# LINEARLY IMPLICIT MULTISTEP METHODS FOR TIME INTEGRATION *

ROSS GLANDON †, MAHESH NARAYANAMURTHI ‡, AND ADRIAN SANDU §

**Abstract.** Time integration methods for solving initial value problems are an important component of many scientific and engineering simulations. Implicit time integrators are desirable for their stability properties, significantly relaxing restrictions on timestep size. However, implicit methods require solutions to one or more systems of nonlinear equations at each timestep, which for large simulations can be prohibitively expensive. This paper introduces a new family of linearly implicit multistep methods (LIMM), which only requires the solution of one linear system per timestep. Order conditions and stability theory for these methods are presented, as well as design and implementation considerations. Practical methods of order up to five are developed that have similar error coefficients, but improved stability regions, when compared to the widely used BDF methods. Numerical testing of a self-starting variable stepsize and variable order implementation of the new LIMM methods shows measurable performance improvement over a similar BDF implementation.

**Key words.** Time integration, ODEs, linear multistep methods, linearly implicit schemes

**AMS subject classifications.** 65L04, 65L05, 65L06

## 1. Introduction.

In this paper we are concerned with the numerical solution of initial value problems (IVP):

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t,\mathbf{y}), \quad t_0 \le t \le t_F, \quad \mathbf{y}(t_0) = y_0; \quad \mathbf{y}(t) \in \mathbb{R}^{\mathrm{N}}, \quad \mathbf{f} : \mathbb{R} \times \mathbb{R}^{\mathrm{N}} \to \mathbb{R}^{\mathrm{N}}. \tag{1.1}$$

Systems of ordinary differential equations (ODEs) of this form appear in a wide variety of scientific and engineering simulations. Some types of simulations, such as those of chemical kinetics or aerosol dynamics [16, 43], can be directly modeled by systems of densely coupled ODEs. Others, such as those of fluid dynamics [30], arise from the semi-discretization in space of partial differential equations (PDEs) via the method of lines, resulting in a large sparse system of ODEs.

Two time integration families of methods are used to discretize (1.1): explicit or implicit. Explicit schemes advance the solution to a new timestep using only information from previous steps, and are simple in structure with very low computational cost per timestep. However, they have stability limitations that result in problem-dependent bounds on the largest allowable stepsizes. For stiff problems, implicit time integration methods that determine solutions at a new timestep using both past and future information, are preferable. These schemes avoid stability-bound stepsize limitations at the cost of solving one or more nonlinear systems per timestep.

As solving large nonlinear systems can be very expensive, many types of linearly implicit methods have been developed that only require solutions of linear systems at each step. Rosenbrock methods [39] (and their many extensions [20, 47, 50]) are

*Submitted to the editors on 5/11/2020.

**Funding:** This work was funded by awards NSF ACI–1709727, NSF CDS&E–MSS 1953113, DOE ASCR DE–SC0021313, AFOSR DDDAS FA9550-17-1-0015, and by the Computational Science Laboratory at Virginia Tech.

†Computational Science Laboratory, Department of Computer Science, Virginia Tech. Blacksburg, Virginia 24060 (rossg42@vt.edu)

‡Computational Science Laboratory, Department of Computer Science, Virginia Tech. Blacksburg, Virginia 24060 (maheshnm@vt.edu)

§Computational Science Laboratory, Department of Computer Science, Virginia Tech. Blacksburg, Virginia 24060 (sandu@cs.vt.edu)

linearized implicit Runge-Kutta methods. Implicit-explicit (IMEX) methods [1, 2, 14, 38, 54–56] couple an implicit scheme for the stiff component with an explicit scheme for the non-stiff component of a split problem. A common splitting treats the nonlinear part of the problem explicitly and the linear part implicitly, therefore avoiding the need for nonlinear solves. Exponential integrators [27, 31–33, 48, 49] also effectively treat a linear-nonlinear problem splitting, with the linear portion solved via an exponential integrating factor. It is also common to linearize implicit methods which would normally require nonlinear solves by taking only a single Newton iteration [46, 51], or to replace parts of the implicit scheme with an extrapolation of past values [12, 13, 18, 52], at the possible cost of order reduction and/or reduced stability.

In this paper we construct linearly implicit multistep methods (LIMM) in much the same way that Rosenbrock methods are obtained from implicit Runge-Kutta methods. We determine order conditions which account for the linearization [6, 25], and solve for a family of $k$-step order $k$ methods for $k = 1, \dots, 5$. LIMM methods have more free coefficients than traditional linear multistep methods with the same number of steps, and this additional freedom enables the optimization of accuracy and stability properties. The new schemes designed herein have linear stability regions larger than, and error constants comparable to, the widely used BDF [15] family of methods.

The remainder of the paper is organized as follows. Section 2 describes the construction of the LIMM general form. Section 3 builds order conditions for $k$-step LIMM methods of order $p$. Section 4 considers the linear stability of LIMM methods, and Section 5 their convergence for step sizes not limited by the stiffness of the system. Section 6 discusses the design of new methods with optimal stability properties and error coefficients, and develops a family of $k$-step order $k$ methods with $k = 1, \dots, 5$. Section 7 provides the details necessary for an efficient implementation of LIMM methods with variable stepsize and variable order. Section 8 reports numerical results comparing LIMM to BDF methods. Finally, Section 9 draws concluding remarks.

**2. Linearly Implicit Multistep Methods.** A linear $k$-step method computes the solution of (1.1) as follows [25, Chapter III.2]:

$$\sum_{i=-1}^{k-1} \alpha_i \, \mathbf{y}_{n-i} = h_n \sum_{i=-1}^{k-1} \beta_i \, \mathbf{f}(t_{n-i}, \mathbf{y}_{n-i}), \tag{2.1}$$

where the numerical solution $\mathbf{y}_{n-i} \approx \mathbf{y}(t_{n-i})$ and the step size is $h_n = t_{n+1} - t_n$. The simplest way to obtain a linearly implicit multistep method is to linearize the implicit evaluation of $\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1})$ in (2.1) about $(t_n, \mathbf{y}_n)$, to obtain:

$$\begin{aligned} \sum_{i=-1}^{k-1} \alpha_i \, \mathbf{y}_{n-i} = h_n \sum_{i=0}^{k-1} \beta'_i \, \mathbf{f}(t_{n-i}, \mathbf{y}_{n-i}) \\ + h_n \beta_{-1} \, \mathbf{f}_{\mathbf{y}}(t_n, \mathbf{y}_n) \, (\mathbf{y}_{n+1} - \mathbf{y}_n) + h_n^2 \beta_{-1} \frac{\partial \mathbf{f}}{\partial t}(t_n, \mathbf{y}_n), \end{aligned} \tag{2.2}$$

where $\beta'_0 = \beta_0 + \beta_{-1}$, $\beta'_i = \beta_i$ for $i = 1, \dots, k-1$.

*Remark* 2.1 (Autonomous and non-autonomous forms). In order to simplify the notation in the remainder of the paper, we will make use of the autonomous form of (1.1), with $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y})$ and the notation

$$\mathbf{f}_n \coloneqq \mathbf{f}(\mathbf{y}_n), \quad \mathbf{J}_n \coloneqq \mathbf{f}_{\mathbf{y}}(\mathbf{y}_n).$$

As the original non-autonomous form can be recovered by stacking $t$ with the vector $\mathbf{y}$ and making corresponding changes to $\mathbf{f}$ and $\mathbf{J}$, the methods developed for autonomous problems are also applicable to non-autonomous problems.

Now, because the direct linearization approach in (2.2) maintains the same number of degrees of freedom, one can expect a degradation of accuracy and stability properties of (2.2) when compared to the standard nonlinear scheme (2.1) with the same number of steps. In order to increase the number of degrees of freedom one can generalize the approach by retaining in the formulation past linearized steps scaled by new $\mu$ coefficients:

$$\sum_{i=-1}^{k-1} \alpha_i \, \mathbf{y}_{n-i} = h_n \sum_{i=0}^{k-1} \beta_i \, \mathbf{f}_{n-i} + h_n \sum_{i=-1}^{k-1} \mu_i \, \mathbf{J}_{n-i-1} \, \mathbf{y}_{n-i}. \tag{2.3}$$

However, method (2.3) requires the storage of multiple past Jacobian-vector products in addition to past solutions and function values, which is not desirable. For this reason we consider a hybrid of the two approaches (2.2) and (2.3), and, drawing inspiration from Rosenbrock methods [39], [26, Section IV.7], define the following computational process.

DEFINITION 2.2 (LIMM methods). *A linearly implicit multistep method (*LIMM*) advances the numerical solution of (1.1) over one step $[t_n, t_{n+1}]$ as follows:*

$$\sum_{i=-1}^{k-1} \alpha_i \, \mathbf{y}_{n-i} = h_n \sum_{i=0}^{k-1} \beta_i \, \mathbf{f}_{n-i} + h_n \, \mathbf{J}_n \left( \sum_{i=-1}^{k-1} \mu_i \, \mathbf{y}_{n-i} + h_n \sum_{i=0}^{k-1} \nu_i \, \mathbf{f}_{n-i} \right), \tag{2.4}$$

*where, without loss of generality, $\alpha_{-1} = 1$.*

*A linearly implicit multistep method of W-type (*LIMM-W*) advances the numerical solution using (2.4), where the exact Jacobian $\mathbf{J}_n$ is replaced by an arbitrary matrix $\mathbf{A}_n$. This matrix is chosen to ensure the numerical stability of the scheme, but without impacting the order of accuracy of the method.*

*The non-autonomous form of (2.4) adds the following term to the right side of equation (2.4)*

$$h_n \frac{\partial \mathbf{f}}{\partial t}(t_n, \mathbf{y}_n) \left( \sum_{i=-1}^{k-1} \mu_i t_{n-i} + h_n \sum_{i=0}^{k-1} \nu_i \right).$$

We note that $k$-step LIMM schemes (2.4) have $4k+1$ free coefficients (compared to $2k+1$ for classical nonlinear methods (2.1)), and require storing only $k$ past solution and $k$ past function values at each step (the same as nonlinear methods (2.1)). Evaluation of the LIMM numerical solution at each step (2.4) requires to only solve a linear system of equations, expressed in a computationally efficient form as:

$$\begin{aligned} (\mathbf{I} - h_n \, \mu_{-1} \, \mathbf{J}_n) \cdot z &= \sum_{i=0}^{k-1} (\mu_i/\mu_{-1} - \alpha_i) \; \mathbf{y}_{n-i} + h_n \sum_{i=0}^{k-1} (\nu_i/\mu_{-1} + \beta_i) \; \mathbf{f}_{n-i}, \\ \mathbf{y}_{n+1} &= z - \sum_{i=0}^{k-1} (\mu_i/\mu_{-1}) \, \mathbf{y}_{n-i} - h_n \sum_{i=0}^{k-1} (\nu_i/\mu_{-1}) \, \mathbf{f}_{n-i}. \end{aligned}$$

Rosenbrock-W methods [26, Section IV.7] seek to increase the computational efficiency of Rosenbrock methods by replacing the exact Jacobian $\mathbf{J}_n$ with an arbitrary matrix $\mathbf{A}_n$, such that the corresponding linear system at each step can be solved more easily. This concept is extended to linearly implicit multistep methods by considering LIMM-W schemes.

**3. Order Conditions.** Derivation of the order conditions amounts to equating the Taylor series coefficients of the numerical and exact solutions about the current time $t_n$, up to a specific predetermined order. This is the approach taken for building the classical order condition theory for linear multistep methods [25, Chapter III.2]. We discuss a direct Taylor series approach to obtain order conditions for LIMM-W schemes in Section 3.1. However, constructing Taylor series by repeatedly differentiating the numerical solution becomes increasingly difficult for high-order schemes, and we use a B-series approach [6,25] to derive LIMM order conditions in Section 3.2.

*Assumption* 3.1 (Sequence of time steps). The linearly implicit scheme (2.4) computes the solution of (1.1) using (possibly) non-uniform time grids $\{t_i\}_{0\le i\le N}$ over $[t_0, t_F]$ (where $t_N = t_F$), with step sizes $h_n = t_{n+1} - t_n$. The ratios of consecutive step sizes are uniformly bounded below and above:

$$\omega_n := \frac{h_n}{h_{n-1}}; \quad \omega_{\min} \le \omega_n \le \omega_{\max} \ \ \forall\, n, \quad \text{where} \quad 0 < \omega_{\min} \le 1 \le \omega_{\max} < \infty. \tag{3.1}$$

It is convenient to express the distances between the solution points used during one step of (2.4) as fractions of the current step size $h_n$:

$$t_{n-i} = t_n - c_i\, h_n, \quad i = -1, \dots, k, \quad \text{where :} \tag{3.2a}$$

$$c_{-1} = -1, \qquad c_0 = 0, \qquad c_i = \sum_{\ell=1}^{i} \prod_{j=0}^{\ell-1} \omega_{n-j}^{-1}, \quad i \ge 1. \tag{3.2b}$$

*Remark* 3.2 (Method coefficients notation). When operating on a non-uniform time grid, the fractions (3.2a) depend on the current step, $c_i(n)$. Moreover, all the coefficients of the method (2.4) depend on the current step, $\alpha_i(n)$, $\beta_i(n)$, $\mu_i(n)$, $\nu_i(n)$, $i \in [-1, k-1]$. In order to simplify the notation, in the remainder of the paper we will leave out the explicit dependency on the step, unless it is needed for clarity, and will explicitly call out when we are considering fixed stepsize methods (as in the latter half of Section 6).

**3.1. Order conditions for LIMM-W methods.** The order conditions for LIMM-W methods (2.4) can be obtained by the standard Taylor series approach, leading to the following result.

THEOREM 3.3 (LIMM-W order conditions). *The* LIMM-W *method* (2.4) *has order of consistency p if and only if its coefficients satisfy:*

$$\sum_{i=-1}^{k-1} \alpha_i = 0, \tag{3.3a}$$

$$\sum_{i=-1}^{k-1} \mu_i = 0, \tag{3.3b}$$

$$\sum_{i=-1}^{k-1} \alpha_i\, c_i^{\ell} + \ell \sum_{i=0}^{k-1} \beta_i\, c_i^{\ell-1} = 0, \qquad \ell = 1, \dots, p, \tag{3.3c}$$

$$\sum_{i=-1}^{k-1} \mu_i\, c_i^{\ell-1} - (\ell-1) \sum_{i=0}^{k-1} \nu_i\, c_i^{\ell-2} = 0, \qquad \ell = 2, \dots, p. \tag{3.3d}$$

*Proof.* We show that the local truncation error is of order $p+1$. To this end we apply the method (2.4) starting with exact past solution values:

$$\begin{aligned}\mathbf{y}_{n+1} + \sum_{i=0}^{k-1} \alpha_i\, \mathbf{y}(t_{n-i}) = h_n \sum_{i=0}^{k-1} \beta_i\, \mathbf{y}'(t_{n-i}) + h_n\, \mathbf{A}_n \sum_{i=0}^{k-1} \mu_i\, \mathbf{y}(t_{n-i}) \\ + h_n\, \mu_{-1}\, \mathbf{A}_n\, \mathbf{y}_{n+1} + h_n\, \mathbf{A}_n \sum_{i=0}^{k-1} \nu_i\, h_n\, \mathbf{f}(t_{n-i}, \mathbf{y}(t_{n-i})),\end{aligned} \tag{3.4}$$

and show that $\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) \sim \mathcal{O}(h_n^{p+1})$. We insert the exact solution in (3.4):

$$\begin{aligned}\sum_{i=-1}^{k-1} \alpha_i\, \mathbf{y}(t_{n-i}) = h_n \sum_{i=0}^{k-1} \beta_i\, \mathbf{y}'(t_{n-i}) + h_n\, \mathbf{A}_n \sum_{i=-1}^{k-1} \mu_i\, \mathbf{y}(t_{n-i}) \\ + h_n^2\, \mathbf{A}_n \sum_{i=0}^{k-1} \nu_i\, \mathbf{y}'(t_{n-i}) + r_n,\end{aligned} \tag{3.5}$$

where $r_n$ is the local residual, and expand in Taylor series about the current time $t_n$

$$\begin{aligned}&\sum_{\ell\geq 0} \sum_{i=-1}^{k-1} \alpha_i\, \frac{(-c_i)^\ell\, h_n^\ell}{\ell!} \mathbf{y}^{(\ell)}(t_n) = \sum_{\ell\geq 1} \sum_{i=0}^{k-1} \beta_i\, \frac{(-c_i)^{\ell-1}\, h_n^\ell}{(\ell-1)!} \mathbf{y}^{(\ell)}(t_n) \\ &+ \mathbf{A}_n \sum_{\ell\geq 1} \sum_{i=-1}^{k-1} \mu_i\, \frac{(-c_i)^{\ell-1}\, h_n^\ell}{(\ell-1)!} \mathbf{y}^{(\ell-1)}(t_n) + \mathbf{A}_n \sum_{\ell\geq 2} \sum_{i=0}^{k-1} \nu_i\, \frac{(-c_i)^{\ell-2}\, h_n^\ell}{(\ell-2)!} \mathbf{y}^{(\ell-1)}(t_n) + r_n.\end{aligned}$$

Equating powers of $h_n$ on both sides of the equality up to power $p$ yields (3.3), and $r_n \sim \mathcal{O}(h_n^{p+1})$. Subtracting (3.5) from (3.10) leads to the local truncation error:

$$\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) = -\left(\mathbf{I} - h_n\, \mathbf{A}_n\right)^{-1}\, r_n, \tag{3.6}$$

and therefore $\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) \sim \mathcal{O}(h_n^{p+1})$ in the asymptotic case $h_n \to 0$ [25, Chapter III.2]. Consequently the method has order $p$. ∎

*Remark* 3.4 (Stiff case). From (3.5) we see that the residual has the form $r_n = q_n + h_n \mathbf{A}_n\, s_n$, where imposing (3.3a), (3.3c) leads to $q_n \sim \mathcal{O}(h_n^{p+1})$, and imposing (3.3b), (3.3d) leads to $s_n \sim \mathcal{O}(h_n^p)$. From (3.6) the local truncation error is

$$\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) = -\left(\mathbf{I} - h_n\, \mathbf{A}_n\right)^{-1}\, (q_n + s_n) + s_n. \tag{3.7}$$

In the asymptotic case where $h_n\, \mathbf{A}_n \to 0$ the $s_n$ components from the two terms cancel, leaving $\mathbf{y}_{n+1} - \mathbf{y}(t_{n+1}) = q_n + \mathcal{O}(h_n)\, s_n$, which recovers Theorem 3.1. Consider now the stiff case where $h_n \to 0$ but $\|h_n\, \mathbf{A}_n\| \not\to 0$, and the above cancellation does not happen. Assume that the matrix $\mathbf{A}_n$ in (2.4) has simple eigenvalues with non-positive real parts, which implies that $\|(\mathbf{I} - h_n \mathbf{A}_n)^{-1}\| \leq C < \infty$ for all step sizes $h_n > 0$. The local error (3.7) is dominated by $s_n \sim \mathcal{O}(h_n^p)$. A simple way to recover the full order is to impose conditions (3.3d) up to order $p+1$, which gives $s_n \sim \mathcal{O}(h_n^{p+1})$.

*Remark* 3.5 (Connection with traditional LMM). Equations (3.3a) and (3.3c) represent the order conditions for a traditional linear multistep method [25, Chapter III.2]. Consequently, the coefficients $\{\alpha_i, \beta_i\}$ correspond to an order $p$ explicit $k$-step method. The coefficients $\mu_i$ and $\nu_i$ are selected such that they only contribute $\mathcal{O}(h_n^{p+1})$ to the local truncation error.

Equation (3.3) pose $2p+1$ constraints for the $4k+1$ free coefficients of the method. Simpler order conditions are possible when $\mathbf{A}_n$ is the exact Jacobian $\mathbf{f}_\mathbf{y}(t_n, \mathbf{y}_n)$, or a well specified approximation of it. In this case the direct Taylor series approach becomes difficult to handle, and we employ the Butcher series machinery.

In practice we may consider only linearly implicit schemes (2.4) with $\nu_i = 0$, $i = 0, \ldots, k-1$, since the remaining $3k+1$ free coefficients offer sufficient degrees of freedom for good method design. Sections 6, 7, and 8 describe construction and testing of methods that make this simplification, and Section 5 considers convergence.

*Remark* 3.6 (Coefficients for variable steps). The order conditions (3.3) are linear in the unknown method coefficients, with the system matrix depending on stepsize fractions $c_j$ (or analogously, on the stepsize ratios $\omega_j$). From (3.2b) we have that

$$\frac{1-\omega_{\max}^{-i}}{\omega_{\max}-1} \leq c_i \leq \frac{\omega_{\min}^{-i}-1}{1-\omega_{\min}} \quad \text{and} \quad c_{i+1} \geq c_i + \omega_{\max}^{-(i+1)} > c_i. \tag{3.8}$$

Consider methods with $p \leq k$. Using (3.8) we conclude that, due to the Vandermonde structure, the system (3.3) has a solution where the method coefficients $\alpha_j$, $\mu_j$, depend continuously on $\beta_j$, $\nu_j$, the stepsize fractions $c_i$ (and analogously, on the stepsize ratios $\omega_i$), and on $k-p$ free parameters [29].

**3.2. Order conditions for LIMM methods with exact Jacobian.** We employ the Butcher series (B-series) formalism to derive order conditions for the LIMM methods (2.4). B-series [25] offer a representation of Taylor series expansions of numerical and exact solutions as expansions over a set of elementary differentials, represented graphically using rooted-trees.

We consider the family of trees $\mathcal{T}_1 = \mathcal{T} \cup \{\emptyset\} \cup \{\tau_\circ\}$, where $\mathcal{T}$ is the set of Butcher T-trees [25], $\emptyset$ denotes the empty tree, $\tau \in \mathcal{T}$ denotes the tree with a single node, and $\tau_\circ$ denotes a special tree with a single fat node (a color different than that of the nodes of $\mathcal{T}$). If $\mathsf{t}_1, \ldots, \mathsf{t}_L \in \mathcal{T}$ then $[\mathsf{t}_1 \ldots \mathsf{t}_L] \in \mathcal{T}$ denotes the tree obtained by joining all $L$ subtrees to a single root. Each tree in $\mathcal{T}$ corresponds to a traditional elementary differential [25]. In addition, $\mathcal{F}(\emptyset)(\mathbf{y}) = y$ and $\mathcal{F}(\tau_\circ)(\mathbf{y}) = \mathbf{J}_n\,\mathbf{y}$. The latter elementary differential appears in the numerical solution, but not in the exact solution.

A B-series expansion over the set of rooted-trees $\mathcal{T}_1$ is [25]:

$$B(\mathsf{a}, \mathbf{y}) = \sum_{\mathsf{t} \in \mathcal{T}_1} \mathsf{a}(\mathsf{t}) \cdot \frac{h^{|\mathsf{t}|}}{\sigma(\mathsf{t})} \mathcal{F}(\mathsf{t})(\mathbf{y}) \tag{3.9}$$

where $\mathsf{a} : \mathcal{T}_1 \mapsto \mathbb{R}$ is a function mapping trees to real values (the coefficients of the B-series), $|\mathsf{t}|$ the order (or number of nodes) of the tree, $\sigma(\mathsf{t})$ is the symmetry of the tree (the number of equivalent rearrangements of $\mathsf{t}$) [25], and $\mathcal{F}(\mathsf{t})(\mathbf{y})$ is the elementary differential corresponding to a tree $\mathsf{t}$, evaluated at $\mathbf{y}$. B-series are the gold standard approach to construct the order conditions for one-step methods such as Runge-Kutta [6, 25, 36, 41], Rosenbrock [50], and exponential [31–33, 49] schemes.

To study the LIMM local truncation error we consider the method (2.4) initialized with the exact solution evaluated at a series of $k$ previous time-steps, $\{t_n, \ldots, t_{n-k+1}\}$:

$$\begin{aligned} \mathbf{y}_{n+1} + \sum_{i=0}^{k-1} \alpha_i\, \mathbf{y}(t_{n-i}) = h_n \sum_{i=0}^{k-1} \beta_i\, \mathbf{y}'(t_{n-i}) + h_n\, \mathbf{f}_\mathbf{y}\big(t_n, \mathbf{y}(t_n)\big) \sum_{i=0}^{k-1} \mu_i\, \mathbf{y}(t_{n-i}) \\ + h_n\, \mu_{-1}\, \mathbf{f}_\mathbf{y}\big(t_n, \mathbf{y}(t_n)\big)\, \mathbf{y}_{n+1} + h_n\, \mathbf{f}_\mathbf{y}\big(t_n, \mathbf{y}(t_n)\big) \sum_{i=0}^{k-1} \nu_i\, \big(h_n\, \mathbf{y}'(t_{n-i})\big)\,. \end{aligned} \tag{3.10}$$

The time-shifted exact solution (1.1) has the following B-series expansion over $\mathcal{T}_1$:

$$\mathbf{y}(t_n + c\,h_n) \sim B(a_{(c)}, \mathbf{y}(t_n)), \qquad a_{(c)}(\mathfrak{t}) = \begin{cases} 0, & \mathfrak{t} = \tau_\circ, \\ 1, & \mathfrak{t} = \emptyset, \\ \frac{c^{|\mathfrak{t}|}}{\gamma(\mathfrak{t})}, & |\mathfrak{t}| \geq 1, \end{cases} \tag{3.11a}$$

where $\mathfrak{t} \in \mathcal{T}$ is a T-tree, and $\gamma(\mathfrak{t})$ is defined as the product of order of the tree $\mathfrak{t}$ and of all its subtrees [25]. Similarly, the time-shifted exact solution derivative of (1.1) has the following B-series expansion over $\mathcal{T}_1$:

$$h_n\,\mathbf{y}'(t_n + c\,h_n) \sim B(\mathrm{D}a_{(c)}, \mathbf{y}(t_n)), \qquad (\mathrm{D}a_{(c)})(\mathfrak{t}) = \begin{cases} 0, & \mathfrak{t} = \tau_\circ, \\ 0, & \mathfrak{t} = \emptyset, \\ \frac{|\mathfrak{t}|\,c^{|\mathfrak{t}|-1}}{\gamma(\mathfrak{t})}, & |\mathfrak{t}| \geq 1, \end{cases} \tag{3.11b}$$

where $\tau \in \mathcal{T}$ is the T-tree with a single node, and $[\mathfrak{u}_1, \ldots, \mathfrak{u}_L]$ denotes the tree where the root has $L$ children, each the root of a subtree $\mathfrak{u}_i$.

The Jacobian matrix times a B-series is another B-series over $\mathcal{T}_1$ [5]:

$$\begin{gathered} h_n\,\mathbf{f}_{\mathbf{y}}(t_n, \mathbf{y}(t_n)) \cdot \mathrm{B}(a, \mathbf{y}(t_n)) \sim \mathrm{B}\big(\mathrm{J}a, \mathbf{y}(t_n)\big), \\ (\mathrm{J}a)(\mathfrak{t}) = \begin{cases} a(\emptyset), & \mathfrak{t} = \tau_\circ, \\ 0, & \mathfrak{t} = \emptyset,\ \mathfrak{t} = \tau, \\ a(\mathfrak{u}), & \text{for } \mathfrak{t} = [\mathfrak{u}],\ \mathfrak{u} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \end{gathered} \tag{3.11c}$$

The numerical solution of (3.10) is a B-series over $\mathcal{T}_1$:

$$\mathbf{y}_{n+1} \sim B(\theta, \mathbf{y}(t_n)). \tag{3.12}$$

Next, in (3.10) replace each quantity by the corresponding B-series. Use the exact solutions (3.11a) and their derivatives (3.11b) at the current and past times $t_{n-k}, \ldots, t_n$, and the numerical solution (3.12) at $t_{n+1}$, to obtain:

$$\begin{aligned} \theta = \mu_{-1}\,(\mathrm{J}\theta) - \sum_{i=0}^{k-1} \alpha_i\,a_{(-c_i)} + \sum_{i=0}^{k-1} \beta_i\,(\mathrm{D}a_{(-c_i)}) \\ + \sum_{i=0}^{k-1} \mu_i\,(\mathrm{J}a_{(-c_i)}) + \sum_{i=0}^{k-1} \nu_i\,(\mathrm{J}(\mathrm{D}a_{(-c_i)})). \end{aligned}$$

This leads to the following recursive definition of $\theta$ over the set $\mathcal{T}_1$ of trees:

$$\theta(\mathfrak{t}) = \begin{cases} -\sum_{i=0}^{k-1} \alpha_i, & \mathfrak{t} = \emptyset, \\ \sum_{i=0}^{k-1} \mu_i + \mu_{-1}\,(-\sum_{i=0}^{k-1} \alpha_i), & \mathfrak{t} = \tau_\circ, \\ -\sum_{i=0}^{k-1} \alpha_i\,(-c_i) + \sum_{i=0}^{k-1} \beta_i, & \mathfrak{t} = \tau, \\ \mu_{-1}\,\theta(\mathfrak{u}_1) - \frac{1}{\gamma(\mathfrak{t})} \sum_{i=0}^{k-1} \alpha_i\,(-c_i)^{|\mathfrak{t}|} & \\ \quad + \frac{|\mathfrak{t}|}{\gamma(\mathfrak{t})} \sum_{i=0}^{k-1} (\beta_i + \mu_i)\,(-c_i)^{|\mathfrak{t}|-1} & \\ \quad + \frac{|\mathfrak{t}|\,(|\mathfrak{t}|-1)}{\gamma(\mathfrak{t})} \sum_{i=0}^{k-1} \nu_i\,(-c_i)^{|\mathfrak{t}|-2}, & \mathfrak{t} = [\mathfrak{u}_1],\ |\mathfrak{u}_1| \geq 1, \\ -\frac{1}{\gamma(\mathfrak{t})} \sum_{i=0}^{k-1} \alpha_i\,(-c_i)^{|\mathfrak{t}|} + \frac{|\mathfrak{t}|}{\gamma(\mathfrak{t})} \sum_{i=0}^{k-1} \beta_i\,(-c_i)^{|\mathfrak{t}|-1}, & \mathfrak{t} = [\mathfrak{u}_1, \ldots, \mathfrak{u}_L],\ L \geq 2. \end{cases} \tag{3.13}$$

To obtain the order conditions we equate the B-series coefficients of the numerical method (3.13) with those of the exact solution up to order $p$:

$$\theta(\mathfrak{t}) = \frac{1}{\gamma(\mathfrak{t})}, \quad \forall\, \mathfrak{t} \in \mathcal{T}_1 \text{ with } |\mathfrak{t}| \leq p.$$

For trees with $|\mathfrak{t}| \leq 1$ we have:

$$\begin{aligned}
\mathfrak{t} = \emptyset: \quad & -\sum_{i=0}^{k-1} \alpha_i = 1 \quad \Leftrightarrow \quad \sum_{i=-1}^{k-1} \alpha_i = 0 \quad (\text{using } \alpha_{-1} = 1), \\
\mathfrak{t} = \tau_\circ: \quad & \sum_{i=-1}^{k} \mu_i = 0 \quad (\text{using condition for } \mathfrak{t} = \emptyset), \\
\mathfrak{t} = \tau: \quad & -\sum_{i=0}^{k-1} \alpha_i \, (-c_i) + \sum_{i=0}^{k-1} \beta_i = 1.
\end{aligned} \tag{3.14}$$

For trees with $|\mathfrak{t}| \geq 2$ with a multiply branched root ($\mathfrak{t} = [\mathfrak{u}_1, \ldots, \mathfrak{u}_L]$, $L \geq 2$) the order condition reads:

$$-\sum_{i=-1}^{k-1} \alpha_i \, (-c_i)^{|\mathfrak{t}|} + |\mathfrak{t}| \sum_{i=0}^{k-1} \beta_i \, (-c_i)^{|\mathfrak{t}|-1} = 0, \tag{3.15}$$

where we used the relations $\alpha_{-1} = 1$ and $c_{-1} = -1$. For trees with $|\mathfrak{t}| \geq 2$ with a singly branched root ($\mathfrak{t} = [\mathfrak{u}_1]$, $|\mathfrak{u}_1| \geq 1$) the order condition reads:

$$-\sum_{i=-1}^{k-1} \alpha_i \, (-c_i)^{|\mathfrak{t}|} + |\mathfrak{t}| \sum_{i=-1}^{k-1} \Big( (\beta_i + \mu_i) \, (-c_i)^{|\mathfrak{t}|-1} + (|\mathfrak{t}| - 1) \, \nu_i \, (-c_i)^{|\mathfrak{t}|-2} \Big) = 0, \tag{3.16}$$

where we formally set $\beta_{-1} = 0$, and use the lower order condition $\theta(\mathfrak{u}_1) = 1/\gamma(\mathfrak{u}_1) = |\mathfrak{t}|/\gamma(\mathfrak{t})$. We make the following observations:

- T-trees of order two, $|\mathfrak{t}| = 2$, have a singly branched root and only condition (3.16) is applied.
- Equations (3.15) and (3.16) depend only on the order of the tree $|\mathfrak{t}|$, but not on the tree topology. For higher orders $|\mathfrak{t}| \geq 3$ there are T-trees with both singly-branched and multiply-branched roots, therefore both order conditions (3.15) and (3.16) apply. The only way both equations (3.15) and (3.16) are satisfied is to require that:

$$\sum_{i=-1}^{k-1} \Big( \mu_i \, (-c_i)^{|\mathfrak{t}|-1} + (|\mathfrak{t}| - 1) \, \nu_i \, (-c_i)^{|\mathfrak{t}|-2} \Big) = 0 \quad \text{for } |\mathfrak{t}| \geq 3.$$

Using $c_{-1} = -1$ (3.2a), $\alpha_{-1} = 1$, and $\beta_{-1} = 0$, we have the following result.

THEOREM 3.7 (LIMM order conditions). *The* LIMM *method* (2.4) *has order of*

*consistency $p \geq 1$ if and only if the coefficients satisfy:*

$$\sum_{i=-1}^{k-1} \alpha_i = 0, \quad \text{(consistency)} \tag{3.17a}$$

$$\sum_{i=-1}^{k} \mu_i = 0, \quad \text{(consistency)} \tag{3.17b}$$

$$\sum_{i=-1}^{k-1} \alpha_i \, c_i + \sum_{i=0}^{k-1} \beta_i = 0, \quad \text{(order one)} \tag{3.17c}$$

$$\sum_{i=-1}^{k-1} \alpha_i \, c_i^2 + 2 \sum_{i=-1}^{k-1} \left( (\beta_i + \mu_i) \, c_i - \nu_i \right) = 0, \quad \text{(order two)} \tag{3.17d}$$

$$\sum_{i=-1}^{k-1} \alpha_i \, c_i^{\ell} + \ell \sum_{i=0}^{k-1} \beta_i \, c_i^{\ell-1} = 0, \quad (\text{order } \ell = 3, \ldots, p) \tag{3.17e}$$

$$\sum_{i=-1}^{k-1} \mu_i \, c_i^{\ell-1} - (\ell - 1) \sum_{i=0}^{k-1} \nu_i \, c_i^{\ell-2} = 0, \quad (\text{order } \ell = 3, \ldots, p). \tag{3.17f}$$

*Remark* 3.8. A comparison of LIMM order conditions (3.17) with LIMM-W order conditions (3.3) reveals that they are the same, except for the second order condition. Specifically, LIMM order condition (3.17d) is the sum of LIMM-W conditions (3.3c) and (3.3d) for $\ell = 2$. There are $2p$ LIMM conditions for order $p$, compared to $2p + 1$ LIMM-W conditions.

**4. Linear Stability Analysis.** To study linear stability we apply the LIMM method (2.4) to the Dahlquist test equation

$$\mathbf{y}' = \lambda \, \mathbf{y}, \quad \mathbf{y}(t_0) = 1,$$

to obtain the numerical solution

$$\sum_{i=-1}^{k-1} \left( \alpha_i - z \, (\beta_i + \mu_i) - z^2 \, \nu_i \right) \mathbf{y}_{n-i} = 0, \quad \text{where } z = h \, \lambda, \ \beta_{-1} = \nu_{-1} = 0. \tag{4.1}$$

To solve this homogeneous linear difference equation we substitute $\zeta^{k-i-1}$ for $\mathbf{y}_{n-i}$. This yields the relation:

$$\begin{aligned} &\varrho(\zeta) - z \, \sigma(\zeta) - z^2 \, \upsilon(\zeta) = 0, \qquad \text{where:} \\ &\varrho(\zeta) \coloneqq \sum_{i=-1}^{k-1} \alpha_i \zeta^{k-i-1}, \quad \sigma(\zeta) \coloneqq \sum_{i=-1}^{k-1} (\beta_i + \mu_i) \, \zeta^{k-i-1}, \quad \upsilon(\zeta) \coloneqq \sum_{i=0}^{k-1} \nu_i \, \zeta^{k-i-1}. \end{aligned} \tag{4.2}$$

The linearly implicit multistep method (2.4) is zero-stable if all the roots of the polynomial $\varrho(\zeta)$ lie on or inside the unit circle, with only simple roots on the unit circle [25, Section III, Def. 3.2].

The stability region of the linearly implicit multistep method (2.4) is a subset of the complex plane defined as [26, Section V, Def. 1.1]:

$$\mathcal{S} \coloneqq \left\{ z \in \mathbf{C} : \begin{array}{l} \text{all roots } \zeta_j(z) \text{ of eqn. (4.2) satisfy } |\zeta_j(z)| \leq 1, \\ \text{and multiple roots satisfy } |\zeta_j(z)| < 1 \end{array} \right\}. \tag{4.3}$$

In order to visualize this stability region we make use of the reverse map

$$z(\zeta, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \begin{cases} \frac{\varrho(\zeta)}{\sigma(\zeta)}, & \nu_0 = \cdots = \nu_{k-1} = 0, \\ \frac{-\sigma(\zeta) \pm \sqrt{\sigma(\zeta)^2 + 4\,\upsilon(\zeta)\,\varrho(\zeta)}}{2\,\upsilon(\zeta)}, & \text{otherwise}, \end{cases} \tag{4.4}$$

and evaluate it for $\zeta = e^{i\theta}$, $0 \leq \theta \leq 2\pi$, to produce the root locus curve. Figure 6.1 shows stability region plots for a set of methods constructed via the optimization approach described in Section 6.

*Remark* 4.1 (Variable step sizes). For variable step size the method coefficients, and therefore the polynomials (4.2), depend on the ratios $\omega_j$ of consecutive step sizes (3.2b) for the last $k$ steps: $\varrho(\omega_{i-k:i-1}, \zeta)$, $\sigma(\omega_{i-k:i-1}, \zeta)$, $\upsilon(\omega_{i-k:i-1}, \zeta)$. Consequently, the stability region (4.3) and the inverse map (4.4) also depend on the step size ratios.

**4.1. Stability matrix.** When all $\nu_i = 0$ the stability region (4.3) is that of an implicit LMM method, and equals the region where the following matrix has all eigenvalues inside the unit disk, with only simple eigenvalues on the unit circle [26, Section V]:

$$\mathbf{M}(\omega_{i-k:i-1}, z) \coloneqq \begin{bmatrix} \frac{-\alpha_{0:k-2}^T + z\,(\beta_{0:k-2} + \mu_{0:k-2})^T}{1 - z\mu_{-1}} & \frac{-\alpha_{k-1} + z\,(\beta_{k-1} + \mu_{k-1})}{1 - z\mu_{-1}} \\ \mathbf{I}_{(k-1)\times(k-1)} & \mathbf{0}_{(k-1)\times 1} \end{bmatrix} \in \mathbb{R}^{k\times k}. \tag{4.5}$$

*Stability over the current step.* The solution stability over the current step requires the stability of the matrix (4.5), which imposes restrictions on the stepsize ratios $\omega_{\min} \leq \omega_i \leq \omega_{\max}$ for particular values of $z$. For $z = 0$, we recover the traditional LMM zero stability condition on $\alpha_{0:k-1}$ [25, Section III.5] [7, 9–11, 24]. For the LIMM(-W) methods constructed in Section 6, the $\alpha_{0:k-1}$ coefficients are chosen to be invariant with respect to $\omega_{i-k:i-1}$, and so always satisfy the zero stability condition. If we consider $z \to -\infty$ as in [11], for the 2-step LIMM method we have $\mathbf{M}_2(\omega_1, \infty) = \left[\begin{smallmatrix} -(\beta_0+\mu_0)/\mu_{-1} & -(\beta_1+\mu_1)/\mu_{-1} \\ 1 & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} (1+2\omega_1-3\omega_1^2)/(1+3\omega_1^2) & 0 \\ 1 & 0 \end{smallmatrix}\right]$, where the second equality comes from substituting the 2-step LIMM coefficients from Table **??**, and the cancelation of the $\beta_1$ and $\mu_1$ coefficients. To guarantee stability, we restrict the spectral radius $\rho\left(\mathbf{M}_2(\omega_1, \infty)\right) \leq 1$, which is satisfied for $\omega_1 \geq 1/3$. In fact, with the help of a computer algebra system, we find that this condition on $\omega_1$ holds for any real $z < 0$.

*Stability over multiple steps.* In case of fixed step size integration the error amplification over $n$ consecutive steps is governed by $\mathbf{M}(\mathbb{1}_k, z)^n$; this matrix is power bounded if $z$ is in the stability region of $\mathbf{M}(\mathbb{1}_k, z)$. In case of variable step size integration the error amplification factor over $n$ consecutive steps is governed by the product of stability matrices $\prod_{i=\ell+1}^{\ell+n} \mathbf{M}(\omega_{i-k:i-1}, z)$. The stability matrices $\mathbf{M}(\omega_{i-k:i-1}, z)$ at different time steps do not share the same eigenvectors, and in general the spectrum of the product cannot be inferred from the spectral radii of individual stability matrices (4.5). Zero-stability (boundedness of the matrix product for $z = 0$) of LMM on non-uniform grids was studied in [7, 9, 10, 19, 23]. Soderlind et al. [45] show that zero-stability of LMM is preserved on smoothly varying grids where $\omega_i = 1 + \mathcal{O}(h_{\max})$. More general stability results (boundedness of the matrix product for a range of $z$'s) are given in [11, 17, 34].

**5. Convergence.** Here we consider LIMM-W methods (2.4) with $\nu_i = 0$, $i = 0, \ldots, k-1$, satisfying (3.3) up to order $p$. We write the LIMM-W method (2.4) as an

IMEX LMM scheme applied to a linear-nonlinear partitioning of the system:

$$\sum_{i=-1}^{k-1} \alpha_i \, \mathbf{y}_{n-i} = h_n \sum_{i=0}^{k-1} \beta_i \, (\mathbf{f}_{n-i} - \mathbf{J}_n \, \mathbf{y}_{n-i}) + h_n \sum_{i=-1}^{k-1} (\mu_i + \beta_i) \, \mathbf{J}_n \, \mathbf{y}_{n-i}. \tag{5.1}$$

One sees that the order conditions for the IMEX LMM scheme (5.1) are equivalent to the LIMM-W order conditions (3.3).

For brevity the following discussion considers the autonomous case, however it can immediately be extended to the non-autonomous case.

*Assumption* 5.1 (Linear stiffness). We are concerned with solving stiff systems (1.1) where the Lipschitz constant of the right hand side function $\mathbf{f}(\cdot)$ is large. Assume that, for any $\tau \in [t_0, t_F]$, there is an interval $[\tau - \varepsilon, \tau + \varepsilon]$ such that the right hand side function of the system (1.1) can be locally decomposed into a linear part and a nonlinear remainder:

$$\mathbf{f}(\mathbf{y}(t)) = \mathbf{J}_\tau \, \mathbf{y}(t) + \mathbf{r}_\tau(\mathbf{y}(t)) \qquad \forall t \in (\tau - \varepsilon, \tau + \varepsilon), \tag{5.2}$$

where the matrix $\mathbf{J}_\tau$ is diagonalizable, $\mathbf{J}_\tau \, \mathbf{X}_\tau = \mathbf{X}_\tau \, \mathbf{\Lambda}_\tau$ with nonsingular $\mathbf{X}_\tau$, and all its eigenvalues have negative real parts. We choose $\mathbf{J}_\tau$ to capture all the stiffness of the system in a vicinity of the exact trajectory, such that the remaining nonlinear parts $\mathbf{r}_\tau(\mathbf{y})$ are non-stiff. Specifically, we assume that the residual $\mathbf{r}_\tau(\mathbf{y})$ has a smooth Jacobian in a vicinity of the solution:

$$\left\| \frac{d\mathbf{r}_\tau(\mathbf{y})}{d\mathbf{y}} \right\| \le \mathrm{L}_\tau \quad \Rightarrow \quad \|\mathbf{r}_\tau(\mathbf{y}) - \mathbf{r}_\tau(\mathbf{z})\| \le \mathrm{L}_\tau \, \|\mathbf{y} - \mathbf{z}\|, \tag{5.3}$$

and that the residual Lipschitz constant $\mathrm{L}_\tau$ is of moderate size, and much smaller than the Lipschitz constant of the right hand side function $\mathbf{f}(\cdot)$.

Equations (5.2), (5.3) mean that the stiffness is due to linear dynamics only, and that the stiff directions of the system evolution do not change too rapidly along a trajectory [42]. Due to compactness, there is a finite number of subintervals $(\tau_i - \varepsilon_i, \tau_i + \varepsilon_i)$, $i = 1, \ldots, M$ with $\tau_i > \tau_j$ for $j > i$, that cover $[t_0, t_F]$. The integration interval is covered by a finite number of disjoint closed subintervals

$$\begin{aligned} &[t_0, t_F] = \cup_{i=1}^{M} [\theta_{i-1}, \theta_i], \quad \theta_0 = t_0, \quad \theta_M = t_F, \\ &\max\{\tau_i - \varepsilon_i, \tau_{i-1}\} < \theta_{i-1} < \min\{\tau_{i-1} + \varepsilon_{i-1}, \tau_i\}, \quad 2 \le i \le M, \\ &[\theta_{i-1}, \theta_i] \subset (\tau_i - \varepsilon_i, \tau_i + \varepsilon_i) \quad \forall \, i. \end{aligned} \tag{5.4}$$

The integration converges over $[t_0, t_F]$ iff it converges over each $[\theta_{i-1}, \theta_i]$, $i = 1, \ldots, M$. On each subinterval $[\theta_{i-1}, \theta_i]$ the system admits a decomposition (5.2) with matrix $\mathbf{J}_{\tau_i}$. Without loss of generality, we carry out the convergence analysis on a single subinterval $[\theta_{i-1}, \theta_i]$ and a single decomposition (5.2); we drop the subscripts $\tau_i$, and denote by $\mathbf{J}_*$ the corresponding matrix with eigenvalues with negative real parts.

We note that the Jacobian of the nonlinear remainder is $\mathbf{r}_\mathbf{y}(\mathbf{y}) = \mathbf{J}(\mathbf{y}) - \mathbf{J}_*$, and that $\|\mathbf{r}_\mathbf{y}\| \le \mathrm{L}_*$ in a vicinity of the solution from (5.3).

We make the following assumptions on the method.

*Assumption* 5.2 (Method properties). The LIMM-W scheme (5.1) applied with fixed step sizes is $A(\alpha)$ stable, for $0 \le \alpha \le \pi/2$. The fixed step stability matrix (4.5) $\mathbf{M}(\mathbb{1}, z)$ is strictly stable at the origin and at infinity [34]. Therefore $\mathbf{M}(\mathbb{1}, 0)$ has a single eigenvalue of modulus one, and has a spectral radius smaller than one at infinity, $\rho_\infty \coloneqq \rho(\mathbf{M}(\mathbb{1}, \infty)) < 1$.

The integration of (1.1) sequence of step sizes is carried out using sequences of step sizes $\{h_i\}$ with $\sum_i h_i = t_F - t_0$, that follow Assumption 3.1. We make the following additional assumptions on the sequence of time steps.

*Assumption* 5.3 (Step ratio bounds). Assumption 5.2 implies that $\mu_{-1} > 0$ for constant step sizes. The step size bounds $\omega_{\min}$, $\omega_{\max}$ are such that $\mu_{-1}$ remains positive for all possible step sequences,

$$\min_{\omega_{\min} \le \omega_{i-k}, \ldots, \omega_{i-1} \le \omega_{\max}} \mu_{-1}(\omega_{i-k:i-1}) > 0.$$

*Assumption* 5.4 (Linear stability over each step). For each step $z_i = h_i \lambda \in \mathcal{S}$ for any eigenvalue $\lambda$ of $\mathbf{J}_*$.

*Assumption* 5.5 (Linear stability across step size sequences). Consider applying the method (5.1) to solve the linear stiff ODE $\mathbf{y}' = \mathbf{J}_* \mathbf{y}$ with the sequence of steps $\{h_i\}$. The sequence of steps is stable, i.e., there exists $C_{\mathbf{M}} > 0$ such that:

$$\| \textstyle\prod_{i=\ell_1}^{\ell_2} \mathbf{M}(\omega_{i-k:i-1}, h_i \mathbf{J}_*) \| \le C_{\mathbf{M}}, \quad \forall h_i : \textstyle\sum_{i=\ell_1}^{\ell_2} h_i \le t_F - t_0, \ \forall \ell_2 \ge \ell_1. \tag{5.4}$$

*Remark* 5.6 (Sufficient conditions for stability). A sufficient condition for stability (5.4) is that each matrix has a norm smaller than one, $\|\mathbf{M}(\omega'_{1:k}, z)\| \le 1$ for all $\omega_{\min} \le \omega'_i \le \omega_{\max}$ and $z = h\lambda$, $h > 0$, and $\lambda$ any eigenvalue of $\mathbf{J}_*$. Butcher and Heard [7] take this approach and find suitable matrix norms for analyzing the stability of BDF2 and BDF3 methods. Taking this approach for the 2-step LIMM-W method constructed in Section 6 using the variable coefficients from Table **??**, we find that the matrix norm $\|\mathbf{M}_2(\omega'_1, z)\|_{\mathbf{R}} = \left\|\mathbf{R}^{-1}\mathbf{M}_2(\omega'_1, z)\mathbf{R}\right\|_1 \le 1$ for all real $z < 0$ and for all $\omega'_1 \in [\omega_{\min}, \omega_{\max}]$, where

$$\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 1 & \frac{120209304}{13204873} \end{bmatrix}, \quad \omega_{\min} = \tfrac{13204873}{133414177} \approx 0.1, \quad \omega_{\max} = \tfrac{10262629 + \sqrt{484910319930169}}{20525258} \approx 1.57.$$

*Remark* 5.7 (General conditions for stability). General sufficient conditions for linear stability are given by Ostermann et al. [34]. They consider the case where the sequence of time steps satisfy Assumptions 3.1, 5.3, 5.4, with $\omega_{\min} = \omega_{\max}^{-1}$ and $\omega_{\max} < \rho_\infty^{-1/2}$. If there are constants $\Delta, C > 0$ such that $\prod_{i=\ell_1}^{\ell_2} (1 + \Delta\,|\omega_j - 1|)^2 \le C$ for any $\ell_2 \ge \ell_1$, and Assumptions 5.1 and 5.2 hold, then the stability equation (5.4) is satisfied cf. [34, Theorem 2].

THEOREM 5.8 (Convergence). *Apply an order $p$* LIMM-W *scheme* (5.1) *to solve the system* (1.1)*, using step size sequences that satisfy Assumption 3.1. Denote the global numerical errors by $\Delta\mathbf{y}_n := \mathbf{y}_n - \mathbf{y}(t_n)$. If Assumptions 5.1, 5.2, 5.3, 5.4, 5.5 hold then the numerical solution converges with order $p$ to the exact solution for sequences of sufficiently small step sizes $h_j \le h_*$,*

$$\|\Delta\mathbf{y}_n\| = \mathcal{O}\left(h_{\max}^p\right) \quad \forall n, \quad h_{\max} = \max_{1 \le j \le n-1} h_j \le h_*,$$

*whenever the initial conditions are sufficiently accurate, $\|\Delta\mathbf{y}_n\| = \mathcal{O}\left(h_{\max}^p\right)$ for $i = 1, \ldots, k$. The upper bound $h_*$ is independent of the stiffness of the system.*

*Proof.* Using Remark 3.6 and the assumption on the step size ratio bounds, the method coefficients depend continuously on step size ratios and remain uniformly bounded: $|\alpha_i(\omega_{n-k:n-1})|$, $|\beta_i(\omega_{n-k:n-1})|$, $|\mu_i(\omega_{n-k:n-1})| \le const$ for all $\omega_{\min} \le \omega_j \le \omega_{\max}$. For notation brevity in the remaining part of the proof we omit the explicit dependency of method coefficients on step size ratios.

Since $\mu_{-1} > 0$ and all the eigenvalues of $\mathbf{J}_*$ have non-positive real parts the following matrix is uniformly bounded:

$$\mathbf{T}_n \coloneqq \mathbf{I}_{\mathrm{N}} - h_n\,\mu_{-1}\,\mathbf{J}_*, \quad \|\mathbf{T}_n^{-1}\| \le C_{\mathbf{T}}, \quad \forall\, h_n.$$

Using the linear-nonlinear splitting (5.2), and the fact that $(\mathbf{r}_{\mathbf{y}})_n = \mathbf{J}_n - \mathbf{J}_*$, the method (5.1) reads:

$$\begin{aligned}\mathbf{T}_n\,\mathbf{y}_{n+1} = &\sum_{i=-1}^{k-1} \left(-\alpha_i + (\beta_i + \mu_i)\,h_n \mathbf{J}_*\right)\,\mathbf{y}_{n-i} \\ &+ h_n \sum_{i=0}^{k-1} \beta_i\,\mathbf{r}_{n-i} + h_n \sum_{i=0}^{k-1} \mu_i\,(\mathbf{r}_{\mathbf{y}})_n\,\mathbf{y}_{n-i}.\end{aligned} \tag{5.5}$$

Replace $\mathbf{y}_{n-i}$ by the exact solution $\mathbf{y}(t_{n-i})$ into the method (5.5), and $\mathbf{f}_{n-i}$ by $\mathbf{f}(t_{n-i}, \mathbf{y}(t_{n-i}))$ (but keep the same $(\mathbf{r}_{\mathbf{y}})_n = \mathbf{J}_n - \mathbf{J}_*$ since the choice of $\mathbf{J}_n$ does not change the order of accuracy); subtracting the numerical solution (5.5) leads to the following recurrence of global errors:

$$\begin{aligned}\Delta\mathbf{y}_{n+1} &= h_n\,\boldsymbol{\delta}_n + h_n\,\boldsymbol{\zeta}_n + \boldsymbol{\theta}_n + \mathbf{T}_n^{-1}\,R_n, \\ \boldsymbol{\delta}_n &= \mathbf{T}_n^{-1} \sum_{i=0}^{k-1} \beta_i\,\Delta\mathbf{r}_n(\mathbf{y}_{n-i}), \qquad \boldsymbol{\zeta}_n = \mathbf{T}_n^{-1}\,(\mathbf{r}_{\mathbf{y}})_n \sum_{i=-1}^{k-1} \mu_i\,\Delta\mathbf{y}_{n-i}, \\ \boldsymbol{\theta}_n &= \mathbf{T}_n^{-1} \sum_{i=0}^{k-1} \left(-\alpha_i + (\beta_i + \mu_i)\,h_n \mathbf{J}_*\right)\,\Delta\mathbf{y}_{n-i},\end{aligned} \tag{5.6}$$

where the local truncation error is $R_n \sim \mathcal{O}(h_n^{p+1})$.

Let $\mathbb{Y}_n = [\mathbf{y}_n^T \ldots \mathbf{y}_{n-k+1}^T]^T$ be the solution history vector. Taking norms in (5.6) we have that:

$$\begin{aligned}&\|\boldsymbol{\delta}_n\| \le C_{\mathbf{T}}\,\mathrm{L}_* \sum_{i=0}^{k-1} |\beta_i| \max_{i=0,\ldots,k-1} \|\Delta\mathbf{y}_{n-i}\| \le C_1\,\|\Delta\mathbb{Y}_n\|, \\ &\|\mathbf{T}_n^{-1}\,h_n\mathbf{J}_*\| = \|(\mu_{-1})^{-1}\,\left(\mathbf{T}_n^{-1} - \mathbf{I}\right)\| \le \mu_{-1}^{-1}\,(1 + C_{\mathbf{T}}), \\ &\|\boldsymbol{\theta}_n\| \le \left(C_{\mathbf{T}} \sum_{i=0}^{k-1} |\alpha_i| + \frac{1 + C_{\mathbf{T}}}{\mu_{-1}} \sum_{i=0}^{k-1} |\beta_i + \mu_i|\right) \max_{i=0,\ldots,k-1} \|\Delta\mathbf{y}_{n-i}\| \le C_2\,\|\Delta\mathbb{Y}_n\|.\end{aligned}$$

Consider the step size bound $h_*$ that is independent of the stiffness of the system:

$$h_* < \left(C_{\mathbf{T}}\,\mathrm{L}\,\mu_{-1}\right)^{-1} \quad \Rightarrow \quad h_*\,C_{\mathbf{T}}\,\mathrm{L}\,\mu_{-1} = C_3 < 1.$$

For any $h_n \le h_*$ the error in the solution is bounded by:

$$\begin{aligned}\|\Delta\mathbf{y}_{n+1}\| &\le \frac{h_*\,C_1 + C_2}{1 - C_3}\,\|\Delta\mathbb{Y}_n\| + \frac{h_*\,C_{\mathbf{T}}\,\mathrm{L}}{1 - C_3} \sum_{i=0}^{k-1} |\mu_i| \max_{i=0,\ldots,k-1} \|\Delta\mathbf{y}_{n-i}\| + \frac{C_{\mathbf{T}}\,\|R_n\|}{1 - C_3} \\ &\Rightarrow \quad \|\boldsymbol{\zeta}_n\| \le C_4\,\|\Delta\mathbb{Y}_n\| + C_5\,\|R_n\|,\end{aligned}$$

where all constants $C_1$ to $C_5$ are independent of the stiffness of the system.

The global error vector obeys the recurrence:

$$
\begin{aligned}
\Delta\mathbb{Y}_{n+1} &= \mathbf{M}(\omega_{n-k:n-1}, h_n\mathbf{J}_*)\,\Delta\mathbb{Y}_n + \mathbf{e}_1 \otimes \left(h_n\left(\boldsymbol{\delta}_n + \boldsymbol{\zeta}_n\right) + \mathbf{T}_n^{-1}\,R_n\right) \\
&= \prod_{i=k}^{n} \mathbf{M}(\omega_{i-k:i-1}, h_i\mathbf{J}_*)\,\Delta\mathbb{Y}_k \\
&\quad + \sum_{\ell=k+1}^{n}\prod_{i=n-\ell}^{n} \mathbf{M}(\omega_{i-k:i-1}, h_i\mathbf{J}_*)\cdot\mathbf{e}_1 \otimes \left(h_\ell\left(\boldsymbol{\delta}_\ell + \boldsymbol{\zeta}_\ell\right) + \mathbf{T}_\ell^{-1}\,R_\ell\right),
\end{aligned}
\tag{5.7}
$$

where $e_1 \in \mathbb{R}^k$ has the first entry equal to one and all other entries equal to zero, and $\otimes$ is the Kronecker product. From (5.4) the products of stability matrices are bounded by the constant $C_{\mathbf{M}}$. Taking norms leads to the following global error bounds:

$$
\|\Delta\mathbb{Y}_{n+1}\| \quad\leq\quad C_{\mathbf{M}}\,\|\Delta\mathbb{Y}_k\| + C_{\mathbf{M}}\,(C_1+C_4)\sum_{\ell=k+1}^{n}\left(h_{\max}\,\|\Delta\mathbb{Y}_\ell\| + \rho_\ell\right), \quad \rho_\ell \sim \mathcal{O}\big(h_{\max}^{p+1}\big),
$$

where $h_{\max} := \max_\ell h_\ell$. Solving this recurrence inequality by standard techniques, and using the fact that the initial state is $\Delta\mathbb{Y}_k \sim \mathcal{O}(h_{\max}^p)$, proves the result. ∎

*Remark* 5.9 (Non-stiff case). Theorem 5.8 is also valid for the non-stiff case where the Lipschitz constant of $\mathbf{f}(\cdot)$ is moderate.

*Remark* 5.10 (LIMM applied to index-1 DAEs). Consider the index-1 differential-algebraic equation (DAE, [26])

$$
\mathbf{x}' = \mathbf{f}(\mathbf{x},\mathbf{z}), \quad 0 = \mathbf{g}(\mathbf{x},\mathbf{z}), \quad \Rightarrow \quad \mathbf{z} = G(\mathbf{x}), \quad \mathbf{z}' = -\mathbf{g}_{\mathbf{z}}^{-1}\,\mathbf{g}_{\mathbf{x}}\,\mathbf{f},
\tag{5.8}
$$

where the sub-Jacobian $\mathbf{g}_{\mathbf{z}}$ is nonsingular and has a negative logarithmic norm in a neighborhood of the exact solution. When starting with consistent initial values the exact solution of (5.8) is smooth. Here we consider LIMM methods (2.4) with $\nu_i = 0$, $i = 0,\ldots,k-1$ satisfying (3.3b) up to order $p$, and strictly stable at infinity. Application to (5.8) gives (after rearranging the equations):

$$
\begin{aligned}
\sum_{i=-1}^{k-1}\alpha_i\,\mathbf{x}_{n-i} \quad &= \quad h_n\sum_{i=0}^{k-1}\beta_i\,\mathbf{f}_{n-i} + h_n\,(\mathbf{f}_{\mathbf{x}} - \mathbf{f}_{\mathbf{z}}\,\mathbf{g}_{\mathbf{z}}^{-1}\,\mathbf{g}_{\mathbf{x}})_n\sum_{i=-1}^{k-1}\mu_i\,\mathbf{x}_{n-i} \\
&\qquad -h_n\,(\mathbf{f}_{\mathbf{z}}\,\mathbf{g}_{\mathbf{z}}^{-1})_n\sum_{i=0}^{k-1}\beta_i\,\mathbf{g}_{n-i},
\end{aligned}
\tag{5.9a}
$$

$$
\sum_{i=-1}^{k-1}(\beta_i+\mu_i)\,\mathbf{g}_{n-i} \quad = \quad \sum_{i=-1}^{k-1}\mu_i\left(\mathbf{g}_{n-i} - (\mathbf{g}_{\mathbf{x}})_n\,\mathbf{x}_{n-i} + (\mathbf{g}_{\mathbf{z}})_n\,\mathbf{z}_{n-i}\right).
\tag{5.9b}
$$

The order conditions (3.3b)–(3.3d) imply that $\sum_{i=-1}^{k-1}\mu_i\,\phi(t_{n-i}) = \mathcal{O}(h_{\max}^p)$ for any smooth function $\phi(t)$, where $h_{\max} = \max h_n$. Under the assumption that the global error is smooth, and recalling that the exact solution is smooth, one can fit smooth curves through the numerical solutions $\mathbf{z}_{n-i}$ and $\mathbf{x}_{n-i}$ for $i = -1,\ldots,k$; in this case the right hand side of (5.9b) is $\mathcal{O}(h_{\max}^p)$. Since $\mu_i+\beta_i$ is a stable recurrence with roots strictly smaller than one [26, Chapter VI.2], $\mathbf{g}_n \sim \mathcal{O}(h_{\max}^p)$ for all $n$, and consequently $\mathbf{z}_n = G(\mathbf{x}_n) + \mathcal{O}(h_{\max}^p)$. Substituting these into (5.9a) shows that (5.9a) is the LIMM method applied to the reduced ODE, plus a perturbation of the size of the local truncation error $\mathcal{O}(h_{\max}^{p+1})$. Consequently, the global errors for the differential variable are $\mathbf{x}_n - \mathbf{x}(t_n) \sim \mathcal{O}(h_{\max}^p)$, and using again the fact that $\mathbf{z}_n = G(\mathbf{x}_n) + \mathcal{O}(h_{\max}^p)$ we have $\mathbf{z}_n - \mathbf{z}(t_n) \sim \mathcal{O}(h_{\max}^p)$.

**6. Construction of Optimized LIMM Schemes.** In order to construct practical LIMM schemes we seek to satisfy two requirements: have a small local truncation error and a large numerical stability region (4.3). For multistep methods, these two requirements are frequently at odds with each other; thus, it is vital that they are considered together when designing a method. To quantify the local truncation error of a method of order $p \geq 1$ we consider the (scaled) residuals of the $(p+1)$-st order conditions (3.3c) and (3.3d):

$$\rho_{p+1,a} = \sum_{i=-1}^{k-1} \alpha_i \, c_i^{p+1} + (p+1) \sum_{i=0}^{k-1} \beta_i \, c_i^p, \tag{6.1a}$$

$$\rho_{p+1,b} = (p+1) \sum_{i=-1}^{k-1} \mu_i \, c_i^p - (p+1)\, p \sum_{i=0}^{k-1} \nu_i \, c_i^{p-1}. \tag{6.1b}$$

If $p = 1$ and exact Jacobians are used then one only needs to consider the sum of the two residuals. To quantify stability observe that the LIMM method is $A(\phi)$ stable with a stability angle:

$$\phi(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \underset{0 \leq \theta \leq 2\pi}{\operatorname{argmin}} \left| \arg\left( -z\left( e^{i\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu} \right) \right) \right|. \tag{6.2}$$

We design practical linearly implicit multistep methods using a multiobjective genetic optimization algorithm to simultaneously maximize the $A(\phi)$ stability angle (6.2), and minimize the local error of the method. Maximizing the stability angle $\phi$ is equivalent to minimizing the first objective function:

$$\Phi_1 \left( \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu} \right) = \left( 1 + \phi(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu}) \right)^{-1}. \tag{6.3}$$

To maximize accuracy we minimize the second objective function:

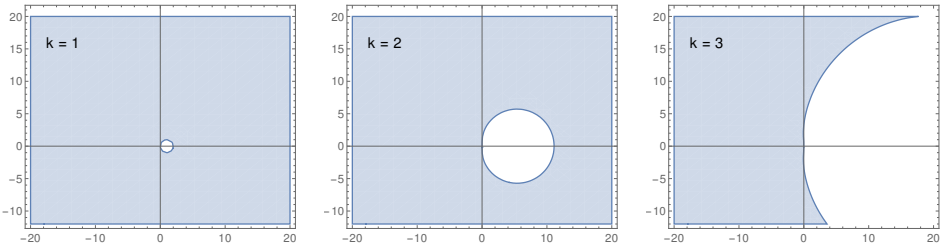
$$\Phi_2 \left( \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\nu} \right) = \rho_{p+1,a}^2 + \rho_{p+1,b}^2. \tag{6.4}$$

In order to simplify the search space, we choose to set $\nu_i = 0$, $i = 0, ..., k-1$. Then, we make use of the method order conditions to write the coefficients $\alpha_i$, $\beta_i$, and $\mu_i$ in terms of the $c_i$'s and a smaller subset of free parameters. Then, to get fixed stepsize coefficients, we substitute $c_i = i$ in the coefficient expressions and in the residuals $\rho_{p+1,a}$ and $\rho_{p+1,b}$ (we can later retrieve the variable stepsize coefficients by leaving the $c_i$'s and substituting only for the free parameters). We also make one additional simplification: by setting $\sigma(0) = 0$, we guarantee that as $z \to -\infty$, the stability function $\zeta(z) \to 0$ (which is a condition for L-stability [26, Section IV.3]). These simplifications are all directly embedded in the computation of both (6.3) and (6.4).

Finally, the resulting method must also be zero-stable, so we apply a nonlinear inequality constraint to enforce that roots of $\varrho(\zeta)$, the first characteristic polynomial of the method, fall on or inside the unit circle depending on multiplicity.

Making use of Matlab's optimization toolbox, we use a genetic algorithm to produce a population of good candidate methods for each of orders 2-5. From the set of candidates we select a method of each order that has the appropriate balance between stability angle and error coefficient. To acquire exact coefficients, we rationalize the optimal free parameters and substitute into the original coefficient expressions. Fixed stepsize coefficients for the selected LIMM-W methods are presented in Table 6.2, with error coefficients and stability angles compared with BDF in Table 6.1. Figure 6.1

plots the stability regions of the selected LIMM-W methods in the complex plane. Coefficients for the fixed stepsize LIMM methods and the corresponding stability regions can be found in the supplementary material in Table **??** and Figure **??**. The full variable stepsize coefficient expressions can be found in the supplementary material in Sections **??** and **??**.




(a) Order one method. (b) Order two method. (c) Order three method.

(d) Order four method. (e) Order five method.

FIG. 6.1. *Stability regions for fixed stepsize* LIMM-W *methods of orders one through five. Orders one and two are A-stable, and orders three to five are $A(\phi)$-stable with $\phi$'s listed in Table 6.1.*

TABLE 6.1

*Characteristics of k-step fixed stepsize* LIMM, LIMM-W, *BDF, and explicit/implicit Adams methods for $k = 1, \ldots, 5$. Implicit Adams methods are of order $k + 1$; all others are of order $k$. For* LIMM *and* LIMM-W, *stability angles are computed from (6.2), and error constants are computed as in (7.3).*

| | k | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| explicit Adams | A($\phi$)-stability angle | n/a | n/a | n/a | n/a | n/a |
| | Error constant | 0.5 | 0.416667 | 0.375 | 0.348611 | 0.329861 |
| implicit Adams | A($\phi$)-stability angle | 90. | n/a | n/a | n/a | n/a |
| | Error constant | 0.083333 | 0.041666 | 0.026389 | 0.01875 | 0.014269 |
| BDF | A($\phi$)-stability angle | 90. | 90. | 86.03 | 73.35 | 51.84 |
| | Error constant | 0.5 | 0.333333 | 0.25 | 0.2 | 0.166667 |
| LIMM | A($\phi$)-stability angle | 90. | 90. | 87.7849 | 78.0742 | 72.9999 |
| | Error constant | 0.5 | 0.222222 | 0.167344 | 0.204625 | 0.217405 |
| LIMM-W | A($\phi$)-stability angle | 90. | 90. | 87.3899 | 77.9101 | 70.3168 |
| | Error constant | 0.5 | 0.424915 | 0.403238 | 0.380873 | 0.365325 |

**7. Variable Stepsize and Variable Order Implementation.** In this section we discuss the details necessary to build an efficient self-starting variable stepsize and variable order implementation of LIMM methods. We followed these steps to implement the new schemes in our software package MATLODE [3], which is a Matlab version of FATODE [53].

Following [25, Chapter III.7], a key component required for adapting stepsize and order is an estimate $e_{n+1}^{(k,h)}$ of the local truncation error when the solution $\mathbf{y}_{n+1}$ is computed with an order $k$ LIMM scheme and step size $h_n$. Based on this error estimate one can determine whether to accept or reject the current step, and can estimate the optimal stepsize for a method of order $k$ using:

$$h_{\text{opt}}^{(k)} = h_n \, \|e_{n+1}^{(k,h)}\|^{-1/(k+1)}. \tag{7.1}$$

To determine when a change of method order is needed, we further require the error estimates $e_{n+1}^{(k-1,h)}$ and $e_{n+1}^{(k+1,h)}$ for the order $k-1$ and $k+1$ methods, respectively. Then, we can select for the next step the method order which gives the best balance between a low estimated error and a large optimal timestep.

In traditional linear multistep methods [25, Chapter III.7] a Taylor expansion reveals that

$$e_{n+1}^{(k,h)} = C_k(\boldsymbol{c})\, h_n^{k+1}\, \mathbf{y}^{(k+1)}(t_{n+1}) + \mathcal{O}\left(h_n^{k+2}\right), \tag{7.2}$$

where $C_k(\boldsymbol{c})$ is the method error coefficient as a function of the stepsize ratios $\boldsymbol{c} = [c_i]_{i=-1}^{k}$ (3.2a), derived from residuals on the $k+1$ order conditions. In the LIMM case different trees in the B-series expansion contribute differently to the $\mathcal{O}(h_n^{k+1})$ error term in (7.2), as can be seen from (3.3c)–(3.3d). In order to build practical error estimators we will use an expansion of the form (7.2) with our error constant computed from the two residuals (6.1):

$$C_k(\boldsymbol{c}) = \frac{1}{(k+1)!} \max\left(|\rho_{k+1,a}(\boldsymbol{c})|\,,\, |\rho_{k+1,a}(\boldsymbol{c}) + \rho_{k+1,b}(\boldsymbol{c})|\right), \tag{7.3}$$

where the sum of the residuals is used because, for LIMM, (6.1b) always appears on trees alongside (6.1a), as described in the observations below (3.16). We use past solutions for $\mathbf{y}_{n+1}, \ldots, \mathbf{y}_{n-k}$ to approximate the $(k+1)$-st time derivative of the solution via divided differences:

$$\delta^{k+1}\mathbf{y}\left[t_{n+1}, \ldots, t_{n-k}\right] = \frac{\mathbf{y}^{(k+1)}(\xi)}{(k+1)!}, \quad \xi \in [t_{n-k}, t_{n+1}]. \tag{7.4}$$

Putting together equations (7.2), (7.3), and (7.4) we arrive at the following estimate for the local error.

DEFINITION 7.1 (Local error estimate). *For a $k$-step order $k$ LIMM method a practical local truncation error estimate is:*

$$e_{n+1}^{(k,h)} = (k+1)!\, C_k(\boldsymbol{c})\, h_n^{k+1}\, \delta^{k+1}\mathbf{y}\left[t_{n+1}, \ldots, t_{n-k}\right], \tag{7.5}$$

*using method error coefficient $C_k(\boldsymbol{c})$ (7.3), the current stepsize $h_n$, and the $(k+1)$-st order divided difference of the solution at $t_{n+1}, \ldots, t_{n-k}$. Due to (7.3), (7.5) may slightly overestimate the local errors in the asymptotic regime.*

This error estimate is convenient for our purpose, as we can compute estimates for methods of order $k-1$ and $k+1$ by simply using their error coefficients and changing the order of the divided difference. Note that the divided difference of order $k+1$ requires $\mathbf{y}_{n+1}$ and a history of $k+1$ values of $\mathbf{y}_n, \ldots, \mathbf{y}_{n-k}$, whereas the $k$-step LIMM method only requires a history of $k$ values of $\mathbf{y}$ in order to compute $\mathbf{y}_{n+1}$. Divided differences for the higher order methods require storing additional past solution values.

Instead of building divided differences at each step from the history of solutions, it is more efficient to maintain a history of the divided differences and build new ones using the following formulas (similar to updating the Nordsieck vector in a variety of multistep codes [4, 8, 35]). First, after computing $\mathbf{y}_{n+1}$ via the LIMM method, we produce the $(k+1)$-st divided difference as follows:

(7.6)
$$\delta^{k+1}\mathbf{y}\left[t_{n+1}, \ldots, t_{n-k}\right] = \frac{\mathbf{y}_{n+1} - \mathbf{y}_n}{\prod_{j=0}^{k} \sum_{l=0}^{j} (t_{n+1-l} - t_{n-l})} - \sum_{i=1}^{k} \frac{\delta^i \mathbf{y}\left[t_n, \ldots, t_{n-i}\right]}{\prod_{j=i}^{k} \sum_{l=0}^{j} (t_{n+1-l} - t_{n-l})}.$$

Then, through direct application of the recursive definition, one computes the updated values of order $k$ and $k+2$ divided differences.

To avoid the redundant saving of both the history of solution values and of divided differences, one can reformulate the LIMM method (2.4) in terms of divided differences:

(7.7)
$$\begin{aligned}(\mathbf{I} - h_n\, \mu_{-1}(\boldsymbol{c})\, \mathbf{J}_n)\, \mathbf{y}_{n+1} = &- \sum_{i=0}^{k-1} h_n^i\, \widehat{\alpha}_i(\boldsymbol{c})\, \delta^i \mathbf{y}\left[t_n, \ldots, t_{n-i}\right] \\ &+ h\, \mathbf{J}_n \sum_{i=0}^{k-1} h_n^i\, \widehat{\mu}_i(\boldsymbol{c})\, \delta^i \mathbf{y}\left[t_n, \ldots, t_{n-i}\right] + h_n \sum_{i=0}^{k-1} h_n^i\, \widehat{\beta}_i(\boldsymbol{c})\, \delta^i \mathbf{f}\left[t_n, \ldots, t_{n-i}\right],\end{aligned}$$

where $\delta^i \mathbf{f}\left[t_n, \ldots, t_{n-i}\right]$ are divided differences constructed from evaluations of the right-hand side function $\mathbf{f}(t_n, \mathbf{y}_n)$ evaluated at $(t_n, \mathbf{y}_n)$, and $\widehat{\alpha}_i$, $\widehat{\mu}_i$, $\widehat{\beta}_i$ are transformed coefficients (as functions of the timestep ratios). The transformed coefficients can be computed from the original method coefficients as

(7.8)
$$\widehat{\alpha}_i(\boldsymbol{c}) = (-1)^i \sum_{j=i}^{k-1} \alpha_j(\boldsymbol{c}) \prod_{l=1}^{i} \sum_{m=l}^{j} (c_m - c_{m-1}).$$

The $\widehat{\mu}_i$ and $\widehat{\beta}_i$ can be computed from the $\mu$'s and $\beta$'s in the same manner.

*Remark* 7.2 (The first step). The first step of integration requires additional care. Even starting with a one-step order one method requires an order two divided difference for error estimation (and a difference of order three for estimation of the order two method's error). We initialize our divided differences as follows:

$$\delta^0 \mathbf{y}[t_0] = \mathbf{y}_0, \quad \delta^1 \mathbf{y}\left[t_0\right] \approx \mathbf{f}_0.$$

Then, following the computation of $\mathbf{y}_1$, we approximate the next divided differences

$$\delta^2 \mathbf{y}\left[t_1, t_0\right] \approx \left((\mathbf{y}_1 - \mathbf{y}_0)/h_0 - \mathbf{f}_0\right)/\left(2h_0\right), \quad \delta^3 \mathbf{y}\left[t_1, t_0\right] \approx \left(\delta^2 \mathbf{y}\left[t_1, t_0\right] - \mathbf{J}_0 \mathbf{f}_0\right)/\left(3h_0\right).$$

*Remark* 7.3 (Stability for variable stepsize methods). Evaluating the stability properties of variable stepsize multistep methods is challenging (see [9–11, 24] for analysis of variable stepsize BDF methods). Specifically, explicit formulas for the

bounds $\omega_{\min}$ and $\omega_{\max}$ in Theorem 5.8 that ensure stability for a general multistep methods are not available. Since an in-depth analysis is outside the scope of the current work, we follow the best practices of other variable stepsize multistep implementations [25, Section III.5], by ensuring the base fixed stepsize method (from which the variable stepsize one is derived) is stable, and limiting stepsize increases to only occur after $k$ successful steps at the current stepsize.

**8. Numerical Results.** In this section we present numerical results for LIMM methods including both convergence and performance experiments. First, we demonstrate the convergence order of the fixed stepsize LIMM and LIMM-W methods, then we examine the performance of the self-starting variable stepsize and variable order implementation of the LIMM methods which was detailed in section 7.

The convergence tests use fixed stepsize implementations of LIMM and LIMM-W, with the built-in Matlab time integrator `ode15s` used as a starter method. For the performance tests, our primary point of comparison is the BDF methods [25, Section III.1], one of the most widely used families of implicit multistep methods. We use a self-starting variable stepsize and order implementation of BDF built in the same framework as the LIMM implementation, using the same error controller and also the error estimator from definition 7.1. For comparison we include results for SDIRK4a [26, Section IV.6] (a 5-stage, fourth order singly diagonally implicit Runge-Kutta method) and RODAS4 [26, Section VI.4] (a 6-stage, fourth order Rosenbrock method) as examples of high-order single step implicit methods. The variable stepsize LIMM-(W) and BDF implementations, along with the SDIRK4a and RODAS4 methods can be found in our MATLODE package [3], in the `limm_dev` branch. We also provide comparisons against the Matlab built-in `ode15s` in both BDF and NDF modes.

Each time integrator is tested by sweeping through the same series of tolerances from $10^{-2}, 10^{-3}, ..., 10^{-10}$, with equal relative and absolute tolerances. When GMRES is used for linear system solves, it uses a tolerance that is one tenth of the integrator tolerance. All tests are performed in Matlab (version 2019a) on a workstation with dual Intel Xeon E5-2650 v3 processors with 20 cores (40 threads) and 128GB of memory. Test problems are drawn from the ODE Test Problems package [37], which contains Matlab implementations of a variety of ODEs and PDEs suitable for testing time integrators. Reference solutions are obtained by applying the Matlab built-in `ode15s` integrator when full Jacobians are available, or an implementation of RODAS4 using GMRES when only Jacobian-vector products are available; both are applied with the tightest possible tolerance of 100 times the machine precision.

**8.1. Convergence: Lorenz-96 ODE.** To test the convergence order of the fixed stepsize LIMM and LIMM-W methods, we use the Lorenz-96 problem [28], a system of nonlinear ODEs defined as

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F(t), \quad i = 1, \ldots, N, \tag{8.1}$$

with periodic boundary conditions $x_{N+1} = x_1$, $x_0 = x_N$, $x_{-1} = x_{N-1}$ and forcing function $F(t) = 8 + 4\cos(3\pi t)$. We select $N = 40$, and timespan $t \in [0, 0.5]$.

Figure 8.1 shows the convergence results for LIMM orders 1-5 as solid lines, and for LIMM-W orders 1-5 as dashed lines. The legend contains the slopes of the linear interpolants for each test, demonstrating that each method achieves its theoretical order. LIMM-W shows slightly smaller total errors.

**8.2. Performance: Gray–Scott reaction-diffusion model.** We run performance comparisons using the Gray–Scott reaction-diffusion model, which describes a

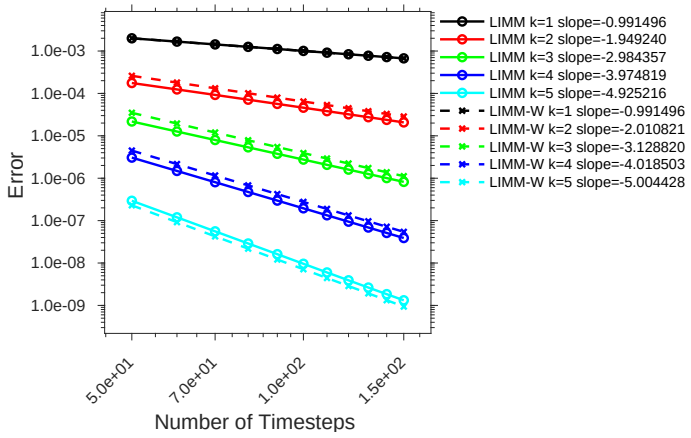



FIG. 8.1. *Fixed stepsize results for the Lorenz-96 problem (8.1) showing the orders of convergence for* LIMM-W *(dashed lines) and* LIMM *(solid lines) methods for orders 1-5.*

two species chemical reactions with retirement [21, 22]:

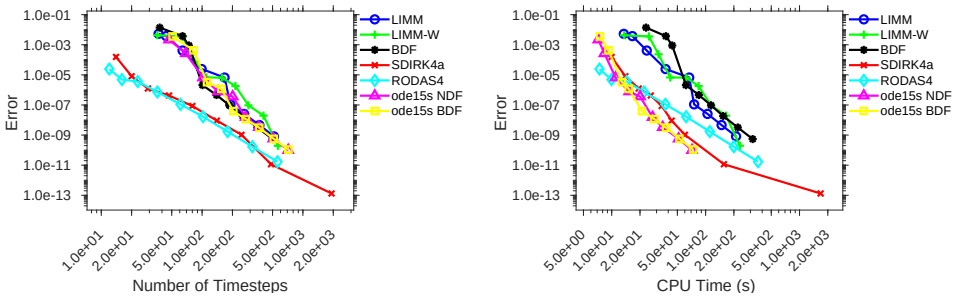
$$\frac{\partial u}{\partial t} = \varepsilon_1 \Delta u - uv^2 + F(1-u), \qquad \frac{\partial v}{\partial t} = \varepsilon_2 \Delta v + uv^2 - (F+k)v, \tag{8.2}$$

where $\varepsilon_1 = 0.2$ and $\varepsilon_2 = 0.1$ are diffusion rates, and $F = 0.04$ and $k = 0.06$ are reaction rates. We use a second order finite difference spatial discretization with periodic boundary conditions on a $128 \times 128$ 2D grid, which brings (8.2) to ODE form (1.1) with $N = 2 \times 128 \times 128$. The simulation time interval is $t \in [0, 2]$.

Results on the Gray–Scott model are given in Figure 8.2 for our implementations of LIMM, LIMM-W, BDF, SDIRK4a, RODAS4, as well as for the `ode15s` implementations of BDF and NDF [44]. Because the full Jacobian is available, all methods use a direct $LU$-factorization, that is reused for methods requiring multiple linear solves per step. The error vs. timestep count results in Figure 8.2a reveal that all five multistep integrators achieve similar levels of error with similar numbers of timesteps, thus none of these multistep methods or implementations have a clear stability or error advantage over the others in this test. The one-step methods SDIRK4a and RODAS4 do show better errors with fewer timesteps, likely as a result of their greater flexibility in choosing timestep sizes. The CPU time comparison in Figure 8.2b reveals the advantage of the built-in `ode15s` implementations, as they benefit from a massive amount of software optimizations. Comparing the results for our implementation of multistep integrators, we can see a small performance advantage for LIMM and LIMM-W over the similar BDF implementation, due to the need for only one linear system solve per timestep. Given a similar level of software optimization, we might expect the LIMM methods to hold the same advantage over the `ode15s` BDF implementation. The SDIRK4a and RODAS4 also show very good performance, due to the lower timestep count and reuse one $LU$-factorization per timestep.

**8.3. Performance: quasi-geostrophic model.** Next, we compare method performance using the 1.5-layer quasi-geostrophic (QG) model [40], which provides a simplified representation of ocean dynamics:

$$\frac{\partial q}{\partial t} = -\psi_x - \varepsilon J(\psi, q) - A\Delta^3 \psi + 2\pi \sin(2\pi y), \qquad q = \Delta\psi - F\psi, \tag{8.3}$$




(a) Variable stepsize convergence results.

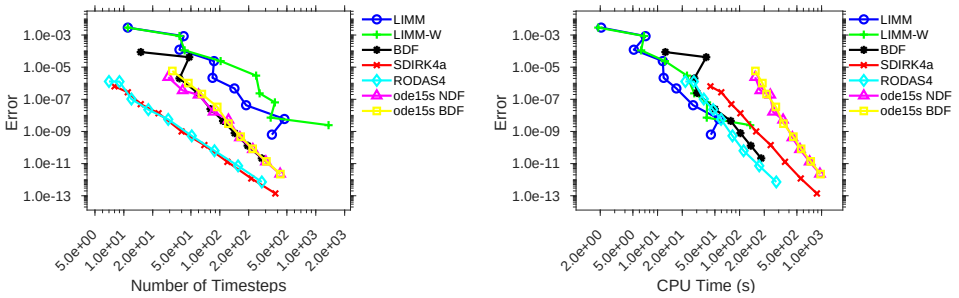
(b) Work-precision diagram.

FIG. 8.2. *Performance results for different integrators applied to the 2D Gray-Scott reaction-diffusion problem* (8.2).

where $J(\psi, q) = \psi_x q_y - \psi_y q_x$, and $F = 1600$, $\varepsilon = 10^{-5}$ and $A = 10^{-5}$ are constants. The implementation discretizes the system in terms of the stream function $\psi$, on the spatial domain $(x, y) \in [0, 1]^2$, using second order central finite differences and homogeneous Dirichlet boundary conditions on a $127 \times 127$ grid. Integration is performed over the time span $t \in [0, 0.01]$. Due to the required solution of a Helmholtz equation, a portion of the QG Jacobian is dense; only Jacobian-vector products are available and all integrators use GMRES as linear solver, except for `ode15s` which cannot use iterative linear solvers. Instead, both BDF and NDF results for `ode15s` make use of a numerical Jacobian approximation.

Figure 8.3 shows errors versus timestep count and CPU times for LIMM, LIMM-W, BDF, SDIRK4a, RODAS4, and `ode15s` applied to the QG model. First, we notice the jagged behavior of both the LIMM and BDF methods, especially at looser tolerances; this is an artifact of the variable order methods, where methods of lower order are preferred for their better stability. When the tolerances are tightened, the graphs smoothen, as the algorithm prefers the better asymptotic errors of the higher order methods. The LIMM and LIMM-W methods are particularly sensitive to this, possibly due to the use of GMRES, as inexact linear solves could degrade stability or lead to LIMM order reduction. Despite this, Figure 8.3b shows that LIMM retains its CPU time advantage over BDF, and also matches or outperforms RODAS4 and SDIRK4a for similar levels of error, as the small stability advantages that BDF, SDIRK and Rosenbrock methods cannot overcome their larger cost per timestep. The built-in `ode15s` methods both provide very poor performance for this test case, as they do not support the use of direct Jacobian-vector products and are forced to build numerical Jacobian approximations to use with direct solves.

**9. Conclusions.** Classical implicit linear multistep methods require the solution of a nonlinear system of equations at each timestep. This work develops the LIMM class of linearly implicit multistep methods, which only require the solution of one linear system per timestep. Order conditions for variable stepsize LIMM methods of arbitrary order are constructed via Butcher trees and B-series operations. Order conditions for LIMM-W schemes that can use arbitrary approximations of the Jacobian matrix are developed by direct series expansion. A LIMM method has twice as many coefficients as a traditional linear multistep method (with the same number of steps), and this additional freedom enables the construction of methods with excellent accuracy and stability properties. We discuss the optimal design of LIMM methods, and




(a) Variable stepsize convergence results. (b) Work-precision diagram.

FIG. 8.3. *Performance results for different integrators applied to the 1.5 layer QG model* (8.3).

develop a set of $k$-step order $k$ LIMM methods for $k = 1, \ldots, 5$. We also discuss the details of a self-starting variable stepsize and variable order implementation of these LIMM methods. Numerical experiments demonstrate the convergence at the theoretical orders for the fixed stepsize LIMM methods. Moreover, our variable stepsize LIMM implementation outperforms a comparable BDF implementation for nonlinear problems.

## REFERENCES

[1] G. AKRIVIS, M. CROUZEIX, AND C. MAKRIDAKIS, *Implicit-explicit multistep methods for quasilinear parabolic equations.*, Numerische Mathematik, 82 (1999), pp. 521–541, https://doi.org/10.1007/s002110050429.

[2] G. AKRIVIS AND C. LUBICH, *Fully implicit, linearly implicit and implicit–explicit backward difference formulae for quasi-linear parabolic equations*, Numerische Mathematik, 131 (2015), pp. 713–735, https://doi.org/10.1007/s00211-015-0702-0.

[3] A. AUGUSTINE AND A. SANDU, *MATLODE: A Matlab suite for ODE integration and sensitivity analysis*, Submitted to ACM TOMS, (2017), https://github.com/ComputationalScienceLaboratory/MATLODE.

[4] P. N. BROWN, G. D. BYRNE, AND A. C. HINDMARSH, *VODE: A variable-coefficient ODE solver*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 1038–1051, https://doi.org/10.1137/0910062.

[5] J. BUTCHER, *Trees, B-series and exponential integrators*, IMA Journal of Numerical Analysis, 30 (2009), pp. 131–140, https://doi.org/10.1093/imanum/drn086.

[6] J. BUTCHER, *Numerical Methods for Ordinary Differential Equations*, Wiley, 3 ed., 2016, https://doi.org/10.1002/9781119121534.

[7] J. C. BUTCHER AND A. D. HEARD, *Stability of numerical methods for ordinary differential equations*, Numerical Algorithms, 31 (2002), pp. 59–73, https://doi.org/10.1023/A:1021108006254.

[8] G. D. BYRNE AND A. C. HINDMARSH, *A polyalgorithm for the numerical solution of ordinary differential equations*, ACM Trans. Math. Softw., 1 (1975), pp. 71–96, https://doi.org/10.1145/355626.355636.

[9] M. CALVO, T. GRANDE, AND R. GRIGORIEFF, *On the zero stability of the variable order variable stepsize BDF-formulas*, Numerische Mathematik, 57 (1990), pp. 39–50, https://doi.org/10.1007/BF01386395.

[10] M. CALVO, F. LISBONA, AND J. MONTIJANO, *On the stability of variable-stepsize Nordsieck BDF methods*, SIAM Journal on Numerical Analysis, 24 (1987), pp. 844–854, https://doi.org/10.1137/0724054.

[11] M. CALVO, J. MONTIJANO, AND L. RÁNDEZ, *A0-stability of variable stepsize BDF methods*, Journal of Computational and Applied Mathematics, 45 (1993), pp. 29 – 39, https://doi.org/10.1016/0377-0427(93)90262-A.

[12] A. CARDONE, Z. JACKIEWICZ, A. SANDU, AND H. ZHANG, *Extrapolated implicit-explicit Runge–*

*Kutta methods*, Mathematical Modelling and Analysis, 19 (2014), pp. 18–43, https://doi.org/10.3846/13926292.2014.892903.

[13] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang, *Extrapolation-based implicit-explicit general linear methods*, Numerical Algorithms, 65 (2014), pp. 377–399, https://doi.org/10.1007/s11075-013-9759-y.

[14] E. Constantinescu and A. Sandu, *Extrapolated implicit-explicit time stepping*, SIAM Journal on Scientific Computing, 31 (2010), pp. 4452–4477, https://doi.org/10.1137/080732833.

[15] C. F. Curtiss and J. O. Hirschfelder, *Integration of stiff equations*, Proceedings of the National Academy of Sciences of the United States of America, 38 (1952), p. 235, https://doi.org/10.1073/pnas.38.3.235.

[16] D. Daescu, A. Sandu, and G. Carmichael, *Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: II – Numerical validation and applications*, Atmospheric Environment, 37 (2003), pp. 5097–5114, http://dx.doi.org/10.1016/j.atmosenv.2003.08.020.

[17] E. Emmrich, *Stability and error of the variable two-step BDF for semilinear parabolic problems*, Journal of Applied Mathematics and Computing, 19 (2005), pp. 33–55, https://doi.org/10.1007/BF02935787.

[18] D. Forti and L. Dedè, *Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a high performance computing framework*, Computers & Fluids, 117 (2015), pp. 168 – 182, https://doi.org/10.1016/j.compfluid.2015.05.011.

[19] C. W. Gear and K. W. Tu, *The effect of variable mesh size on the stability of multistep methods*, SIAM Journal on Numerical Analysis, 11 (1974), pp. 1025–1043, http://www.jstor.org/stable/2156040.

[20] R. Glandon, P. Tranquilli, and A. Sandu, *Biorthogonal Rosenbrock-Krylov time discretization methods*, Applied Numerical Mathematics, 150 (2020), pp. 233–251, https://doi.org/10.1016/j.apnum.2019.09.003.

[21] P. Gray and S. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Isolas and other forms of multistability*, Chemical Engineering Science, 38 (1983), pp. 29 – 43, https://doi.org/10.1016/0009-2509(83)80132-8.

[22] P. Gray and S. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system $A + 2B \rightarrow 3B$; $B \rightarrow C$*, Chemical Engineering Science, 39 (1984), pp. 1087 – 1097, https://doi.org/10.1016/0009-2509(84)87017-7.

[23] R. D. Grigorieff, *Stability of multistep-methods on variable grids*, Numerische Mathematik, 42 (1983), pp. 359–377, https://doi.org/10.1007/BF01389580.

[24] N. Guglielmi and M. Zennaro, *On the zero-stability of variable stepsize multistep methods: the spectral radius approach*, Numerische Mathematik, 88 (2001), pp. 445–458, https://doi.org/10.1007/s211-001-8010-0.

[25] E. Hairer, S. Norsett, and G. Wanner, *Solving ordinary differential equations I: Nonstiff problems*, no. 8 in Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 1993, https://doi.org/10.1007/978-3-540-78862-1.

[26] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, no. 14 in Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 2 ed., 1996, https://doi.org/10.1007/978-3-642-05221-7.

[27] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM Journal on Scientific Computing, 19 (1998), pp. 1552–1574, https://doi.org/10.1137/S1064827595295337.

[28] E. N. Lorenz, *Predictability – a problem partly solved*, Cambridge University Press, 2006, pp. 40–58, https://doi.org/10.1017/CBO9780511617652.004.

[29] N. Macon and A. Spitzbart, *Inverses of vandermonde matrices*, The American Mathematical Monthly, 65 (1958), pp. 95–100, http://www.jstor.org/stable/2308881.

[30] B. R. Munson, *Fundamentals Of Fluid Mechanics*, John Wiley & Sons, Inc., Hoboken, NJ, 7 ed., 2013.

[31] M. Narayanamurthi and A. Sandu, *Efficient implementation of partitioned stiff exponential Runge-Kutta methods*, Applied Numerical Mathematics, 152 (2020), pp. 141–158, https://doi.org/10.1016/j.apnum.2020.01.010.

[32] M. Narayanamurthi and A. Sandu, *Partitioned exponential methods for coupled multiphysics systems*, Applied Numerical Mathematics, 161 (2021), pp. 178 – 207, https://doi.org/10.1016/j.apnum.2020.10.020.

[33] M. Narayanamurthi, P. Tranquilli, A. Sandu, and M. Tokman, *EPIRK-W and EPIRK-K time integration methods*, Journal of Scientific Computing, 78 (2019), pp. 167–201, https://doi.org/10.1007/s10915-018-0761-3.

[34] A. Ostermann, M. Thalhammer, and G. Kirlinger, *Stability of linear multistep methods and*

*applications to nonlinear parabolic problems*, Applied Numerical Mathematics, 48 (2004), pp. 389–407, https://doi.org/10.1016/j.apnum.2003.10.004. Workshop on Innovative Time Integrators for PDEs.

[35] K. Radhakrishnan and A. C. Hindmarsh, *Description and use of LSODE, the Livermore solver for ordinary differential equations*, Tech. Report UCRL-ID-113855, Lawrence Livermore National Laboratory, Livermore, CA, 1993, https://ntrs.nasa.gov/search.jsp?R=19940030753.

[36] S. Roberts, J. Loffeld, A. Sarshar, C. Woodward, and A. Sandu, *Implicit multirate GARK methods*, In print, Journal of Scientific Computing CSL-TR-19-5, Computational Science Laboratory, Virginia Tech, 2020, https://arxiv.org/abs/1910.14079.

[37] S. Roberts, A. A. Popov, and A. Sandu, *ODE Test Problems: a MATLAB suite of initial value problems*, arXiv e-prints, (2019), arXiv:1901.04098, https://arxiv.org/abs/1901.04098.

[38] S. Roberts, A. Sarshar, and A. Sandu, *Parallel implicit-explicit general linear methods*, Communications on Applied Mathematics and Computation, in print (2020), https://doi.org/10.1007/s42967-020-00083-5.

[39] H. H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, The Computer Journal, 5 (1963), pp. 329–330, https://doi.org/10.1093/comjnl/5.4.329.

[40] P. Sakov and P. R. Oke, *A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters*, Tellus A, 60 (2007), pp. 361–371, https://doi.org/10.1111/j.1600-0870.2007.00299.x.

[41] A. Sandu, *A class of multirate infinitesimal GARK methods*, SIAM Journal on Numerical Analysis, 57 (2019), pp. 2300–2327, https://doi.org/10.1137/18M1205492.

[42] A. Sandu, *Convergence results for implicit-explicit general linear methods*, Applied Numerical Mathematics, 156 (2020), https://doi.org/10.1016/j.apnum.2020.04.005.

[43] A. Sandu and C. T. Borden, *A framework for the numerical treatment of aerosol dynamics*, Applied Numerical Mathematics, 45 (2003), pp. 475–497, https://doi.org/10.1016/S0168-9274(02)00251-9.

[44] L. F. Shampine and M. W. Reichelt, *The MATLAB ODE suite*, SIAM Journal on Scientific Computing, 18 (1997), pp. 1–22, https://doi.org/10.1137/S1064827594276424.

[45] G. Söderlind, I. Fekete, and I. Faragó, *On the zero-stability of multistep methods on smooth nonuniform grids*, BIT Numerical Mathematics, 58 (2018), pp. 1125–1143, https://doi.org/10.1007/s10543-018-0716-y.

[46] L. Song and C. Yang, *Convergence of a second-order linearized BDF–IPDG for nonlinear parabolic equations with discontinuous coefficients*, Journal of Scientific Computing, 70 (2017), pp. 662–685, https://doi.org/10.1007/s10915-016-0261-2.

[47] T. Steihaug and A. Wolfbrandt, *An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations*, Mathematics of Computation, 33 (1979), pp. 521–534, https://doi.org/10.1090/S0025-5718-1979-0521273-8.

[48] M. Tokman, *A new class of exponential propagation iterative methods of Runge–Kutta type (EPIRK)*, Journal of Computational Physics, 230 (2011), pp. 8762—8778, https://doi.org/10.1016/j.jcp.2011.08.023.

[49] P. Tranquilli and A. Sandu, *Exponential-Krylov methods for ordinary differential equations*, Journal of Computational Physics, 278 (2014), pp. 31–46, https://doi.org/10.1016/j.jcp.2014.08.013.

[50] P. Tranquilli and A. Sandu, *Rosenbrock-Krylov methods for large systems of differential equations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1313–A1338, https://doi.org/10.1137/130923336.

[51] C. Yang, *Convergence of a linearized second-order BDF–FEM for nonlinear parabolic interface problems*, Computers & Mathematics with Applications, 70 (2015), pp. 265 – 281, https://doi.org/10.1016/j.camwa.2015.05.006.

[52] C. Yao, Y. Lin, C. Wang, and Y. Kou, *A third order linearized BDF scheme for Maxwell's equations with nonlinear conductivity using finite element method*, International Journal of Numerical Analysis and Modeling, 14 (2017), pp. 511–531, http://global-sci.org/intro/article_detail/ijnam/10047.html.

[53] H. Zhang and A. Sandu, *FATODE: A library for forward, adjoint and tangent linear integration of stiff systems*, SIAM Journal on Scientific Computing, 36 (2014), pp. C504–C523, https://doi.org/10.1137/130912335.

[54] H. Zhang, A. Sandu, and S. Blaise, *Partitioned and implicit-explicit general linear methods for ordinary differential equations*, Journal of Scientific Computing, 61 (2014), pp. 119–144, https://doi.org/10.1007/s10915-014-9819-z.

[55] H. Zhang, A. Sandu, and S. Blaise, *High order implicit–explicit general linear methods with optimized stability regions*, SIAM Journal on Scientific Computing, 38 (2016), pp. A1430–A1453, https://doi.org/10.1137/15M1018897.

[56] E. Zharovsky, A. Sandu, and H. Zhang, *A class of IMEX two-step Runge-Kutta methods*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 321–341, https://doi.org/10.1137/130937883.

TABLE 6.2
*Exact coefficients for k-step fixed stepsize* LIMM-W *methods of order k, for* $k = 1 \dots 5$.

| $i$ | $-1$ | $0$ |
|---|---|---|
| $\alpha_i$ | $1$ | $-1$ |
| $\beta_i$ | $0$ | $1$ |
| $\mu_i$ | $1$ | $-1$ |

(a) LIMM-W 1-step order 1 coefficients.

| $i$ | $-1$ | $0$ | $1$ |
|---|---|---|---|
| $\alpha_i$ | $1$ | $-\frac{146619050}{133414177}$ | $\frac{13204873}{133414177}$ |
| $\beta_i$ | $0$ | $\frac{193518829}{133414177}$ | $-\frac{73309525}{133414177}$ |
| $\mu_i$ | $\frac{73309525}{133414177}$ | $-\frac{146619050}{133414177}$ | $\frac{73309525}{133414177}$ |

(b) LIMM-W 2-step order two coefficients.

| $i$ | $-1$ | $0$ | $1$ | $2$ |
|---|---|---|---|---|
| $\alpha_i$ | $1$ | $-\frac{192592391}{118869921}$ | $\frac{41981416}{61945353}$ | $-\frac{5229175002546}{90906657005273}$ |
| $\beta_i$ | $0$ | $\frac{16233524076078647}{9817918956569484}$ | $-\frac{4193351041739980}{2454479739142371}$ | $\frac{4833530710149845}{9817918956569484}$ |
| $\mu_i$ | $\frac{4833530710149845}{9817918956569484}$ | $-\frac{4833530710149845}{3272639652189828}$ | $\frac{4833530710149845}{3272639652189828}$ | $-\frac{4833530710149845}{9817918956569484}$ |

(c) LIMM-W 3-step order three coefficients.

| $i$ | $-1$ | $0$ |
|---|---|---|
| $\alpha_i$ | $1$ | $-\frac{68547635}{35752838}$ |
| $\beta_i$ | $0$ | $\frac{136586035293284691}{70863342514650928}$ |
| $\mu_i$ | $\frac{719593273725529014067099}{1590107007076830596400464}$ | $-\frac{719593273725529014067099}{3975267517692076491000116}$ |
| $i$ | $1$ | $2$ |
| $\alpha_i$ | $\frac{332147775}{246829693}$ | $-\frac{120323842}{247754257}$ |
| $\beta_i$ | $-\frac{4675749204985773774031537}{1590107007076830596400464}$ | $\frac{3052167106160890365719135}{1590107007076830596400464}$ |
| $\mu_i$ | $\frac{2158779821176587042201297}{7950535035384152982002 32}$ | $-\frac{719593273725529014067099}{3975267517692076491000116}$ |
| $i$ | $3$ | |
| $\alpha_i$ | $\frac{11382486133370227314625}{19876337588460382455005 8}$ | |
| $\beta_i$ | $-\frac{719593273725529014067099}{1590107007076830596400464}$ | |
| $\mu_i$ | $\frac{719593273725529014067099}{1590107007076830596400464}$ | |

(d) LIMM-W 4-step order 4 coefficients.

| $i$ | $-1$ | $0$ |
|---|---|---|
| $\alpha_i$ | $1$ | $-\frac{170476503}{75237041}$ |
| $\beta_i$ | $0$ | $\frac{331771538883068227418188877246 6725}{153316057707823400216955030318 6624}$ |
| $\mu_i$ | $\frac{659152962863648794216719015147251}{153316057707823400216955030318 6624}$ | $-\frac{329576481431824397108359507573 6255}{153316057707823400216955030318 6624}$ |
| $i$ | $1$ | $2$ |
| $\alpha_i$ | $\frac{124149029}{52265116}$ | $-\frac{53697673}{39342191}$ |
| $\beta_i$ | $-\frac{3387422206381293505203420155442595}{7665802885391170010847751515933 12}$ | $\frac{2946833511207935757036598656340 35}{6388169071159308342373126263277 6}$ |
| $\mu_i$ | $\frac{329576481431824397108359507573 6255}{7665802885391170010847751515933 12}$ | $-\frac{329576481431824397108359507573 6255}{7665802885391170010847751515933 12}$ |
| $i$ | $3$ | $4$ |
| $\alpha_i$ | $\frac{67073128}{206463953}$ | $-\frac{221958277447939858892136346 6455}{3194084535579654171186563131 6388}$ |
| $\beta_i$ | $-\frac{1632980052046035774065588376123413}{7665802885391170010847751515933 12}$ | $\frac{659152962863648794216719015147251}{153316057707823400216955030318 6624}$ |
| $\mu_i$ | $\frac{329576481431824397108359507573 6255}{153316057707823400216955030318 6624}$ | $-\frac{659152962863648794216719015147251}{153316057707823400216955030318 6624}$ |

(e) LIMM-W 5-step order 5 coefficients.