Commun Nonlinear Sci Numer Simulat 61 (2018) 37–53

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

Research paper

A new subdivision algorithm for the flow propagation using polynomial algebras

Daniel Pérez-Palau[a,*], Gerard Gómez[b], Josep J. Masdemont[c]

[a] CNES, 18 avenue Edouard Belin, Toulouse 31401, France
[b] Universitat de Barcelona & IEEC, Gran Via 585, Barcelona 08007, Spain
[c] Universitat Politècnica de Catalunya & IEEC, Diagonal 647, Barcelona 08028, Spain

A R T I C L E   I N F O

Article history:
Received 13 March 2017
Revised 22 January 2018
Accepted 23 January 2018
Available online 31 January 2018

MSC:
10.070

Keywords:
Jet transport
Polynomial algebra
Subdivision algorithms
ODE polynomial propagation

A B S T R A C T

The Jet Transport method has emerged as a powerful tool for the numerical integration of ordinary differential equations; it uses polynomial expansions to approximate the flow map associated to the differential equation in the neighbourhood of a reference solution. One of the main drawbacks of the method is that the region of accuracy becomes smaller along the integration. In this paper we introduce a procedure to determine a ball covering the set of given initial conditions that keeps the accuracy of the integration within a selected threshold. The paper gives detailed explanations of the algorithm illustrated with some examples of applicability, as well as a comparison with a previous existing method for the same purpose.

1. Introduction

Let $\dot{x} = f(t, x)$, $x \in \mathbb{R}^n$, be a system of ordinary differential equations (ODE), and $\phi_t(t_0, x_0)$ the associated flow map: if $x(t)$ denotes the solution of the ODE such that $x(t_0) = x_0$, then $x(t) = \phi_t(t_0, x_0)$. The Jet Transport (JT), also known as Differential Algebra, procedure is a semi-numerical method that propagates a neighbourhood U of x_0 instead of the single initial condition x_0; this is, at the first time step h of the propagation, the initial condition x_0 is replaced by a polynomial of degree one $P_{t_0,x_0}(\xi) = x_0 + \xi \in \mathbb{R}^n$ that parameterises U, and a higher degree polynomial approximation $P_{t_0+h,x_0}(\xi)$ of $\phi_{t_0+h}(t_0, x_0 + \xi)$ is computed. This resulting polynomial is propagated in the next step, and the procedure is repeated recursively. The basic idea of the method is shown schematically in Fig. 1.

The propagated polynomials $P_{t,\,x}(\xi)$ are computed using an implementation of a numerical integration method for ODEs in which the real number floating point arithmetic is replaced by a polynomial algebra (i.e. all the arithmetic operations are done using truncated polynomials up to a certain degree). The polynomials $P_{t,\,x}(\xi)$ provide, up to a certain order, the solutions of the variational equations associated to the ODE without writing and integrating them explicitly. Therefore, the only tools that are needed are: a numerical integration method for ODEs and a polynomial algebra package.

For common numerical integration methods, such as Runge–Kutta, Taylor or symplectic, the step-size selection is done according to a local truncation error estimate. For the JT procedures it is also necessary to control the size of U. This is the

* Corresponding author.
E-mail address: daniel@maia.ub.es (D. Pérez-Palau).

https://doi.org/10.1016/j.cnsns.2018.01.013
1007-5704/

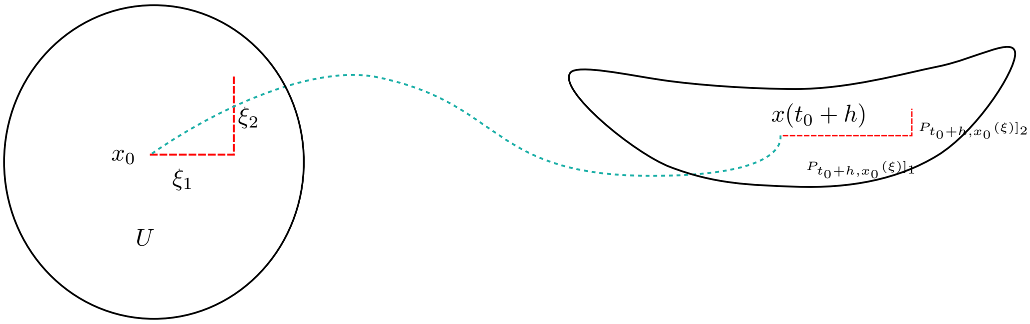

Fig. 1. Schematic idea of the JT propagation procedure. The circle on the left represents the neighbourhood U parameterised by (ξ_1, ξ_2), and the right hand side ellipse is its image at time t_0+h. In the figure $x(t_0+h)$ stands for $\phi_{t_0+h}(t_0, x_0)$, and the image of $x_0+(\xi_1, \xi_2) \in U$ is the point $P_{t_0+h,x_0}(\xi) = x(t_0+h) + (P_{t_0+h,x_0}(\xi)\,|_1, P_{t_0+h,x_0}(\xi)\,|_2)$.

problem we address in the present paper. In the proposed approach, whenever necessary, a subdivision strategy is applied either increasing the number of polynomials or splitting regions for the propagation.

Jet Transport methods where introduced by Berz [7] to study beam dynamics in particle accelerators problems and, since then, have been used in several other fields, for instance: in Astrodynamics to study the close approaches of Near Earth Asteroids [3,4], to compute a Gaussian particle filter for spacecraft navigation [14], or to detect structures in dynamical systems using several indicators [11]. JT methods have also been used to compute the time evolution of probability density functions (PDF) according to the flow associated to an ordinary differential equation [5,10,17].

Currently, there are several implementations of the Jet Transport/Differential Algebra methods. Some of the most well known are: COSY INFINITY [9]; TIDES [1]; CAPD DynSys Library [15], and DACE, which is an implementation developed from COSY specially adapted to the space community [12]. For this work we have used our own one [10], which includes all the basic algebra and functions needed for the implementation of the numerical integration method, as well as some additional ones such as the inversion of functions and an equation solver.

One of the main drawbacks of the JT methods is that the region where their accuracy is below a certain threshold becomes smaller along the integration. This paper presents a new procedure to determine how and when the region and associated polynomials can be split to maintain the required accuracy. The method is based on the covering of the propagated states by new neighbourhoods. The algorithm presented contains some ideas similar to the ones of the interval enclosure methods [2,6], in which the results of the operations are assured to be in a given interval. In the current algorithm, the results of the operations are given by polynomials, which are assured to be in a given region. The new algorithm is compared with the Automatic Domain Splitting (ADS) approach, an algorithm developed by Wittig et al. [16] for this same problem, in which the basic idea is the division and rescaling of the propagation polynomials along the integration.

It must be noted that both approaches mentioned have two main steps: the first one detects when the splitting should be done, and the second accounts for how the division of neighbourhoods or polynomials is implemented. In both cases the procedures work schematically as follows. First a certain neighbourhood is propagated until some condition breaks because the accuracy of the propagation is below some fixed tolerance. Up to that point the method is a usual JT flow propagation. When the condition breaks the second part of the algorithm starts: the polynomial or the set of points, depending on the selected strategy, splits. After the division, the usual Jet Transport flow propagation is started with new initial conditions.

The paper is organised as follows: in Section 2 we briefly introduce the step control strategy for the integrator used as well as how to determine if the polynomials are accurate enough; in Section 3 we review the (ADS) approach; Section 4 describes the procedure that we have developed; in Section 5 we give some numerical tests and comparisons between both methods, while the last section ends with some remarks and conclusions.

2. Step-size control in polynomial algebra propagation

In this section we briefly discuss how to determine the step size in a JT procedure. As numerical integration method we have used Taylor's method for ODE (as implemented by Jorba and Zou [8]). Given a certain accuracy level ε, the optimal step of this method is given by (see [13] for details)

$$h_{opt} = \min\left\{\left(\frac{\varepsilon e^2 \|x^{(1)}\|_\infty}{\|x^{(n-1)}\|_\infty}\right)^{\frac{1}{n-2}}, \left(\frac{\varepsilon \|x^{(1)}\|_\infty}{\|x^{(n)}\|_\infty}\right)^{\frac{1}{n-1}}\right\},$$

where $x^{(i)}$ are the coefficients of Taylor's method solution written in powers of the step size h at each step, this is:

$$\phi_{t_n+h}(t_n, x_n) = \sum_{i=0}^{n} x^{(i)}(t_n, x_n) h^i.$$