A Runge-Kutta method for computing guaranteed solutions of ODEs Comparison with Taylor Series methods.

Olivier Bouissou
olivier.bouissou@cea.fr

In this article we present a new approach for the computation of guaranteed solutions of ODEs, based on a classical Runge-Kutta method with a precise error approximation.

The interest in validated methods for solving ODEs has recently increased in many areas, such as state estimation or verification of hybrid systems. Existing validation tools mostly use Taylor Series methods, either with intervals like VNODE or Taylor models like COSY.

Our method has been developed in order to be used within a validation tool for hybrid systems, and thus focuses more on long-term stability and accuracy than execution time, without however neglecting it. Each computation step consists of two stages: a prediction stage where approximate values for the solution are computed, and a correcting stage where an overapproximation of the global error is calculated. The first stage uses classical RK4 formulae and multiprecision arithmetic in order to get an accurate result. The second stage represents our main contribution. The global error after n steps is expressed as a function of the error after $n-1$ steps and the step size h. An overapproximation of this error is then computed and from this value we modify h in order to keep the global error under user-defined bounds. This separation between the computation of the next step, which uses multiprecision arithmetic, and the computation of the global error using interval arithmetic makes it possible to avoid the well known wrapping effect and then guarantees long term stability of the method.

In this article, I will present how we compute the error bounds and compare the formulae we get with the ones that give the error in Taylor Series methods. I will also present the implementation of our algorithm in a C++ library called GRKLib for Guaranteed Runge-Kutta library. This library uses the formal derivation tool GiNaC in order to compute the error functions and the MPFR library for multiprecision computations. The interval arithmetic library used is Profil/BIAS. We have compared our results on classical problems with results from VNODE. This comparison showed a better stability and higher precision for our method, while being as fast as it.