

# Test Report: Runge-Kutta Generator

Alexander Schaap

December 18, 2017

# 1 Revision History

The latest version can be found at <https://github.com/aschaap/cas741>.

Date	Version	Notes
December 18	1.0	Initial (and final) version

## 2 Symbols, Abbreviations and Acronyms

See the Table of Symbols in the SRS at <https://github.com/aschaap/cas741>.

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Introduction</b>	<b>1</b>
<b>4</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
<b>5</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>1</b>
5.1	Usability . . . . .	1
5.2	Performance . . . . .	1
<b>6</b>	<b>Comparison to Existing Implementation</b>	<b>1</b>
<b>7</b>	<b>Unit Testing</b>	<b>2</b>
<b>8</b>	<b>Changes Due to Testing</b>	<b>2</b>
<b>9</b>	<b>Automated Testing</b>	<b>2</b>
<b>10</b>	<b>Trace to Requirements</b>	<b>2</b>
<b>11</b>	<b>Trace to Modules</b>	<b>2</b>
<b>12</b>	<b>Code Coverage Metrics</b>	<b>2</b>

### 3 Introduction

This document will summarize whether the implementation satisfies the requirements implied in the Commonality Analysis available at <https://github.com/aschaap/cas741>.

### 4 Functional Requirements Evaluation

The implementation contains both the RK2 and RK4 methods, and allows users to choose between them. It generates a function that will return numerical approximations for a given RK method, ODE, interval, step size, and initial condition.

### 5 Nonfunctional Requirements Evaluation

Since this is a library, the primary nonfunctional requirement is performance.

#### 5.1 Usability

Usability was not a goal, but the implementation ensures the exposed function interface is minimalistic.

#### 5.2 Performance

Performance evaluation has been moved to phase 2, and will therefore be added later.

### 6 Comparison to Existing Implementation

Comparing the output of the system tests to the same equation in WolframAlpha shows that this implementation is close for slopes that are not too steep. The infinite values returned for the higher order ODE are also accurate, since the function appears undefined at these points on the interval. For the stiff ODE, the results are predictably all over the place.

## **7 Unit Testing**

The OUnit framework available for OCaml (and by extension, MetaOCaml) has been employed to run some unit tests.

## **8 Changes Due to Testing**

N/A.

## **9 Automated Testing**

GNU Make has been employed to both run the unit tests and generate output for the user to manually examine.

## **10 Trace to Requirements**

N/A. There are no explicit requirements, since a Commonality Analysis has been done rather than a requirements specification.

## **11 Trace to Modules**

N/A. This library has been decomposed according to binding times, rather than the traditional modularization. See the Design document available at <https://github.com/aschaap/cas741> for more details and justification for this.

## **12 Code Coverage Metrics**

Due to time constraints, this has also been moved to phase 2.