solution to the ODE is the output of the function. The responsibilities of the user and the system are as follows:

- User Responsibilities:
  - Provide the correct program call, while adhering to conventions of the program's prototype
  - Provide the input details of the ODE to be solved, ensuring no errors in data entry
  - Declaration of the ODE method to be used in solving the ODE
- LODES Responsibilities:
  - Detect an improper input, such as invalid characters in the ODE statement and incomplete input arguments
  - Detect a data type mismatch where applicable, such as a string of characters in a floating point argument
  - Calculate the solution to the ODE problem

3.2 Potential User Characteristics

The end user of LODES should have an understanding of undergraduate Level 1 Calculus.

3.3 Potential System Constraints

There are no system constraints.

4 Commonalities

This section first presents the background and motivation of the program family, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the theories, definitions, assumptions, and finally the instance models as variabilities.

4.1 Background Overview

LODES is a software library developed to be provide a means to solve ODE problems using numerical methods. It can be used to solve different variations of ODEs given their initial values. It can be implemented to find the most accurate method (the method which produces the least error in their scientific computing implementation).

4.2 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:

- Initial values: A "starting point" (x_0, y_0) of known values that exists in the domain of the solution
- Final values: An "ending point" $(x_\mathrm{k}, y_\mathrm{k})$ of unknown value y_k that exists in the domain of the solution
- Step size: The measure of arbitrary positive value (h) from the starting point (x_0, y_0) of the domain to the next $(x_0 + h, y_1)$
- Derivative: The amount by which a a function changes at any given point as an instantaneous rate of change
- Numerical Analysis: A branch of mathematics and computer science wherein the solutions are numerical approximations taking into account the errors involved in the process.
- Numerical Approximation:

4.3 Goal Statements

GS1: Given an ordinary differential equation (ODE) represented by $dy/dx = f(x, y)$, the set of initial values x_0 and y_0 that satisfy $y(x_0) = y_0$, and x_k, return y_k such that $y(x_\mathrm{k}) = y_\mathrm{k}$ (the final values), where $y(x)$ is a function, $f(x, y)$ is a function, and x is an independent variable.

4.4 Theoretical Models

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

Number	T1
Label	**Ordinary Differential Equation**
Equation	$\frac{dy}{dx} = y' = f(x, y)$
Description	The above model gives the definition of an ordinary differential equation. A differential equation is an equation, where the unknown is a function and both the function and its derivatives (rate of change) appear in the equation. An ordinary differential equation is a differential equation involving only ordinary derivatives with respect to a single independent variable. For an arbitrary ODE, the true solution will, in general, be unknown. Numerical methods are used to find numerical approximations of the solution to the ODE.
Source	http://users.math.msu.edu/users/gnagy/teaching/ode.pdf
Ref. By	T2

Number	T2
Label	**Existence and Uniqueness of the Solution**
Equations	$\frac{dy}{dx} = f(x, y)$ [T1] $y(x_0) = y_0$ Assuming f and df/dy are continuous in $R = \{(x, y) : a < x < b, c < y < d\}$, where R is a rectangle and a, b, c, d are its vertices The initial value problem has a unique solution in some interval $x_0 - h < x < x_0 + h$
Description	The above theoretical model shows that when an equation satisfies the initial values, it is assured that a solution to the initial value problem exists. It is desirable to know whether or not the equation has an existing solution before effort is made to solve it. As well, the theoretical model states that if a solution is found, then it is the only solution to the initial value problem.
Source	Nagle, et al, "Solutions and Initial Value Problems," in *Fundamentals of Differential Equations and Boundary Value Problems, 3rd ed. USA: Addison Wesley Longman, 2000, ch. 1, p. 12.*
Ref. By	T??

5 Variabilities

This section presents the variabilities in LODES. It gives the assumptions for the input, the variabilities in the calculations, and finally the target output.

5.1 Input Assumptions

This section focuses on the variabilities and assumptions in the inputs of LODES.

Input Variability	Parameter of Variation
Allowed program family calls	{Euler's Method, Heun's Method, Fourth Order Taylor Series, Fourth Order Runge-Kutta}
Allowed order of $\frac{dy}{dx}$	{First}
Allowed dimensions of $\frac{dy}{dx}$	set of $\mathbb{N}$
Allowed values of h	set of positive $\mathbb{R}$
Allowed values of x_0	the set of $\mathbb{R}$
Allowed values of x_k	the set of $\mathbb{R}$
Allowed values of y_0	the set of $\mathbb{R}$

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —SS]

6 Likely Changes

LC1: [Give the likely changes, with a reference to the related assumption (aref), as appropriate. —SS]

7 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" may have to be modified as well. Table 2 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 3 shows the dependencies of instance models, requirements, and data constraints on each other. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —SS]

	T??	T??	T??	GD??	GD??	DD??	DD??	DD??	DD??	IM??	IM??	IM??	IM
T??													
T??			X										
T??													
GD??													
GD??	X												
DD??				X									
DD??				X									
DD??													
DD??								X					
IM??					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 2: Traceability Matrix Showing the Connections Between Items of Different Sections

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. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure **??** shows the dependencies of instance models, requirements, and data constraints on each other.

	IM??	IM??	IM??	IM??	??	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R??			X	X			
R??		X					
R??		X					

Table 3: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD??		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 4: Traceability Matrix Showing the Connections Between Assumptions and Other Items

8 Appendix

[Your report may require an appendix. For instance, this is a good point to show the values of the symbolic parameters introduced in the report. —SS]

8.1 Symbolic Parameters

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —SS]