Applied Mathematics and Computation 268 (2015) 227–245

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Automatic implementation of the numerical Taylor series method: A Mathematica and Sage approach

A. Abad [a], R. Barrio [a,b], M. Marco-Buzunariz [c], M. Rodríguez [a,b,d,*]

[a] Grupo de Mecánica Espacial - IUMA, University of Zaragoza, Spain
[b] Computational Dynamics group - IUMA, University of Zaragoza, Spain
[c] Departamento de Matemáticas, University of Zaragoza, Spain
[d] Centro Universitario de la Defensa Zaragoza and IUMA, Academia General Militar. Zaragoza, Spain

ARTICLE INFO

Keywords:
Taylor series method
TIDES
ODEs
Automatic differentiation
Sage
Mathematica

ABSTRACT

In the last few years, the requirements in the numerical solution of ordinary differential equations in physics and in dynamical systems have pointed to new kind of methods capable to maintain geometric properties of the equations, or looking for high-precision, or solving variational equations. One method that can solve most of these problems is the Taylor series method. TIDES is a free software based on the Taylor series method that uses an optimized variable-stepsize variable-order formulation. The kernel of this software consists of a C library that permits to compute up to any precision level (by using multiple precision libraries for high precision when needed) the solution of an ordinary differential system from a C driver program containing the equations of the ODE. In this paper we present the symbolic methods, implemented in a computer algebra system (CAS), used to write, automatically, the code based on the automatic differentiation processes that integrates a particular differential system by means of the Taylor method. The precompiler has been written in Mathematica and Sage (which includes it by default since version 6.4). The software has been done to be extremely easy to use. The Mathematica version also permits to compute in a direct way not only the solution of the differential system, but also the partial derivatives, up to any order, of the solution with respect to the initial conditions or any parameter of the system.

1. Introduction

The Taylor series method, already used by Newton [33] and detailed by Euler [17] in the XVIII century, is a classical method for solving ordinary differential equations. However, this method has been scarcely used in numerical analysis until last few years, mainly because each problem requires a specific formulation. Recently, the method has been rediscovered and it has been applied in Celestial Mechanics [16,24,40] (where it is called recurrent power series method), in dynamical systems [7,9] to find periodic orbits, in numerical analysis applied to ordinary differential equations (ODEs) [4,8,10,11,13,14,39], and more recently applied to differential-algebraic equations (DAEs) [4,28–30,34,35]. In all these problems the Taylor series method (from now on TSM) has proven its applicability.

The use of automatic differentiation (AD) techniques combined with the algebra of series, suggested in [27], opened a new approach to solve the difficulties in the formulation of each particular problem. Now we may apply the symbolic computational

* Corresponding author. Tel.: +34 976739856.
E-mail addresses: abad@unizar.es (A. Abad), rbarrio@unizar.es (R. Barrio), mmarco@unizar.es (M. Marco-Buzunariz), marcos@unizar.es (M. Rodríguez).

http://dx.doi.org/10.1016/j.amc.2015.06.042
0096-3003/

tools to systematize the application of the TSM. In this context there are several software tools different from the one used in this paper: *ATOMFT* [13] and *Taylor* [23]. *ATOMFT* is written in Fortran 77; it consists of a preprocessor and a run-time library. The preprocessor transforms Fortran-like statements into a Fortran form suitable for the run-time library. *Taylor* makes a similar task, it uses a parser written with the help of the Unix tools Lex and Yacc to analyze the differential equation, written on a natural language, and creates C code with the integration method. Taylor supports several extended precision arithmetics.

The use of some extended automatic differentiation formulas [5] in the obtention of the Taylor series coefficients has allowed, in the last few years, the direct obtention of the solution of differential equations and partial derivatives of the solutions of any order without an explicit generation of the variational equations, which can be quite cumbersome for high orders. This extended Taylor series method has been used in [6,12] to solve directly variational equations in the study of several chaos indicators.

Taking into account the usefulness of the TSM, and the characteristics of the new problems where it is applied, we developed TIDES [1,2], a new free software that includes all the possibilities of the previous ones and adds new characteristics to handle new problems, like the solution of the variational equations, and others. TIDES has two different parts (pieces of software): the Mathematica [44] package MathTIDES and the C library libTIDES. MathTIDES is a preprocessor that writes, automatically from a Mathematica session, the files containing the code with the iterative scheme to obtain the Taylor series of the variables. Recently the authors also implemented some of the MathTIDES functionality in Sage [42], which includes it by default since version 6.4. These files, together with the library libTIDES, form the TSM Integrator. Combining TIDES with the MPFR library ([18], http://www.mpfr.org) we have a multiple precision version of the integrator. The software TIDES can be freely obtained by visiting the web page https://sourceforge.net/projects/tidesodes or from the package installation tool of Sage.

We remark that with the inclusion of TIDES the software Sage is nowadays the only algebraic manipulator equipped with a numerical software suitable to obtain the numerical solution of ordinary differential equations with arbitrary high precision with a single command out of the box. As shown in the paper, the solution of a system with, lets say, 1000 precision digits is quite simple and becomes an approachable problem.

Among the main characteristics of TIDES we can consider the following:

- The expression of the ODE system may contain combinations of the following operations and/or functions: $+, -, *, /, a^x$, x^a $(a \in \mathbb{R})$, $\sin$, $\cos$, $\tan$, $\sinh$, $\cosh$, $\tanh$, asin, acos, atan, asinh, acosh, atanh, log.
- The differential equation may contain parameters.
- Together with the solution of the ODE, i.e, the time evolution of the variables, TIDES can also give the evolution of partial derivatives, up to any order, of the variables with respect to the initial conditions and/or the parameters, functions of the variables, and partial derivatives, up to any order, of the functions of the variables with respect to the initial conditions and/or parameters.
- The user of TIDES can choose the number of digits of precision of the solution.

In this paper we describe only the symbolic part of TIDES, i.e. the preprocessor MathTIDES and the corresponding functionality in Sage, but not the numerical characteristics implemented in libTIDES. A detailed description of libTIDES can be found in [1].

The present paper is organized as follows: Section 2 reviews the TSM and how TIDES is used, from the formulation of the problem until its resolution, Sections 3 and 4 present the Mathematica and Sage preprocessors and Section 5 compares both, Section 6 presents the expressions of the algebra of series extended to compute partial derivatives, Section 7 defines the concept of Linked function and its relation with the AD-techniques and extends this concept to handle the algebra of series, Section 8 analyzes the data structures necessary to implement the TSM and how MathTIDES computes the value of these structures for each particular problem, Section 9 presents some conclusions and finally, in the appendixes appear the codes for the Lorenz system.

2. The Taylor series method and TIDES

Let us consider the initial value problem:

$$\frac{d\boldsymbol{y}(t)}{dt} = \boldsymbol{f}(t, \boldsymbol{y}(t); \boldsymbol{p}), \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \qquad t \in \mathbb{R}, \boldsymbol{y} \in \mathbb{R}^s, \boldsymbol{p} \in \mathbb{R}^k, \tag{1}$$

where $\boldsymbol{p}$ represents a vector of $k(\geq 0)$ parameters.

If the Taylor series expansion of the solution is given by the expression

$$\boldsymbol{y}(t) = \sum_i \boldsymbol{y}^{[i]} h^i, \quad h = t - t_0, \quad \boldsymbol{y}^{[i]} := \boldsymbol{y}^{[i]}(t_0) = \frac{1}{i!} \frac{d\boldsymbol{y}^{(i)}(t_0)}{dt}, \tag{2}$$

then, substituting $\boldsymbol{y}$ by its power series expansion in (1) we have

$$\sum_i (i+1) \boldsymbol{y}^{[i+1]} h^i \simeq \boldsymbol{f}\left(t, \sum_i \boldsymbol{y}^{[i]} h^i; \boldsymbol{p}\right) \simeq \sum_i \boldsymbol{f}^{[i]}(t_0, \boldsymbol{y}^{[0]}, \ldots, \boldsymbol{y}^{[i]}; \boldsymbol{p})\, h^i, \tag{3}$$