Native Julia Solvers for Ordinary Differential Equations Boundary Value Problem: A GSoC proposal

Yingbo Ma

March 31, 2017

Contents

1 Synopsis

2 The Project

2.1 Basic concepts for nonexperts

An ordinary differential equation (ODE) is an equality relationship between a function $y(x)$ and its derivatives, and an nth order ODE can be written as

$$F(x, y, y', \cdots, y^{(n)}) = 0$$

.

Physics and engineering often raise ODE problems. Many physics phenomenon can be reduced into a ODE. Newton's second law of motion, namely $F = ma$ can rewrite into a ODE as $m\frac{d^2x}{dt^2} = F(x)$, since a can depend on time. ODEs can be really hard to solve. There are many case that a ODE does not have an analytical solution. Therefore, numerical methods need to be used to solve ODEs, by approximating the solution. There are two kinds of problems in ODE. One is initial value problem (IVP) and the other is boundary value problem (BVP). For instance, the ODE

$$m\frac{d^2x}{dt^2} = -kx(t)$$

describes the motion for a harmonic oscillator. If the initial position x_0 and initial velocity $\frac{dx_0}{dt}$ is known, then it is an IVP problem. If the condition at the "boundary" is know, for instance, initial position x_0 and final position x_1, then it is a BVP problem. The solvers that I am going to work on solves BVP problem.

2.1.1 Introduction

The project that I propose to work in Google's Summer of Code project is the native Julia implementation of some BVP solving methods for ODE, namely, collocation method and shooting method.

2.2 Project Goals

2.2.1 Goal 1: Implement BVP related data structure

A data structure to describe the BVP problem, namely, "BVProblem". It contains the information

$$\begin{aligned}&F(x, y, y', \cdots, y^{(n)}) = 0\\&\text{domin: } x \in [a, b]\\&\text{boundary condition: Dirichlet, Neumann, Robin.}\end{aligned}$$

It can be defined by

$$\text{prob} = \text{BVProblem}(f, domin, bc)$$

.