Popular Abstract

Adrian Zafari

March 11, 2024

The study of differential equations has been a topic of great interest to many mathematicians over the centuries, having brought with it great developments in engineering and the sciences. With the advent of the 20th century and the gradual automation of the laborious task of computation, mankind has strived to develop ever more efficient and effective means to ensure progress. Numerical methods for solving differential equations as a result saw a great rise in prominence during this time given that with the aid of computers, we could begin to automate the process of solving different equations, allowing us to even contend with problems that could not be solved analytically. In order to best be able to harness the capabilities of computers, particularly due to hardware limitations, innovations needed to be made.

One such innovation is the humble controller, an algorithm which harnesses the simple principle of wanting to reach and maintain a selected standard for performance at all possible times in a dynamical system. These controllers may be used in situations where automation is needed such as ensuring that the cabin temperature of an aeroplane is kept within a certain threshold to ensure the wellbeing of the passengers and crew in spite of changing climate conditions. Mathematicians in the early 20th century saw apt grounds for the implementation of controllers in order to automate the normally sensitive time-stepping process, which is responsible for the generation of the time grid on which a given differential equation is evaluated. Should this time grid not be suitable for the problem, one could simply get completely wrong solutions. Thus instead of using a fixed time step and wasting precious computational power and time spent waiting, one may use an adaptive controller-based time-stepping scheme which automatically conforms to the demands imposed on the solver by the problem.

This work studies precisely the implementation of such a solver, using the error in the evaluation of the current time step to predict the best possible step size for the next time step. It begins with an introduction to some of the underlying theory required to contend with numerical solvers, the problems they will solve, and a bit of a deeper look at what exactly controllers are, focusing on the study of a specific type of controller known as a PID controller. Having implemented the solver in Python, it is tested for the aforementioned problems to explore its efficacy and reliability. Included is a comparison of the implemented solver with results produced by SciPy and for two problem cases, the implementation is compared with another in Julia which shares many of the same characteristics. Finally, the work is concluded with a discussion on the considerations made with respect to the Python and Julia solvers as well as the difficulties that were experienced working with the two implementations.