

Computational Science Laboratory
Technical Report CSL-TR-16 -7
November 22, 2016

Paul Tranquilli, Adrian Sandu and Hong
Zhang

“LIRK-W: Linearly-implicit
Runge-Kutta methods with
approximate matrix factorization”

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: sandu@cs.vt.edu
Web: <http://cs1.cs.vt.edu>



Compute the Future



LIRK-W: LINEARLY-IMPLICIT RUNGE-KUTTA METHODS WITH APPROXIMATE MATRIX FACTORIZATION

PAUL TRANQUILLI ^{†¶}, ADRIAN SANDU ^{‡¶}, AND HONG ZHANG ^{§ ¶}

Abstract. This paper develops a new class of linearly implicit time integration schemes called Linearly-Implicit Runge-Kutta-W (LIRK-W) methods. These schemes are based on an implicit-explicit approach which does not require a splitting of the right hand side and allow for arbitrary, time dependent, and stage varying approximations of the linear systems appearing in the method. Several formulations of LIRK-W schemes, each designed for specific approximation types, and their associated order condition theories are presented.

1. Introduction. A standard approach to the solution of systems of partial differential equations is the method of lines technique, in which a discretization method such as finite differences, finite volumes, or finite elements is used to approximate derivatives in space to arrive at the semi-discrete initial value problem (1.1)

$$(1.1) \quad \frac{dy}{dt} = F(y), \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_0; \quad y(t), \quad F(y) \in \mathbb{R}^N.$$

The system (1.1) can be evolved through time using a time integration scheme to approximate solutions at discrete times $t_n < t_i < t_F$. For problems with stiff dynamics, or where mesh refinement leads to unfortunate Courant-Friedrichs-Lewy (CFL) numbers, explicit methods may not be suitable.

Implicit methods, such as Backwards Differentiation or Runge-Kutta methods improve upon the stability of explicit methods, at the cost of requiring one or more non-linear solves at each timestep. The approximation of solutions of these nonlinear systems represents the bulk of the computational cost of the time integration process.

Linearly implicit methods, such as Rosenbrock [8, section IV.7] or linearly-implicit Runge-Kutta [6, 4, 2, 1], schemes replace the need to solve a nonlinear system with the solution of more computationally efficient linear systems. Once again, however, the cost of approximating the solution of these linear systems represents a disproportionately large percentage of the overall method cost. Rosenbrock-W [8, section IV.7] [12] methods attempt to alleviate this burden by allowing for the use of arbitrary approximations, and so permit relatively cheap, and inaccurate, solutions of the linear system while maintaining full order of convergence. Unfortunately, while these methods permit arbitrary approximations, the order condition theory on which they are built does not account for variations in the method of approximation between stages of the method.

For this reason, many approximation techniques may not be available in the context of Rosenbrock-W methods. Primary among them is the use of Krylov based iterative solvers, such as GMRES [14], which computes the solution of several different, nearby linear systems when the iteration procedure is truncated early [16]. Similarly, making use of an approximate matrix factorization (AMF) to accelerate the solution of the linear systems also leads to a reduction in the order of the Rosenbrock-W scheme. Several families of time integration schemes (ROWMAP [13] and Rosenbrock-Krylov

[†]ptranq@vt.edu

[‡]sandu@cs.vt.edu

[§]zhang@vt.edu

[¶]Computational Science Laboratory, Department of Computer Science, Virginia Tech. 2202 Kraft Drive, Blacksburg, Virginia 24060.

[17]) have been constructed to couple Krylov solvers and Rosenbrock-W methods to alleviate order reduction in the case of GMRES like approximations.

Here we present a new family of time integration schemes, called linearly-implicit Runge-Kutta-W (LIRK-W) methods, based on an implicit-explicit (IMEX) [3] approach, which allow for arbitrary, time dependent, and stage varying approximations of the linear systems appearing in the method. The rest of the paper is laid out as follows: In section 2 we motivate, and present, the general form of a LIRK-W method; in section 2.3 we review the use of AMF to accelerate the solution of linear systems in the context of PDEs; in section 3 we present the order condition theory for a LIRK-W method; in section 4 we give linear stability results for a LIRK-W method; in section 5 we derive a specific LIRK-W method; and finally in section 6 we present numerical results.

2. Linearly-Implicit Runge-Kutta-W (LIRK-W) Methods.

2.1. Implicit-explicit Runge Kutta methods. Consider a splitting of the right hand side $F(y)$ of the initial value problem (1.1)

$$(2.1) \quad \frac{dy}{dt} = F(y) = f(y) + g(y), \quad t_0 \leq t \leq t_F,$$

such that $f(y)$ represents slow dynamics, unlikely to impact the stability of the numerical integration, and $g(y)$ contains the fast dynamics. An implicit-explicit Runge-Kutta (IMEX-RK) method applies different discretizations to the the two terms, and integrates the non-stiff component $f(y)$ explicitly and stiff component $g(y)$ implicitly [15]:

$$(2.2a) \quad Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} f(Y_j) + h \sum_{j=1}^i \hat{a}_{i,j} g(Y_j), \quad i = 1, \dots, s,$$

$$(2.2b) \quad y_{n+1} = y_n + h \sum_{j=1}^s b_j f(Y_j) + h \sum_{j=1}^s \hat{b}_j g(Y_j).$$

The fact that only $g(y)$ is treated implicitly has the benefit of reducing the per-timestep cost as compared to implicit Runge-Kutta methods, since one only solves non-linear systems containing $g(y)$ instead of the entire of $F(y)$. In the same time (2.2) has considerably better stability properties than explicit Runge-Kutta schemes since the stiff dynamics is integrated implicitly.

Unfortunately, the application of (2.2) requires to first partition the system $F(t, y)$ into non-stiff and stiff parts (2.1), and to provide the Jacobian operator corresponding to the stiff term. These steps can be difficult to achieve for systems implemented in large legacy codes.

2.2. Arbitrary linear approximations of the stiff term. Here we propose the new class of Linearly-Implicit Runge-Kutta-W (LIRK-W) time integrators that make use of an alternative partitioning, based on a linear/non-linear splitting of the right hand side operator:

$$(2.3) \quad \frac{dy}{dt} = \mathbf{L}y + (F(y) - \mathbf{L}y), \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_n; \quad y(t), F(y) \in \mathbb{R}^N, \quad \mathbf{L} \in \mathbb{R}^{N \times N}.$$

where $\mathbf{L}y$ ideally captures the stiff dynamics of $F(y)$. In this way we will seek to treat implicitly the linear terms \mathbf{L} that capture the stiffness of the system, and to

treat explicitly the remaining nonlinear part. The IMEX-RK method (2.2) applied to (2.3), after some rearranging of terms, reads:

$$(2.4a) \quad (\mathbf{I} - h \widehat{a}_{i,i} \mathbf{L}) Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \mathbf{L} \sum_{j=1}^{i-1} (\widehat{a}_{i,j} - a_{i,j}) Y_j,$$

$$(2.4b) \quad y_{n+1} = y_n + h \sum_{j=1}^s b_j F(Y_j) + h \mathbf{L} \sum_{j=1}^s (\widehat{b}_j - b_j) Y_j.$$

Using the notation

$$\gamma_{i,j} = (\widehat{a}_{i,j} - a_{i,j}), \quad \text{and} \quad g_i = (\widehat{b}_i - b_i)$$

in equation (2.4) leads to the standard form of a Linearly-Implicit Runge-Kutta-W (LIRK-W) method:

$$(2.5a) \quad (\mathbf{I} - h \gamma_{i,i} \mathbf{L}) Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \mathbf{L} \sum_{j=1}^{i-1} \gamma_{i,j} Y_j,$$

$$(2.5b) \quad y_{n+1} = y_n + h \sum_{j=1}^s b_j F(Y_j) + h \mathbf{L} \sum_{j=1}^s g_j Y_j.$$

It is clear from equation (2.3) that the linear operator \mathbf{L} can be arbitrary, since collecting terms leads back to the original form of the initial value problem (1.1). Additionally, (2.5) makes exclusive use of the entire right-hand-side vector $F(y)$, there is no splitting necessary. There are several benefits of this framework. First, the stiff terms that are integrated implicitly are linear, so no solutions of nonlinear systems are required. Next, LIRK-W methods will be designed to preserve accuracy for any matrix \mathbf{L} . The selected \mathbf{L} is only used to ensure numerical stability; its structure is arbitrary and can be chosen to ensure computational efficiency on the hardware at hand. With this in mind, an ideal choice of the linear operator is one which approximates the stiff dynamics of the system, $\mathbf{L} \approx \partial g(t, y) / \partial y$, in order to improve stability of the numerical integration, and where solutions of linear systems containing \mathbf{L} can be computed efficiently.

LIRK-W methods are similar to Rosenbrock-W schemes, in that they depend only on linear solves and permit arbitrary matrices. However, as will be discussed more thoroughly in section 2.5, there are several advantages to the LIRK-W framework.

2.3. Approximate Matrix Factorization. Approximate Matrix Factorization (AMF) is often employed to speed up the solution of linear systems required when computing stage vectors (2.2a) or (2.5a) of implicit methods:

$$(2.6) \quad (\mathbf{I} - h \gamma_{i,i} \mathbf{L}) k_i = d_i.$$

AMF splits the matrix \mathbf{L} , usually the Jacobian of the right hand side vector in (1.1), into a sum of parts

$$(2.7) \quad \mathbf{L} = \sum_{r=1}^R \mathbf{L}^{\{r\}},$$

and approximates the solution to (2.6) by replacing the matrix as follows:

$$(2.8) \quad \begin{aligned} (\mathbf{I} - h \gamma_{i,i} \mathbf{L}) &\approx (\mathbf{I} - h \gamma_{i,i} \tilde{\mathbf{L}}) = \prod_{r=1}^R (\mathbf{I} - h \gamma_{i,i} \mathbf{L}^{\{r\}}), \\ k_i &= (\mathbf{I} - h \gamma_{i,i} \mathbf{L})^{-1} d_i \approx \prod_{r=1}^R (\mathbf{I} - h \gamma_{i,i} \mathbf{L}^{\{r\}})^{-1} d_i. \end{aligned}$$

The solution of one large linear system with matrix \mathbf{L} is replaced by solving in succession R small linear systems with matrices $\mathbf{L}^{\{r\}}$, potentially leading to considerable computational savings.

The approximation (2.8) implicitly defines the matrix $\tilde{\mathbf{L}}$ as

$$(2.9) \quad \tilde{\mathbf{L}} = \mathbf{L} + \sum_{k=2}^R (-h \gamma_{i,i})^{k-1} \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq R} \mathbf{L}_{i_1} \mathbf{L}_{i_2} \dots \mathbf{L}_{i_k}.$$

For example, if \mathbf{L} is the discrete two-dimensional Laplacian, then in (2.7) the parts $\mathbf{L}^{\{1\}}$ and $\mathbf{L}^{\{2\}}$ can correspond to derivatives along the x_1 and x_2 directions respectively. In this case the AMF approximation corresponds to the alternating directions factorization [5, 11]

$$\mathbf{I} - h \gamma \tilde{\mathbf{L}} := (\mathbf{I} - h \gamma \mathbf{L}^{\{1\}}) (\mathbf{I} - h \gamma \mathbf{L}^{\{2\}}), \quad \tilde{\mathbf{L}} = \mathbf{L} - h \gamma \mathbf{L}^{\{1\}} \mathbf{L}^{\{2\}}.$$

In this way it is possible to solve individually much simpler one-dimensional systems corresponding to each individual \mathbf{L}_i . An alternating directions factorization is not the only choice possible, but it does motivate many of our design decisions for LIRK-W methods.

2.4. Alternative formulations of LIRK-W methods. We note from (2.9) that the approximate matrix $\tilde{\mathbf{L}}(h)$ depends on the step size, i.e., depends on time. The general form (2.5) making use of a time dependent, stage varying matrix \mathbf{L} can be interpreted in several ways. In the most straightforward variation, we can let there be one \mathbf{L}_i per stage, and make use of these matrices everywhere that \mathbf{L} appears in (2.5), we will call this a type 1 LIRK-W method

$$(2.10a) \quad Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \sum_{j=1}^i \gamma_{i,j} \mathbf{L}_j Y_j,$$

$$(2.10b) \quad y_{n+1} = y_n + h \sum_{i=1}^s b_i F(Y_i) + h \sum_{i=1}^s g_i \mathbf{L}_i Y_i,$$

with $\mathbf{L}_i = \tilde{\mathbf{L}}(t_n + h \gamma_{i,i})$.

Alternatively, we can attempt to interpret (2.5) to account for an AMF form of \mathbf{L} , where there exists two forms of \mathbf{L} : equation (2.7), which has clear advantages for a multiplication by \mathbf{L} ; and equation (2.8), which has clear advantages when used in the inversion of the left hand side of equation (2.5a). A LIRK-W method of type 2

exploits the natural features of an AMF form of \mathbf{L} and has the general form

$$(2.11a) \quad Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \mathbf{L} \sum_{j=1}^{i-1} \gamma_{i,j} Y_j + h \gamma_{i,i} \mathbf{L}_i Y_i,$$

$$(2.11b) \quad y_{n+1} = y_n + h \sum_{j=1}^s b_j F(Y_j) + h \mathbf{L} \sum_{j=1}^s g_j Y_j,$$

where $\mathbf{L}_i = \mathbf{L}(t_n + h\gamma_{i,i})$.

More variations of (2.5) are possible, and different choices for constructing \mathbf{L}_i may motivate alternatives to type 1 or type 2 LIRK-W schemes.

2.5. LIRK-W methods are not Rosenbrock-W methods. As was discussed briefly above, LIRK-W methods share many common traits with a Rosenbrock, or Rosenbrock-W, approach. Primary among them is that LIRK-W methods require only the solution of linear systems, and permit arbitrary matrices \mathbf{L} . The defining characteristic of LIRK-W schemes, in contrast to Rosenbrock-W methods, is the allowance for the use of several different approximations in a single timestep. Additionally, the arbitrary linear term, \mathbf{L}_i , may depend on time so that it is possible to directly approximate $(\mathbf{I} - h\gamma_{i,i}\mathbf{L}_i)^{-1}$ without violating assumptions inherent in the order condition framework.

One example of this, is the use of a GMRES like approximation to the linear systems $(\mathbf{I} - h\gamma_{i,i}\mathbf{J})x = b_i$. Early truncation of the GMRES procedure may lead to loss of order for Rosenbrock-W schemes since each linear system is solved using a different subspace, and is essentially the same as solving s systems with s different left-hand-sides. It is possible to formulate a LIRK-W method for exactly this scenario, one possible approach is:

$$(2.12a) \quad Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \mathbf{L}_i \sum_{j=1}^i \gamma_{i,j} Y_j$$

$$(2.12b) \quad y_{n+1} = y_n + h \sum_{j=1}^s b_j F(Y_j) + h \mathbf{L} \sum_{j=1}^s g_j Y_j,$$

which we will refer to as type 3 LIRK-W method.

3. Order Conditions for Linearly-Implicit Runge-Kutta-W Methods.

We construct classical order conditions for the various types of LIRK-W methods by matching the Taylor series expansion of the exact solution of equation (1.1) with the Taylor series expansion of the numerical solutions. To do so we make use of Butcher trees [7, section II.2], with suitable modifications to handle the particular aspects of LIRK-W methods. These modifications lead to the new family of LW -trees, which we discuss next.

3.1. LW-trees. These trees contain three different colored vertices: meagre vertices are filled and represent an appearance of $F(y)$ and its derivatives, fat vertices are empty and represent an appearance of the linear term \mathbf{L} , and finally square vertices represent a differentiation in time of the fat node they terminate in. We refer to the set of these trees as LW_i -trees, where i denotes the type of LIRK-W method being discussed.

For LIRK-W methods of all types, we make use of the standard notation that the tree $[\tau_1, \dots, \tau_m]_{\bullet}$ is constructed by attaching the subtrees τ_1, \dots, τ_m to a meagre node

as its root. Similarly, the $[\tau]_{\circ}$ is constructed by attaching the subtree τ to a fat node, and new to this manuscript is the notation that $[\tau]_{\theta_p}$ is constructed by attaching the subtree τ to a fat node prepended by p square nodes. Figure 3.1 shows how trees can be constructed from sub-trees.

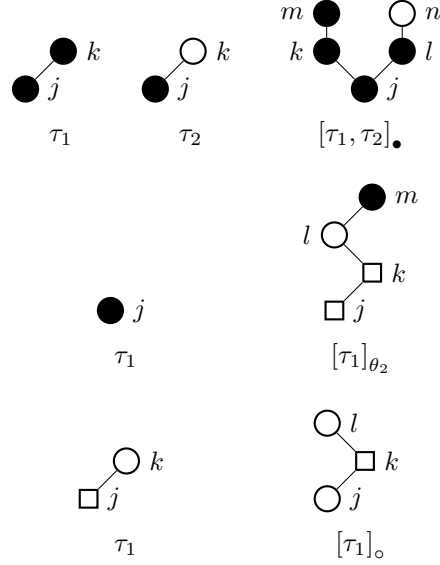


FIG. 3.1. Illustration of the recursive definition of LW-trees.

To derive the expansion of the numerical solution for type 1 and 2 methods we make use of Faà di Bruno's formula [7, section II.2] to compute general high-order derivatives of the initial value problem right-hand-side vector $F(y)$

$$(3.1) \quad (F(Y_j))^{(q-1)} \Big|_{h=0} = \sum_{m \geq 1} \frac{\partial^m f}{\partial y^m} \left(Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)} \right), \quad \mu_1 + \dots + \mu_m = q - 1.$$

3.2. Order conditions for LIRK-W methods of type 1. Here we derive order conditions for LIRK-W methods of type 1 (2.10).

We can make use of the expansion (3.1) to construct derivatives of the type 1 LIRK-W scheme to arrive at the result that

(3.2a)

$$(Y_i)^{(q)} \Big|_{h=0} = q \sum_{j=1}^{i-1} a_{i,j} \sum_{m \geq 1} \frac{\partial^m F}{\partial y^m} \left(Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)} \right) + q \sum_{j=1}^i \gamma_{i,j} \sum_{k=0}^{q-1} \binom{q-1}{k} \gamma_{j,j}^k \left(\frac{d^k}{dt^k} \mathbf{L} \right) Y_j^{(q-1-k)}$$

(3.2b)

$$(y_{n+1})^{(q)} \Big|_{h=0} = q \sum_{i=1}^s b_i \sum_{m \geq 1} \frac{\partial^m F}{\partial y^m} \left(Y_i^{(\mu_1)}, \dots, Y_i^{(\mu_m)} \right) + q \sum_{i=1}^s g_i \sum_{k=0}^{q-1} \binom{q-1}{k} \gamma_{i,i}^k \left(\frac{d^k}{dt^k} \mathbf{L} \right) Y_i^{(q-1-k)}$$

Leading to the definition for the LW_1 trees

$$LW_1 = \left\{ \begin{array}{l} N_3\text{-trees: fat vertices are singly branched, and} \\ \quad \quad \quad \text{square vertices are singly branched} \\ \quad \quad \quad \text{with square or fat children} \end{array} \right\}$$

Figures 3.2 and 3.3, give the LW_1 -trees, τ_i , representing the elementary differentials, $f(\tau_i)$, present in the Taylor expansion of the exact and numerical solutions, as well as the coefficients of these terms, $\Phi^1(\tau_i)$ and $P(\tau_i)$ respectively.

i	1	2	3	4	5
τ_i					
$f(\tau_i)$	f^J	$\mathbf{L}_{JK}y^K$	$f_K^J f^K$	$f_K^J \mathbf{L}_{KL}y^L$	$\mathbf{L}_{JK}f^K$
$\Phi^1(\tau_i)$	b_j	g_j	$b_j a_{j,k}$	$b_j \gamma_{j,k}$	$g_j a_{j,k}$
$\Phi^2(\tau_i)$	b_j	g_j	$b_j a_{j,k}$	$b_j \gamma_{j,k}$	$g_j a_{j,k}$
$\Phi^3(\tau_i)$	b_j	g_j	$b_j a_{j,k}$	$b_j \gamma_{j,k}$	$g_j a_{j,k}$
$P(\tau_i)$	1	0	1/2	0	0
i	6	7	8	9	10
τ_i					
$f(\tau_i)$	$\mathbf{L}_{JK}\mathbf{L}_{KLY}^L$	$\mathbf{L}'_{JK}y^K$	$f_{KL}^J f^K f^L$	$f_{KL}^J \mathbf{L}_{KMY}^M f^L$	$f_{KL}^J \mathbf{L}_{KMY}^M \mathbf{L}_{LNY}^N$
$\Phi^1(\tau_i)$	$g_j \gamma_{j,k}$	$g_k \gamma_{k,k}$	$b_j a_{j,k} a_{j,l}$	$b_j \gamma_{j,k} a_{j,l}$	$b_j \gamma_{j,k} \gamma_{j,l}$
$\Phi^2(\tau_i)$	$g_j \gamma_{j,k}$	N/A	$b_j a_{j,k} a_{j,l}$	$b_j \gamma_{j,k} a_{j,l}$	$b_j \gamma_{j,k} \gamma_{j,l}$
$\Phi^3(\tau_i)$	$g_j \gamma_{j,k}$	N/A	$b_j a_{j,k} a_{j,l}$	$b_j \gamma_{j,k} a_{j,l}$	$b_j \gamma_{j,k} \gamma_{j,l}$
$P(\tau_i)$	0	0	1/3	0	0
i	11	12	13	14	15
τ_i					
$f(\tau_i)$	$f_K^J f_L^K f^L$	$f_K^J f_L^K \mathbf{L}_{LMY}^M$	$f_K^J \mathbf{L}_{KL} \mathbf{L}_{LMY}^M$	$f_K^J \mathbf{L}_{KL} f^L$	$\mathbf{L}_{JK} \mathbf{L}_{KL} f^L$
$\Phi^1(\tau_i)$	$b_j a_{j,k} a_{k,l}$	$b_j a_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} a_{k,l}$	$g_j \gamma_{j,k} a_{k,l}$
$\Phi^2(\tau_i)$	$b_j a_{j,k} a_{k,l}$	$b_j a_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} a_{k,l}$	$g_j \gamma_{j,k} a_{k,l}$
$\Phi^3(\tau_i)$	$b_j a_{j,k} a_{k,l}$	$b_j a_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} \gamma_{k,l}$	$b_j \gamma_{j,k} a_{k,l}$	$g_j \gamma_{j,k} a_{k,l}$
$P(\tau_i)$	1/6	0	0	0	0

FIG. 3.2. Trees and order conditions for LIRK-W methods up to order three.

The recursive formulas in equation (3.3) map the visual representation of a given LW_1 -tree to the left hand side of the order condition, $\Phi_j(\tau)$, corresponding to that

i	16	17	18	19
τ_i				
$f(\tau_i)$	$\mathbf{L}_{JK} f_L^K f^L$	$\mathbf{L}_{JK} f_L^K \mathbf{L}_{LM} y^M$	$\mathbf{L}_{JK} \mathbf{L}_{KL} \mathbf{L}_{LM} y^M$	$\mathbf{L}_{JM}'' y^M$
$\Phi^1(\tau_i)$	$g_j a_{j,k} a_{k,l}$	$g_j a_{j,k} \gamma_{k,l}$	$g_j \gamma_{j,k} \gamma_{k,l}$	$g_l \gamma_{l,l} \gamma_{l,l}$
$\Phi^2(\tau_i)$	$g_j a_{j,k} a_{k,l}$	$g_j a_{j,k} \gamma_{k,l}$	$g_j \gamma_{j,k} \gamma_{k,l}$	N/A
$\Phi^3(\tau_i)$	$g_j a_{j,k} a_{k,l}$	$g_j a_{j,k} \gamma_{k,l}$	$g_j \gamma_{j,k} \gamma_{k,l}$	N/A
$P(\tau_i)$	0	0	0	0
i	20	21	22	23
τ_i				
$f(\tau_i)$	$\mathbf{L}_{JK} \mathbf{L}'_{KM} y^M$	$\mathbf{L}'_{JL} \mathbf{L}_{LM} y^M$	$\mathbf{L}'_{JL} f^L$	$f_K^J \mathbf{L}'_{KM} y^M$
$\Phi^1(\tau_i)$	$g_j \gamma_{j,l} \gamma_{l,l}$	$g_k \gamma_{k,k} \gamma_{k,l}$	$g_k \gamma_{k,k} a_{k,l}$	$b_j \gamma_{j,l} \gamma_{l,l}$
$\Phi^2(\tau_i)$	$g_j \gamma_{j,j} \gamma_{j,j}$	N/A	N/A	$b_j \gamma_{j,j} \gamma_{j,j}$
$\Phi^3(\tau_i)$	$g_j \gamma_{j,j} \gamma_{j,l}$	N/A	N/A	$b_j \gamma_{j,j} \gamma_{j,l}$
$P(\tau_i)$	0	0	0	0

FIG. 3.3. Trees and order conditions for LIRK-W methods up to order three (Continued).

tree.

$$(3.3a) \quad \Phi_j^1(\tau) = \begin{cases} b_j \bar{\Phi}_j^1(\tau_1) \cdots \bar{\Phi}_j^1(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ g_j \bar{\Phi}_j^1(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ g_j \gamma_{j,j}^p \bar{\Phi}_j^1(\tau_1) & \text{if } \tau = [\tau_1]_{\theta_p}, \\ b_j & \text{if } \tau = [\]_{\bullet}, \\ g_j & \text{if } \tau = [\]_{\circ}, \end{cases}$$

$$(3.3b) \quad \bar{\Phi}_j^1(\tau) = \begin{cases} a_{j,k} \bar{\Phi}_k^1(\tau_1) \cdots \bar{\Phi}_k^1(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ \gamma_{j,k} \bar{\Phi}_k^1(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ \gamma_{j,k} \gamma_{k,k}^p \bar{\Phi}_k^1(\tau_1) & \text{if } \tau = [\tau_1]_{\theta_p}, \\ a_{j,k} & \text{if } \tau = [\]_{\bullet}, \\ \gamma_{j,k} & \text{if } \tau = [\]_{\circ}, \end{cases}$$

(3.3c)

Similarly, the recursive formula in equation (3.4) maps the visual representation of an LW_1 -tree to its corresponding elementary differential in the Taylor series expansion

of the numerical solution computed using a LIRK-W scheme (2.10).

$$(3.4) \quad F^J(\tau)(y) = \begin{cases} \sum_{K_1, \dots, K_m} f_{K_1, \dots, K_m}^J(y) \cdot (F^{K_1}(\tau_1) \dots F^{K_m}(\tau_m))(y) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet} \\ \sum_K \mathbf{L}_{JK} F^K(\tau_1)(y) & \text{if } \tau = [\tau_1]_{\circ} \\ \sum_K \left(\frac{d^p}{dt^p} \mathbf{L}_{JK} \right) F^K(\tau_1)(y) & \text{if } \tau = [\tau_1]_{\theta_p} \\ \sum_K \mathbf{L}_{JK} y^K & \text{if } \tau = []_{\circ} \\ \sum_K \left(\frac{d^p}{dt^p} \mathbf{L}_{JK} \right) y^K & \text{if } \tau = []_{\theta_p} \end{cases}$$

REMARK 1. We note that the Taylor series expansion of the exact solution will be comprised of only terms containing the function $F(t, y)$ and its derivatives, and so these terms will be represented by trees containing only meagre nodes. Alternatively, the Taylor series expansion of the numerical solution will be comprised of terms containing the function $F(t, y)$ and its derivatives, as well as terms containing \mathbf{L} and its derivatives. For this reason, the order conditions corresponding to trees containing fat and square nodes will be set to zero, to eliminate any contribution of these terms to the error of the LIRK-W method.

THEOREM 3.1 (Order conditions for LIRK-W methods of type 1). A LIRK-W method of type 1 has order p iff the following order conditions hold:

$$(3.5) \quad \sum_j \Phi_j^1(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau \in T \quad \text{with } \rho(\tau) \leq p,$$

$$(3.6) \quad \sum_j \Phi_j^1(\tau) = 0 \quad \forall \tau \in LW_1 \setminus T \quad \text{with } \rho(\tau) \leq p.$$

Here $\rho(t)$ is the number of vertices of the tree t , and $\gamma(t)$ is the “product of $\rho(t)$ and all orders of the trees which appear, if the roots, one after another, are removed from t ” [7, Section II.2].

Proof. The proof follows from our discussion, equation (3.2) and the order conditions of Rosenbrock-W methods [8, Section IV.7]. \square

3.3. Order conditions for LIRK-W methods of type 2. Here we derive order conditions for LIRK-W methods of type 2 (2.11).

We once again make use of the expansion (3.1) to construct derivatives of the type 2 LIRK-W scheme to arrive at the result that

$$(3.7a) \quad (Y_i)^{(q)} \Big|_{h=0} = q \sum_{j=1}^{i-1} a_{i,j} \sum_{m \geq 1} \frac{\partial^m f}{\partial y^m} (Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)}) + q \mathbf{L} \sum_{j=1}^s (Y_j)^{(q-1)} \\ + q \sum_{k=0}^{q-1} \binom{q-1}{k} \gamma_{i,i}^{k+1} \left(\frac{d^k}{dt^k} \mathbf{L} \right) (Y_i)^{(q-1-k)}$$

$$(3.7b) \quad (y_{n+1})^{(q)} \Big|_{h=0} = q \sum_{j=1}^s b_j \sum_{m \geq 1} \frac{\partial^m f}{\partial y^m} (Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)}) + q \mathbf{L} \sum_{j=1}^s g_j (Y_j)^{(q-1)}$$

Leading to the definition of LW_2 -trees

$$LW_2 = \left\{ \begin{array}{l} N_3\text{-trees: fat vertices are singly branched, and} \\ \text{no tree has a square root, and} \\ \text{square vertices are singly branched} \\ \text{with square or fat children.} \end{array} \right\}$$

Figures 3.2 and 3.3, give the LW_2 -trees, τ_i , representing the elementary differentials, $f(\tau_i)$, present in the Taylor expansion of the exact and numerical solutions, as well as the coefficients of these terms, $\Phi^2(\tau_i)$ and $P(\tau_i)$ respectively.

REMARK 2. LW_2 -trees are a subset of LW_1 -trees, and so figures 3.2 and 3.3 contains trees which are not present in an expansion of the numerical solution of a LIRK-W method of type 2. These trees can be identified by the presence of a square root, and the corresponding entry for $\Phi^2(\tau_i)$ being listed as "N/A". In addition, $\Phi^1(\tau_i)$ and $\Phi^2(\tau_i)$ are identical, except in the case where τ_i contains a square node.

The recursive formulas in equation (3.8) map the visual representation of a given LW -tree to the left hand side of the order condition, $\Phi_j(\tau)$, corresponding to that tree.

$$(3.8a) \quad \Phi_j^2(\tau) = \begin{cases} b_j \bar{\Phi}_j^2(\tau_1) \cdots \bar{\Phi}_j^2(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ g_j \bar{\Phi}_j^2(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ b_j & \text{if } \tau = [\]_{\bullet}, \\ g_j & \text{if } \tau = [\]_{\circ}, \end{cases}$$

$$(3.8b) \quad \bar{\Phi}_j^2(\tau) = \begin{cases} a_{j,k} \bar{\Phi}_k^2(\tau_1) \cdots \bar{\Phi}_k^2(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ \gamma_{j,k} \bar{\Phi}_k^2(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ \gamma_{jj}^{p+1} \bar{\Phi}_j^2(\tau_1) & \text{if } \tau = [\tau_1]_{\theta_p}, \\ a_{j,k} & \text{if } \tau = [\]_{\bullet}, \\ \gamma_{j,k} & \text{if } \tau = [\]_{\circ}, \end{cases}$$

LW_1 and LW_2 map from trees to elementary differentials in the same way, so that once again equation (3.4) maps the visual representation of an LW_2 -tree to its corresponding elementary differential in the Taylor series expansion of the numerical solution using the LIRK-W scheme of type 2 in equation (2.11).

Making use of the same logic as for LIRK-W methods of type-1 we have that for LIRK-W methods of type 2

THEOREM 3.2 (Order conditions for LIRK-W methods of type 2). *A LIRK-W method of type 1 has order p iff the following order conditions hold:*

$$(3.9a) \quad \sum_j \Phi_j^2(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau \in T \quad \text{with } \rho(\tau) \leq p,$$

$$(3.9b) \quad \sum_j \Phi_j^2(\tau) = 0 \quad \forall \tau \in LW_2 \setminus T \quad \text{with } \rho(\tau) \leq p.$$

Here $\rho(t)$ is the number of vertices of the tree t , and $\gamma(t)$ is the "product of $\rho(t)$ and all orders of the trees which appear, if the roots, one after another, are removed from t " [7, Section II.2].

Proof. The proof follows from our discussion, equation (3.7) and the order conditions of Rosenbrock-W methods [8, Section IV.7]. \square

3.4. Order conditions for LIRK-W methods of type 3. Here we derive order conditions for LIRK-W methods of type 3 (2.12). Making use of the expansion (3.1) we see that

(3.10a)

$$(Y_i)^{(q)} \Big|_{h=0} = q \sum_{j=1}^{i-1} a_{i,j} \sum_{m \geq 1} \frac{\partial^m F}{\partial y^m} \left(Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)} \right) + q \sum_{j=1}^i \gamma_{i,j} \sum_{k=0}^{q-1} \binom{q-1}{k} \gamma_{i,i}^k \left(\frac{d^k}{dt^k} \mathbf{L} \right) Y_j^{(q-1-k)}$$

$$(3.10b) \quad (y_{n+1})^{(q)} \Big|_{h=0} = q \sum_{j=1}^s b_j \sum_{m >= 1} \frac{\partial^m f}{\partial y^m} \left(Y_j^{(\mu_1)}, \dots, Y_j^{(\mu_m)} \right) + q \mathbf{L} \sum_{j=1}^s g_j (Y_j)^{(q-1)}$$

Figures 3.2 and 3.3, give the LW_3 -trees, τ_i , representing the elementary differentials, $f(\tau_i)$, present in the Taylor expansion of the exact and numerical solutions, as well as the coefficients of these terms, $\Phi^3(\tau_i)$ and $P(\tau_i)$ respectively. The LW_3 -trees are defined exactly as the LW_2 -trees with only $\Phi^3(\tau)$ differing from its type 2 counterpart. The recursive definition of $\Phi^3(\tau_i)$ is given by (3.11).

$$(3.11a) \quad \Phi_j^3(\tau) = \begin{cases} b_j \bar{\Phi}_j^3(\tau_1) \cdots \bar{\Phi}_j^3(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ g_j \bar{\Phi}_j^3(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ b_j & \text{if } \tau = []_{\bullet}, \\ g_j & \text{if } \tau = []_{\circ}, \end{cases}$$

$$(3.11b) \quad \bar{\Phi}_j^2(\tau) = \begin{cases} a_{j,k} \bar{\Phi}_k^3(\tau_1) \cdots \bar{\Phi}_k^3(\tau_m) & \text{if } \tau = [\tau_1, \dots, \tau_m]_{\bullet}, \\ \gamma_{j,k} \bar{\Phi}_k^3(\tau_1) & \text{if } \tau = [\tau_1]_{\circ}, \\ \gamma_{jj}^p \gamma_{jk} \bar{\Phi}_k^3(\tau_1) & \text{if } \tau = [\tau_1]_{\theta_p}, \\ a_{j,k} & \text{if } \tau = []_{\bullet}, \\ \gamma_{j,k} & \text{if } \tau = []_{\circ}, \end{cases}$$

Similar to the reasoning for type 1 and 2 methods the order conditions for type 3 methods can be summarized by the following theorem.

THEOREM 3.3 (Order conditions for LIRK-W methods of type 3). *A LIRK-W method of type 1 has order p iff the following order conditions hold:*

$$(3.12a) \quad \sum_j \Phi_j^3(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau \in T \quad \text{with } \rho(\tau) \leq p,$$

$$(3.12b) \quad \sum_j \bar{\Phi}_j^3(\tau) = 0 \quad \forall \tau \in LW_3 \setminus T \quad \text{with } \rho(\tau) \leq p.$$

Here $\rho(t)$ is the number of vertices of the tree t , and $\gamma(t)$ is the ‘‘product of $\rho(t)$ and all orders of the trees which appear, if the roots, one after another, are removed from t ’’ [7, Section II.2].

Proof. The proof follows from our discussion, equation (3.10) and the order conditions of Rosenbrock-W methods [8, Section IV.7]. \square

4. Linear stability of LIRK-W methods. To examine the linear stability of the proposed method we solve the linear test problem

$$(4.1) \quad \frac{dy}{dt} = \mathbf{L}y + (\mathbf{J} - \mathbf{L})y, \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_n; \quad y(t) \in \mathbb{R}^N \quad \mathbf{J}, \mathbf{L} \in \mathbb{R}^{N \times N}.$$

and apply the method (2.4) in compact form

$$(4.2a) \quad \mathbf{Y} = \mathbf{1} \otimes y_n + \left[\mathbf{A} \otimes (h\mathbf{J} - h\mathbf{L}(h)) + \widehat{\mathbf{A}} \otimes h\mathbf{L}(h) \right] \mathbf{Y},$$

$$(4.2b) \quad y_{n+1} = y_n + \left[\mathbf{b}^T \otimes (h\mathbf{J} - h\mathbf{L}(h)) + \widehat{\mathbf{b}}^T \otimes h\mathbf{L}(h) \right] \mathbf{Y},$$

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,s} \\ \vdots & \ddots & \vdots \\ a_{s,1} & \cdots & a_{s,s} \end{bmatrix}, \quad \widehat{\mathbf{A}} = \begin{bmatrix} \widehat{a}_{1,1} & \cdots & \widehat{a}_{1,s} \\ \vdots & \ddots & \vdots \\ \widehat{a}_{s,1} & \cdots & \widehat{a}_{s,s} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_s \end{bmatrix}, \quad \widehat{\mathbf{b}} = \begin{bmatrix} \widehat{b}_1 \\ \vdots \\ \widehat{b}_s \end{bmatrix}$$

and

$$\mathbf{Y} = [Y_1^T \quad \cdots \quad Y_s^T]^T.$$

4.1. Type 1 methods. Application of a type 1 method (2.10) to (4.1) gives:

$$Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} (\mathbf{J} - \mathbf{L}_j) Y_j + h \sum_{j=1}^i \widehat{a}_{i,j} \mathbf{L}_j Y_j,$$

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i (\mathbf{J} - \mathbf{L}_i) Y_i + h \sum_{i=1}^s \widehat{b}_i \mathbf{L}_i Y_i.$$

In compact notation we have:

$$\mathbf{Y} = \mathbf{1} \otimes y_n + \left[(\mathbf{A} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{J} - h\mathbf{L}_i) + (\widehat{\mathbf{A}} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{L}_i) \right] \mathbf{Y},$$

$$y_{n+1} = y_n + \left[(\mathbf{b}^T \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{J} - h\mathbf{L}_i) + (\widehat{\mathbf{b}}^T \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{L}_i) \right] \mathbf{Y}.$$

Solving for y_{n+1} in terms of y_n leads to the transfer matrix:

$$y_{n+1} = R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) y_n,$$

$$R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) = \mathbf{I} + \left[(\mathbf{b}^T \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{J} - h\mathbf{L}_i) + (\widehat{\mathbf{b}}^T \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{L}_i) \right] \cdot$$

$$\left[\mathbf{I} - (\mathbf{A} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{J} - h\mathbf{L}_i) - (\widehat{\mathbf{A}} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{L}_i) \right]^{-1} (\mathbf{1} \otimes \mathbf{I}).$$

For a stiffly accurate method with

$$(4.3) \quad \mathbf{b}^T = \mathbf{e}_s^T \mathbf{A} \quad \text{and} \quad \widehat{\mathbf{b}}^T = \mathbf{e}_s^T \widehat{\mathbf{A}},$$

standard calculations give:

$$R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) = (\mathbf{e}_s^T \otimes \mathbf{I}) \cdot \left[\mathbf{I} - (\mathbf{A} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{J} - h\mathbf{L}_i) - (\widehat{\mathbf{A}} \otimes \mathbf{I}_N) \text{blkdiag}_{i=1,\dots,s} (h\mathbf{L}_i) \right]^{-1} (\mathbf{1} \otimes \mathbf{I}).$$

This matrix goes to zero for stiff linear parts:

$$\|h\mathbf{L}_i\| \rightarrow \infty, \quad i = 1, \dots, s \quad \Rightarrow \quad R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) \rightarrow 0.$$

4.2. Type 2 methods. Application of a type 2 method (2.11) to (4.1) gives:

$$Y_i = y_n + h \sum_{j=1}^{i-1} a_{i,j} F(Y_j) + h \mathbf{L} \sum_{j=1}^{i-1} \gamma_{i,j} Y_j + h \gamma_{i,i} \mathbf{L}_i Y_i,$$

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i F(Y_i) + h \mathbf{L} \sum_{i=1}^s g_i Y_i.$$

In compact form:

$$\mathbf{Y} = \mathbf{1} \otimes y_n + \left[\mathbf{A} \otimes (h\mathbf{J} - h\mathbf{L}) + \widehat{\mathbf{A}} \otimes h\mathbf{L} + \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right] \mathbf{Y},$$

$$y_{n+1} = y_n + \left[\mathbf{b}^T \otimes (h\mathbf{J} - h\mathbf{L}) + \widehat{\mathbf{b}}^T \otimes h\mathbf{L} \right] \mathbf{Y},$$

where blkdiag denotes a block-diagonal matrix with the blocks given by its arguments; the block indices should be clear from the context. The transfer matrix reads:

$$R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) = \mathbf{I} + \left[(\mathbf{b}^T \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes (h\mathbf{J} - h\mathbf{L})) + (\widehat{\mathbf{b}}^T \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes h\mathbf{L}) \right] \cdot$$

$$\left[\mathbf{I} - (\mathbf{A} \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes (h\mathbf{J} - h\mathbf{L})) - (\widehat{\mathbf{A}} \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes h\mathbf{L}) \right. \\ \left. - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]^{-1} (\mathbf{1} \otimes \mathbf{I}).$$

For a stiffly accurate method (4.3) standard calculations give:

$$R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) = (\mathbf{e}_s^T \otimes \mathbf{I}) \cdot \left[\mathbf{I} - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]_{i=1, \dots, s}$$

$$\cdot \left[\mathbf{I} - (\mathbf{A} \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes (h\mathbf{J} - h\mathbf{L})) - (\widehat{\mathbf{A}} \otimes \mathbf{I}_N) (\mathbf{I}_s \otimes h\mathbf{L}) \right. \\ \left. - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]_{i=1, \dots, s}^{-1} (\mathbf{1} \otimes \mathbf{I})$$

$$= (\mathbf{e}_s^T \otimes \mathbf{I}) \cdot \left[\mathbf{I} - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]_{i=1, \dots, s}$$

$$\cdot \left[\mathbf{I} - \mathbf{A} \otimes (h\mathbf{J}) - \text{tril}(\mathbf{\Gamma}) \otimes (h\mathbf{L}) - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]_{i=1, \dots, s}^{-1} (\mathbf{1} \otimes \mathbf{I})$$

$$= (\mathbf{e}_s^T \otimes \mathbf{I}) \cdot$$

$$\cdot \left[\mathbf{I} - \left(\mathbf{A} \otimes (h\mathbf{J}) + \text{tril}(\mathbf{\Gamma}) \otimes (h\mathbf{L}) \right) \left[\mathbf{I} - \text{blkdiag}(\gamma_{i,i}(h\mathbf{L}_i - h\mathbf{L})) \right]_{i=1, \dots, s}^{-1} \right]^{-1} (\mathbf{1} \otimes \mathbf{I}),$$

where we denote

$$\text{tril}(\mathbf{\Gamma}) = \mathbf{\Gamma} - \text{diag}_{1=1, \dots, s}(\gamma_{i,i}).$$

The stability matrix goes to zero when $\mathbf{L}_i \approx \mathbf{L}$ in the sense that the difference increases slower than the matrices with increasing stiffness:

$$\frac{\|h\mathbf{L}\|}{\|h\mathbf{L} - h\mathbf{L}_i\|} \rightarrow \infty, \quad i = 1, \dots, s \quad \Rightarrow \quad R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) \rightarrow 0.$$

or

$$\frac{\|h\mathbf{J}\|}{\|h\mathbf{L} - h\mathbf{L}_i\|} \rightarrow \infty, \quad i = 1, \dots, s \quad \Rightarrow \quad R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) \rightarrow 0.$$

Alternatively, the stability matrix goes to one when the time dependent linear term $h\mathbf{L}_i$ grows more quickly than either $h\mathbf{J}$ or $h\mathbf{L}$ with increasing stiffness, so that

$$\frac{\|h\mathbf{L}\| + \|h\mathbf{J}\|}{\|h\mathbf{L}_i\|} \rightarrow 0, \quad i = 1, \dots, s \quad \Rightarrow \quad R(h\mathbf{J}, h\mathbf{L}_1, \dots, h\mathbf{L}_s) \rightarrow 1.$$

From this it is clear that the stability of the method is dependent on the choice of both \mathbf{L} and \mathbf{L}_i .

5. Construction of Practical LIRK-W Methods.

5.1. A third-order LIRK-W method of type 1. For stability considerations we seek methods which are stiffly accurate (4.3), i.e.,

$$(5.1) \quad b_i = a_{s,i} \quad \text{and} \quad \hat{b}_i = \hat{a}_{s,i} \quad \text{for} \quad i = 1, \dots, s.$$

Moreover we wish to make use of an approximate matrix factorization, specifically the alternating directions approximation. Our choice of general form (2.10) and the structure of $\tilde{\mathbf{L}}_i$ in equation (2.9) naturally leads to the choice that

$$(5.2) \quad \hat{a}_{i,i} = \sum_{j=1}^{i-1} a_{i,j}.$$

Finally, for convenience we require that

$$(5.3) \quad \sum_{j=1}^{i-1} a_{i,j} = \sum_{j=1}^i \hat{a}_{i,j}$$

The requirements imposed by (5.1), (5.2), and (5.3) taken together imply that a method with general form (2.5) must have the additional following properties:

$$(5.4) \quad \sum_{j=1}^i \gamma_{i,j} = 0, \quad \gamma_{i,i} = \sum_{j=1}^{i-1} a_{i,j}, \quad \sum_{j=1}^{i-1} \gamma_{i,j} = -\gamma_{i,i}, \quad \sum_{i=1}^s g_i = 0, \quad g_i = \gamma_{s,i}.$$

Interestingly, the first of these conditions, implied by (5.3), automatically satisfies all order conditions coming from trees in figures 3.2 and 3.3 which end in a fat node not directly preceded by a square node. Additionally, the simplifications in equations (5.1), (5.2), (5.3), and (5.4) means that several order conditions reduce to the same as those coming from other trees, so that:

$$\Phi(\tau_5) = \Phi(\tau_7), \quad \Phi(\tau_{14}) = \Phi(\tau_{23}), \quad \Phi(\tau_{15}) = \Phi(\tau_{20}), \quad \Phi(\tau_{22}) = \Phi(\tau_{19}).$$

In this way number of order conditions required to construct such a method of order three reduces from the original twenty-three to a much more reasonable nine. Figure 5.1 shows these trees and the order conditions associated with them.

$$\begin{aligned}
a &= \begin{bmatrix} 0 & & & & \\ 0.520300000000000 & 0 & & & \\ 0.026500000000000 & 0.938000000000000 & 0 & & \\ 0.122175553766880 & 0.105600000000000 & 0.018300000000000 & 0 & \\ -0.033950868284890 & 0.218016324016351 & 0.258600000000000 & 0.557334544268539 & 0 \end{bmatrix} \\
\gamma &= \begin{bmatrix} 0 & & & & \\ -0.520300000000000 & 0.520300000000000 & & & \\ 0.911500000000000 & -1.876000000000000 & 0.964500000000000 & & \\ -0.401069249711528 & 0.663393695944647 & -0.508400000000000 & 0.246075553766880 & \\ -0.155925222099085 & -0.084089256959580 & -1.070724285228281 & 0.310738764286946 & 1 \end{bmatrix} \\
b &= [-0.033950868284890 \quad 0.218016324016351 \quad 0.258600000000000 \quad 0.557334544268539 \quad 0] \\
g &= [-0.155925222099085 \quad -0.084089256959580 \quad -1.070724285228281 \quad 0.310738764286946 \quad 1]
\end{aligned}$$

TABLE 5.1
Method coefficients for a third order LIRK-W method of type 1.

These conditions, with significant simplification coming from equations (5.1), (5.2), (5.3), and (5.4) as well as some algebraic manipulation are given for a five-stage method as:

$$\begin{aligned}
(5.5a) \quad & a_{5,1} + a_{5,2} + a_{5,3} + a_{5,4} = 1 \\
(5.5b) \quad & a_{5,2}a_2 + a_{5,3}a_3 + a_{5,4}a_4 = \frac{1}{2} \\
(5.5c) \quad & \gamma_{5,2}a_2 + \gamma_{5,3}a_3 + \gamma_{5,4}a_4 = -1 \\
(5.5d) \quad & a_{5,2}a_2^2 + a_{5,3}a_3^2 + a_{5,4}a_4^2 = \frac{1}{3} \\
(5.5e) \quad & a_{5,3}a_{3,2}a_2 + a_{5,4}a_{4,3}a_3 + a_{5,4}a_{4,2}a_2 = \frac{1}{6} \\
(5.5f) \quad & a_{5,3}\gamma_{3,2}a_2 + a_{5,4}\gamma_{4,3}a_3 + a_{5,4}\gamma_{4,2}a_2 = -\frac{1}{3} \\
(5.5g) \quad & \gamma_{5,3}\gamma_{3,2}a_2 + \gamma_{5,4}\gamma_{4,3}a_3 + \gamma_{5,4}\gamma_{4,2}a_2 = 1 \\
(5.5h) \quad & \gamma_{5,3}a_{3,2}a_2 + \gamma_{5,4}a_{4,3}a_3 + \gamma_{5,4}a_{4,2}a_2 = -\frac{1}{2} \\
(5.5i) \quad & \gamma_{5,2}a_2^2 + \gamma_{5,3}a_3^2 + \gamma_{5,4}a_4^2 = -1
\end{aligned}$$

Table 5.1 gives the coefficients for a type 1 LIRK-W method of third order.

5.2. Third-order LIRK-W methods of type 2. For stability considerations we again seek a method which is stiffly accurate, and so satisfies (5.1), as well as the condition that (5.3) to reduce the number of required order conditions. Due to the difference in general forms we no longer impose the condition (5.2), which provides the freedom to impose the condition that

$$(5.6) \quad \gamma_{i,i} = \begin{cases} 0 & \text{for } i = 1 \\ \gamma & \text{for } i = 2, \dots, s \end{cases}$$

τ_1		f^J	$\sum b_j = 1$
τ_3		$f_K^J f^K$	$\sum b_j a_{j,k} = 1/2$
τ_5		$\mathbf{L}_{JK} f^K$	$\sum g_j a_{j,k} = 0$
τ_8		$f_{KL}^J f^K f^L$	$\sum b_j a_{j,k} a_{j,l} = 1/3$
τ_{11}		$f_K^J f_L^K f^L$	$\sum b_j a_{j,k} a_{k,l} = 1/6$
τ_{14}		$f_K^J \mathbf{L}_{KL} f^L$	$\sum b_j \gamma_{j,k} a_{k,l} = 0$
τ_{15}		$\mathbf{L}_{JK} \mathbf{L}_{KL} f^L$	$\sum g_j \gamma_{j,k} a_{k,l} = 0$
τ_{16}		$\mathbf{L}_{JK} f_L^K f^L$	$\sum g_j a_{j,k} a_{k,l} = 0$
τ_{22}		$\mathbf{L}_{JK} \mathbf{L}'_{KL} y^L$	$\sum g_j \gamma_{j,j}^2 = 0$

FIG. 5.1. Butcher trees and LIRK-W conditions up to order three for a type 1 method.

Taking requirements imposed by (5.1), (5.3), and (5.6) taken together imply that a method with general form (2.11) must have the additional following properties:

$$(5.7) \quad \sum_{j=1}^i \gamma_{i,j} = 0, \quad \sum_{j=1}^{i-1} \gamma_{i,j} = -\gamma_{i,i}, \quad \sum_{i=1}^s g_i = 0, \quad g_i = \gamma_{s,i}.$$

Once again the first of these conditions, implied by (5.3), automatically satisfies all order conditions coming from trees in figures 3.2 and 3.3 which end in a fat node not directly preceded by a square node. In this way the number of required order conditions required to construct such a method of order three reduces from the original nineteen to a much more reasonable ten. Figure 5.2 shows these trees and the order conditions associated with them.

Examining the order conditions coming from the first and last two trees of figure 5.2 leads to the additional strategic choice that

$$(5.8) \quad a_{5,1} = 1, \quad \gamma_{5,1} = 0$$

so that the order conditions coming from τ_{18} and τ_{19} are automatically satisfied.




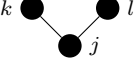
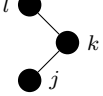
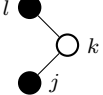
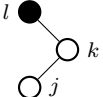
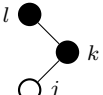
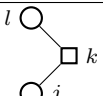
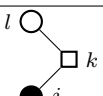
τ_1		f^J	$\sum b_j = 1$
τ_3		$f_K^J f^K$	$\sum b_j a_{j,k} = 1/2$
τ_5		$\mathbf{L}_{JK} f^K$	$\sum g_j a_{j,k} = 0$
τ_8		$f_{KL}^J f^K f^L$	$\sum b_j a_{j,k} a_{j,l} = 1/3$
τ_{11}		$f_K^J f_L^K f^L$	$\sum b_j a_{j,k} a_{k,l} = 1/6$
τ_{14}		$f_K^J \mathbf{L}_{KL} f^L$	$\sum b_j \gamma_{j,k} a_{k,l} = 0$
τ_{15}		$\mathbf{L}_{JK} \mathbf{L}_{KL} f^L$	$\sum g_j \gamma_{j,k} a_{k,l} = 0$
τ_{16}		$\mathbf{L}_{JK} f_L^K f^L$	$\sum g_j a_{j,k} a_{k,l} = 0$
τ_{18}		$\mathbf{L}_{JK} \mathbf{L}'_{KL} y^L$	$g_j \gamma_{j,j}^2 = 0$
τ_{19}		$f_K^J \mathbf{L}'_{KL} y^L$	$b_j \gamma_{j,j}^2 = 0$

FIG. 5.2. Butcher trees and LIRK-W conditions up to order three for a type 2 method.

These conditions, with significant simplification coming from equations (5.1), (5.3), (5.7), and (5.8) as well as some algebraic manipulation are given for a five-stage method as:

$$\begin{aligned}
a &= \begin{bmatrix} 0 & & & & & & \\ & \frac{1}{6} & & & & & \\ & \frac{1}{3} - \frac{9\gamma+2\gamma_{5,4}}{3(5\gamma+2\gamma_{5,4})} & & & & & \\ \frac{1}{2} - \frac{2}{3}(3a_{4,3} - 1) - a_{4,3} & & \frac{9\gamma+2\gamma_{5,4}}{3(5\gamma+2\gamma_{5,4})} & & 0 & & \\ & 1 & \frac{2}{3}(3a_{4,3} - 1) & & a_{4,3} & & 0 \\ & & -\frac{3}{2} & & 0 & & \frac{3}{2} \\ & & & & & & 0 \end{bmatrix} \\
\gamma &= \begin{bmatrix} 0 & & & & & & \\ & -\gamma & & & & & \\ & \frac{2(3\gamma^2+\gamma\gamma_{5,4})}{5\gamma+2\gamma_{5,4}} - \gamma & & & & & \\ 2(\gamma + \gamma_{4,3}) - \gamma - \gamma_{4,3} & & -\frac{2(3\gamma^2+\gamma\gamma_{5,4})}{5\gamma+2\gamma_{5,4}} & & \gamma & & \\ & 0 & 4\gamma + \gamma_{5,4} & & -5\gamma - 2\gamma_{5,4} & & \gamma_{5,4} \\ & & & & & & \gamma \end{bmatrix} \\
b_i &= [1 \quad -\frac{3}{2} \quad 0 \quad \frac{3}{2} \quad 0] \\
g_i &= [0 \quad 4\gamma + \gamma_{5,4} \quad -5\gamma - 2\gamma_{5,4} \quad \gamma_{5,4} \quad \gamma]
\end{aligned}$$

TABLE 5.2
Method coefficients for a third order LIRK-W method of type 2.

$$\begin{aligned}
(5.9a) \quad & a_{5,2} + a_{5,3} + a_{5,4} = 0 \\
(5.9b) \quad & \gamma_{5,2} + \gamma_{5,3} + \gamma_{5,4} = -\gamma \\
(5.9c) \quad & a_{5,2}a_2 + a_{5,3}a_3 + a_{5,4}a_4 = \frac{1}{2} \\
(5.9d) \quad & \gamma_{5,2}a_2 + \gamma_{5,3}a_3 + \gamma_{5,4}a_4 = -\gamma \\
(5.9e) \quad & a_{5,2}a_2^2 + a_{5,3}a_3^2 + a_{5,4}a_4^2 = \frac{1}{3} \\
(5.9f) \quad & a_{5,3}a_{3,2}a_2 + a_{5,4}a_{4,2}a_2 + a_{5,4}a_{4,3}a_3 = \frac{1}{6} \\
(5.9g) \quad & a_{5,3}\gamma_{3,2}a_2 + a_{5,4}\gamma_{4,2}a_2 + a_{5,4}\gamma_{4,3}a_3 = -\frac{1}{2}\gamma \\
(5.9h) \quad & \gamma_{5,3}\gamma_{3,2}a_2 + \gamma_{5,4}\gamma_{4,2}a_2 + \gamma_{5,4}\gamma_{4,3}a_3 = \gamma^2 \\
(5.9i) \quad & \gamma_{5,3}a_{3,2}a_2 + \gamma_{5,4}a_{4,2}a_2 + \gamma_{5,4}a_{4,3}a_3 = -\frac{1}{2}\gamma
\end{aligned}$$

6. Numerical Experiments. We show here the convergence of both type 1 and type 2 LIRK-W schemes when applied to the shallow water equations on a sphere. The shallow water equations in spherical coordinates are

$$(6.1a) \quad \frac{\partial u}{\partial t} + \frac{1}{a \cos \theta} \left(u \frac{\partial u}{\partial \lambda} + v \cos \theta \frac{\partial u}{\partial \theta} \right) - \left(f + \frac{u \tan \theta}{a} \right) a + \frac{g}{a \cos \theta} \frac{\partial h}{\partial \lambda} = 0$$

$$(6.1b) \quad \frac{\partial v}{\partial t} + \frac{1}{a \cos \theta} \left(u \frac{\partial v}{\partial \lambda} + v \cos \theta \frac{\partial v}{\partial \theta} \right) + \left(f + \frac{u \tan \theta}{a} \right) u + \frac{g}{a} \frac{\partial h}{\partial \theta} = 0$$

$$(6.1c) \quad \frac{\partial h}{\partial t} + \frac{1}{a \cos \theta} \left(\frac{\partial(hu)}{\partial \lambda} + \frac{\partial(hv \cos \theta)}{\partial \theta} \right) = 0.$$

Where $f = 2\Omega \sin \theta$, h is the height of the atmosphere, u is the zonal wind component, v is the meridional wind component, θ and λ are the latitudinal and longitudinal di-

rections, a is the radius of the earth, Ω is the rotational velocity of the earth, and g is the gravitational constant. The space discretization is performed using the unstaggered Turkel-Zwas scheme [10, 9], with 72 nodes in the longitudinal direction and 36 nodes in the latitudinal direction. Figure 6.1 confirms the third order convergence of type 1 and type 2 LIRK-W schemes applied to the spherical shallow water equations, when $\mathbf{L} = \mathbf{J}$.

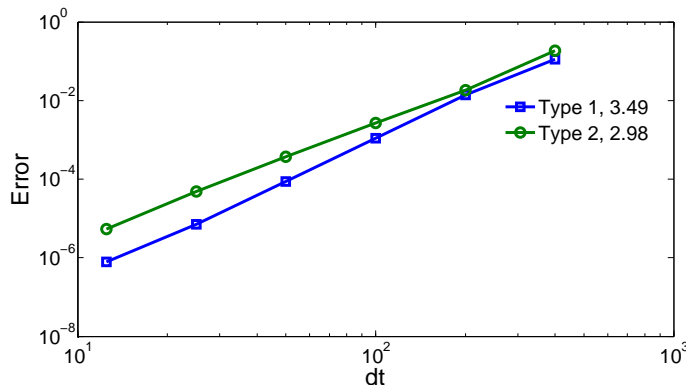


FIG. 6.1. Convergence diagram for LIRK-W methods applied to spherical shallow water equations.

The numerical exploration of these schemes is ongoing, and work is currently being done to apply these methods to large computational fluid dynamics models. This exploration will include the application of several different formulations of the time dependent, and stage varying linear term $\mathbf{L}_i y$.

7. Conclusions. We have presented a new class of time-integration schemes, which permit an arbitrary, time dependent, and stage varying linear approximation of the stiff system dynamics, called Linearly-Implicit Runge-Kutta-W methods. These schemes maintain high-order when making use of an approximate matrix factorization, and can be reformulated to permit other time dependent approximations. Future work that explores new formulations of the method and order conditions to permit new, or different, approximations may lead to more efficient integrators.

References.

REFERENCES

- [1] G. AKRIVIS AND M. CROUZEIX, Linearly implicit methods for nonlinear parabolic equations, Mathematics of Computation, 73 (2004), pp. pp. 613–635.
- [2] G. AKRIVIS, O. KARAKASHIAN, AND F. KARAKATSANI, Linearly implicit methods for nonlinear evolution equations, Numerische Mathematik, 94 (2003), pp. 403–418.
- [3] U. ASCHER, S. RUUTH, AND R. SPITERI, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Applied Numerical Mathematics, 25 (1997), pp. 151–167.
- [4] M. CALVO, J. DE FRUTOS, AND J. NOVO, Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations, Applied Numerical Mathematics, 37 (2001), pp. 535–549.
- [5] J. DOUGLAS JR., Alternating direction methods for three space variables, Numerische Mathematik, 4 (1962), pp. 41–63.
- [6] I. GROOMS AND K. JULIEN, Linearly implicit methods for nonlinear PDEs with linear dispersion and dissipation, Journal of Computational Physics, 230 (2011), pp. 3630 – 3650.

- [7] E. HAIRER, S. NORSETT, AND G. WANNER, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer, 2008.
- [8] E. HAIRER AND G. WANNER, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, 2002.
- [9] I. NAVON AND J. YU, Exshall: A turkel-zwas explicit large time-step fortran program for solving the shallow-water equations in spherical coordinates, *Computers & Geosciences*, 17 (1991), pp. 1311 – 1343.
- [10] I. M. NAVON AND R. DE VILLIERS, The application of the turkelzwas explicit large time-step scheme to a hemispheric barotropic model with constraint restoration, *Monthly Weather Review*, 115 (1987), pp. 1036–1052.
- [11] D. PEACEMAN AND H. RACHFORD JR., The numerical solution of parabolic and elliptic differential equations, *Journal of the Society for Industrial and Applied Mathematics*, 3 (1955), pp. 28–41.
- [12] J. RANG AND L. ANGERMANN, New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1, *BIT Numerical Mathematics*, 45 (2005), pp. 761–787.
- [13] R. WEINER, B. SCHMITT, AND H. PODHAISKY, ROWMAP—a ROW-code with Krylov techniques for large stiff ODEs, *Applied Numerical Mathematics*, 25 (1997), pp. 303–319.
- [14] Y. SAAD, Iterative methods for sparse linear systems, PWS Pub. Co., Boston, 1996.
- [15] A. SANDU AND M. GÜNTHER, A generalized-structure approach to additive Runge-Kutta methods, *SIAM Journal on Numerical Analysis*, 53 (2015), pp. 17–42.
- [16] P. TRANQUILLI, S. R. GLANDON, A. SARSHAR, AND A. SANDU, Analytical jacobian-vector products for the matrix-free time integration of partial differential equations, *Journal of Computational and Applied Mathematics*, (2016), pp. –.
- [17] P. TRANQUILLI AND A. SANDU, Rosenbrock-krylov methods for large systems of differential equations, *SIAM J. Scientific Computing*, 36 (2014).