# RKF45 Generator

Alexander Schaap

October 2, 2017

# 1 Revision History

| Date | Version | Notes |
|---|---|---|
| October 2, 2017 | 1.0 | Initial version |

# 2 Reference Material

This section records information for easy reference.

## 2.1 Table of Units

Since the application of this software is dependent on the program using the functions provided by this program family, no physical units are used throughout this document.

## 2.2 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 |
| $A_{\text{in}}$ | $m^2$ | 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]

## 2.3 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 |
| rkf45.ml | Family of programs based on the RK4 / RKF45 method(s) |
| T | Theoretical Model |

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

# Contents

# 3 Introduction

This document provides an overview of the commonality analysis (CA) for the rkf45.ml program family. This program family calculates approximations in the form of splines to given ordinary differential equations (ODEs) using a fourth-order Runge-Kutta / Runge-Kutta-Fehlberg method. It does most of this at compile-time, generating a different family member for each given combination of ODE, interval, step size, and initial values. The current section describes the purpose of this document, the scope of this family, the organization of the remainder of the document and the characteristics of the intended reader.

## 3.1 Purpose of Document

The main purpose of this document is to provide sufficient information to understand what rkf45.ml does. The goals and theoretical models used in the rkf45.ml implementations are provided, as are assumptions and unambiguous definitions.

## 3.2 Scope of the Family

## 3.3 Characteristics of Intended Reader

The reader is expected to have some undergraduate STEM background. Ideally, they have been exposed to some calculus and programming courses.

## 3.4 Organization of Document

# 4 General System Description

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

## 4.1 Potential System Contexts

[Your system context will likely include an explicit list of user and system responsibilities —SS]

- User Responsibilities:
  - Provide a non-stiff continuous ODE to rkf45.ml for which an accurate solution can be obtained using RK4 (see T2)
  - Correctly process the resulting interpolated spline function
- rkf45.ml Responsibilities:

– Generate (in a type-safe manner) a family member which contains a spline that approximates the given ODE.

–

### 4.2 Potential User Characteristics

The most common user of rkf45.ml will be other programs. However, one or more programmers are needed to write the code that calls the function(s) provided by this family.

These programmers therefore should have an understanding of undergraduate Level 1 Calculus.

### 4.3 Potential System Constraints

The responsibility of generating family members in a type-safe way restricts the system to the MetaOCaml extension of the OCaml programming language.

## 5 Commonalities

### 5.1 Background Overview

There are various numerical methods for approximating ordinary differential equation (ODE), such as the Runge-Kutta methods (which include Euler's method).

### 5.2 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:

•

### 5.3 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —SS]

| | |
|---|---|
| Number | DD1 |
| Label | **Ordinary Differential Equation (ODE)** |
| Symbol | $\mathbf{f}$ |
| Units | $\mathbb{R} \times \mathbb{C}^n \to \mathbb{C}^n$ |
| Equation | $\mathbf{x}' = \mathbf{f}(t, \mathbf{x}(t))$ |
| Description | $\mathbf{f}$ is the equation for which we ultimately want to find a numerical approximation. |
| Sources | **?** |
| Ref. By | IM**??** |

| | |
|---|---|
| Number | DD2 |
| Label | **Initial values** |
| Symbol | $a, b$ |
| Units | $\mathbb{R} \times \mathbb{R}$ |
| Equation | |
| Description | The interval for which to solve the ODE (see DD1). $a$ represents the beginning of the interval, $b$ the end. |
| Sources | **?** |
| Ref. By | IM**??** |

| | |
|---|---|
| Number | DD3 |
| Label | **Interval** |
| Symbol | $\mathbf{x}_0$ |
| Units | $\mathbb{C}^n$ |
| Equation | $\mathbf{x}(t_0) = \mathbf{x}_0$ |
| Description | Initial values for solving ODE (see DD1). |
| Sources | **?** |
| Ref. By | IM**??** |

| | |
|---|---|
| Number | DD4 |
| Label | **Step size** |
| Symbol | $h$ |
| Units | $\mathbb{R}$ |
| Equation | |
| Description | Size of the steps between the opints for which to find approximations using RK4 (T2). |
| Sources | **?** |
| Ref. By | IM**??** |

## 5.4 Goal Statements

Given the non-stiff continuous ODE, the goal statements are:

GS1: Given an interval and the desired number of knots, as well as an initial value, calculate a spline and return a function that uses this spline to solve for specific points on the provided interval.

## 5.5 Theoretical Models

This section focuses on the general equations and laws that rkf45.ml is based on. [Modify the examples below for your problem, and add additional models as appropriate. —SS]

| | |
|---|---|
| Number | T1 |
| Label | **Initial value problem (IVP)** |
| Equation | $\mathbf{x}' = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad$ where<br>• $\mathbf{x} : \mathbb{R} \to \mathbb{C}^n$ is the vector solution as a function of time<br>• $\mathbf{x}_0 \in \mathbb{C}^n$ is the initial condition (see DD2)<br>• $\mathbf{f} : \mathbb{R} \times \mathbb{C}^n \to \mathbb{C}^n$ is the function describing the vector field (see DD1)<br>Given an initial value $\mathbf{x}_0$, the goal is to find approximations on a given interval $[t_a..t_b]$. |
| Description | The standard form of an initial value problem is given above. The issue is that many IVPs are difficult to solve manually (or programmatically) and the correct solutions are often unknown. Numerical methods are close enough to be used in most applications. |
| Source | Corless & Fillion, A Graduate Introduction to Numerical Methods, p. 510,513 [TODO: ref to bib —AS] |
| Ref. By | GD?? |

| Number | T2 |
|---|---|
| Label | **Fourth order Runge-Kutta method (RK4)** |
| Equation | • $t_1 = t_0 + h$<br>• $\mathbf{k}_1 = \mathbf{f}(t_0, \mathbf{x}_0)$ slope at $x_0$<br>• $\mathbf{k}_2 = \mathbf{f}(t_0 + \frac{h}{2}, \mathbf{x}_0 + \frac{h}{2}\mathbf{k}_1)$ slope of the point halfway between $t_0$ and $t_1$ when extrapolating slope $\mathbf{k}_1$ from point $\mathbf{x}_0$<br>• $\mathbf{k}_3 = \mathbf{f}(t_0 + \frac{h}{2}, \mathbf{x}_0 + \frac{h}{2}\mathbf{k}_2)$ slope of the point halfway between $t_0$ and $t_1$ when extrapolating slope $\mathbf{k}_2$ from point $\mathbf{x}_0$<br>• $\mathbf{k}_4 = \mathbf{f}(t_0 + h, \mathbf{x}_0 + h\mathbf{k}_3)$ slope of point at $t_1$ when extrapolating slope $\mathbf{k}_3$ from point $\mathbf{x}_0$<br>• $\mathbf{x}_1 = \mathbf{x}_0 + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$ new point created when extrapolating from point $\mathbf{x}_0$ using a weighted average of the previously calculated slopes |
| Description | The above equations can be used to calculate (an approximation of) a new point given the previous or starting point. |
| Source | Corless & Fillion, A Graduate Introduction to Numerical Methods, p. 618 [TODO: ref to bib —AS] |
| Ref. By | GD?? |

# 6 Variabilities

## 6.1 Assumptions

A1: [Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. —SS]

A2: The given initial values (DD2) are for the beginning of the interval (DD3), represented by $a$.

A3: Initial value vector size is expected to match the ones produced by the ODE (DD1).

A4: The interval's bounds satisfy $a \leq b$.

### 6.2 Calculation

### 6.3 Output

## 7 Traceability Matrices and Graphs

[You will have to add tables. —SS]

# References

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

# 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]