INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Computation of order conditions for symplectic partitioned Runge-Kutta schemes with application to optimal control

J. Frédéric BONNANS — Julien LAURENT-VARIN

N° 5398

Décembre 2004

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--5398--FR+ENG

Computation of order conditions for symplectic partitioned Runge-Kutta schemes with application to optimal control

J. Frédéric BONNANS* , Julien LAURENT-VARIN†

Thème NUM — Systèmes numériques
Projet SYDOCO

Rapport de recherche n° 5398 — Décembre 2004 — 18 pages

Abstract: We discuss the derivation of order conditions for the discretization of (unconstrained) optimal control problems, when the scheme for the state equation is of Runge-Kutta type. This problem appears to be essentially the one of checking order conditions for symplectic partitioned Runge-Kutta schemes. We show that the the computations using bi-coloured trees are naturally expressed in this case in terms of *oriented free tree*. This gives a way to compute them by an appropriate computer program.

Our software is able to compute conditions up to order 7 (we display them up to order 6). The results are in accordance with those of Hager (where they were computed for order up to 4) as well as those of Murua where the number of conditions up to order 7 is stated.

Key-words: Optimal control, Hamiltonian systems, partitioned Runge-Kutta methods, Symplectic schemes, order conditions, P-series, H-trees, Oriented free-trees.

This study is supported by CNES and ONERA, in the framework of the CNES fellowship of the second author.

* J. Frédéric BONNANS: Projet Sydoco, Inria-Rocquencourt, Domaine de Voluceau, BP 105, 78153 Le Chesnay, France (Frederic.Bonnans@inria.fr).

† Julien LAURENT-VARIN: Long-Term Design & System Integration Department, ONERA, and Projet Sydoco, INRIA Rocquencourt, Domaine de Voluceau, BP 105, 78153 Le Chesnay (Julien.Laurent-Varin@inria.fr).

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Calcul des conditions d'ordre de schémas de Runge-Kutta symplectiques partitionnés avec application à la commande optimale

Résumé : L'article traite le calcul des conditions d'ordre pour la discrétisation de problèmes de commande optimale sans contraintes, lorsque le schéma est de type Runge-Kutta. Ce problème se réduit à celui de la vérification de conditions d'ordre de méthodes de Runge-Kutta symplectiques partitionnées. Nous montrons que le calcul basé sur les arbres bicolores avec racine s'exprime de manière naturelle en utilisant des arbres libres orientés. Ceci permet d'exprimer ces conditions par un programme.

Notre code calcule les conditions jusqu'à l'ordre 7 (ils sont présentés jusqu'à l'ordre 6). Les résultats sont en accord avec ceux de Hager (qui a calculé les conditions jusqu'à l'ordre 4) et Murua (qui a dénombré ces conditions jusqu'à l'ordre 7).

Mots-clés : Commande optimale, systèmes hamiltoniens, schémas de Runge-Kutta partitionnés, schémas symplectiques, conditions d'ordre, P-séries, H-arbres, arbres orientés.

1 Introduction

The motivation of this work comes from an analysis by Hager [7] of order conditions for optimal control problems (of an ordinary differential equation). The idea is to discretize the state equation by a Runge-Kutta scheme, with a different value of control associated with each "inner state". Hager observes that the resulting optimality system, after some change of variable, is a partitioned Runge-Kutta scheme. He computes then (by hand, i.e., without computer code) the order conditions for order up to 4. See also the results of [5] and [6] on constrained optimal control problems (a first order state constrained problem, discretized by Euler's scheme, and a control constrained problem with a Runge-Kutta discretization).

There are essentially two hypotheses in the analysis of [7], one on the original problem and the other is a restriction on the scheme. One has to assume that the Hamiltonian is strongly convex w.r.t. the control, or more generally that the second derivative of Hamiltonian w.r.t. the control is invertible. This allows to eliminate the control thanks to the implicit theorem, so that we have an equivalent scheme for the reduced (state, adjoint state) system. The second hypothesis is that none of the coefficients b_i's (of the particular Runge-Kutta scheme) is zero.

The main result of Hager [7] is that, if the original Runge-Kutta scheme is of (global) order p (i.e., when applied to an uncontrolled differential equation) then the resulting scheme has order $q \leq p$, with equality if $p \leq 2$ but strict inequality in some cases whenever $p \geq 3$. In addition, $q = p$ if the scheme is explicit of order at most 4.

For order greater than four, one cannot do computations by hand. It is then useful to rely on the theory of order conditions for partitioned Runge-Kutta scheme. This theory, based on bicolour rooted trees with which are associated certain numbers, is an extension of the original work by Butcher for (non partitioned) Runge-Kutta schemes, see Butcher [4, p. 88].

It appears that the class of partitioned Runge-Kutta schemes coming from the discretization of optimal control problems are in fact partitioned symplectic Runge-Kutta schemes, characterized by relation (4) below. So the question boils down to the one of expressing order conditions for this class. The main result of this paper is that we can obtain the desired expressions using a "calculus" on oriented free trees. To be specific, with each oriented free tree are associated some weights, and the main operation is to "split" any rooted tree into a sum (with coefficients ± 1) of such oriented free trees.

The paper is organized as follows. In the next section we detail the discretization of optimal control problems by Runge-Kutta schemes, and show the relation with partitioned symplectic Runge-Kutta schemes. satisfying (4). Then in section 3 we review the theory of order conditions for partitioned Runge-Kutta schemes. Section 4 introduces oriented free trees, and shows how the order conditions can be expressed in terms of the latter. Finally section 5 discusses the numerical implementation, and displays the results for order up to 6 and the number of conditions for order up to 7.

2 Discretization of unconstrained optimal control problems

Let f and Φ be C^∞ functions $I\!R^m \times I\!R^n \to I\!R^n$ and $cR^n \to I\!R$, respectively. Consider the following unconstrained optimal control problem:

$$\left\{ \begin{array}{l} \text{Min } \Phi(y(T)); \\ \dot{y}(t) = f(u(t), y(t)), \quad t \in [0,T]; \\ y(0) = y^0. \end{array} \right. \tag{P}$$

We restrict the analysis to continuous control functions. Denote by $H(u,y,p) := p \cdot f(u,y)$ the *pseudo-Hamiltonian* of the problem. The first order necessary optimality conditions of this problem may be written in the following form:

$$\left\{ \begin{array}{rcl} \dot{y}(t) & = & f(u(t), y(t)), \\ \dot{p}(t) & = & H_y(u(t), y(t), p(t)), \\ 0 & = & H_u(u(t), y(t), p(t)), \\ p(T) = \Phi'(y(T)), & & y(0) = y^0. \end{array} \right\} \quad t \in [0,T], \tag{OC}$$

We say that $(\bar{u}, \bar{y}, \bar{p})$ is an extremal if it satisfies (OC) ($\bar{u}$ being a continuous function). Let $(\bar{u}, \bar{y}, \bar{p})$ be an extremal. If

$$u \mapsto H_{uu}(u,y,p) \quad \text{is invertible along the trajectory,} \tag{1}$$

then by the implicit functions theorem, in a small L^∞ neighborhood of this trajectory, we have that $H_u(u(t), y(t), p(t)) = 0$ iff $u = \phi(y(t), p(t))$, where ϕ is a C^∞ mapping. Define the *true Hamiltonian* as $\mathcal{H}(y,p) := H(\phi(y,p), y, p)$. Using $H_u(\phi(y(t), p(t)), y(t), p(t)) = 0$, obtain

$$\mathcal{H}_y(y,p) = H_y(\phi(y,p), y, p); \quad \mathcal{H}_p(y,p) = H_p(\phi(y,p), y, p). \tag{2}$$

Consequently, under hypothesis (1), (OC) is locally equivalent to the *reduced Hamiltonian system*

$$\begin{array}{rcll} \dot{y}(t) & = & \mathcal{H}_p(\phi(y,p), y, p), & t \in [0,T], \\ -\dot{p}(t) & = & \mathcal{H}_y(\phi(y,p), y, p), & t \in [0,T], \\ p(T) & = & \Phi'(y(T)), & y(0) = y^0. \end{array} \tag{3}$$

We now turn to the discussion of the discretization of the optimal control problem (P). The Runge-Kutta discretization considered in [7] is

$$\left\{ \begin{array}{rcl} \text{Min } \Phi(y_N); & & \\ y_{k+1} & = & y_k + h_k \sum_{i=1}^s b_i f(u_{ki}, y_{ki}), \\ y_{ki} & = & y_k + h_k \sum_{j=1}^s a_{ij} f(u_{kj}, y_{kj}), \\ y_0 & = & y^0, \end{array} \right. \tag{DP$_1$}$$

for all $k = 0$ to $N-1$ and $i = 1$ to s. Here $h_k > 0$ is the site of the kth time step, and (a,b) is the set of Runge-Kutta coefficients. Let us underline the choice of using different

values of controls u_{kj} associated with inner states y_{kj}; this contrasts with other approaches in which the discretization of controls is coarser than the one of the state, i.e., for control is taken constant, or only affine on each time step (e.g. [1, 2, 3]).

We may rewrite (DP_1) under the equivalent form

$$\left\{\begin{array}{l} \text{Min } \Phi(y_N); \\ 0 = h_k \displaystyle\sum_{i=1}^{s} b_i K_{ki} + y_k - y_{k+1}, \\ 0 = f(u_{ki}, y_k + h_k \textstyle\sum_{j=1}^{s} a_{ij} K_{kj}) - K_{ki}, \\ 0 = y^0 - y_0. \end{array}\right. \qquad (DP_2)$$

In the expression below we contract $y_k + h_k \sum_{j=1}^{s} a_{ij} K_{kj}$ into y_{ki}. The Lagrangian function associated with (DP_2) is:

$$\Phi(y_N) + \sum_{k=0}^{N-1} \left\{ p_{k+1} \cdot \left(h_k \sum_{i=1}^{s} b_i K_{ki} + y_k - y_{k+1} \right) + \sum_{i=1}^{s} \xi_{ki} \cdot (f(u_{ki}, y_{ki}) - K_{ki}) \right\} + p^0 \cdot (y^0 - y_0).$$

Here p_{k+1}, ξ_{ki}, and p^0 are the Lagrange multipliers associated with the constraints of (DP_2). Variables p_k will be interpreted as the discretization of co-state of continuous formulation. Setting to zero the derivative of this Lagrangian function w.r.t. the primal variables y_N, y_0, y_k for $k = 1$ to $N-1$, K_{ki} and u_{ki} for $k = 0 \ldots N-1$, $\quad i = 1 \ldots s$, we obtain

$$\begin{array}{rcl} p_N & = & \Phi'(y_N), \\ p_1 & = & p^0, \\ p_k - p_{k+1} & = & \sum_{i=1}^{s} f_y(u_{ki}, y_{ki})^\top \xi_{ki}, \\ 0 & = & h_k b_i p_{k+1} + h_k \displaystyle\sum_{j=1}^{s} a_{ji} f_y(u_{kj}, y_{kj})^\top \xi_{kj} - \xi_{ki}, \\ 0 & = & f_u(u_{ki}, y_{kj})^\top \xi_{ki}, \quad k = 0 \ldots N-1, \quad i = 1 \ldots s. \end{array}$$

Using now the hypothesis that $b_i \neq 0$, set $p_{ki} := \xi_{ki}/(h_k b_i)$ for all $k = 0$ to $N-1$, and $i = 1$ to s. Eliminating ξ_{ki} from the above relations, we obtain the equivalent optimality conditions

$$\left\{\begin{array}{rclrcl} y_{k+1} & = & y_k + h_k \sum_{i=1}^{s} b_i f(u_{ki}, y_{ki}), & y_{ki} & = & y_k + h_k \sum_{j=1}^{s} a_{ij} f(u_{kj}, y_{kj}), \\ p_{k+1} & = & p_k - h_k \sum_{i=1}^{s} \hat{b}_i H_y(u_{ki}, y_{ki}, p_{ki}), & p_{ki} & = & p_k - h_k \sum_{j=1}^{s} \hat{a}_{ij} H_y(u_{kj}, y_{kj}, p_{kj}), \\ 0 & = & H_u(u_{ki}, y_{ki}, p_{ki}), & y_0 & = & y^0, \qquad p_N = \Phi'(y_N), \end{array}\right. \qquad (DOC)$$

where coefficients $\hat{b}$ and $\hat{a}$ are defined by the following relations:

$$\hat{b}_i := b_i, \quad \hat{a}_{ij} := b_j - \frac{b_j}{b_i} a_{ji}, \qquad \text{for all } i = 1, \ldots, s \text{ and } j = 1, \ldots, s. \qquad (4)$$

If the algebraic constraints $H_u(u_{ki}, y_{ki}, p_{ki}) = 0$ are locally equivalent to $u_{ki} = \phi(y_{ki}, p_{ki})$, then (DOC) is equivalent to the same partitioned Runge-Kutta scheme applied to the reduced system (3). This approach based on formulation (DP_2) is slightly simpler, but equivalent to the one of Hager [7].

It is said that a partitioned Runge-Kutta scheme (or more generally any one step scheme) is symplectic of the corresponding flow is symplectic. It is known that partitioned Runge-Kutta schemes satisfying (4) are symplectic, see [10, Theorem 4.6]. We obtain that the scheme obtained by discretization of problem (P) is symplectic. In particular the following diagram commutes, when we use the above discretization:

$$\begin{array}{ccc} (P) & \xrightarrow{\text{discretization}} & (DP) \\ \text{optimality conditions} \downarrow & & \text{optimality conditions} \downarrow \\ (OC) & \xrightarrow{\text{discretization}} & (DOC) \end{array} \qquad (D)$$

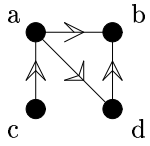
For a detailed presentation of partitioned Runge-Kutta and symplectic methods we refer to the books [8, 9].

3 Order conditions for partitioned Runge-Kutta schemes

A basic tool in the study of order conditions for Runge-Kutta schemes is the theory of B-series and associated calculus on rooted trees, due to Butcher (see [4]). For partitioned Runge-Kutta schemes an extension of this theory is the one of expansion in P-series, and the associated calculus on bicoloured rooted trees, see [8]. The latter allows to state the order conditions in terms of coefficients a, b, $\hat{a}$, and $\hat{b}$, of the following type: certain polynomials (which are in fact sum of monomials with unit coefficients) in these variables have to be equal to certain fractional numbers. The substitution of $\hat{a}$, and $\hat{b}$ (using (4)) would give complicated expressions, since instead of each monomial we would have a sum of rational fractions. We will show that among all these terms it is sufficient to express a condition on a "principal term" since the other terms of the sum are already determined by previous conditions. This allows a tremendous simplification that permits us to display these conditions up to order 6, in terms of a and b alone.

In order to fix notation we will need some definitions. A *graph* G is a pair (V, E) where V had a finite cardinal and $E \subset \mathcal{P}(V)$. Elements of V (resp. E) are called vertices (resp. edges). Here $\mathcal{P}(V)$ denotes is the set of subset of V. The set of *graph* will be denote $\bar{\mathcal{G}}$. An *oriented graph* G is a pair (V, E) where V is as before and and $E \subset V \times V$; elements of E are called oriented edges. The set of *oriented graph* will be denote $\mathcal{G}$.

Example. $V = \{a, b, c, d\}$ and $E = \{(a, b), (c, a), (d, b), (a, d)\}$



A path of a graph is a finite sequence of vertices (at least two) such that there exists an edge between two successive vertices. A graph is *connected* if any pair of vertices are

member of at least one path. A *circuit* is a path whose first and last elements coincide. A *tree* G is a connected graph without circuits. The set of trees will be denote $\bar{\mathcal{T}}$. A *rooted tree* t is a pair of a tree and of one of its vertices, called the root, and denoted $r(t)$. The leaves are the vertices with only one adjacent edge, other than the root. We may identify a rooted tree with an oriented graph, where all edges point in direction of leaves.

The set of *rooted tree* will be denote $\mathcal{T}^*$. We sometimes speak of free trees, as opposed to rooted trees. We define in a similar manner oriented trees, also called H-trees in [12]. The set of *oriented free tree* is denoted $\mathcal{T}$.

Bicoloured graphs are graphs together with a mapping that to each vertex v associates a colour $c(v)$, of value B or W (black and white). The set of *bi-coloured oriented graphs* (resp. *bi-coloured rooted trees*) will be denote $\mathcal{BG}$ (resp. $\mathcal{BT}^*$).

We note $V_B = c^{-1}(\{B\})$ (resp. $V_W = c^{-1}(\{W\})$) the set of black (resp. white) vertices and E_B (resp. E_W) the set of edges ending on black (resp. white) vertice.

For a given rooted tree, whose vertices are associated with letters $i, j, \ldots$ it is convenient to denote

$$\begin{array}{rcl} \tilde{b}_{i_k} & = & b_{i_k} \text{ if vertex } k \text{ is white, } \hat{b}_{i_k} \text{ otherwise.} \\ \tilde{a}_{i_k i_\ell} & = & a_{ij} \text{ if vertex } \ell \text{ is white, } \hat{a}_{ij} \text{ otherwise.} \end{array} \tag{5}$$

Here i_k associates with each vertex $k \in \{1, \ldots, \#t\}$ a variable i_k (varying from 1 to s in the subsequent expressions).

With each rooted tree t is associated a value $\gamma(t)$ in an inductive way: $\gamma(\bullet) = 1$, and

$$\gamma(t) = \#t \times \gamma(t_1) \times \cdots \times \gamma(t_m), \tag{6}$$

where by $t_1, \ldots, t_m$ we denote the branches of t (the connected graphs obtained by removing the root from t, viewed as rooted trees, the root being the vertex whose one adjacent edge has been removed).

Next, we need to recall the definition of the elementary weight of a tree t, where t is a bicoloured rooted graph, for given Runge-Kutta coefficients a and b:

$$\phi(t) = \sum_{i=1}^{s} \tilde{b}_i \phi_i(t). \tag{7}$$

The above functions $\phi_i(t)$ are themself defined in an inductive way, by $\phi_i(\bullet) = \phi_i(\circ) = 1$ and

$$\begin{array}{rcl} \phi_i(t) & = & \psi_i(t_1) \times \cdots \times \psi_i(t_m), \\ \psi_i(t_k) & = & \sum_{j_k=1}^{s} \tilde{a}_{ij_k} \phi_{j_k} \end{array} \tag{8}$$

We have the following result (see [8, Thm III.2.5]):

Theorem 1. *A partitioned Runge-Kutta scheme has (global) order q iff*

$$\phi(t) = \frac{1}{\gamma(t)}, \quad \textit{for all } t \in \mathcal{BT}^*, \ \#t \le q. \tag{9}$$

Substituting the expressions of $\hat{a}$ and $\hat{b}$ in (4) in the formula of $\phi(t)$ would give complicated expressions. We show in the next section how to state equivalent, but simpler conditions.

4 Calculus on oriented free trees

In this section we choose as convention that the set of vertices V is of the form : $\{1,\dots,\#V\}$ we do not lose generality. Then we start by stating alternative expressions of the elementary weight $\phi(t)$, where t is a rooted bicoloured tree, defined in (9), for a general partitioned scheme. Taking into account the definition of functions $\psi_i(t_k)$, obtain the more explicit expression

$$\phi(t) = \sum_{i_1=1}^{s} \tilde{b}_{i_1} \sum_{i_2,\dots,i_{\#t}=1}^{s} \prod_{(k,\ell)\in E} \tilde{a}_{i_k i_\ell}. \tag{10}$$

Here again i_k associates with each vertex $k \in \{1,\dots,\#t\}$ (vertex 1 is the root) a variable i_k (varying from 1 to s). Formula (10) may be written as

$$\phi(t) = \sum_{v\in V} \sum_{i_v=1}^{s} \prod_{k\in V} \tilde{b}_{i_k} \prod_{(k,\ell)\in E} \frac{\tilde{a}_{i_k i_\ell}}{\tilde{b}_{i_\ell}}. \tag{11}$$

Indeed, all $\tilde{b}_{i_\ell}$ in the above expression cancel except $\tilde{b}_{i_1}$, so that (10) and (11) coincide. Observe, however, that the above formula makes sense for (non necessarily connected) bicoloured oriented graphs. Any such graph t is a finite union of connected graphs with empty intersection of vertices, called *connected components* of t, and denoted $\{t^q, q \in Q\}$ (not to be confused with branches of a rooted tree, denoted t_i).

Lemma 2. *The elementary weight of a bi-coloured oriented graph $t = (V, E, c)$, with connected components $\{t^q, q \in Q\}$, is the product of the elementary weights of its connected components, i.e.,*

$$\phi(t) = \prod_{q\in Q} \phi(t^q). \tag{12}$$

Proof. We have that

$$\begin{aligned}
\phi(t) &= \sum_{v\in V} \sum_{i_v=1}^{s} \prod_{k\in V} \tilde{b}_{i_k} \prod_{(k,\ell)\in E} \frac{\tilde{a}_{i_k i_\ell}}{\tilde{b}_{i_\ell}} \\
&= \sum_{v\in V} \sum_{i_v=1}^{s} \prod_{q\in Q} \left(\prod_{k\in V_q} \tilde{b}_{i_k} \prod_{(k,\ell)\in E_q} \frac{\tilde{a}_{i_k i_\ell}}{\tilde{b}_{i_\ell}} \right) \\
&= \prod_{q\in Q} \left(\sum_{v\in V_q} \sum_{i_v=1}^{s} \left(\prod_{k\in V_q} \tilde{b}_{i_k} \prod_{(x,y)\in E_q} \frac{\tilde{a}_{i_k i_\ell}}{\tilde{b}_{i_\ell}} \right) \right),
\end{aligned}$$

where the last equality uses the identity (valid for arbitrary families of sets I_q and functions A_q)

$$\prod_{q\in Q} \left(\sum_{i\in I_q} A_q(i) \right) = \sum_{r\in Q} \sum_{i\in I_r} \left(\prod_{q\in Q} A_q(i) \right). \tag{13}$$

The conclusion follows. □

Given an oriented graph $t = (V, E)$, and $F \subset E$, the set of arcs in opposite direction to those of F is denoted as

$$E^{\top} := \{(x, y) \in V \times V; \ \ (y, x) \in E\}. \tag{14}$$

Theorem 3. *The elementary weight of a bi-coloured oriented graph $t = (V, E, c)$, when (4) holds, satisfies*

$$\phi(t) = \sum_{\hat{E}_B \in \mathcal{P}(E_B)} (-1)^{\#\hat{E}_B} \phi(V, E_W \cup \hat{E}_B^{\top}). \tag{15}$$

Proof. Substituting the expressions of $\hat{a}$ and $\hat{b}$ in (4), we may write elementary weights as follows:

$$\phi(t) = \sum_{v \in V} \sum_{i_v=1}^{s} \prod_{k \in V} b_{i_k} \prod_{(k,\ell) \in E_W} \frac{a_{i_k i_\ell}}{b_{i_\ell}} \prod_{(k,\ell) \in E_B} \left(1 - \frac{a_{i_\ell i_k}}{b_{i_k}}\right). \tag{16}$$

Expanding the last term, get

$$\phi(t) = \sum_{\hat{E}_B \in P(E_B)} (-1)^{\#\hat{E}_B} \sum_{v \in V} \sum_{i_v=1}^{s} \prod_{k \in V} b_{i_k} \prod_{(k,\ell) \in E_W} \frac{a_{i_k i_\ell}}{b_{i_\ell}} \prod_{(k,\ell) \in \hat{E}_B} \frac{a_{i_\ell i_k}}{b_{i_k}}. \tag{17}$$

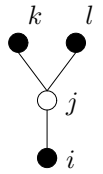
The conclusion follows. □

Our procedure computes order conditions in a recursive way. When dealing with order p, all conditions of order smaller than p have already been obtained. We claim that elementary weights of all terms in the sum of (15) have already been computed, except (perhaps) for the one in which $\hat{E}_B = E_B$.

Indeed, for an arc in $E_B \setminus \hat{E}_B$ then there is no contribution from this arc: we can interpret this as a deletion of the arc, whereas for an arc in E_B, we obtain the usual term (for a non partitioned Runge-Kutta scheme) except that we have $a_{i_\ell i_k}$ instead of $a_{i_k i_\ell}$, which would be the usual term if the direction of arc had been changed, and an associated minus sign.

Now whenever an arc is deleted, it follows from lemma 2 that the term is a product of elementary weights for trees of smaller size. Since black nodes have all possible locations, when conditions of order less than p were computed, the elementary weights of all trees of weight at most $p - 1$ have been computed.

The idea now is to focus on this graph interpretation, and compute order conditions. With theorem 15 we can associate to a bi-coloured tree a set of oriented graph for which only one is connected. This means that we can obtain order conditions for each graph. Let us now take a tutorial sample to show how the method works.

Bi-coloured tree t	
$\phi(t) = 1/\gamma(t)$	$\sum_{i,j,k,l=1}^{s} b_i \hat{a}_{ij} a_{jk} a_{jl} = \frac{1}{12}$
$\phi(t) = \phi(t_1) - \phi(t_2)$	$\sum_{i,j,k,l=1}^{s} b_i b_j a_{jk} a_{jl} - \sum_{i,j,k,l=1}^{s} b_j a_{ji} a_{jk} a_{jl}$
$t_1 - t_2$	

This example illustrates how we can compute order conditions associate with a given graph. In this example t_1 has two connected components, and hence, $\phi(t_1)$ is the product of two elementary weights that have been already evaluated (when dealing with order 4 conditions). Therefore the the new condition can be expressed as $\phi(t_2) = \phi(t_1) - 1/\gamma(t)$, where the right-hand-side reduces to a rational number.

Note that we get one order condition for each bi-coloured graph. Since there are less oriented free tree than bi-coloured trees, it may happen that all terms of the sum in (15) are already evaluated.

Remark. The above discussion is coherent with the result of Murua [12]: there exist as many order conditions for order n for an integration method as there exist oriented free tree with n nodes.

5 Implementation and numerical results

5.1 Implementation

Given a rooted bicoloured tree t, denote by $(\sigma_i, g_i)_i$ the associated family of oriented graphs obtained by the operation of either cutting of reversing black edges; denote by g^* the connected element of this family (obtained by reversing all black edges). Let $v(g)$ be the value of the elementary weight, stored in a data base called **Bank**. Since elementary weights

are rational numbers, we store them as a pair of irreducible integers. The procedure for obtaining order conditions at a given order n may be written as

OrderGen(n)
For $t \in \mathcal{BT}^*$ ordered by increasing degree, with $|t| \leq n$
  Compute the family of oriented graphs$(\sigma_i, g_i)_i$
  Evaluate elementary weights of all no-connected g_i
  If $g^* \in$ **Bank**
    Then check if $\sum_i \sigma_i v(g_i) + \sigma^* v(g^*) = 1/\gamma(t)$
    Else Bank $\leftarrow$ **Bank** $\cup \{v(g^*) = \sigma^*(1/\gamma(t) - \sum_i \sigma_i v(g_i))\}$
End For
Return Bank

Remark. The Gauss-Legendre Runge-Kutta method (in which a and b are equal to $\hat{a}$ and $\hat{b}$) is known to be symplectic, of order $2s$ where s is the number of stages. Therefore the check of compatibility conditions that occurs when g^* already belongs to **Bank** has to be satisfied. We use it only as a debugging tool.

Our implementation uses the program for trees generation of Li and Ruskey [11]. Then we build all possible colorations for a given tree, and apply procedure **OrderGen**. The code is written in the C language. The coloration procedure is as follows:

ColorGen(t)
If $t = \tau$ return [W,B]
Else Express t as links from root to branches $t = [t_1^{n_1} ... t_s^{n_s}]$
  For i=1 to s ColorGen(t_i)
  Combine all coloration into Tab
Return(Tab)

Remark. Let $n_c(t_i)$ denote the number of coloration of branch t_i. Then the number of colorations of t is

$$\prod_{i=1}^{s} \frac{n_c(t_i)!}{n_i!(n_c(t_i) - n_i)!}$$

This explain the combinatorial explosion and the limitation of our result to $n = 7$.

Remark. Most of the computing time is spent in the search of a given graph (whenever it exists) in the bank. Our implementation checks bijection with all graphs of same number of vertices and edges. Obviously this procedure could be improved. For order greater than 6, conditions are too numerous to be displayed. However, it can be useful to formulate them in order to compute the order for specific values of a and b.

Table 1: Number of order conditions

Order	1	2	3	4	5	6	7
Simple	1	1	2	4	9	20	48
Symplectic	1	1	3	8	27	91	350
Partitioned	2	4	14	52	214	916	4116

5.2 Computational result

We first display the number of order conditions in table 1. Observe the rapid increase of these numbers with p for symplectic schemes, and even more with general partitioned schemes.

In the next tables, we use the usual notations $d_j = \sum_i b_i a_{ij}$ and $c_i = \sum_j a_{ij}$. All indexes in the sum vary from 1 to s, where s is the number of stages in the Runge-Kutta method. All (latex source of) tables are generated automatically by the computer code. Conditions for order 1 to 4 were already obtained by Hager [7].

Table 2: Ordre 1

Graph	Condition
	$\sum b_i = 1$

Table 3: Ordre 2

Graph	Condition
	$\sum d_j = \frac{1}{2}$

Table 4: Ordre 3

Graph	Condition	Graph	Condition
	$\sum c_j d_j = \frac{1}{6}$		$\sum b_i c_i^2 = \frac{1}{3}$
	$\sum \frac{1}{b_k} d_k^2 = \frac{1}{3}$		

Table 5: Ordre 4

Graph	Condition	Graph	Condition
	$\sum \frac{1}{b_k} a_{lk} d_k d_l = \frac{1}{8}$		$\sum a_{jk} d_j c_k = \frac{1}{24}$
	$\sum \frac{b_i}{b_k} a_{ik} c_i d_k = \frac{5}{24}$		$\sum b_i a_{ij} c_i c_j = \frac{1}{8}$
	$\sum c_j^2 d_j = \frac{1}{12}$		$\sum b_i c_i^3 = \frac{1}{4}$
	$\sum \frac{1}{b_k} c_k d_k^2 = \frac{1}{12}$		$\sum \frac{1}{b_l^2} d_l^3 = \frac{1}{4}$

Table 6: Ordre 5

Graph	Condition	Graph	Condition
	$\sum \frac{b_i}{b_k} a_{ik} a_{il} d_k c_l = \frac{3}{40}$		$\sum b_i a_{ik} a_{ij} c_j c_k = \frac{1}{20}$
	$\sum a_{lk} a_{kj} c_j d_l = \frac{1}{120}$		$\sum b_i a_{ik} a_{kj} c_i c_j = \frac{1}{30}$
	$\sum \frac{b_i}{b_k} a_{lk} a_{il} c_i d_k = \frac{11}{120}$		$\sum \frac{b_i}{b_k} a_{lk} a_{ik} c_i d_l = \frac{3}{40}$
	$\sum \frac{b_i b_j}{b_k} a_{jk} a_{ik} c_i c_j = \frac{2}{15}$		$\sum \frac{b_i}{b_l b_m} a_{im} a_{il} d_l d_m = \frac{2}{15}$
	$\sum \frac{1}{b_k} a_{ml} a_{lk} d_k d_m = \frac{1}{30}$		$\sum \frac{1}{b_k} a_{mk} a_{lk} d_l d_m = \frac{1}{20}$
	$\sum \frac{1}{b_l b_m} a_{lm} d_l^2 d_m = \frac{1}{15}$		$\sum \frac{1}{b_k} a_{kl} d_k^2 c_l = \frac{1}{60}$
	$\sum \frac{1}{b_l^2} a_{ml} d_l^2 d_m = \frac{1}{10}$		$\sum \frac{b_i}{b_l^2} a_{il} c_i d_l^2 = \frac{3}{20}$
	$\sum \frac{1}{b_k} a_{lk} d_k c_l d_l = \frac{7}{120}$		$\sum a_{jk} c_j d_j c_k = \frac{1}{40}$
	$\sum \frac{1}{b_k} a_{lk} c_k d_k d_l = \frac{1}{40}$		$\sum \frac{b_i}{b_k} a_{ik} c_i c_k d_k = \frac{7}{120}$

Table 6: Ordre 5

Graph	Condition	Graph	Condition
	$\sum \frac{b_i}{b_k} a_{ik} c_i^2 d_k = \frac{3}{20}$		$\sum b_i a_{ij} c_i^2 c_j = \frac{1}{10}$
	$\sum a_{kj} c_j^2 d_k = \frac{1}{60}$		$\sum b_i a_{ij} c_i c_j^2 = \frac{1}{15}$
	$\sum c_j^3 d_j = \frac{1}{20}$		$\sum b_i c_i^4 = \frac{1}{5}$
	$\sum \frac{1}{b_k} c_k^2 d_k^2 = \frac{1}{30}$		$\sum \frac{1}{b_l^2} c_l d_l^3 = \frac{1}{20}$
	$\sum \frac{1}{b_m^3} d_m^4 = \frac{1}{5}$		

Table 7: Ordre 6

Graph	Condition	Graph	Condition
	$\sum \frac{b_i}{b_l b_m} a_{im} a_{il} a_{nl} d_m d_n = \frac{7}{144}$		$\sum \frac{b_i}{b_k} a_{im} a_{ik} a_{lk} d_l c_m = \frac{19}{720}$
	$\sum \frac{1}{b_k} a_{nm} a_{mk} a_{lk} d_l d_n = \frac{1}{72}$		$\sum \frac{b_i}{b_k} a_{im} a_{mk} a_{lk} c_i d_l = \frac{13}{360}$
	$\sum \frac{1}{b_k} a_{nk} a_{mn} a_{lm} d_k d_l = \frac{1}{144}$		$\sum a_{lm} a_{kl} a_{jk} d_j c_m = \frac{1}{720}$
	$\sum \frac{b_i}{b_k} a_{ik} a_{mk} a_{lm} c_i d_l = \frac{7}{360}$		$\sum \frac{b_i}{b_l b_m} a_{im} a_{in} a_{nl} d_l d_m = \frac{1}{18}$
	$\sum \frac{b_i}{b_k} a_{im} a_{il} a_{lk} d_k c_m = \frac{13}{360}$		$\sum \frac{b_i}{b_k} a_{im} a_{ml} a_{lk} c_i d_k = \frac{19}{720}$
	$\sum \frac{b_i}{b_k} a_{lm} a_{il} a_{ik} d_k c_m = \frac{7}{360}$		$\sum \frac{b_i b_j}{b_l b_m} a_{jm} a_{im} a_{il} c_j d_l = \frac{61}{720}$
	$\sum \frac{b_i b_j}{b_k} a_{jl} a_{jk} a_{ik} c_i c_l = \frac{7}{144}$		$\sum \frac{b_i b_j}{b_k} a_{jl} a_{lk} a_{ik} c_i c_j = \frac{1}{18}$
	$\sum b_i a_{kl} a_{jk} a_{ij} c_i c_l = \frac{1}{144}$		$\sum b_i a_{il} a_{ik} a_{kj} c_j c_l = \frac{1}{72}$

Table 7: Ordre 6

Graph	Condition	Graph	Condition
	$\sum \frac{b_i}{b_k} a_{ik} a_{il} d_k c_l^2 = \frac{7}{180}$		$\sum b_i a_{ik} a_{ij} c_j^2 c_k = \frac{1}{36}$
	$\sum a_{lk} a_{kj} c_j^2 d_l = \frac{1}{360}$		$\sum b_i a_{ik} a_{kj} c_i c_j^2 = \frac{1}{72}$
	$\sum \frac{b_i}{b_k} a_{lk} a_{il} c_i^2 d_k = \frac{13}{180}$		$\sum b_i a_{jk} a_{ij} c_i^2 c_k = \frac{1}{36}$
	$\sum \frac{b_i}{b_k} a_{lk} a_{ik} c_i^2 d_l = \frac{19}{360}$		$\sum \frac{b_i b_j}{b_k} a_{jk} a_{ik} c_i^2 c_j = \frac{7}{72}$
	$\sum \frac{b_i}{b_l b_m} a_{im} a_{il} c_l d_l d_m = \frac{13}{360}$		$\sum \frac{b_i}{b_k} a_{il} a_{ik} c_k d_k c_l = \frac{1}{45}$
	$\sum \frac{1}{b_k} a_{ml} a_{lk} c_k d_k d_m = \frac{1}{180}$		$\sum \frac{b_i}{b_k} a_{il} a_{lk} c_i c_k d_k = \frac{7}{360}$
	$\sum \frac{1}{b_k} a_{mk} a_{lm} d_k c_l d_l = \frac{7}{360}$		$\sum a_{kl} a_{jk} c_j d_j c_l = \frac{1}{180}$
	$\sum \frac{1}{b_k} a_{mk} a_{lk} c_l d_l d_m = \frac{1}{45}$		$\sum \frac{b_i}{b_k} a_{ik} a_{lk} c_i c_l d_l = \frac{13}{360}$
	$\sum \frac{b_i}{b_m^2 b_n} a_{in} a_{im} d_m^2 d_n = \frac{7}{72}$		$\sum \frac{b_i}{b_l^2} a_{im} a_{il} d_l^2 c_m = \frac{19}{360}$
	$\sum \frac{1}{b_l^2} a_{nm} a_{ml} d_l^2 d_n = \frac{1}{36}$		$\sum \frac{b_i}{b_l^2} a_{im} a_{ml} c_i d_l^2 = \frac{13}{180}$
	$\sum \frac{1}{b_l b_m} a_{nm} a_{ln} d_l^2 d_m = \frac{1}{72}$		$\sum \frac{1}{b_k} a_{lm} a_{kl} d_k^2 c_m = \frac{1}{360}$
	$\sum \frac{1}{b_l b_m} a_{nm} a_{lm} d_l^2 d_n = \frac{1}{36}$		$\sum \frac{b_i}{b_l b_m} a_{im} a_{lm} c_i d_l^2 = \frac{7}{180}$
	$\sum \frac{1}{b_k} a_{lk} a_{lm} d_k d_l c_m = \frac{1}{60}$		$\sum a_{jl} a_{jk} d_j c_k c_l = \frac{1}{120}$
	$\sum \frac{1}{b_k} a_{mk} a_{kl} d_k c_l d_m = \frac{1}{240}$		$\sum \frac{b_i}{b_k} a_{ik} a_{kl} c_i d_k c_l = \frac{1}{80}$
	$\sum \frac{b_i}{b_k} a_{ik} a_{il} c_i d_k c_l = \frac{7}{120}$		$\sum b_i a_{ik} a_{ij} c_i c_j c_k = \frac{1}{24}$

Table 7: Ordre 6

Graph	Condition	Graph	Condition
	$\sum a_{lj}a_{jk}c_jc_kd_l = \frac{1}{240}$		$\sum b_ia_{ij}a_{jk}c_ic_jc_k = \frac{1}{48}$
	$\sum \frac{b_i}{b_lb_m}a_{lm}a_{il}c_id_ld_m = \frac{11}{240}$		$\sum \frac{b_i}{b_l^2}a_{ml}a_{il}c_id_ld_m = \frac{7}{120}$
	$\sum \frac{b_ib_j}{b_l^2}a_{jl}a_{il}c_ic_jd_l = \frac{11}{120}$		$\sum \frac{b_i}{b_k}a_{lk}a_{il}c_id_kc_l = \frac{11}{240}$
	$\sum \frac{b_i}{b_k}a_{lk}a_{ik}c_ic_kd_l = \frac{1}{60}$		$\sum \frac{b_ib_j}{b_k}a_{jk}a_{ik}c_ic_jc_k = \frac{1}{24}$
	$\sum \frac{1}{b_lb_m}a_{nm}a_{nl}d_ld_md_n = \frac{1}{24}$		$\sum \frac{1}{b_lb_m}a_{nl}a_{lm}d_ld_md_n = \frac{1}{48}$
	$\sum \frac{b_i}{b_lb_m}a_{im}a_{il}c_id_ld_m = \frac{11}{120}$		$\sum \frac{1}{b_k}a_{ml}a_{lk}d_kc_ld_m = \frac{1}{80}$
	$\sum \frac{1}{b_l^2}a_{nl}a_{ml}d_ld_md_n = \frac{1}{24}$		$\sum \frac{1}{b_k}a_{mk}a_{lk}c_kd_ld_m = \frac{1}{120}$
	$\sum \frac{1}{b_m^2b_n}a_{mn}d_m^3d_n = \frac{1}{24}$		$\sum \frac{1}{b_l^2}a_{lm}d_l^3c_m = \frac{1}{120}$
	$\sum \frac{1}{b_m^3}a_{nm}d_m^3d_n = \frac{1}{12}$		$\sum \frac{b_i}{b_m^3}a_{im}c_id_m^3 = \frac{7}{60}$
	$\sum \frac{1}{b_lb_m}a_{lm}c_ld_l^2d_m = \frac{1}{40}$		$\sum \frac{1}{b_k}a_{kl}c_kd_k^2c_l = \frac{1}{120}$
	$\sum \frac{1}{b_l^2}a_{ml}c_ld_l^2d_m = \frac{1}{60}$		$\sum \frac{b_i}{b_l^2}a_{il}c_ic_ld_l^2 = \frac{1}{30}$
	$\sum \frac{1}{b_k}a_{lk}d_kc_l^2d_l = \frac{1}{30}$		$\sum a_{jk}c_j^2d_jc_k = \frac{1}{60}$
	$\sum \frac{1}{b_k}a_{lk}c_k^2d_kd_l = \frac{1}{120}$		$\sum \frac{b_i}{b_k}a_{ik}c_ic_k^2d_k = \frac{1}{40}$
	$\sum \frac{b_i}{b_k}a_{ik}c_i^3d_k = \frac{7}{60}$		$\sum b_ia_{ij}c_i^3c_j = \frac{1}{12}$
	$\sum a_{kj}c_j^3d_k = \frac{1}{120}$		$\sum b_ia_{ij}c_ic_j^3 = \frac{1}{24}$

Table 7: Ordre 6

Graph	Condition	Graph	Condition
	$\sum \frac{1}{b_l b_m} a_{lm} d_l^2 c_m d_m = \frac{1}{90}$		$\sum \frac{1}{b_k} a_{kl} d_k^2 c_l^2 = \frac{1}{180}$
	$\sum \frac{1}{b_m^2 b_n} a_{nm} d_m^2 d_n^2 = \frac{1}{18}$		$\sum \frac{1}{b_l^2} a_{ml} d_l^2 c_m d_m = \frac{2}{45}$
	$\sum \frac{b_i}{b_l^2} a_{il} c_i^2 d_l^2 = \frac{19}{180}$		$\sum \frac{1}{b_k} a_{lk} c_k d_k c_l d_l = \frac{1}{72}$
	$\sum a_{jk} c_j d_j c_k^2 = \frac{1}{90}$		$\sum \frac{b_i}{b_k} a_{ik} c_i^2 c_k d_k = \frac{2}{45}$
	$\sum b_i a_{ij} c_i^2 c_j^2 = \frac{1}{18}$		$\sum c_j^4 d_j = \frac{1}{30}$
	$\sum b_i c_i^5 = \frac{1}{6}$		$\sum \frac{1}{b_k} c_k^3 d_k^2 = \frac{1}{60}$
	$\sum \frac{1}{b_l^2} c_l^2 d_l^3 = \frac{1}{60}$		$\sum \frac{1}{b_m^3} c_m d_m^4 = \frac{1}{30}$
	$\sum \frac{1}{b_n^4} d_n^5 = \frac{1}{6}$		

References

[1] J.T. Betts. Survey of numerical methods for trajectory optimization. *AIAA J. of Guidance, Control and Dynamics*, 21:193–207, 1998.

[2] J.T. Betts. *Practical methods for optimal control using nonlinear programming.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[3] J.F. Bonnans and G. Launay. Large scale direct optimal control applied to a re-entry problem. *AIAA J. of Guidance, Control and Dynamics*, 21:996–1000, 1998.

[4] J. C. Butcher. *Numerical methods for ordinary differential equations.* A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, 2003.

[5] A. L. Dontchev and William W. Hager. The Euler approximation in state constrained optimal control. *Math. Comp.*, 70(233):173–203, 2001.

[6] A. L. Dontchev, William W. Hager, and Vladimir M. Veliov. Second-order Runge-Kutta approximations in control constrained optimal control. *SIAM J. Numer. Anal.*, 38(1):202–226 (electronic), 2000.

[7] W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numer. Math.*, 87(2):247–282, 2000.

[8] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration.* Springer-Verlag, Berlin, 2002.

[9] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I.* Springer-Verlag, Berlin, second edition, 1993.

[10] E. Hairer and G. Wanner. *Solving ordinary differential equations. II.* Springer-Verlag, Berlin, second edition, 1996.

[11] G. Li and F. Ruskey. The advantages of forward thinking in generating rooted and free trees. In *10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).* http://www.theory.csc.UVic.CA/ fruskey/ ou http://www.theory.csc.UVic.CA/ cos/.

[12] A. Murua. On order conditions for partitioned symplectic methods. *SIAM J. Numer. Anal.*, 34(6):2204–2211, 1997.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399