Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 218 (2008) 543–555
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Symbolic derivation of order conditions for hybrid Numerov-type methods solving $y'' = f(x, y)$

I.Th. Famelis[a,*], Ch. Tsitouras[b]

[a] *Department of Mathematics, TEI of Athens, GR 122 10 Egaleo, Greece*
[b] *Department of Applied Sciences, TEI of Chalkis, GR 34400 Psahna, Greece*

Received 26 September 2006; received in revised form 10 September 2007

Abstract

Numerov-type ODE solvers are widely used for the numerical treatment of second-order initial value problems. In this work we present a powerful and efficient symbolic code in MATHEMATICA for the derivation of their order conditions and principal truncation error terms. The relative tree theory for such order conditions is presented along with the elements of combinatorial mathematics, partitions of integer numbers and computer algebra which are the basis of the implementation of the symbolic code.

Keywords: Numerov-type methods; Order conditions; Rooted trees; Integer partitions; Truncation error; MATHEMATICA®

1. Introduction

Second-order ordinary differential equations (ODEs) that do not involve y',

$$y'' = f(t, y), \quad y(t_0) = y^{[0]}, \quad y'(t_0) = y'^{[0]}, \tag{1}$$

where $f : \mathfrak{R}^N \longrightarrow \mathfrak{R}^N$ and $y^{[0]},\ y'^{[0]} \in \mathfrak{R}^N$, are widely used to model physical problems. Thus, methods for the numerical treatment of such ODEs are of great importance. There exist various classes of methods for the numerical solution of such problems. For instance, Problem (1) can be treated using a Runge–Kutta–Nyström method, or if we transform it in a system of first-order ODEs, it can be solved by a Runge–Kutta method [11]. One of the most widely used methods for solving (1) is the Numerov which attains fourth algebraic and sixth phase-lag order [2,8]. This method is implicit and its implementation involves computations of Jacobians and solutions of nonlinear systems of equations. So, many authors proposed explicit modifications of the Numerov method which are usually called hybrid or two-step Numerov-type methods. The construction of such methods requires the derivation and the solution of equations called "order conditions". Such a procedure is a tedious task since the number of order conditions to be derived and then solved increases as the order of a method increases. The order conditions are nonlinear expressions which involve the methods coefficients. So, constructing a specific method requires the solution of a system of nonlinear equations. For high order methods, we usually use symbolic computations to solve some of the equations resulting

* Corresponding author.
E-mail addresses: ifamelis@teiath.gr (I.Th. Famelis), tsitoura@teihal.gr (Ch. Tsitouras).
URLs: http://math.teiath.gr/ifamelis/ (I.Th. Famelis), http://users.ntua.gr/tsitoura/ (Ch. Tsitouras).

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.09.017

in exact expressions for coefficients that involve other coefficients. Then the numerical treatment of the remaining system, usually by powerful minimization algorithms, yields solutions that fail to satisfy the order conditions with an acceptable accuracy. Nevertheless, the outcome can be useful as a set of initial values for the next phase of our work. Computer Algebra systems, such as MATHEMATICA [30], provide the capability to apply numerical methods asking the results to satisfy a lot more than 16 digits of accuracy. So, using the numerical results as good guesses we hope and usually manage to specify coefficients that satisfy the order conditions with very high acceptable accuracy. Therefore, for both the derivation and the solution of the order conditions the use of a Computer Algebra systems is needed.

In the literature, computer codes for generating Runge–Kutta trees, order conditions and truncation order coefficients can be found. Keipers [14] program, written in MATHEMATICA language, was probably the first but it was limited in deriving low order conditions. Hosea [13] presented a code named RKTEC, written in ANSI C, which is available from `Netlib`. This was based on a recurrence procedure due to Albrecht [1] that generates order conditions. A new perspective was introduced by Harisson [12] and Papakostas [18] as it was acknowledged in [23]. They suggested the use of tensor notation which resulted in very interesting symbolic codes. That early package due to Papakostas had been a powerful tool for the research work of our group [21,29,20,25] when truncation error calculations were needed. It can be asked by an e-mail from the present authors. Sofroniou [23] as well has published an integrated package for deriving Runge–Kutta order conditions. Then Papakostas, in his Ph.D. Thesis [19], proposed that in such codes the derivation of trees should be avoided. Following his suggestions, we have presented [10] a very efficient code for the derivation of Runge–Kutta order conditions. Finally, a MATLAB code for generating Runge–Kutta trees, order conditions and truncation error coefficients is due to Cameron [7].

For the class of Runge–Kutta–Nyström methods a first code to generate RKN trees is due to Okunbor [16]. This specific code, which was based on the Keipers program philosophy, fails at high orders. Following the same lines of our previous work, our team presented a powerful and efficient symbolic package for the derivation of Runge–Kutta–Nyström [28] order conditions and principal truncation error terms. Here, we present the first symbolic package for the derivation of order conditions and principal local truncation error terms for two-step Numerov-type methods.

In the following section we outline the theory of construction of two-step Numerov-type methods. Then we present the elements of combinatorial mathematics and tree theory [15,17,22] which have been used to approach the construction of our symbolic program. In our approach, constructing the trees as matrix products results in a very fast, neat and cheap in memory usage code.

2. Hybrid Numerov methods

Two-step Numerov-type methods proceed to the evaluation of $y^{[k+1]}$ as an estimation of $y(t_{k+1})=y(t_k+h)$, according to the following formulae:

$$
\begin{aligned}
&Y^{[1]} = (1-c_1)y^{[k]} + c_1 y^{[k-1]} + h^2 \sum_{j=1}^{s} a_{1j} f_{k-c_j},\\
&f_{k-c_1} = f(t_k - c_1 h, Y^{[1]}),\\
&Y^{[2]} = (1-c_2)y^{[k]} + c_2 y^{[k-1]} + h^2 \sum_{j=1}^{s} a_{2j} f_{k-c_j},\\
&f_{k-c_2} = f(t_k - c_2 h, Y^{[2]}),\\
&\vdots \qquad\qquad \vdots\\
&Y^{[s]} = (1-c_s)y^{[k]} + c_s y^{[k-1]} + h^2 \sum_{j=1}^{s} a_{sj} f_{k-c_j},\\
&f_{k+c_s} = f(t_k - c_s h, Y^{[s]}),\\
&y^{[k+1]} = 2y^{[k]} - y^{[k-1]} + h^2 \sum_{j=1}^{s} b_j \ f_{k-c_j},
\end{aligned}
\tag{2}
$$