Validated Computation of the Local Truncation Error of Runge-Kutta Methods with Automatic Differentiation

Olivier Mullier, Alexandre Chapoutot, Julien Alexandre Dit Sandretto

Validated Computation of the Local Truncation Error of Runge-Kutta Methods with Automatic Differentiation

Olivier Mullier* Alexandre Chapoutot† Julien Alexandre Dit Sandretto‡

April 11, 2016

1 Introduction

Validated numerical integration is an appealing approach to produce rigorous results on Initial Value Problem of Ordinary Differential Equations (ODE). An IVP-ODE is defined by

$$\begin{cases} \dot{\mathbf{y}} = f(t, \mathbf{y}) \\ \mathbf{y}(0) \in \mathcal{Y}_0 \subseteq \mathbb{R}^n, \ t \in [0, t_{\text{end}}] \end{cases} . \tag{1}$$

The set $\mathcal{Y}_0$ of initial conditions is used to model some (bounded) uncertainties. For a given initial condition $\mathbf{y}_0$, the solution when it exists is denoted $\mathbf{y}(t; \mathbf{y}_0)$. The goal of validated (or rigorous) numerical integration methods is to compute the set of solution, or an over-approximation, of the solutions of (1), *i.e.*, $\{\mathbf{y}(t; \mathbf{y}_0) : \forall \mathbf{y}_0 \in \mathcal{Y}_0, \forall t \in [0, t_{\text{end}}]\}$.

Most of the rigorous techniques defined so far, since Ramon Moore's seminal work [1], are based on Taylor series approach, see for example [2, 3, 4, 5] and the references therein. Nervertheless, it is unlikely that only one kind of methods is adapted to all various classes of ODE. So, more recent work [6, 7, 8, 9, 10] deals with the adaptation of Runge-Kutta methods to be validated methods in order to try to benefit the good properties such as A-stability. A generic s-step Runge Kutta method for IVP-ODE is defined by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^{s} b_i \mathbf{k}_i \tag{2}$$

with

$$\mathbf{k}_i = f(t_n + c_i h, \mathbf{y}_n + \sum_{j=1}^{s} a_{ij} \mathbf{k}_j), \quad i = 1, \ldots, s \ . \tag{3}$$

The real coefficients c_i, $a_{i,j}$ and b_i fully characterize a Runge-Kutta methods, see [11].

The challenge to make a Runge-Kutta validated is to bound the *local truncation error* (LTE), *i.e.*, the distance between the true solution $\mathbf{y}(t_n; \mathbf{y}_{n-1})$ at time t_n with y_{n-1} as intial conditions and the numerical solution $\mathbf{y}_n$ starting from the same initial condition so to bound $\mathbf{y}(t_n; \mathbf{y}_{n-1}) - \mathbf{y}_n$.

Following the *order condition*, see [12], a Runge-Kutta method is of order p if the p first terms of the Taylor form associated to the numerical solution y_n are equal to the terms of the exact solution of (1) that is $y(t; y_{n-1})$ assuming the same initial condition. In this case the LTE corresponds to the difference of the two Taylor remainders. Now, the challenge is to compute these Taylor remainders.

In this paper, we propose a novel approach to bound the LTE based on the order condition which is usable for explicit and implicit Runge-Kutta methods. More precisely, our approach is an instance of the algorithm defined in [13] and applied in the context of validated numerical integration methods based on Runge-Kutta methods.

2 Bounding the local truncation error

One of the great ideas of John Butcher in [12] is to express on the same basis of *elementary differentials* the Taylor expansion of the exact solution of (1) and those of the numerical solution. Those elementary differential are made of sums of partial derivatives of f with respect to the components of $\mathbf{y}$. An other great idea of John Butcher in [12] is to relate these partial derivatives of order q to a combinatorial problem to enumerate all the trees τ with exactly q nodes. From the structure of a tree τ one can map a particular partial derivative, see [12] for more details. In consequence, one has the three following theorems which are used to express the order condition of Runge-Kutta methods. In theorems 2.1 and 2.2, τ is a rooted tree, $F(\tau)$ is the elementary differential associated to τ, $r(\tau)$ is the order of τ (the number of vertices it contains), $\gamma(\tau)$ is the density, $\alpha(\tau)$ is the number of equivalent trees and $\psi(\tau)$ the elementary weight of τ based on the coefficients c_i, a_{ij} and b_i defining a Runge-Kutta method.

*Corresponding author, ENSTA ParisTech, Palaiseau France, olivier.mullier@polytechnique.edu
†ENSTA ParisTech, Palaiseau, France, alexandre.chapoutot@ensta-paristech.fr
‡ENSTA ParisTech, Palaiseau, France, julien.alexandre-dit-sandretto@ensta-paristech.fr

Theorem 2.1. *The q-th derivative w.r.t. time of the* ***exact solution*** *is given by*

$$\mathbf{y}^{(q)} = \sum_{r(\tau)=q} \alpha(\tau) F(\tau)(\mathbf{y}_0) \ .$$

Theorem 2.2. *The q-th derivative w.r.t. time of the* ***numerical solution*** *is given by*

$$\mathbf{y}_1^{(q)} = \sum_{r(\tau)=q} \gamma(\tau)\phi(\tau)\alpha(\tau) F(\tau)(\mathbf{y}_0) \ .$$

Theorem 2.3 (Order condition)**.** *A Runge-Kutta method has order p iff*

$$\phi(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau, r(\tau) \leqslant p \ .$$

In consequence of Theorem 2.3, the LTE of a Runge-Kutta method is defined by

$$y(t_j; t_{j-1}) - y_j = \frac{h^{p+1}}{(p+1)!} \sum_{r(\tau)=p+1} \alpha(\tau)\big[1 - \gamma(\tau)\psi(\tau)\big] F(\tau)(\mathbf{y}(\xi)) \quad \xi \in]t_{j-1}, t_j[\ . \tag{4}$$

Based on (4), validated numerical Runge-Kutta methods were successfully developped in [14] mainly using a symbolic generation of the elementaty differentials. The limitation of the symbolic computation approach is that the number of partial derivatives of f increases exponentially with the order q and the syntactic expression may increase also exponentially so this approach does not scale up. During our investigation to overcome these limitations the work [13] was published. It defines an algorithm to compute B-series, *i.e.*, sums of elementary differentials, only as an abstract mathematical object without targeting a particular use. Our contribution is then to instanciate the algorithm defined in [13] to fit our purpose to make validated Runge-Kutta methods. This algorithm will scale up as it is based on automatic differention techniques, more precisely, the techniques defined in [15]. It is also based on a particular factorization of the elementary differential in order to reduce the computation of the same sub-expressions appearing in different elementary differentials.

Note that in (4), the value of $\mathbf{y}(\xi)$ is bounded following classical approach in validated numerical integration methods. More precisely, a variant of the Picard operator, see [2], is used in combination with interval arithmetic. Hence, it is possible to bound the value of the (4) in order to make any explicit and implicit Runge-Kutta validated.

3 Conclusion

In this paper, the computation of a guaranteed outer approximation of the local truncation error of a Runge Kutta method is presented. Combining the computation of the LTE with the classical two step integration method that is the use of the Picard-Lindelöf operator to enclose all the solutions on one step, and the computation of the approximated solution makes possible the validated simulation of an ordinary differential equation with any Runge-Kutta method (implicit or explicit). This new method benefits from the previous work on the computation of B-series from Bartha *et alii*.

References

[1] R.E. Moore. *Interval Analysis*. Series in Automatic Computation. Prentice Hall, 1966.

[2] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105(1):21 – 68, 1999.

[3] Kyoko Makino and Martin Berz. COSY INFINITY version 9. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 558(1):346 – 350, 2006.

[4] Youdong Lin and Mark A. Stadtherr. Validated solutions of initial value problems for parametric odes. *Appl. Numer. Math.*, 57(10):1145–1162, 2007.

[5] Tomáš Dzetkulič. Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. *Numerical Algorithms*, 69(1):183–205, 2015.

[6] Karol Gajda, Andrzej Marciniak, and Barbara Szyszka. Three- and four-stage implicit interval methods of Runge-Kutta type. *Computational Methods in Science and Technology*, 6(1):41–59, 2000.

[7] Andrzej Marciniak and Barbara Szyszka. On representations of coefficients in implicit interval methods of runge-kutta type. *Computational Methods in Science and Technology*, 10(1):57–71, 2004.

[8] Andrzej Marciniak. Implicit interval methods for solving the initial value problem. *Numerical Algorithms*, 37(1-4):241–251, 2004.

[9] Olivier Bouissou and Matthieu Martel. GRKLib: a Guaranteed Runge Kutta Library. In *Scientific Computing, Computer Arithmetic and Validated Numerics*, 2006.

[10] Olivier Bouissou, Alexandre Chapoutot, and Adel Djoudi. Enclosing temporal evolution of dynamical systems using numerical methods. In *NASA Formal Methods*, number 7871 in LNCS, pages 108–123. Springer, 2013.

[11] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Computational Mathematics. Springer, 2006.

[12] John C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *Journal of the Australian Mathematical Society*, 3:185–201, 5 1963.

[13] Ferenc Bartha and Hans Z Munthe-Kaas. Computing of B-series by automatic differentiation. *Discrete and continuous dynamical systems*, 34(3):903–914, 2014.

[14] Alexandre Chapoutot, Julien Alexandre dit Sandretto, and Olivier Mullier. Validated Explicit and Implicit Runge-Kutta Methods. In *Small Workshop on Interval Methods*, Prague, Czech Republic, June 2015.

[15] Andreas Griewank, Jena Utke, and Andrea Walther. Evaluating higher derivative tensors by forward propagation of univariate taylor series. *Mathematics of Computation*, 69(231):1117–1130, 2000.