# Analysis Report

Ordinary Differential Equation Solvers

### Overview

This programming assignment focuses on building different types of solvers for finding the solutions to ordinary differential equations (ODEs). Three methods (Forward Euler, standard Runge-Kutta 34 (RK34) and time-adaptive Runge-Kutta (RK34A)) are used to build the solvers.

### Overall Analysis

#### Code Reuse

The program functions were built in an incremental fashion, with suitable tests built into each relevant file. We maximized code reuse by importing functions from previous programming assignments, as indicated in the comments of the programs.

#### Testing

##### System Functions

The outputs of each system function were compared to Hacker Practice implementations and to additional MATLAB implementations to verify that identical values were obtained, within a certain tolerance limit.

##### Solver Functions

All the solvers were first tested on a simple exponential function for which the ground truth is known. Subsequently, they were implemented on the system equations obtained from the Simple RC circuit and the Amplifier circuit. Since the ground truth for these functions are not readily known, a secondary implementation on MATLAB was used to compare values and verify values.

All files for the secondary MATLAB implementation can be found in a sub-folder in the repo.

#### Ease of Function Calls

Since the solver steps are identical for each system, with the exception of the system function call itself, we decided to build generic solver functions that will be able to take in a void function pointer to any system with the standard function signature given by (phi , current\_values , time , march). Future users will be able to use the solvers for any custom defined system as long as it follows the given function signature.

Furthermore, instead of running a different function call for each type of solver, we created a “parent” ODE Solver function that allows the user to pass in the type of solver and the function to be solved as parameters. For example, when considering the simple RC circuit, and wanting to solve this system’s ODE using the Forward Euler method, the function would look like:

ODE\_Solver(simpleCircuit, slope, current\_values , time , march)

and the simpleCircuit function pointer would be defined as such:

(\*simpleCircuit)(vector\*, vector\*, double , double) = simple\_RC\_circuit

where simple\_RC\_circuit would be the originally defined function which takes in the same type of variables as given in the function pointer definition.