

ORDER CONDITIONS FOR NONLINEARLY PARTITIONED RUNGE–KUTTA METHODS*

BRIAN K. TRAN[†], BEN S. SOUTHWORTH[†], AND TOMMASO BUVOLI[‡]

Abstract. Recently, a new class of nonlinearly partitioned Runge–Kutta (NPRK) methods was proposed for nonlinearly partitioned systems of autonomous ordinary differential equations $y' = F(y, y)$. The target class of problems are those in which different scales, stiffnesses, or physics are coupled in a nonlinear way, wherein the desired partition cannot be written in a classical additive or component-wise fashion. Here we use a rooted-tree analysis to derive full-order conditions for NPRK_M methods, where M denotes the number of nonlinear partitions. Due to the nonlinear coupling and thereby the mixed product differentials, it turns out that the standard node-colored rooted-tree analysis used in analyzing ODE integrators does not naturally apply. Instead we develop a new edge-colored rooted-tree framework to address the nonlinear coupling. The resulting order conditions are enumerated, are provided directly for up to fourth order with $M = 2$ and third order with $M = 3$, and are related to existing order conditions of additive and partitioned RK methods. We conclude with an example that shows how the nonlinear order conditions can be used to obtain an embedded estimate of the state-dependent nonlinear coupling strength in a dynamical system.

Key words. Runge–Kutta, order conditions, time integration, nonlinear coupling

AMS subject classifications. 65L05, 65L06, 65L70

1. Introduction. Nonlinearly partitioned Runge–Kutta (NPRK) methods [5] are a newly proposed family of time integrators for solving the *partitioned* initial value problem

$$(1.1) \quad y' = F(y, y), \quad y(t_0) = y_0.$$

NPRK methods can treat each argument of $F(y, y)$ with a different level of implicitness and may be interpreted as nonlinear generalizations of additive Runge–Kutta (ARK) methods [2, 11, 12]. An s -stage NPRK method for the partitioned initial value problem (1.1) is

$$(1.2) \quad \begin{aligned} Y_i &= y_n + h \sum_{j=1}^s \sum_{k=1}^s a_{ijk} F(Y_j, Y_k), \quad i = 1, \dots, s, \\ y_{n+1} &= y_n + h \sum_{i=1}^s \sum_{j=1}^s b_{ij} F(Y_i, Y_j), \end{aligned}$$

where Y_i are stage values, the rank-3 tensor a_{ijk} takes the place of the classical Runge–Kutta matrix a_{ij} , and the matrix b_{ij} replaces the classical weight vector b_i . To motivate the definition of the partitioned initial value problem (1.1), we first provide a simple motivating example (for another example see Section 5, where we consider the Lotka–Volterra equation).

*Received September 25, 2024. Accepted February 22, 2025. Published online on March 13, 2025. Recommended by Peter Benner. Corresponding author: B. K. Tran.

B. K. T. was supported by the Marc Kac Postdoctoral Fellowship at the Center for Nonlinear Studies at Los Alamos National Laboratory. B. S. S. was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20220174ER. T. B. was funded by the National Science Foundation under grant NSF-OIA-2327484. Los Alamos National Laboratory report number LA-UR-24-20566.

[†]Los Alamos National Laboratory, Theoretical Division, Los Alamos, New Mexico 87545
 ({btran, southworth}@lanl.gov).

[‡]Tulane University, Department of Mathematics, New Orleans, LA 70118 (tbuvoli@tulane.edu).



This work is published by ETNA and licensed under the Creative Commons license [CC BY 4.0](#).

EXAMPLE 1.1 (Viscous Burgers' equation). To provide intuition for how partitioned initial value problems can lead to more efficient methods, we consider the viscous Burgers' equation with viscosity coefficient $\epsilon \geq 0$ given by

$$u_t = \epsilon u_{xx} + uu_x,$$

which, after spatial semi-discretization, takes the form

$$(1.3) \quad \mathbf{u}' = \epsilon D\mathbf{u} + \text{diag}(\mathbf{u})A\mathbf{u} =: f(\mathbf{u}),$$

where D and A denote the discrete diffusion and advection matrices, respectively. It is instructive to consider three standard first-order methods for integrating (1.3):

$$\begin{aligned} \text{Implicit:} & \quad \mathbf{u}_{n+1} = \mathbf{u}_n + hf(\mathbf{u}_{n+1}), \\ \text{Rosenbrock:} & \quad (I - hJ_n)(\mathbf{u}_{n+1} - \mathbf{u}_n) = hf(\mathbf{u}_n), \\ \text{IMEX:} & \quad \mathbf{u}_{n+1} = \mathbf{u}_n + h\epsilon D\mathbf{u}_{n+1} + h\text{diag}(\mathbf{u}_n)A\mathbf{u}_n, \end{aligned}$$

where J_n denotes the Jacobian of f at \mathbf{u}_n . The implicit method is of course stable independent of ϵ but requires a nonlinear solve. The stability and accuracy of the Rosenbrock method are comparable to that of the implicit method, but it only requires a linear solve involving the Jacobian [5]. The IMEX method requires only a linear solve; however, it reduces to the explicit Euler method when the viscosity coefficient satisfies $\epsilon \rightarrow 0$, and therefore it can suffer from stability issues for small ϵ . In [5] we show how formulating (1.3) as a partitioned initial value problem and subsequently applying an NPRK method can retain the advantages of each of the above methods while eliminating their respective disadvantages. Namely, note that (1.3) can be expressed as a partitioned initial value problem $\mathbf{u}' = F(\mathbf{u}, \mathbf{u})$, where

$$F(\mathbf{u}, \mathbf{v}) := \epsilon D\mathbf{u} + \text{diag}(\mathbf{v})A\mathbf{u}.$$

A corresponding first-order NPRK method is given by

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + hF(\mathbf{u}_{n+1}, \mathbf{u}_n) \\ &= \mathbf{u}_n + h\epsilon D\mathbf{u}_{n+1} + \text{diag}(\mathbf{u}_n)A\mathbf{u}_{n+1}. \end{aligned}$$

Note that this method is linearly implicit for all ϵ and provides the same accuracy and stability as the implicit and Rosenbrock method while only requiring a linear solve at each step and avoiding computing the Jacobian (for numerical results comparing these methods, see [5]).

The NPRK framework facilitates efficient integration of equations with stiff terms that cannot be isolated additively or component-wise. Specifically, any unpartitioned system $y' = f(y)$ can be converted into (1.1) by selecting a function $F(y, z)$ that satisfies $f(y) = F(y, y)$; if integrated with an IMEX-NPRK method (i.e., $a_{ijk} = 0$ for $j > i$ and $k \geq i$), then implicit solves are only required over the first argument of F , e.g., as discussed in Example 1.1. This approach has been previously applied to solve a number of problems using the related, though less general, family of semi-implicit integrators [3, 4, 16]. In [5], we had success applying NPRK methods to nonlinear partitions of challenging thermal radiative transfer and radiation hydrodynamics problems as studied in [15, 16], and we have ongoing work applying the framework to other multiphysics systems.

In this paper, order conditions are derived for the NPRK method (1.2) and the more general NPRK $_M$ method, described later in (2.4), that allows for M nonlinear partitions. We derive the order conditions using edge-colored rooted trees that represent the elementary differentials in the Taylor series expansion of (1.1). This work generalizes our previous results from [5], which has only investigated order conditions for a simple subclass of NPRK methods called sequentially-coupled methods.

1.1. Main contribution. This paper is the first to analyze the full-order conditions for the NPRK method (1.2) and the more general class of NPRK_M methods (2.5). In our previous work [5], we have only considered a simplified family of NPRK methods whose order conditions are equivalent to those of ARK methods. In this work, we show that ARK order conditions are insufficient for the general class of NPRK methods due to nonlinear coupling terms between the arguments of a nonlinear partition.

We first review NPRK methods in Section 2. To derive order conditions for NPRK methods, we introduce a new edge-colored rooted tree framework and use it to first derive order conditions for two partitions in Section 3.2, Theorem 3.2, and then generalize this to M -partitions in Section 3.3, Theorem 3.4. Full-order conditions for up to 5 partitions and 8th order are enumerated in Section 3.4. Interestingly, an NPRK_M method has $1/M$ times the number of order conditions of an ARK_M method, which is due to the inherent tensorial structure of an NPRK tableau. Furthermore, in Section 3.4, we present an algorithm for computing NPRK order conditions, and explicit order conditions up to fourth order for $M = 2$ and third order for $M = 3$ are provided in Appendix A.

NPRK methods and the underlying order conditions are then related to ARK and PRK methods in Section 4. By relating NPRK order conditions to ARK order conditions, we show that NPRK order conditions contain new order conditions corresponding to nonlinear couplings that vanish when considering additive partitions. As an application of these nonlinear order conditions, in Section 5, we show through an example how to obtain an embedded estimate for the nonlinear coupling strength in a system. The code for all of the numerical results obtained in the paper, including numerical and symbolic implementations of the NPRK order conditions, as well as for the numerical example is available at [17].

2. Nonlinearly partitioned Runge–Kutta methods. We introduce several definitions and properties of NPRK methods (1.2) that are relevant for determining and analyzing order conditions.

2.1. Nonlinear partitions. Consider an ordinary differential equation (ODE) $\dot{y} = f(y)$ on a vector space X , specified by a vector field $f : X \rightarrow X$. We say that a mapping $F : X \times X \rightarrow X$ is a *nonlinear partition* of $f : X \rightarrow X$ if

$$F(y, y) = f(y) \quad \text{for all } y \in X.$$

Nonlinear partitions for a given $f(y)$ are not unique, and different choices will affect the stability, accuracy, and computational efficiency of an NPRK method. For example, even a simple scalar function like $f(y) = y^2$ can be nonlinearly partitioned into an infinite number of ways: $F(u, v) = u^p v^{2-p}$, $p \in \mathbb{R}$, is one such example.

2.2. The underlying RK and ARK integration methods. For certain nonlinear partitions, the NPRK methods (1.2) reduce to simpler RK families.

DEFINITION 2.1. A classical RK integrator (a_{ij}, b_i, c_i) is an underlying RK method of an NPRK integrator if (1.2) reduces to the Runge–Kutta method corresponding to (a_{ij}, b_i, c_i) when the function $F(x, y)$ depends only on x or only on y .

If $F(u, v) = f(u)$, then (1.2) reduces to the classical RK method

$$(2.1) \quad a_{ij}^{\{1\}} = \sum_{k=1}^s a_{ijk}, \quad b_i^{\{1\}} = \sum_{j=1}^s b_{ij}, \quad c_i^{\{1\}} = \sum_{j=1}^s a_{ij}^{\{1\}}.$$

Similarly, if $F(u, v) = f(v)$, then (1.2) reduces to

$$(2.2) \quad a_{ik}^{\{2\}} = \sum_{j=1}^s a_{ijk}, \quad b_j^{\{2\}} = \sum_{i=1}^s b_{ij}, \quad c_i^{\{2\}} = \sum_{k=1}^s a_{ik}^{\{2\}} = c_i^{\{1\}}.$$

If the NPRK method (1.2) is applied to a nonlinear partition with additive structure

$$(2.3) \quad F(u, v) = F^{\{1\}}(u) + F^{\{2\}}(v),$$

then it reduces to an ARK method [11], which we call the *underlying ARK method*.

DEFINITION 2.2. *The underlying ARK method of (1.2) is*

$$(2.4) \quad \begin{aligned} Y_i &= y_n + h \sum_{j=1}^s \left[a_{ij}^{\{1\}} F^{\{1\}}(Y_j) + a_{ij}^{\{2\}} F^{\{2\}}(Y_j) \right], \\ y_{n+1} &= y_n + h \sum_{i=1}^s \left[b_i^{\{1\}} F^{\{1\}}(Y_i) + b_i^{\{2\}} F^{\{2\}}(Y_i) \right], \end{aligned}$$

whose coefficients are defined in (2.1) and (2.2).

Since NPRK methods maintain their order-of-accuracy for all nonlinear partitions, it follows that an order- q accurate NPRK method must have underlying methods (RK and ARK) that are also order- q accurate. In Section 3, we will distinguish the resulting RK and ARK order conditions from those that are due to a nonlinear coupling between the arguments of a nonlinear partition.

2.3. Generality of ARK and NPRK. NPRK methods generalize ARK methods in the sense that they allow for nonlinearly partitioned right-hand sides (1.1) but reduce to their underlying ARK method (2.4) on additively partitioned right-hand sides. We briefly discuss the generality of these two method families.

- *NPRK to ARK.* The NPRK family only contains ARK methods with shared abscissa $c_i^{\{1\}} = c_i^{\{2\}}$. An ARK method with differing abscissa cannot be an underlying ARK method because the abscissa of all underlying ARK methods are all formed by summing over the same indices of the tensor a_{ijk} :

$$c_i^{\{1\}} = \sum_j (\sum_k a_{ijk}) = \sum_k (\sum_j a_{ijk}) = c_i^{\{2\}}.$$

- *ARK to NPRK.* Given any ARK method \mathcal{M} with tableaux $(a^{\{1\}}, b^{\{1\}}, c^{\{1\}})$, $(a^{\{2\}}, b^{\{2\}}, c^{\{2\}})$, shared abscissa $c_i^{\{1\}} = c_i^{\{2\}}$, and $s > 1$ stages. It is then possible to construct a family of NPRK methods with \mathcal{M} as their underlying ARK method. This restriction imposes $2s^2$ conditions on the tensor a_{ijk} and $2s$ conditions on the matrix b_j , leading to a total of $(s^3 - 2s^2)$ free parameters in a_{ijk} and $s^2 - 2s$ free parameters in b_{ij} .

2.4. M -nonlinearly partitioned Runge–Kutta methods. The NPRK framework trivially generalizes to the M -component nonlinearly partitioned equation $y' = F(y, \dots, y)$, where F has M arguments. The ansatz for an s -stage NPRK $_M$ method is

$$(2.5) \quad \begin{aligned} Y_i &= y_n + h \sum_{j_1, \dots, j_M=1}^s a_{i, j_1, \dots, j_M} F(Y_{j_1}, \dots, Y_{j_M}), \quad i = 1, \dots, s, \\ y_{n+1} &= y_n + h \sum_{j_1, \dots, j_M=1}^s b_{j_1, \dots, j_M} F(Y_{j_1}, \dots, Y_{j_M}), \end{aligned}$$

where the tensors a and b are now of rank $M + 1$ and M , respectively. The method (2.5) nonlinearly combines M underlying RK methods and reduces to an ARK $_M$ method [11] when applied to an additively partitioned system $F(Y_1, \dots, Y_M) = \sum_{k=1}^M F^{\{k\}}(Y_k)$.

3. NPRK order conditions. In this section we present order conditions for the NPRK method (1.2) and the more general NPRK_M method (2.5). We begin by computing third-order NPRK conditions via Taylor expansions and then introduce edge-color rooted trees for determining order conditions for NPRK_M methods.

3.1. Order conditions from Taylor series. The most direct approach for deriving order conditions is to compare successive terms in the Taylor expansion of the exact solution to those of an NPRK method. In doing so, we will also see that the derivatives and order conditions are related to rooted trees with colored edges.

Let $y(t)$ denote the exact solution to the initial-value problem (1.1). In order to expand $y(t_0 + h)$ into a Taylor series at t_0 , we first compute the derivatives of y up to third order,

$$\begin{aligned}
 y' &= F(y, y), \\
 y'' &= D_1F(y, y)y' + D_2F(y, y)y', \\
 y''' &= D_{11}F(y, y)[y', y'] + 2D_{12}F(y, y)[y', y'] + D_{22}F(y, y)[y', y'] \\
 &\quad + D_1F(y, y)D_1F(y, y)y' + D_1F(y, y)D_2F(y, y)y' \\
 &\quad + D_2F(y, y)D_1F(y, y)y' + D_2F(y, y)D_2F(y, y)y'.
 \end{aligned}$$

Here, $D_iF(y_1, y_2) : X \rightarrow X$ denotes the derivative of F at (y_1, y_2) with respect to the i^{th} argument, which is a linear mapping whose action on $z \in X$ we simply write as $D_iF(y_1, y_2)z$. Similarly, $D_{ij}F(y_1, y_2) : X \times X \rightarrow X$ denotes the second derivative of F at (y_1, y_2) with respect to the j^{th} and then the i^{th} argument, which is a bilinear mapping whose action on $u, v \in X$ we write as $D_{ij}F(y_1, y_2)[u, v]$. Note that in the above identity, any instance of y' appearing on the right-hand side can be replaced with $F(y, y)$.

For brevity, throughout this section, unless the argument of F or its derivatives are written explicitly, we will assume that they are evaluated at (y_0, y_0) . Thus, the Taylor expansion of $y(t_0 + h)$ at t_0 is given by

$$\begin{aligned}
 y(t_0 + h) &= y(t_0) + hy'(t_0) + \frac{h^2}{2}y''(t_0) + \frac{h^3}{3!}y'''(t_0) + \mathcal{O}(h^4) \\
 &= y_0 + hF + \frac{h^2}{2} \left(D_1FF + D_2FF \right) \\
 &\quad + \frac{h^3}{3!} \left[D_{11}F[F, F] + 2D_{12}F[F, F] + D_{22}F[F, F] + D_1FD_1FF \right. \\
 &\quad \left. + D_1FD_2FF + D_2FD_1FF + D_2FD_2FF \right] + \mathcal{O}(h^4).
 \end{aligned}$$

Each summand in the derivatives (e.g., D_1FF or $D_{12}F[F, F]$) is called an *elementary differential*. We can find a pattern in the elementary differentials using edge-colored rooted trees, as shown in Table 3.1; we formalize this relationship in Section 3.2. For the moment, we simply remark that: (1) each factor in an elementary differential corresponds to a node, (2) the order of the derivatives (1 for D_i and 2 for D_{ij}) determines the number of outward edges, and (3) the edge color depends on whether differentiation is conducted with respect to the first or the second argument.

TABLE 3.1

Derivatives of y up to order 3 and the associated edge-colored rooted trees. The notation F_i and F_{ij} , respectively, abbreviates $D_i F$ and $D_{ij} F$. Thin dashed edges represent differentiation with respect to the first argument, while thick solid edges represent differentiation with respect to the second argument. Note that the third-order trees for $F_{12}[F, F]$ and $F_{21}[F, F]$ lead to a single-order condition.

Order 1	Order 2	Order 3			
\bullet F	$\begin{array}{c} \vdots \\ F_1[F] \end{array}$ $\begin{array}{c} \\ F_2[F] \end{array}$	$\begin{array}{c} \vee \\ F_{11}[F, F] \end{array}$	$\begin{array}{c} \vee \\ F_{12}[F, F] \end{array}$	$\begin{array}{c} \vee \\ F_{21}[F, F] \end{array}$	$\begin{array}{c} \vee \\ F_{22}[F, F] \end{array}$
		$\begin{array}{c} \vdots \\ F_1[F_1[F]] \end{array}$	$\begin{array}{c} \vdots \\ F_1[F_2[F]] \end{array}$	$\begin{array}{c} \vdots \\ F_2[F_1[F]] \end{array}$	$\begin{array}{c} \vdots \\ F_2[F_2[F]] \end{array}$

Next we expand the numerical solution obtained by applying one step of (1.2). A direct calculation yields

$$\begin{aligned}
 y_1 = & y_0 + h \sum_{ij} b_{ij} F + h^2 \sum_{ijkl} b_{ij} a_{ikl} D_1 F F + h^2 \sum_{ijkl} b_{ij} a_{jkl} D_2 F F \\
 & + h^3 \sum_{ijkluv} b_{ij} a_{ikl} a_{kuv} D_1 F D_1 F F + h^3 \sum_{ijkluv} b_{ij} a_{ikl} a_{luv} D_1 F D_2 F F \\
 & + h^3 \sum_{ijkluv} b_{ij} a_{jkl} a_{kuv} D_2 F D_1 F F + h^3 \sum_{ijkluv} b_{ij} a_{jkl} a_{luv} D_2 F D_2 F F \\
 & + \frac{h^3}{2} \sum_{ijkluv} b_{ij} a_{ikl} a_{iuv} D_{11} F[F, F] + \frac{h^3}{2} \sum_{ijkluv} b_{ij} a_{ikl} a_{juv} 2D_{12} F[F, F] \\
 & + \frac{h^3}{2} \sum_{ijkluv} b_{ij} a_{jkl} a_{juv} D_{22} F[F, F] + \mathcal{O}(h^4).
 \end{aligned}$$

Comparing the above expansions of $y(t_0 + h)$ and y_1 yields the order conditions up to order 3 in Table 3.2.

3.2. Order conditions via edge-colored rooted trees. The starting point for a systematic derivation of NPRK order conditions is to observe that for $y' = F(y, y)$, the total time derivative is given by

$$\frac{d}{dt}(\cdot) = D_1(\cdot) \cdot F + D_2(\cdot) \cdot F,$$

where $\cdot F$ denotes the insertion of F into the multilinear operator obtained after differentiation.

Based on this observation, we can represent the derivatives of y via edge-colored rooted trees. Every edge in a tree has one of two colors, which we represent graphically as dashed or solid, corresponding to differentiation with respect to the first or second argument, respectively, i.e.,

$$D_1(\cdot) = \begin{array}{c} \vdots \\ | \\ \vdots \end{array}, \quad D_2(\cdot) = \begin{array}{c} | \\ | \\ | \end{array}.$$

Letting $F = \bullet$, we can differentiate by grafting onto a rooted tree a dashed edge connected to a node and a solid edge connected to a node,

$$D_1(\cdot) \cdot F = \begin{array}{c} \bullet \\ \vdots \\ | \\ \vdots \end{array}, \quad D_2(\cdot) \cdot F = \begin{array}{c} \bullet \\ | \\ | \\ | \end{array}.$$

TABLE 3.2

The order conditions are $\Phi(\tau) = 1/\gamma(\tau)$. The horizontal dashed lines separate conditions of orders 1, 2, and 3. The order conditions consist of the well-known ARK order conditions, along with two additional conditions corresponding to a nonlinear coupling between the arguments of F .

Tree τ	Elementary Differential	Elementary Weight $\Phi(\tau)$	$\Phi(\tau)$ in terms of (2.1), (2.2)	$\gamma(\tau)$
\bullet	F	$\sum_{ij} b_{ij}$	$= \sum_i b_i^{\{1\}} = \sum_i b_i^{\{2\}}$	1
\vdots	$F_1[F]$	$\sum_{ijkl} b_{ij} a_{ikl}$	$= \sum_i b_i^{\{1\}} c_i$	2
\vdots	$F_2[F]$	$\sum_{ijkl} b_{ij} a_{jkl}$	$= \sum_i b_i^{\{2\}} c_i$	2
\vee	$F_{11}[F, F]$	$\sum_{ijkluv} b_{ij} a_{ikl} a_{iuv}$	$= \sum_i b_i^{\{1\}} c_i c_i$	3
\vee	$F_{12}[F, F]$	$\sum_{ijkluv} b_{ij} a_{ikl} a_{juv}$	n.a.	3
\vee	$F_{22}[F, F]$	$\sum_{ijkluv} b_{ij} a_{jkl} a_{juv}$	$= \sum_i b_i^{\{2\}} c_i c_i$	3
\vdots	$F_1[F_1[F]]$	$\sum_{ijkluv} b_{ij} a_{ikl} a_{kuv}$	$= \sum_{ij} b_i^{\{1\}} a_{ij}^{\{1\}} c_j$	6
\vdots	$F_1[F_2[F]]$	$\sum_{ijkluv} b_{ij} a_{ikl} a_{luv}$	$= \sum_{ij} b_i^{\{1\}} a_{ij}^{\{2\}} c_j$	6
\vdots	$F_2[F_1[F]]$	$\sum_{ijkluv} b_{ij} a_{jkl} a_{kuv}$	$= \sum_{ij} b_i^{\{2\}} a_{ij}^{\{2\}} c_j$	6
\vdots	$F_2[F_2[F]]$	$\sum_{ijkluv} b_{ij} a_{jkl} a_{luv}$	$= \sum_{ij} b_i^{\{2\}} a_{ij}^{\{2\}} c_j$	6

For example, the derivatives of y up to third order are given by

$$\begin{aligned}
 y' &= F = \bullet, \\
 y'' &= D_1 F F + D_2 F F = \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} + \begin{array}{c} \vdots \\ \vdots \end{array}, \\
 y''' &= \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} + \begin{array}{c} \vdots \\ \vdots \end{array} + \begin{array}{c} \vdots \\ \vdots \end{array} + \begin{array}{c} \vdots \\ \vdots \end{array} \\
 &\quad + \begin{array}{c} \vee \\ \vee \end{array} + \begin{array}{c} \vee \\ \vee \end{array} + \begin{array}{c} \vee \\ \vee \end{array} + \begin{array}{c} \vee \\ \vee \end{array}.
 \end{aligned}$$

$=_{D_1 F D_1 F F}$ $=_{D_2 F D_1 F F}$ $=_{D_1 F D_2 F F}$ $=_{D_2 F D_2 F F}$
 $=_{D_{11} F[F, F]}$ $=_{D_{21} F[F, F]}$ $=_{D_{12} F[F, F]}$ $=_{D_{22} F[F, F]}$

Given such an edge-colored tree τ , we denote by $\mathcal{F}(\tau)(y)$ the corresponding elementary differential obtained by differentiation and insertion of F as described above, e.g.,

$$\mathcal{F}(\bullet)(y) = F(y, y),$$

$$\mathcal{F} \left(\begin{array}{c} \bullet \\ \vdots \\ \bullet \end{array} \right) (y) = D_1 F(y, y) F(y, y).$$

Observe that, by symmetry of partial differentiation, the trees

$$\underbrace{\begin{array}{c} \bullet \quad \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array}}_{=D_{21}F[F,F]}, \quad \underbrace{\begin{array}{c} \bullet \quad \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array}}_{=D_{12}F[F,F]}$$

have the same elementary differential. We also have the usual internal symmetries of rooted trees. We define such trees to be equivalent and introduce a symmetry factor $\alpha(\tau)$ counting the elements of the equivalence class of such trees, e.g.,

$$\alpha \left(\begin{array}{c} \bullet \quad \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array} \right) = 2.$$

To be more precise, we introduce a recursive definition of edge-colored rooted trees, defining such a tree in terms of its children. Let τ be an edge-colored rooted tree, and let the order $|\tau|$ be the number of its nodes. Let τ_1, \dots, τ_m be the trees obtained by removing the root node and the edges connecting these trees to the root node. Let the edges connecting the root node of τ to τ_1, \dots, τ_m have colors $a_1, \dots, a_m \in \{1, 2\}$, respectively. Then, we express the tree τ as

$$\tau = [\tau_1|_{a_1}, \dots, \tau_m|_{a_m}].$$

The elementary differential of τ can then be defined recursively as

$$\begin{aligned} \mathcal{F}(\bullet)(y) &= F(y, y), \\ \mathcal{F}(\tau)(y) &= D_{a_1 \dots a_m} F(y, y) [\mathcal{F}(\tau_1)(y), \dots, \mathcal{F}(\tau_m)(y)] \quad \text{for } \tau = [\tau_1|_{a_1}, \dots, \tau_m|_{a_m}], \end{aligned}$$

where $D_{a_1 \dots a_m} F(y, y) : X \times \dots \times X \rightarrow X$ denotes the m -multilinear mapping given by the m th derivative of $F(y, y)$ with respect to its a_1, \dots, a_m arguments (for a discussion of higher-order derivatives as multilinear maps, see [6]).

The symmetry factor $\alpha(\tau)$ can be understood from the symmetry of the multilinear operator $D_{a_1 \dots a_m} F$. In determining order conditions, we only want to consider independent elementary differentials, so we only wish to sum over distinct trees corresponding to different elementary differentials. The symmetry factor can be computed recursively as $\alpha(\bullet) = 1$ and

$$\alpha(\tau) = \binom{|\tau| - 1}{|\tau_1|, \dots, |\tau_m|} \alpha(\tau_1) \cdots \alpha(\tau_m) \frac{1}{\mu_1! \cdots \mu_{j_m}!},$$

where $\tau = [\tau_1|_{a_1}, \dots, \tau_m|_{a_m}]$ and μ_1, \dots, μ_{j_m} count the number of mutually equal trees with the same root edge-coloring $\tau_1|_{a_1}, \dots, \tau_m|_{a_m}$, whose representatives we index as $\{1, \dots, j_m\}$. To see this, we consider labeling each non-root node with a number $1, \dots, |\tau| - 1$; the multinomial coefficient gives the number of possible partitions of $1, \dots, |\tau| - 1$ to the trees τ_1, \dots, τ_m . For each tree τ_i , there are $\alpha(\tau_i)$ ways of assigning these labels. Finally, we divide by $\mu_1! \cdots \mu_{j_m}!$ since permutations of equal trees with the same root edge-coloring $\tau_1|_{a_1}, \dots, \tau_m|_{a_m}$ do not change the labeling. This is derived in [1], with the slight modification that, in our case, a recursively defined edge-colored rooted tree $\tau = [\tau_1|_{a_1}, \dots, \tau_m|_{a_m}]$ keeps track of the root edge labelings, so we appropriately modify the definition of the counting

factors μ_1, \dots, μ_{j_m} . Throughout, when we sum over trees of order q , we understand that we are summing over equivalence classes of trees of order q , denoted as $|\tau| = q$. In other words, a sum over non-identified trees of order q can be expressed as a sum over equivalence classes of trees of order q by introducing the symmetry factor α , i.e.,

$$\sum_{\substack{\text{non-identified} \\ \text{trees of order } q}} (\dots) = \sum_{|\tau|=q} \alpha(\tau)(\dots),$$

given that the summand (\dots) is invariant on the equivalence classes. In particular, the elementary differentials are invariant on equivalence classes, due to the symmetry of the multilinear operator $D_{a_1 \dots a_m} F$.

Then, we have that the q^{th} derivative can be expressed as a sum over edge-colored trees of order q .

PROPOSITION 3.1. *For the exact solution $y(t)$, its q^{th} derivative is given by*

$$(3.1) \quad \left. \frac{d^q}{dt^q} y \right|_{t_0} = \sum_{|\tau|=q} \alpha(\tau) \mathcal{F}(\tau)(y_0).$$

Note that the stages can be viewed as functions of h , $Y_i = Y_i(h)$, implicitly defined by (1.2). Now, we define the following function of h , $g_{ij}(h) := hF(Y_i(h), Y_j(h))$. By the Leibniz rule (see [9, Sect. III.1.1, Eq. (1.8)]), we have

$$(3.2) \quad \left. \frac{d^q}{dh^q} g_{ij} \right|_{h=0} = q \left. \frac{d^{q-1}}{dh^{q-1}} F(Y_i, Y_j) \right|_{h=0}.$$

The factor of q appearing on the right-hand side of (3.2) will produce additional integer factors in the Taylor expansion of the numerical solution at $h = 0$. To account for these additional integer factors, noting that $q = |\tau|$, we define the density of a tree $\gamma(\tau)$ as the product of the order of τ with all orders of trees that appear if roots are successively removed, i.e.,

$$\begin{aligned} \gamma(\bullet) &= 1, \\ \gamma(\tau) &= |\tau| \gamma(\tau_1) \cdots \gamma(\tau_m), \end{aligned}$$

where $\tau = [\tau_1|_{a_1}, \dots, \tau_m|_{a_m}]$.

Furthermore, the above formula (3.2) allows us to inductively construct the elementary weights associated to a tree. To construct the elementary weights, we first note that (3.2) provides an expression for the derivative of the internal stages of an NPRK method (1.2):

$$\left. \frac{d^q}{dh^q} Y_i \right|_{h=0} = q \sum_{kl} a_{ikl} \left. \frac{d^{q-1}}{dh^{q-1}} F(Y_k, Y_l) \right|_{h=0},$$

which is seen by taking the q^{th} derivative of (1.2) at $h = 0$ and substituting (3.2). We associate to $F(Y_k, Y_l)$ a multi-indexed node \bullet_{kl} . Differentiation of $F(Y_k, Y_l)$ with respect to the first argument produces a factor $\sum_{uv} a_{kuv}(\dots)$ (here \dots denotes differentials evaluated at (Y_u, Y_v)) and similarly $\sum_{uv} a_{luv}(\dots)$ for the second argument. Letting a tree τ have root labeled ij , we define the *indexed elementary weight* $\phi_{ij}(\tau)$ as follows: label the remaining nodes in the tree with a multi-index; between a pair of nodes labeled kl below and uv above, if they are connected by a dashed edge, we obtain a factor a_{kuv} and similarly a_{luv} for a solid edge; once we have traversed through all non-rooted nodes in the tree, we sum over the non-rooted indices. That is, differentiation with respect to the first argument produces a

factor a_{kuv} , where the first index k corresponds to the first component of the multi-index of the lower node, and differentiation with respect to the second argument produces a factor a_{luv} , where the first index l corresponds to the second component of the multi-index of the lower node. For example,

$$\phi_{ij} \left(\begin{array}{c} kl \bullet \\ \vdots \\ ij \bullet \end{array} \right) = \sum_{kluv} a_{ikl} a_{j uv}, \quad \phi_{ij} \left(\begin{array}{c} uv \\ \vdots \\ kl \\ \vdots \\ ij \bullet \end{array} \right) = \sum_{kluv} a_{jkl} a_{kuv}$$

(note: the multi-indices kl and uv in the arguments of ϕ_{ij} above are dummy indices only shown for conceptual clarity). The preceding discussion produces a formula for the q^{th} derivative of g_{ij} at $h = 0$,

$$\frac{d^q}{dh^q} g_{ij} \Big|_{h=0} = \sum_{|\tau|=q} \alpha(\tau) \gamma(\tau) \phi_{ij}(\tau) \mathcal{F}(\tau)(y_0).$$

This yields a formula for the q^{th} derivative of $y_1 = y_0 + \sum_{ij} b_{ij} g_{ij}$ at $h = 0$,

$$(3.3) \quad \frac{d^q}{dh^q} y \Big|_{h=0} = \sum_{|\tau|=q} \alpha(\tau) \gamma(\tau) \sum_{ij} b_{ij} \phi_{ij}(\tau) \mathcal{F}(\tau)(y_0).$$

We define the *elementary weight* of a tree as $\Phi(\tau) = \sum_{ij} b_{ij} \phi_{ij}(\tau)$. Comparing equations (3.1) and (3.3), we obtain the order conditions for an NPRK method:

THEOREM 3.2 (NPRK order conditions). *An NPRK method has order p if and only if*

$$\Phi(\tau) = \frac{1}{\gamma(\tau)}$$

for all trees τ such that $|\tau| \leq p$.

Proof. From the above discussion, we have already proven the “if” direction, so it remains to prove the “only if” direction. As is standard (e.g., see [10, Sect. II.2, Thm. 2.13]), it suffices to show that the elementary differentials are independent. To show that they are independent, it suffices to show that for every tree τ (more precisely, for the equivalence class corresponding to τ), there exists a partitioned ODE $y' = F(y, y)$ on \mathbb{R}^k , for some k , with initial condition $y(0)$ such that

$$(3.4) \quad \mathcal{F}_k(\tau)(y(0)) = 1,$$

where $\mathcal{F}_k(\tau)(y(0))$ denotes the k^{th} component of $\mathcal{F}(\tau)(y(0))$, and that

$$\mathcal{F}_k(\sigma)(y(0)) = 0$$

for all other trees $\sigma \neq \tau$. We adapt the proof of the analogous result for Runge–Kutta methods (see [10, Sect. II.2, Thm. 2.13 and Exercise 4]) to account for edge coloring.

For a tree of order q , we define an ODE on $\mathbb{R}^q \ni y = (y_1, \dots, y_q)$ with zero initial condition $y(0) = (0, \dots, 0)$ as follows: For the tree \bullet of order 1, we define the ODE to be simply $y'_1 = 1$. Let τ be a tree of order q , expressed in recursive form

$$\tau = [\tau_{q-1}|_{a_1}, \dots, \tau_{q-m}|_{a_m}],$$

where of course the number of children m is less than q . We define the ODE to be $y' = f(y)$, where the q^{th} component of f is

$$(3.5) \quad f(y)_q = \prod_{k=1}^m y_{q-k}.$$

Let A_1 be the set of all indices $j \in \{1, \dots, m\}$ such that $a_j = 1$, and similarly let A_2 be the set of all indices $j \in \{1, \dots, m\}$ such that $a_j = 2$. Then, define the q^{th} component of the partition, corresponding to (3.5), to be

$$F_q(x, y) = \prod_{k \in A_1} x_{q-k} \prod_{k \in A_2} y_{q-k}.$$

Now, recall that the elementary differential at $y(0) = 0$ is given in recursive form as

$$\mathcal{F}(\tau)(0) = D_{a_1 \dots a_m} F(0, 0) [\mathcal{F}(\tau_{q-1})(0), \dots, \mathcal{F}(\tau_{q-m})(0)].$$

Let us denote by $[D_{a_1 \dots a_m} F(0, 0)]_{j_1, \dots, j_m}^i$ the components of the multi-linear operator $D_{a_1 \dots a_m} F(0, 0)$. Then the only non-zero component is

$$[D_{a_1 \dots a_m} F(0, 0)]_{q-1, \dots, q-m}^q = 1$$

by construction. Now, we finish by induction. Assume that (3.4) holds for all $k < q$. Then

$$\begin{aligned} \mathcal{F}_q(\tau)(0) &= \sum_{j_1, \dots, j_m=1}^q [D_{a_1 \dots a_m} F(0, 0)]_{j_1, \dots, j_m}^q [\mathcal{F}_{j_1}(\tau_{q-1})(0), \dots, \mathcal{F}_{j_m}(\tau_{q-m})(0)] \\ &= [D_{a_1 \dots a_m} F(0, 0)]_{q-1, \dots, q-m}^q [\mathcal{F}_{q-1}(\tau_{q-1})(0), \dots, \mathcal{F}_{q-m}(\tau_{q-m})(0)] = 1. \end{aligned}$$

A similar argument shows that $\mathcal{F}_q(\sigma)(0) = 0$ for all $\sigma \neq \tau$, which completes the proof. \square

REMARK 3.3. Note that, in principle, there is an alternative method to derive the order conditions for NPRK methods by viewing them as PRK methods. Namely, for the ODE $y' = F(y, y)$, $y(0) = y_0$, one can consider the equivalent partitioned ODE $y' = F(y, z)$, $z' = G(y, z)$, $y(0) = y_0 = z(0)$. Subsequently, one can use the order condition theory for PRK methods to derive the order conditions for NPRK methods.

For example, this was done in [5] for a simplified class of sequentially-coupled NPRK methods with a highly sparse NPRK tableau. For an s -stage sequentially-coupled method, it can equivalently be expressed as that of an $(s-1)$ -stage PRK method. However, for a fully dense NPRK tableau, the corresponding PRK method would require s^2 stages; furthermore, there are many constraints that arise due to the duplication of variables. As such, we find this approach unnatural and instead work directly in the NPRK formulation. Furthermore, as we will see in Section 4, this direct approach will naturally reveal a separation between additive and nonlinear order conditions. We use these nonlinear order conditions in Section 5 to show how the nonlinear coupling strength can be estimated.

3.3. Generalization to M partitions. Analogous to how two-component ARK methods can be generalized to M components [7], NPRK methods can be generalized to M partitions. The order conditions follow in a conceptually similar manner to the $M = 2$ case.

Let M be a positive integer. Consider the initial value problem

$$y' = f(y), \quad y(t_0) = y_0,$$

specified by a vector field $f : X \rightarrow X$. We say a map

$$F : \underbrace{X \times \dots \times X}_{M \text{ times}} \rightarrow X$$

is an M -partition of $f : X \rightarrow X$ if $F(y, \dots, y) = f(y)$ for all $y \in X$. Given an M -partition F of f , the ansatz for an s -stage NPRK $_M$ method for the above initial value problem is

$$Y_{i_0} = y_n + h \sum_{i_1 \dots i_M=1}^s a_{i_0 i_1 \dots i_M} F(Y_{i_1}, \dots, Y_{i_M}), \quad i_0 = 1, \dots, s,$$

$$y_{n+1} = y_n + h \sum_{i_1 \dots i_M=1}^s b_{i_1 \dots i_M} F(Y_{i_1}, \dots, Y_{i_M}).$$

The rank- $(M + 1)$ tensor $a_{i_0 i_1 \dots i_M}$ replaces the classical RK matrix a_{ij} , and the rank- M tensor $b_{i_1 \dots i_M}$ replaces the classical RK weight vector b_i . The choice $M = 1$ reproduces the classical RK methods, and the choice $M = 2$ reproduces the NPRK methods as defined previously.

Every NPRK $_M$ method has M underlying RK schemes, where the r^{th} scheme, $r = 1, \dots, M$, is given by using a trivial partition in the r^{th} argument, $F(Y_{i_1}, \dots, Y_{i_M}) = f(Y_{i_r})$. The corresponding RK coefficients are

$$a_{i_0 i_r}^{\{r\}} = \sum_{i_1, \dots, \widehat{i_r}, \dots, i_M=1}^s a_{i_0 i_1 \dots i_M}, \quad b_{i_r}^{\{r\}} = \sum_{i_1, \dots, \widehat{i_r}, \dots, i_M=1}^s b_{i_1 \dots i_M}$$

(where $\widehat{i_r}$ denotes omission of that index in the sum).

To derive the order conditions, we note that the time derivative is given by

$$\frac{d}{dt}(\cdot) = D_1(\cdot) \cdot F + \dots + D_M(\cdot) \cdot F,$$

where again D_i denotes differentiation with respect to the k^{th} argument, $k = 1, \dots, M$; for example,

$$y'' = \frac{d}{dt} F(y, \dots, y)$$

$$= D_1 F(y, \dots, y) \cdot F(y, \dots, y) + \dots + D_M F(y, \dots, y) \cdot F(y, \dots, y).$$

The order conditions can be obtained analogously to the $M = 2$ case. Namely, we consider edge-colored rooted trees where each edge can be colored by one of M colors from $\{1, \dots, M\}$. We represent this graphically by writing the color adjacent to the edge. Differentiation is then given by grafting onto a rooted tree a colored edge connected to a node:

$$D_1(\cdot) \cdot F = \begin{array}{c} \bullet \\ | \\ 1 \end{array}, \quad \dots, \quad D_M(\cdot) \cdot F = \begin{array}{c} \bullet \\ | \\ M \end{array}.$$

For example, for two colors $a, b \in \{1, \dots, M\}$,

$$\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a \quad b \\ \bullet \end{array} = D_{ab} F[F, F], \quad \begin{array}{c} \bullet \\ | \\ b \\ \bullet \\ | \\ a \\ \bullet \end{array} = D_a F D_b F F.$$

where x is an indeterminate in a formal power series.

We will additionally count the number of coupling conditions for an NPRK_M method, i.e., order conditions that do not correspond to order conditions for the M underlying RK methods; we discuss this further in Section 4. This is straightforward as the order conditions for the M underlying RK methods arise from only considering trees whose edges have a single color. Thus, for an NPRK_M method, there are $M\alpha_i^{[1]}$ order conditions for the underlying methods. Denote the number of coupling conditions for an NPRK_M method by $\tilde{\sigma}_i^{[M]}$ ($M \geq 2$). Then,

$$\tilde{\sigma}_i^{[M]} = \sigma_i^{[M]} - M\alpha_i^{[1]}.$$

The enumeration of both the total number of order conditions and the number of coupling conditions is shown in Table 3.3 for the number of partitions M up to 5 and order i up to 8.

TABLE 3.3
Enumeration of NPRK_M order conditions for up to 5 partitions and order up to 8.

M	$\sigma_i^{[M]}$	i							
		1	2	3	4	5	6	7	8
1	$\sigma_i^{[1]}$	1	1	2	4	9	20	48	115
2	$\sigma_i^{[2]}$	1	2	7	26	107	458	2058	9498
	$\tilde{\sigma}_i^{[2]}$	0	0	3	18	89	418	1962	9268
3	$\sigma_i^{[3]}$	1	3	15	82	495	3144	20875	142773
	$\tilde{\sigma}_i^{[3]}$	0	0	9	70	468	3084	20731	142428
4	$\sigma_i^{[4]}$	1	4	26	188	1499	12628	111064	1006840
	$\tilde{\sigma}_i^{[4]}$	0	0	18	172	1463	12548	110872	1006380
5	$\sigma_i^{[5]}$	1	5	40	360	3570	37476	410490	4635330
	$\tilde{\sigma}_i^{[5]}$	0	0	30	340	3525	37376	410250	4634755

REMARK 3.5. Observe that the number of NPRK_M order conditions is $1/M$ times the number of ARK_M order conditions, i.e.,

$$\sigma_i^{[M]} = M\tilde{\sigma}_i^{[M]}.$$

This can be explained by the fact that each NPRK_M tree corresponds to M ARK_M trees. To see this, given an M -edge-colored tree, for the root node create M trees by coloring the root with one of M colors; for every other node, color them with the color of the edge below it and finally remove all of the edge colors (note that this procedure is purely for enumeration; it does not map NPRK elementary differentials correctly to ARK elementary differentials). By this procedure, the set of all equivalence classes of NPRK_M trees corresponds to the set of all equivalence classes of ARK_M trees with a fixed root color. In the reverse direction, fixing the root color of an ARK_M tree, given an ARK_M tree with that root color, remove the coloring of the root node, color each edge by the color of the node above the edge, and finally remove all node colorings. Then this will produce an NPRK_M tree.

In terms of tableaux coefficients, the reduced number of order conditions for NPRK vs. ARK methods, despite the larger number of mixed differentials to account for, can be seen as arising from the tensorial nature of the NPRK tableaux, which we discuss further in Section 4.

Computation of order conditions. We will now present an algorithm for computing the order conditions. To be concrete, we consider the case of $M = 2$ partitions, although the case of a higher number of partitions follows similarly. We know each NPRK order condition can be obtained by an equivalence class of NPRK trees, as discussed in Section 3.2. To choose a representative of each equivalence class, we begin with a canonical representative of each equivalence class of ARK_2 trees given by a lexicographic ordering of the level and color sequences as described in [13]. By considering ARK_2 trees with a fixed root coloring, the procedure described in Remark 3.5 puts these trees in correspondence with canonical representatives of NPRK_2 trees. We utilize the `Julia` package `RootedTrees.jl` [14] to generate the canonical representative of each tree τ , whose canonical representative is specified by a level sequence $L(\tau)$ and a color sequence $C(\tau)$ describing the level of each node in lexicographic order and the color of each node, respectively. For the k^{th} element of the level sequence, $L(\tau)_k$, we denote by $\text{Parent}(\tau, k)$ the index of the parent node of node k , i.e., the largest index p in $L(\tau)$ such that $L(\tau)_k - 1 = L(\tau)_p$.

REMARK 3.6. Note that `RootedTrees.jl` colors canonical bicolored rooted trees with 0 and 1. We will let the colors 0 and 1 correspond to differentiation with respect to the first and second arguments, respectively. Furthermore, since we only consider ARK_2 trees with a fixed root color, we will just consider canonical trees where the root is colored by 0, as a matter of convention. As we will see in Algorithm 1 below, the first index of the color sequence is never used.

We then apply the method of obtaining the order conditions described in Section 3.2. Given a canonically represented tree as described above, we label each node with indices $i_k j_k$, where k is the integer where the node appears in the level sequence. For an upper node with indices $i_k j_k$ and a lower node with indices $i_l j_l$ connected by an edge of color $b \in \{0, 1\}$, we obtain a factor $a_{i_l i_k j_k}$ if $b = 0$ or $a_{j_l i_k j_k}$ if $b = 1$. We thus loop through all non-rooted nodes in the tree, multiply the factors obtained from them, multiply by $b_{i_1 j_1}$, where $i_1 j_1$ are the indices of the root node, and sum over all indices to obtain the left-hand side of the order condition, i.e., the elementary weight. The right-hand side of the order condition is given by the reciprocal of the density of the tree, $1/\gamma(\tau)$, which can also be obtained from `RootedTrees.jl`. This algorithm is summarized in Algorithm 1.

In addition to a numerical implementation of Algorithm 1 in `Julia`, we also implemented a symbolic version of this algorithm in `Mathematica` to generate the order conditions symbolically. For example, the third-order and fourth-order conditions for $M = 2$ are shown in Appendix A, equations (A.1) and (A.2), respectively. Third-order conditions for $M = 3$ are shown in (A.3). Both the numerical and symbolic implementations are available at [17].

4. Relation to additive order conditions. As we have seen, an additive partition for an NPRK method results in an ARK method. In this section, we will relate the previously obtained order conditions for NPRK methods to order conditions for ARK methods (see, for example, [1, 11] and [9, Section III.2]). Note this discussion equally applies to PRK methods, as they can be expressed as ARK methods.

Recall that we have shown that an NPRK method with tableaux (a_{ijk}, b_{ij}) with an additive partition reduces to a two-component ARK method, with the pair of tableaux $(a^{\{1\}}, b^{\{1\}}, c^{\{1\}})$ and $(a^{\{2\}}, b^{\{2\}}, c^{\{2\}})$ (see Section 2.2). We will now state a partial converse.

PROPOSITION 4.1. *Consider a two-component ARK method with the pair of tableaux $(a^{\{1\}}, b^{\{1\}}, c^{\{1\}})$ and $(a^{\{2\}}, b^{\{2\}}, c^{\{2\}})$. Additionally, assume that both tableaux are at least of first order $\sum_i b_i^{\{1\}} = 1 = \sum_j b_j^{\{2\}}$, have the same number of stages s , and satisfy $c^{\{1\}} = c^{\{2\}} = c$. Then, the method can be expressed as an NPRK method with an additive*

Algorithm 1 Order condition for a 2-NPRK tree.

Require: $a_{ijk}, b_{ij}, i, j, k = 1, \dots, s$

Require: Canonical tree τ , specified by (lexicographically ordered) level sequence $L(\tau)$, color sequence $C(\tau)$, density $\gamma(\tau)$.

$N \leftarrow \text{Length}(L(\tau))$

sum $\leftarrow 0$

for $i_1, j_1, \dots, i_N, i_N = 1; i_1, j_1, \dots, i_N, j_N \leq s; i_1, j_1, \dots, i_N, j_N++$ **do**

 prod $\leftarrow 1$

for $k = N; k \geq 2; k--$ **do**

if $C(\tau)_k == 0$ **then**

 prod \leftarrow prod $\ast a_{i_{\text{Parent}(\tau, k)} i_k j_k}$

else

 prod \leftarrow prod $\ast a_{j_{\text{Parent}(\tau, k)} i_k j_k}$

end if

end for

 sum \leftarrow sum $+ b_{i_1 j_1} \ast$ prod

end for

return sum $== 1/\gamma(\tau)$

partition with coefficients

$$(4.1a) \quad a_{ijk} = \frac{a_{ij}^{\{1\}}}{s} + \frac{a_{ik}^{\{2\}}}{s} - \frac{c_i}{s^2},$$

$$(4.1b) \quad b_{ij} = \frac{b_i^{\{1\}}}{s} + \frac{b_j^{\{2\}}}{s} - \frac{1}{s^2}.$$

Proof. We simply have to verify that the sums of a (respectively b) over its second and third (respectively second) indices reproduces the pair of underlying RK tableaux.

$$\sum_{k=1}^s a_{ijk} = \sum_{k=1}^s \left(\frac{a_{ij}^{\{1\}}}{s} + \frac{a_{ik}^{\{2\}}}{s} - \frac{c_i}{s^2} \right) = a_{ij}^{\{1\}} + \frac{c_i^{\{2\}}}{s} - \frac{c_i}{s} = a_{ij}^{\{1\}},$$

$$\sum_{j=1}^s a_{ijk} = \sum_{j=1}^s \left(\frac{a_{ij}^{\{1\}}}{s} + \frac{a_{ik}^{\{2\}}}{s} - \frac{c_i}{s^2} \right) = \frac{c_i^{\{1\}}}{s} + \frac{a_{ik}^{\{2\}}}{s} - \frac{c_i}{s} = a_{ik}^{\{2\}},$$

$$\sum_j b_{ij} = \sum_j \left(\frac{b_i^{\{1\}}}{s} + \frac{b_j^{\{2\}}}{s} - \frac{1}{s^2} \right) = b_i^{\{1\}} + \frac{1}{s} - \frac{1}{s} = b_i^{\{1\}},$$

$$\sum_i b_{ij} = \sum_i \left(\frac{b_i^{\{1\}}}{s} + \frac{b_j^{\{2\}}}{s} - \frac{1}{s^2} \right) = \frac{1}{s} + b_j^{\{2\}} - \frac{1}{s} = b_j^{\{2\}}. \quad \square$$

REMARK 4.2. Note that there can be different NPRK methods that have the same underlying ARK method. To see this, consider for example an ARK method with $b_i^{\{1\}} = b_i^{\{2\}}$, $i = 1, \dots, s \geq 2$. Then, the ARK method can be expressed as an NPRK method with the above choice of a and b , (4.1a)–(4.1b). Note that this choice of b is not generally diagonal. On the other hand, it can also be expressed as an NPRK method with the above choice of a and

$$(4.2) \quad b_{ij} = b_i^{\{1\}} \delta_{ij},$$

which is diagonal. These two possible choices correspond to different NPRK methods when applied to a general nonlinear partition but reduce to the same ARK method given an additive partition.

In light of this discussion, we are able to relate ARK order conditions with NPRK order conditions with additive partitions.

Relation to ARK order conditions. We will now investigate the relation of NPRK order conditions to ARK order conditions. In doing so, we will see that NPRK order conditions imply ARK order conditions but the converse is not true. In essence this is because the linearly separable structure of ARK methods remove some elementary differentials appearing in a general NPRK method; namely, those containing $D_I F$, where $I = i_1 \dots i_n$ is a multi-index such that at least two of its indices are different. In other words, a nonlinear coupling of stages gives rise to new order conditions.

We will first consider the two-component case. Consider an ODE

$$y' = f(y) = f_1(y) + f_2(y),$$

and define the partition

$$F(y_1, y_2) = f_1(y_1) + f_2(y_2).$$

To understand how to relate the ARK and NPRK order conditions, let us start with the first-order condition. For an NPRK method, the first-order condition is

$$\sum_{ij} b_{ij} = 1.$$

On the other hand, for an ARK method with coefficients $(a^{\{1\}}, b^{\{1\}}, c)$ and $(a^{\{2\}}, b^{\{2\}}, c)$, there are two order conditions

$$\sum_i b_i^{\{1\}} = 1 = \sum_j b_j^{\{2\}}.$$

The doubling of the number of order conditions arise from the fact that, for an NPRK method, $b^{\{1\}}$ and $b^{\{2\}}$ are not independent but rather related via the tensor b . Namely,

$$\sum_i b_i^{\{1\}} = \sum_{ij} b_{ij} = \sum_j b_j^{\{2\}}.$$

Thus, we only have to account for one first-order condition for an NPRK method, as the tensorial nature of the NPRK tableau automatically accounts for the other condition. We will visualize the process of converting an NPRK method to an ARK method by relating their rooted trees. For an NPRK method, we utilize edge-colored rooted trees as before; we refer to the dashed edge as having “color 1” and to the solid edge as having “color 2”. For an ARK method, it is standard to utilize node-colored rooted trees. We will color the nodes for an ARK method as

$$\circ, \bullet,$$

where the open node, referred to as “color 1”, corresponds to f_1 and the filled node, referred to as “color 2”, corresponds to f_2 . We can thus visualize the rooted tree with one node for an NPRK method by decomposing it into a pair of rooted trees for an ARK method,

$$\bullet ij \longrightarrow \circ_i, \bullet_j,$$

and thus obtaining a decomposition of the NPRK order condition as two ARK order conditions.

To understand decomposing higher-order NPRK trees into ARK trees, consider the derivative matrices of F expressed in terms of f_1 and f_2

$$\begin{aligned}
 D_1FF &= D_1f_1(f_1 + f_2) = D_1f_1f_1 + D_1f_1f_2, \\
 D_2FF &= D_2f_2(f_1 + f_2) = D_2f_2f_1 + D_2f_2f_2.
 \end{aligned}$$

This gives us a method of mapping an NPRK tree with two nodes connected by a colored edge to two ARK trees:

$$\begin{aligned}
 D_1FF &= \begin{array}{c} \bullet \\ \vdots \\ \bullet \end{array} \rightarrow \begin{array}{c} \circ \\ | \\ \circ \end{array} + \begin{array}{c} \bullet \\ | \\ \circ \end{array} = D_1f_1f_1 + D_1f_1f_2, \\
 D_2FF &= \begin{array}{c} \bullet \\ | \\ \bullet \end{array} \rightarrow \begin{array}{c} \circ \\ | \\ \bullet \end{array} + \begin{array}{c} \bullet \\ | \\ \bullet \end{array} = D_2f_2f_1 + D_2f_2f_2.
 \end{aligned}$$

Namely, given an NPRK tree with two nodes connected by an edge of color $a \in \{1, 2\}$, remove the coloring of the edge, color the lower node a , and form two trees that have upper nodes of color 1 and 2. Observe that the two second-order conditions for an NPRK method,

$$\sum_{ijkl} b_{ij} a_{ikl} = \frac{1}{2} = \sum_{ijkl} b_{ij} a_{jkl},$$

give the four second-order conditions for the underlying ARK method

$$\sum_i b_i^{\{1\}} c_i^{\{2\}} = \sum_i b_i^{\{1\}} c_i^{\{1\}} = \frac{1}{2} = \sum_j b_j^{\{2\}} c_j^{\{2\}} = \sum_j b_j^{\{2\}} c_j^{\{1\}}.$$

Analogous to the first-order case, the tensorial nature of the NPRK method means that one only has to account for two second-order conditions. Of course this is expected because if an NPRK method has order p , then the method with an additive partition must also have order p since the order conditions derived for an NPRK method were independent of the partition. However, we continue to investigate the order conditions at higher orders as it will reveal new nonlinear order conditions that are not present in the ARK order conditions.

To generalize to the higher-order case, we first introduce some terminology. We say that a node of a rooted tree is a *leaf* if it does not connect to any nodes above it, i.e., it has out-degree zero. We say that a node *branches* if it connects to at least two nodes above it, i.e., has out-degree greater than or equal to two. We say that an edge-colored rooted tree is *color-branching* if it contains a node that branches with at least two edges of different colors.

For the higher-order case, we repeat the procedure above from top to bottom: we split an NPRK tree into multiple ARK trees by starting with all possible colorings of the leaves. From there, the edges of the NPRK tree tell us how to color all of the lower nodes, as done above. At first, it may seem that this procedure is ill-defined since a node that branches with two different colors would not have a well-defined color. However, such trees need not be considered as the following proposition shows:

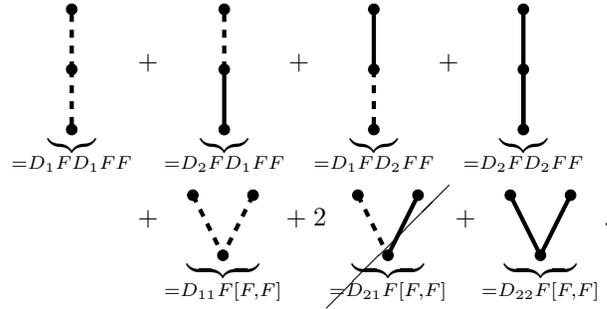
PROPOSITION 4.3. *For an NPRK method with an additive partition, the elementary differential of a color-branching tree vanishes.*

Proof. Given a color-branching tree where one of the nodes branches with two edges of different colors. Thus, the elementary differential contains a factor

$$\dots D_{12}F \dots \quad \text{or} \quad \dots D_{21}F \dots,$$

which vanishes since $D_{12}F = 0 = D_{21}F$ holds true for an additive partition $F(y_1, y_2) = f_1(y_1) + f_2(y_2)$. \square

This method is thus well-defined and allows us to conclude that all of the ARK order conditions of order p are satisfied if the NPRK order conditions of order p are satisfied (again, this is a trivial observation since if an NPRK method has order p , then the method with an additive partition must also have order p). For example, for third-order, we have 6 NPRK trees with nonzero differentials,



It is clear that the first four NPRK trees above correspond to 8 ARK trees. The last two nonzero trees correspond to 6 ARK trees,

$$\begin{aligned}
 D_{11}F[F, F] &= \text{[V-shaped tree with solid lines]} = \text{[V-shaped tree with solid lines]} + 2 \text{[V-shaped tree with solid lines]} + \text{[V-shaped tree with solid lines]} \\
 &= D^2 f_1[f_1, f_1] + 2D^2 f_1[f_1, f_2] + D^2 f_1[f_2, f_2],
 \end{aligned}$$

and similarly

$$D_{22}F[F, F] = D^2 f_2[f_1, f_1] + 2D^2 f_2[f_2, f_1] + D^2 f_2[f_2, f_2].$$

Thus, the 6 non-vanishing third-order NPRK trees give rise to 14 third-order ARK trees, which corresponds to the number of third-order conditions for two-component ARK methods [11].

The above construction shows that every NPRK condition corresponding to trees that are not color-branching can be expressed in terms of ARK order conditions. More precisely, Proposition 4.3 implies that any factor a_{kuv} in the elementary weight of a non-color-branching tree will always have either u or v as a free index, i.e., there are no other factors in the elementary weight depending on either u or v since otherwise the tree must be color-branching. Once the final summation over all non-rooted indices is performed, this means that a_{kuv} can be expressed as either $a_{kv}^{\{2\}}$ or $a_{ku}^{\{1\}}$. Thus, all factors of a appearing in the order condition of a non-color-branching tree can be expressed as either $a^{\{1\}}$ or $a^{\{2\}}$. Finally, since the root node of the tree, indexed by say ij , also cannot be color-branching, one of i or j must be free, and thus, b_{ij} can be expressed as either $b_i^{\{1\}}$ or $b_j^{\{2\}}$ in the order condition.

Conversely, we ask: Given an NPRK method (a, b, c) whose underlying coefficients $(a^{\{1\}}, b^{\{1\}}, c)$, $(a^{\{2\}}, b^{\{2\}}, c)$ satisfy the ARK order conditions to order p . Does the NPRK method satisfy the NPRK order conditions to order p for a generally nonlinear partition? For $p \leq 2$, the answer is affirmative since there are no color-branching trees of order 2. For $p \geq 3$, the answer is negative. This can be seen from the color-branching trees since we know that the order conditions for the non-color-branching trees will be satisfied. It suffices to consider the third-order condition

$$\sum_{ijkluv} b_{ij} a_{ikl} a_{juv} = 1/3$$

corresponding to the color-branching tree



Let $(a^{\{1\}}, b^{\{1\}}, c)$ and $(a^{\{2\}}, b^{\{2\}}, c)$ be coefficients for a (at least) third-order ARK method. Define a_{ijk} and b_{ij} by (4.1a)–(4.1b). Substituting (4.1a)–(4.1b) into the above order condition yields

$$\begin{aligned} \frac{1}{3} &= \sum_{ijkluv} b_{ij} a_{ikl} a_{juv} = \sum_{ijkluv} \left(\frac{b_i^{\{1\}}}{s} + \frac{b_j^{\{2\}}}{s} - \frac{c_i}{s^2} \right) a_{ikl} a_{juv} \\ &= \frac{1}{s} \sum_{ij} b_i^{\{1\}} c_i c_j + \frac{1}{s} \sum_{ij} b_j^{\{2\}} c_j c_i - \frac{1}{s^2} \sum_{ij} c_i c_j \\ &= \frac{1}{2s} \sum_j c_j + \frac{1}{2s} \sum_i c_i - \frac{1}{s^2} \left(\sum_i c_i \right)^2, \end{aligned}$$

where in the second line we used the second-order conditions $\sum_i b_i^{\{1\}} c_i = 1/2 = \sum_j b_j^{\{2\}} c_j$. Letting $x := \sum_i c_i$, then the order condition can be expressed

$$\frac{1}{3} = \frac{x}{s} - \frac{x^2}{s^2}.$$

Viewed as a quadratic in x , the discriminant is $-1/(3s^2)$, which is negative since $s \in \mathbb{R} \setminus \{0\}$. Thus, the above equation has no real solutions, so the condition cannot be satisfied for any choice of x since $c_i \in \mathbb{R}$.

Thus, we have shown that the underlying methods of an NPRK method satisfying ARK order conditions to order p is not sufficient for the NPRK method to have order p , for $p \geq 3$. In essence, this is due to the nonlinear coupling in an NPRK method with a generally nonlinear partition $F(Y_1, Y_2)$, which creates nonlinear order conditions corresponding to color-branching trees, starting at order $p \geq 3$. For ARK methods, the partition has a linearly separable structure which removes such color-branching trees from consideration since their elementary differentials vanish.

Thus, we cannot in general use order- p ARK methods to construct order- p NPRK methods via equations (4.1a)–(4.1b). However, since an ARK method does not arise from a unique NPRK method, it is still possible that some order- p ARK methods can produce an order- p NPRK method.

Specifically, for order 3, consider an order-3 ARK method with coefficients

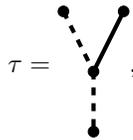
$$(a^{\{1\}}, b^{\{1\}}, c), (a^{\{2\}}, b^{\{2\}}, c) \quad \text{such that } b^{\{1\}} = b^{\{2\}},$$

such as a third-order Lobatto IIIA-IIIIB pair. Then, we can define an NPRK method with coefficients a via equation (4.1a) and b via equation (4.2). It is straightforward to verify that this method does satisfy the third-order condition for the order-3 color-branching tree

$$\sum_{ijkluv} b_{ij} a_{ikl} a_{juv} = 1/3,$$

since the diagonality of b allows one to write this condition in terms of non-color-branching order conditions.

For orders higher than 3, using a diagonal b will not work in general, due to color-branching trees such as



(i.e., color-branching trees which contains a non-rooted color-branching node) whose elementary weight is given by

$$\phi_{ij}(\tau) = \sum_{klmnuv} a_{ikl} a_{kmn} a_{luv}.$$

Here, a_{ikl} can not be expressed in terms of either $a^{\{1\}}$ or $a^{\{2\}}$ since k and l are not free. Despite the fact that an NPRK method does not necessarily have order p even if its underlying integrators have order p due to new nonlinear order conditions, the increased dimensionality of the NPRK tableaux and the reduced number of order conditions, compared to ARK methods, could lead to more flexibility in constructing higher-order NPRK methods. We will explore such constructions in subsequent work.

The generalization of relating NPRK_M order conditions to ARK_M order conditions follows similarly. Namely, the order conditions can be decomposed into the underlying RK order conditions for trees of a single color, linearly separable coupling conditions corresponding to trees which have more than one edge color but are non-color-branching, and nonlinearly separable coupling order conditions corresponding to color-branching trees. This again follows from the fact that the elementary differential for a color-branching tree vanishes, where now edges can have one of M colors, assuming an M -additive partition

$$F(Y_1, \dots, Y_M) = \sum_{i=1}^M f_i(Y_i).$$

An immediate corollary of this discussion is that an NPRK_M method (with generally nonlinear partition) is second-order if its M underlying RK methods are, since there are no trees of order 2 with more than one edge color.

5. Numerical example. In our companion paper [5], we have demonstrated the utility of NPRK methods for solving the viscous Burgers' and gray thermal radiation transport equations, particularly with regard to numerical stability. We presented several NPRK methods with differing levels of stability, order, and implicitness. Here, we will instead consider an example which exemplifies the nonlinear order condition theory developed in this paper. The code for this numerical example is available at [17].

We will show how the nonlinear order condition theory, discussed in Section 4, can be utilized to measure the state-dependent nonlinear coupling strength in a system. We consider the Lotka–Volterra system, with $y := [u, v]^T \in \mathbb{R}^2$,

$$(5.1) \quad \frac{d}{dt} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u - \alpha uv \\ v + \alpha uv \end{bmatrix},$$

where $\alpha \in \mathbb{R}$. Consider the Lobatto IIIA-III B pair with $s = 3$,

$$\left| \begin{array}{c} a^{\{1\}} \\ b^{\{1\}} \end{array} \right| = \begin{array}{ccc|ccc} 0 & 0 & 0 & 1/6 & -1/6 & 0 \\ 5/24 & 1/3 & -1/24 & 1/6 & 1/3 & 0 \\ 1/6 & 2/3 & 1/6 & 1/6 & 5/6 & 0 \\ \hline 1/6 & 2/3 & 1/6 & 1/6 & 2/3 & 1/6 \end{array}, \quad \left| \begin{array}{c} a^{\{2\}} \\ b^{\{2\}} \end{array} \right| = \begin{array}{ccc|ccc} 1/6 & -1/6 & 0 & 1/6 & 1/3 & 0 \\ 1/6 & 1/3 & 0 & 1/6 & 5/6 & 0 \\ 1/6 & 5/6 & 0 & 1/6 & 2/3 & 1/6 \end{array}.$$

As is well known, this method has additive order 4 ([9, Sect. II.2.2, Thm. 2.2]). We use this additive method to construct two NPRK methods as follows: We define Method 1 with NPRK tableau a_{ijk} given by (4.1a) and b_{ij} given by (4.2). We define Method 2 with NPRK tableau a_{ijk} again given by (4.1a) and \tilde{b}_{ij} now given by (4.1b). Method 1 and Method 2 together can be interpreted as an *embedded* NPRK method since they share the same a_{ijk} but have differing b_{ij} . As discussed in the previous section, both methods reduce to the same fourth-order additive method when the partition is additive. For a nonlinear partition, Method 1 will generally be a third-order NPRK method, and Method 2 will generally be a second-order NPRK method. First, we verify the order of both methods with the above Lotka–Volterra system (5.1) with the nonlinear partition

$$(5.2) \quad F \left(\begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \right) = \begin{bmatrix} u_2 - \alpha u_1 v_2 \\ v_1 + \alpha u_2 v_1 \end{bmatrix}.$$

See Figure 5.1. As expected, when $\alpha = 0$, the partition (5.2) becomes additive, and thus both methods are equivalent and exhibit fourth-order convergence. When $\alpha \neq 0$, Method 1 exhibits third-order convergence, whereas Method 2 exhibits second-order convergence.

Now, since both methods reduce to the same additive method, this means that the leading order difference in their accuracy is due to the nonlinear coupling. Particularly, denoting y_{n+1} as the numerical solution for Method 1 and \tilde{y}_{n+1} as the numerical solution for Method 2, both updated from the same y_n , we have an embedded estimate of the nonlinear coupling of the form

$$(5.3) \quad \|y_{n+1} - \tilde{y}_{n+1}\| \leq Ch^3 \left\| \mathcal{F} \left(\begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \right) \right\| + \mathcal{O}(h^4).$$

To demonstrate this embedded estimate, we compute one step of the update by Method 1 and Method 2 for the Lotka–Volterra system with $u(0) = 1 = v(0)$ as a function of α for several choices of the step size h . The difference in the l^1 -norm of the solutions $\|y_1 - \tilde{y}_1\|_{l^1}$ as a function of α is presented in Figure 5.2.

As expected, for each choice of h , as $\alpha \rightarrow 0$, we have $\|y_1 - \tilde{y}_1\|_{l^1} \rightarrow 0$ since in this limit x and y decouple and the partition becomes additive. From here, one might expect that the estimate $\|y_1 - \tilde{y}_1\|_{l^1}$ increases monotonically as α increases. However, this is not the case as seen in Figure 5.2. Particularly, the nonlinear coupling strength also approaches zero as $\alpha \rightarrow 1$. To see why this is the case, note that at the initial state $u(0) = 1 = v(0)$, for $\alpha = 1$, $du/dt = 0$, and thus, $dv/dt = v + \alpha uv$ is no longer nonlinearly coupled to the dynamics of u since u is constant. In other words, the nonlinear coupling strength in a system is state-dependent, which can be measured using the embedded estimate (5.3).

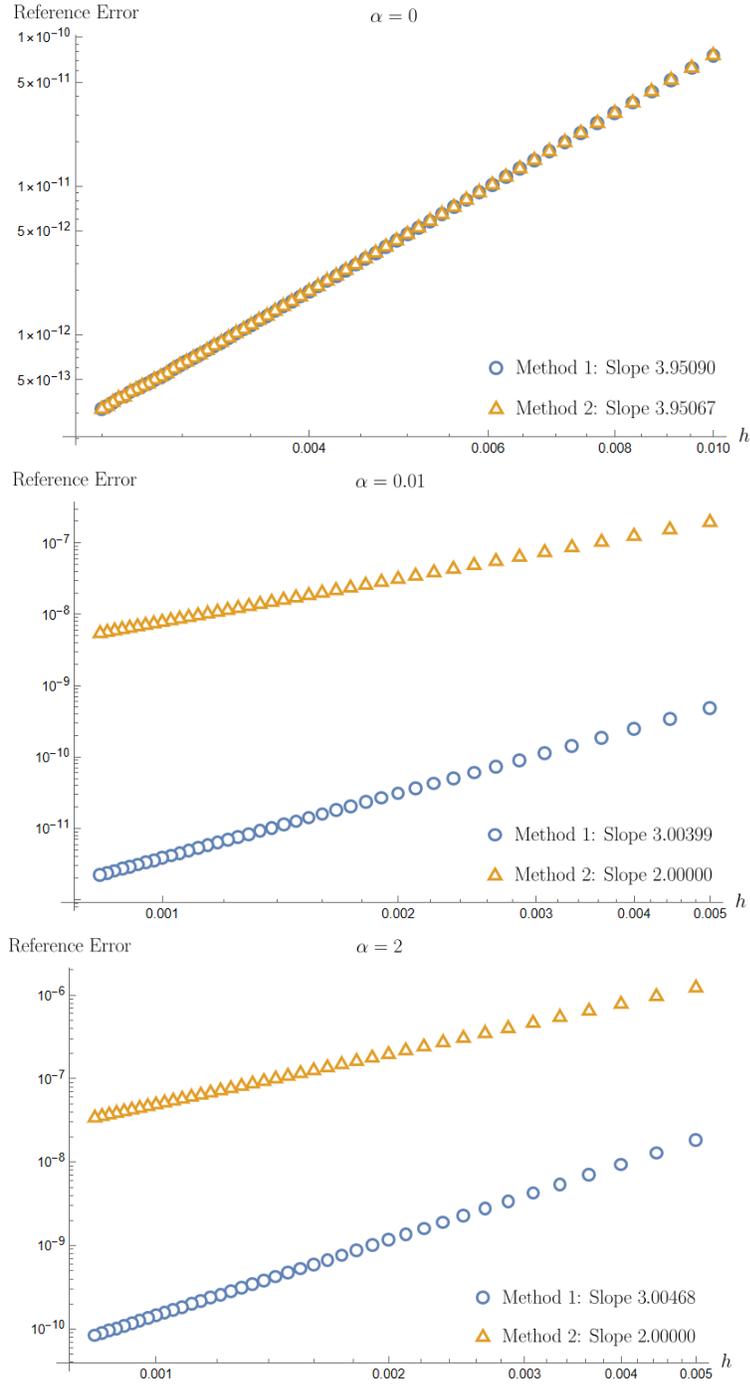


FIG. 5.1. Convergence test for Method 1 and Method 2 for $\alpha = 0, 0.01, 2$ with $u(0) = 1 = v(0)$ and final time $T = 1$. The reference solution was computed using Method 1 with $h = 10^{-4}$. The slopes were determined using linear regression.

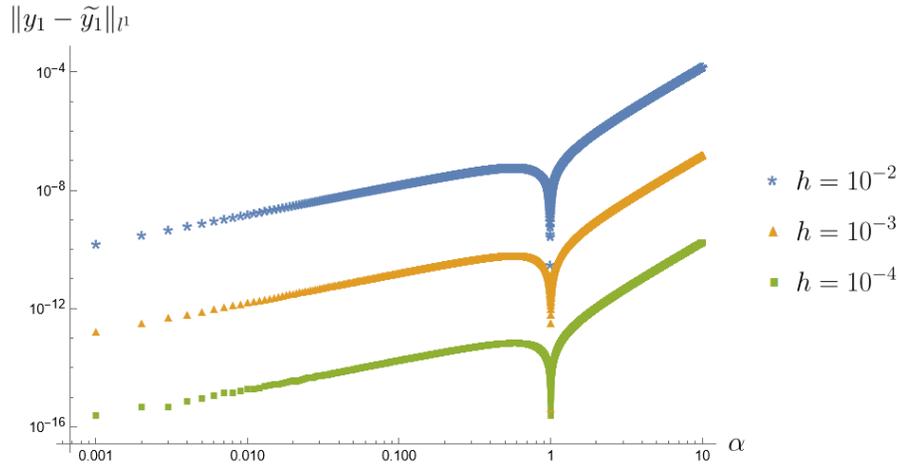


FIG. 5.2. Embedded estimate (5.3) of the nonlinear coupling after one step as a function of α for several choices of the step size $h = 10^{-2}, 10^{-3}, 10^{-4}$.

Stated another way, the leading term in (5.3) vanishes at the state $u(0) = 1 = v(0)$ as $\alpha \rightarrow 1$. To confirm this, we numerically verify the asymptotic scaling of the embedded estimate (5.3) as a function of h for various choices of α . We compute $\|y_1 - \tilde{y}_1\|_{l^1}$ for $u(0) = 1 = v(0)$ as a function of h , where y_1 and \tilde{y}_1 are again the solution after one step of size h from Method 1 and Method 2, respectively. This is done for the choices $\alpha = 0.1, 0.5, 1.0, 1.5, 2.0, 3.0$, displayed in Figure 5.3.

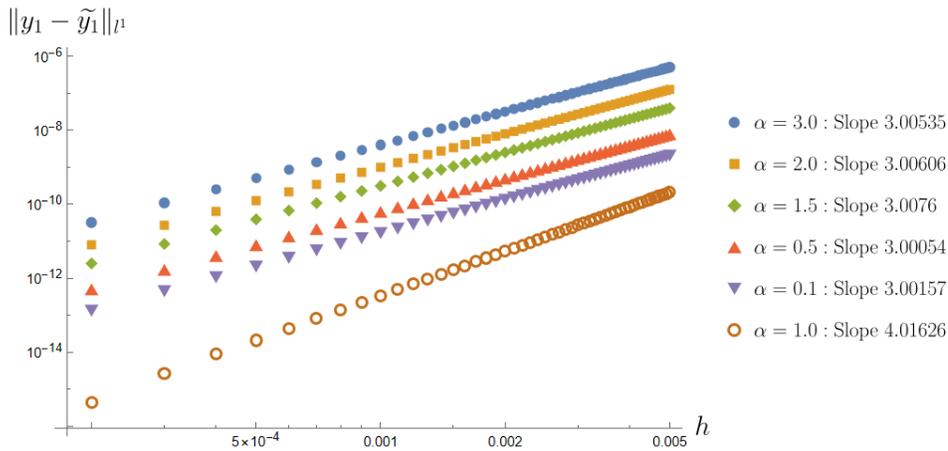


FIG. 5.3. Asymptotic scaling of the embedded estimate (5.3) with respect to h for various choices of α . The slopes were determined using linear regression.

As expected from (5.3), for $\alpha = 0.1, 0.5, 1.5, 2.0, 3.0$, the embedded estimate $\|y_1 - \tilde{y}_1\|_{l^1}$ exhibits third-order convergence in h . Interestingly, for $\alpha = 1.0$, the embedded estimate

exhibits fourth-order convergence. This is again explained by the fact that the leading order term of the embedded estimate (5.3) measures the nonlinear coupling, which vanishes at the current state $u(0) = 1 = v(0)$ for $\alpha = 1$. Thus, in the case $\alpha = 1$ with $u(0) = 1 = v(0)$, (5.3) reduces to $\|y_{n+1} - \tilde{y}_{n+1}\| \leq \mathcal{O}(h^4)$.

Conclusion. In many realistic multiphysics applications, there is a nonlinear coupling of scales and physics (e.g., see [15, 16]), making the resulting equations not directly amenable to classical additive partitions. In this paper, we provide a complete analysis of order conditions for the newly proposed class of NPRK methods via edge-colored rooted trees, which facilitates high-order partitioned integration for nonlinearly partitioned equations. General order conditions are provided for arbitrary order and number of partitions, and explicit conditions for the NPRK tableaux are provided for up to fourth order for two partitions and third order for three partitions. NPRK order conditions are also related to ARK order conditions, and it is shown that for order $p \geq 3$, additional conditions related to a nonlinear coupling cannot be represented in trees used to derive ARK order conditions (although it is possible for an ARK method of order 3 to satisfy the nonlinear coupling conditions when posed as an NPRK method).

Here we provide examples of a second-order and third-order NPRK method constructed from the additive Lobatto IIIA-IIIIB pair, differing only in the nonlinear order conditions, and we show how they can be used to obtain an estimate of the state-dependent nonlinear coupling strength. Our companion paper provides a number of other example methods of second and third order [5]. In future work, we will derive practical NPRK methods of various orders, structures, and number of partitions, focusing on optimizing coefficients for properties related to accuracy and stability. We will also explore the NPRK tensor structure as a simplifying assumption for deriving ARK methods of higher order and number of partitions, due to the reduced number of order conditions compared with classical ARK methods. We have not addressed structure preservation here; future work will also study method properties such as conservation, symplecticity, and adaptivity.

Acknowledgements. The authors would like to thank the reviewers for their helpful comments and suggestions. BKT was supported by the Marc Kac Postdoctoral Fellowship at the Center for Nonlinear Studies at Los Alamos National Laboratory. BSS was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20220174ER. Los Alamos National Laboratory report LA-UR-24-20566. TB was funded by the National Science Foundation under grant NSF-OIA-2327484.

Appendix A. List of order conditions. Below are lists of NPRK_M order conditions for $M = 2$ of third-order (A.1) and of fourth-order (A.2), and for $M = 3$ of third-order (A.3), generated by a symbolic version of Algorithm 1. The summations are understood to run over all present indices from 1 to s . Equations annotated with * denote linearly separable coupling conditions and equations annotated with † denote nonlinear coupling conditions. The code for all results generated in this paper is available at [17].

(A.1) Third-order conditions for $M = 2$.

$$\begin{aligned}
 \sum b_{ij} a_{ikl} a_{kuv} &= \frac{1}{6}, & \sum b_{ij} a_{ikl} a_{iuv} &= \frac{1}{3}, \\
 * \sum b_{ij} a_{kuv} a_{jkl} &= \frac{1}{6}, & \dagger \sum b_{ij} a_{iuv} a_{jkl} &= \frac{1}{3}, \\
 * \sum b_{ij} a_{ikl} a_{luv} &= \frac{1}{6}, & \sum b_{ij} a_{jkl} a_{juv} &= \frac{1}{3}, \\
 \sum b_{ij} a_{jkl} a_{luv} &= \frac{1}{6}.
 \end{aligned}$$

(A.2) Fourth-order conditions for $M = 2$.

$$\begin{aligned}
 \sum b_{ij} a_{ikl} a_{kuv} a_{uab} &= \frac{1}{24}, & \sum b_{ij} a_{ikl} a_{iab} a_{kuv} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{kuv} a_{uab} a_{jkl} &= \frac{1}{24}, & \dagger \sum b_{ij} a_{iab} a_{kuv} a_{jkl} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{ikl} a_{uab} a_{luv} &= \frac{1}{24}, & * \sum b_{ij} a_{ikl} a_{iab} a_{luv} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{uab} a_{jkl} a_{luv} &= \frac{1}{24}, & \dagger \sum b_{ij} a_{iab} a_{jkl} a_{luv} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{ikl} a_{kuv} a_{vab} &= \frac{1}{24}, & \dagger \sum b_{ij} a_{ikl} a_{kuv} a_{jab} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{kuv} a_{jkl} a_{vab} &= \frac{1}{24}, & * \sum b_{ij} a_{kuv} a_{jkl} a_{jab} &= \frac{1}{8}, \\
 * \sum b_{ij} a_{ikl} a_{luv} a_{vab} &= \frac{1}{24}, & \dagger \sum b_{ij} a_{ikl} a_{jab} a_{luv} &= \frac{1}{8}, \\
 \sum b_{ij} a_{jkl} a_{luv} a_{vab} &= \frac{1}{24}, & \sum b_{ij} a_{jkl} a_{jab} a_{luv} &= \frac{1}{8}, \\
 \sum b_{ij} a_{ikl} a_{kuv} a_{kab} &= \frac{1}{12}, & \sum b_{ij} a_{ikl} a_{iuv} a_{iab} &= \frac{1}{4}, \\
 * \sum b_{ij} a_{kuv} a_{kab} a_{jkl} &= \frac{1}{12}, & \dagger \sum b_{ij} a_{iuv} a_{iab} a_{jkl} &= \frac{1}{4}, \\
 \dagger \sum b_{ij} a_{ikl} a_{kab} a_{luv} &= \frac{1}{12}, & \dagger \sum b_{ij} a_{iab} a_{jkl} a_{juv} &= \frac{1}{4}, \\
 \dagger \sum b_{ij} a_{kab} a_{jkl} a_{luv} &= \frac{1}{12}, & \sum b_{ij} a_{jkl} a_{juv} a_{jab} &= \frac{1}{4}, \\
 * \sum b_{ij} a_{ikl} a_{luv} a_{lab} &= \frac{1}{12}, \\
 \sum b_{ij} a_{jkl} a_{luv} a_{lab} &= \frac{1}{12}.
 \end{aligned}$$

(A.3) Third-order conditions for $M = 3$.

$$\begin{array}{ll}
 \sum b_{ijk} a_{iuvw} a_{uabc} = \frac{1}{6}, & \sum b_{ijk} a_{iuvw} a_{iabc} = \frac{1}{3}, \\
 * \sum b_{ijk} a_{iuvw} a_{vabc} = \frac{1}{6}, & \dagger \sum b_{ijk} a_{iabc} a_{juvw} = \frac{1}{3}, \\
 * \sum b_{ijk} a_{iuvw} a_{wabc} = \frac{1}{6}, & \dagger \sum b_{ijk} a_{iabc} a_{kuvw} = \frac{1}{3}, \\
 * \sum b_{ijk} a_{uabc} a_{juvw} = \frac{1}{6}, & \sum b_{ijk} a_{juvw} a_{jabc} = \frac{1}{3}, \\
 \sum b_{ijk} a_{juvw} a_{vabc} = \frac{1}{6}, & \dagger \sum b_{ijk} a_{jabc} a_{kuvw} = \frac{1}{3}, \\
 * \sum b_{ijk} a_{juvw} a_{wabc} = \frac{1}{6}, & \sum b_{ijk} a_{kuvw} a_{kabc} = \frac{1}{3}, \\
 * \sum b_{ijk} a_{uabc} a_{kuvw} = \frac{1}{6}, & \\
 * \sum b_{ijk} a_{vabc} a_{kuvw} = \frac{1}{6}, & \\
 \sum b_{ijk} a_{kuvw} a_{wabc} = \frac{1}{6}. &
 \end{array}$$

REFERENCES

- [1] A. L. ARAUJO, A. MURUA, AND J. M. SANZ-SERNA, *Symplectic methods based on decompositions*, SIAM J. Numer. Anal., 34 (1997), pp. 1926–1947.
- [2] U. M. ASCHER, S. J. RUUTH, AND R. J. SPITERI, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math., 25 (1997), pp. 151–167.
- [3] S. BOSCARINO, J. QIU, G. RUSSO, AND T. XIONG, *High order semi-implicit WENO schemes for all-Mach full Euler system of gas dynamics*, SIAM J. Sci. Comput., 44 (2022), pp. B368–B394.
- [4] W. BOSCHERI AND M. TAVELLI, *High order semi-implicit schemes for viscous compressible flows in 3D*, Appl. Math. Comput., 434 (2022), Paper No. 127457, 19 pages.
- [5] T. BUVOLI AND B. S. SOUTHWORTH, *A new class of Runge-Kutta methods for nonlinearly partitioned systems*, Preprint on arXiv, 2024. <https://arxiv.org/abs/2401.04859>
- [6] H. CARTAN, *Differential Calculus*, Kershaw Publishing Company, London, 1971.
- [7] G. J. COOPER AND A. SAYFY, *Additive methods for the numerical solution of ordinary differential equations*, Math. Comp., 35 (1980), pp. 1159–1172.
- [8] L. FOISSY, *Algebraic structures on typed decorated rooted trees*, SIGMA Symmetry Integrability Geom. Methods Appl., 17 (2021), Paper No. 086, 28 pages.
- [9] E. HAIRER, G. WANNER, AND C. LUBICH, *Geometric Numerical Integration*, 2nd ed., Springer, Berlin, 2006.
- [10] E. HAIRER, G. WANNER, AND S. P. NØRSETT, *Solving Ordinary Differential Equations. I.*, Springer, Berlin, 1993.
- [11] C. A. KENNEDY AND M. H. CARPENTER, *Additive Runge–Kutta schemes for convection–diffusion–reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
- [12] ———, *Higher-order additive Runge–Kutta schemes for ordinary differential equations*, Appl. Numer. Math., 136 (2019), pp. 183–205.
- [13] D. I. KETCHESON AND H. RANOCHA, *Computing with B-series*, ACM Trans. Math. Software, 49 (2023), Paper No. 13, 23 pages.
- [14] H. RANOCHA ET AL., *RootedTrees.jl: A collection of functionality around rooted trees to generate order conditions for Runge-Kutta methods in Julia for differential equations and scientific machine learning (SciML) Software*, May, 2019. <https://github.com/SciML/RootedTrees.jl>
- [15] B. S. SOUTHWORTH, S. OLIVIER, H. PARK, AND T. BUVOLI, *One-sweep moment-based semi-implicit-explicit integration for gray thermal radiation transport*, J. Comput. Phys., 517 (2024), Paper No. 113349, 19 pages.

- [16] B. S. SOUTHWORTH, H. PARK, S. TOKAREVA, AND M. CHAREST, *Implicit-explicit Runge-Kutta for radiation hydrodynamics I: Gray diffusion*, J. Comput. Phys., 518 (2024), Paper No. 113339, 17 pages.
- [17] B. K. TRAN, B. S. SOUTHWORTH, AND T. BUVOLI, *NPRK-order-conditions*, Software, 2025.
<https://github.com/buvoli/NPRK-order-conditions.git>