

## ESTIMATING GLOBAL ERRORS IN TIME STEPPING\*

EMIL CONSTANTINESCU<sup>†</sup>

**Abstract.** This study introduces new time-stepping strategies with built-in global error estimators. The new methods propagate the defect along with the numerical solution much like solving for the correction or Zadunaisky’s procedure; however, the proposed approach allows for overlapped internal computations and, therefore, represents a generalization of the classical numerical schemes for solving differential equations with global error estimation. The resulting algorithms can be effectively represented as general linear methods. We present a few explicit self-starting schemes akin to Runge-Kutta methods with global error estimation and illustrate the theoretical considerations on several examples.

**Key words.** time integration, local and global error estimation, general linear methods

**AMS subject classifications.** 65L05, 65L06, 65L20, 65L70

**1. Introduction.** The global error or *a posteriori* error represents the actual numerical error resulting after applying a time-stepping algorithm. Calculating this error and controlling it by step-size reduction are generally viewed as expensive processes, and therefore in practice only local error or the error from one step to the next is used for error estimation and control [2, 45, 47, 58]. It is known that local error estimation is not always suitable, especially for problems with unstable modes or long integration times [41, 46]. This aspect prompts us to revisit global error estimation in order to make it more transparent and practical, which ultimately leads to better error control and reliable accuracy.

In this study we introduce and analyze efficient strategies for estimating global errors for time-stepping algorithms. We present a unifying approach that includes most of the classical strategies as particular cases, and we develop new algorithms that fall under general linear time-stepping schemes. One of the most comprehensive surveys for global error estimation is by Skeel [72]. We focus on a subset of the methods discussed therein and generalize some of the results presented there.

Global error estimation in time stepping has a long history [1, 28–32, 34, 50, 56, 57, 59, 62, 64–68, 77]. *A posteriori* global error estimation has been recently discussed in [3, 19, 35, 52]. Step-size control with multimethod Runge-Kutta (RK) is discussed in [20, 67, 69, 70]. Global error estimation for stiff problems is discussed in [27, 52, 53, 75, 76]. Adjoint methods for global error estimation for PDEs are analyzed in [23, 37]. These studies cover most of the types of strategy that have been proposed to address global error estimation. The Zadunaisky procedure [80] and the related procedure for solving for the correction [72] are arguably the most popular global-error estimation strategies [1, 50]. The work of Dormand et al. [28, 31] relies on this procedure and is extended to a composition of RK methods in [29]. Further extensions are introduced by Makazaga and Murua [57]. Shampine [67] proposes using multiple methods to estimate global errors.

Recent work on global error control by Kulikov [52, 53] extends the quasi-consistency

---

\*This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program under contract DE-AC02-06CH11357, FWP #56706 and #57K87.

<sup>†</sup>Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439 (emconsta@mcs.anl.gov).

property introduced by Skeel [71] and recently advanced by Kulikov [51]. Moreover, these ideas were extended to peer methods by using a peer-triplets strategy by Kulikov and Weiner [54, 55, 78, 79]. These strategies seem to give very good results in terms of global error control on prototypical problems. The general linear (GL)-based algorithm proposed in this study bears a more general representation of the methods discussed above. We will also show how Runge-Kutta triplets [31] and by proxy the peer triplets [78] are naturally represented as GL methods.

Our work builds on similar ideas introduced by Shampine [67] and Zadunaisky [80] and the followups in the sense that the strategy evolves the defect along with the solution; however, in our strategy the internal calculations of the two quantities can be overlapped by using a single scheme to evolve them simultaneously. Therefore, the new method automatically integrates the local truncation error or defect. Previous strategies can be cast as particular cases of the one introduced in this study when the overlapping part is omitted. This leads to new types of schemes that are naturally represented as GL methods, which are perfectly suited for this strategy, as we demonstrate.

The general linear methods introduced in this study have built-in asymptotically correct global and local error estimators. These methods propagate at least two quantities; one of them is the solution, and the other one can be either the global error or equivalently another solution that can be used to determine the global errors. We show that two new elements are required for GL methods to propagate global errors: (i) a particular relation between the truncation error of the two quantities and (ii) a decoupling property between the errors of its two outputs. The GL framework encapsulates all linear time-stepping algorithms and provides a platform for robust algorithms with built-in error estimates. Moreover, this encapsulated treatment simplifies the analysis of compound schemes used for global error analysis; for instance, stability analysis turns out to be much simpler in this representation.

We consider the first-order system of nonautonomous ordinary differential equations

$$(1.1) \quad y(t)' = f(t, y(t)); \quad y(t_0) = y_0, \quad t_0 < t \leq T, \quad y \in \mathbb{R}^m, f : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m,$$

of size  $m$  with  $y_0$  given. We will use the tensor notation denoting the components in (1.1) by  $y^{\{j\}}$  and  $f^{\{j\}}$ ,  $j = 1, 2, \dots, m$ . We will often consider nonautonomous systems because the exposition is less cluttered. In order to convert (1.1) to autonomous form, the system can be augmented with  $(y^{\{m+1\}})' = 1$ , with  $y^{\{m+1\}}(t_0) = t_0$ ; hence,  $t = y^{\{m+1\}}(t)$ . This is likely not a restrictive theoretical assumption, but there can be exceptions [60]; however, in practice it is preferable to treat the temporal components separately. For brevity, we will refer to (1.1) in both autonomous and nonautonomous forms depending on the context.

The purpose of this study is to analyze strategies for estimating the global error at every time step  $n$ ,

$$(1.2) \quad \varepsilon(t_n) = y(t_n) - y_n, \quad n = 1, 2, \dots, T/\Delta t,$$

that is, the difference between the exact solution  $y(t_n)$  and a numerical approximation  $y_n$ . *A priori* and *a posteriori* error bounds under appropriate smoothness assumptions are well known [41, 44]. This study focuses on efficient *a posteriori* estimates of  $\varepsilon(t_n)$ .

This study has several limitations. Stiff differential equations are not directly addressed. Although the theory presented here applies to the stiff case, additional constraints are believed to be necessary to preserve the asymptotic correctness of

the error estimators. Moreover, we do not discuss global error control. Although algorithms introduced in this study work well with variable time steps, we do not address error control strategies here. This is a topic for a future study; however, local error control can be used as before with global error estimates being a diagnostic of the output.

We aim to bring a self-contained view of global error estimation. New results are interlaced with classical theory to provide a contained picture for this topic. We also illustrate the connection among different strategies. The proposed algorithm generalizes all the strategies reviewed in this study and provides a robust instrument for estimating *a posteriori* errors in numerical integration. Section 2 introduces the background for the theoretical developments and discusses different strategies to estimate the global errors, which include developments that form the basis of the proposed approach. In Sec. 3 we discuss the general linear methods that are used to represent practical algorithms. The analysis of these schemes and examples are provided in Sec. 4. In Sec. 5 we discuss the relationship between the approach introduced here and related strategies and show how the latter are particular instantiations of the former. Several numerical experiments are presented in Sec. 6, and concluding remarks are discussed in Sec. 7.

**2. Global error estimation.** Let us consider a one-step linear numerical discretization method for (1.1),

$$(2.1) \quad y_{n+1} = y_n + \Delta t \Phi(t_n, y_n, \Delta t_n), \quad y_0 = y(t_0), \quad n = 1, 2, \dots, T/\Delta t,$$

where  $\Phi$  is called the Taylor increment function with  $\Phi(t_n, y_n, 0) = f(t_n, y(t_n))$ . We denote the time series obtained via (2.1) with step  $\Delta t$  by  $\{y_{\Delta t}\}$ . A method of order  $p$  for a sufficiently smooth function  $f$  satisfies

$$(2.2a) \quad \|y(t_n + \Delta t) - y_{n+1}\| \leq C_1 \Delta t^{p+1}, \quad y_n = y(t_n),$$

for a constant  $C_1$ . The local error then satisfies

$$(2.2b) \quad y(t + \Delta t) - y(t) - \Delta t \Phi(t, y(t), \Delta t) = d_{p+1}(t) \Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}).$$

The following classical result states the bounds on the global errors.

**THEOREM 2.1.** *Let  $U$  be a neighborhood of  $\{(t, y(t)) | t_0 \leq t \leq T\}$ , where  $y(t)$  is the exact solution of (1.1) and there exists a constant  $L$  such that  $\|f(t, y) - f(t, z)\| \leq L\|y - z\|$  and (2.2) is satisfied for  $(t, x), (t, y) \in U$ . Then*

$$(2.3) \quad \|\varepsilon(t)\| \leq \Delta t^p \frac{C_2}{L} \left( e^{L(t-t_0)} - 1 \right)$$

for a constant  $C_2$ .

This is proved in several treatises [41, 44, 73]. Under sufficient smoothness assumptions [40, 44], it follows that the *global error* satisfies

$$(2.4) \quad \varepsilon(t) = y(t) - y_{\Delta t}(t) = e_p(t) \Delta t^p + o(\Delta t^p),$$

where  $y_n := y_{\Delta t}(t)$  at  $t = t_0 + n\Delta t$ . These results are obtained by comparing the expansions of the exact and the numerical solutions. To alleviate the analysis difficulties that come with large  $p$ , we use the B-series representation of the derivatives.

DEFINITION 2.2 (Rooted trees and labeled trees [5,22]). Let  $\mathcal{T}$  be a set of ordered indexes  $\mathcal{T}_q = \{j_1 < j_2 < j_3 < \dots < j_q\}$  with cardinality  $q$ . A labeled tree of order  $q$  is a mapping  $\tau : \mathcal{T}_q \setminus \{j_1\} \rightarrow \mathcal{T}_q$  such that  $\tau(j) < j, \forall j \in \mathcal{T}_q \setminus \{j_1\}$ . The set of all labeled trees of order  $q$  is denoted by  $LT_q$ . The order of a tree is denoted by  $\rho(\tau) = q$ . Furthermore, we define an equivalence class of order  $q$  as the permutation  $\sigma : \mathcal{T}_q \rightarrow \mathcal{T}_q$  such that  $\sigma(j) = j, \tau_k \sigma = \sigma \tau_\ell, \tau_k, \tau_\ell \in LT_q$ . These unlabeled trees of order  $q$  are denoted by  $T_q$ , and the number of different monotonic labelings of  $\tau \in T_q$  is denoted by  $\alpha(\tau)$ . Also,  $T_q^\# = T_q \cup \emptyset$ , where  $\emptyset$  is the empty tree and the only one with  $\rho(\emptyset) = 0$ .

DEFINITION 2.3 (Elementary differentials [5,22]). For a labeled tree  $\tau \in LT_q$  we call an elementary differential the expression

$$(2.5) \quad F^{\{K_1\}}(\tau)(y) = \sum_{K_2, K_3, \dots, K_q} \prod_{i=1}^q f_{\tau^{-1}(K_i)}^{\{K_i\}},$$

where  $K_1, K_2, \dots, K_q = 1, 2, \dots, m$ , and  $f_{K_1, K_2, \dots, K_r}^{\{J\}} = \partial^r f^{\{J\}} / \partial y^{\{K_1\}} \partial y^{\{K_2\}} \dots \partial y^{\{K_r\}}$ . We denote by  $F(\tau)(y) = [F^{\{1\}}(\tau)(y), F^{\{2\}}(\tau)(y), \dots, F^{\{m\}}(\tau)(y)]^T$ . We use the graphical notation to represent derivatives discussed in [14,41].

Example. The tree  corresponds to  $f'f''(f, f)$ . The trees of order 4 are  $T_4 = \left\{ \begin{array}{c} \text{tree 1} \\ \text{tree 2} \\ \text{tree 3} \\ \text{tree 4} \end{array} \right\}$ ,  $\alpha(\tau) = 1$  for  $\tau \in T_4 \setminus \left\{ \begin{array}{c} \text{tree 1} \end{array} \right\}$ ,  $\alpha \left( \begin{array}{c} \text{tree 1} \end{array} \right) = 3$ .

DEFINITION 2.4 (B-series [43]). Let  $a : T \rightarrow \mathbb{R}$  be a mapping between the tree set and real numbers. The following is called a B-series:

$$(2.6) \quad \begin{aligned} B(a, y) &= a(\emptyset)y + \Delta t a(\bullet) f(y) + \frac{\Delta t^2}{2} a(\begin{array}{c} \bullet \\ \bullet \end{array}) F(\begin{array}{c} \bullet \\ \bullet \end{array})(y) + \dots \\ &= \sum_{\tau \in T} \frac{\Delta t^{\rho(\tau)} \alpha(\tau)}{\rho(\tau)!} a(\tau) F(\tau)(y), \end{aligned}$$

where  $T = \{\emptyset\} \cup T_1 \cup T_2 \cup \dots$ .

The exact solution of an ODE system is a B-series [43]. Formally we have the following result.

THEOREM 2.5 (Exact solution as B-series [43]). The exact solution of (1.1) satisfies

$$y^{(q)}(\tau) = \sum_{\tau \in T} \alpha(\tau) F(\tau)(y).$$

Therefore the exact solution is given by (2.6) with  $a(\tau) = 1$ , and the coefficient of  $\Delta t^{\rho(\tau)} F(\tau)(y)$  in the expansion is given by  $\frac{\alpha(\tau)}{\rho(\tau)!}, \forall \tau \in T_k, k = 1, 2, \dots, p$ .

The elementary weights in the expression of the B-series are independent. The following result captures this aspect.

LEMMA 2.6 (Independence of elementary differentials [14]). The elementary differentials are independent. Moreover, the values of the distinct elementary differentials for  $(y^{\{j\}})' = \prod_{j=1}^k (y^{\{j\}})^{m_j} / m_j!, y^{\{j\}}(t_0) = 0$  are given by  $F(\tau_i)(y(t_0)) = e_i$ , where  $k$  is the number of resulting trees when the root is removed and  $m_j$  is the number of copies of  $\tau_j$ .

The order of the numerical method can be defined in terms of a B-series as follows.

DEFINITION 2.7 (Order of time-stepping methods). A numerical method applied to (1.1) with  $f$   $p$ -times continuous differentiable is of order  $p$  if the expansion of the numerical solution satisfies (2.6) with  $\rho(\tau) \leq p$ .

**2.1. Error equation.** We now analyze the propagation of numerical errors through the time-stepping processes.

**THEOREM 2.8** (Asymptotic expansion of the global errors [41,44]). *Suppose that method (2.1) possesses an expansion (2.2b) under smoothness conditions of Theorem 2.1. Then the global error has an asymptotic expansion of form*

$$(2.7) \quad y(t) - y_{\Delta t}(t) = e_p(t)\Delta t^p + \cdots + e_N(t)\Delta t^N + E_{\Delta t}(t)\Delta t^{N+1},$$

where  $E_{\Delta t}(t)$  is bounded on  $t_0 < t \leq T$  and  $0 \leq \Delta t \leq \Delta T$  for some  $\Delta T$ , and  $e_p(t)$  satisfies

$$(2.8) \quad e'_p(t) = \frac{\partial f}{\partial y}(t, y) \cdot e_p(t) + d_{p+1}(t), \quad e_p(t_0) = 0.$$

The other  $e_j(t)$  terms satisfy similar equations.

*Proof.* Consider a perturbed method  $\widehat{y}_{\Delta t}(t) := y_{\Delta t}(t) + e_p(t)\Delta t^p$ . Then  $\widehat{y}_{\Delta t}(t)$  can be represented as the numerical solution of a new method:  $\widehat{y}_{n+1} = \widehat{y}_n + \Delta t \widehat{\Phi}(t_n, \widehat{y}_n, \Delta t)$ . By comparison with (2.1) we obtain  $\{\widehat{y}_n\}$ ,

$$(2.9) \quad \widehat{\Phi}(t, \widehat{y}_n, \Delta t) = \Phi(t_n, \widehat{y}_n - e_p(t_n)\Delta t^p, \Delta t) + (e_p(t_n + \Delta t) - e_p(t_n))\Delta t^{p-1}.$$

Expanding the local error of the perturbed method with the Taylor function defined by (2.9) yields in general

$$(2.10) \quad \begin{aligned} & y(t + \Delta t) - y(t) - \Delta t \widehat{\Phi}(t, y(t), \Delta t) \\ &= \left( d_{p+1}(t) + \frac{\partial f}{\partial y}(t, y) e_p(t) - e'_p(t) \right) \Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}). \end{aligned}$$

We take  $e_p(t)$  to satisfy (2.8), so that by Theorem 2.1 it follows that

$$(2.11) \quad y(t) - y_{\Delta t}(t) = e_p(t)\Delta t^p + \mathcal{O}(\Delta t^{p+1})$$

determines the asymptotic expansion. For more details see [41].  $\square$

Equations for the next terms in the global error expansion can be obtained by using the same procedure; however, this is not pursued in this study.

**2.1.1. Estimating global errors using two methods.** We now introduce the general global error estimation strategy used in this study. This approach relies on propagating two solutions through a linear time-stepping process that has the property of maintaining a fixed ratio between the truncation error terms. This result will play a crucial role in constructing the new methods discussed in Sec. 4 and can be stated as follows.

**THEOREM 2.9** (Global error estimation with two methods). *Consider numerical solutions  $\{y_n\}$  and  $\{\tilde{y}_n\}$  of (1.1) obtained by two time-stepping methods started from the same exact initial condition under the conditions of Theorem 2.8. If the local errors of the two methods with increments  $\Phi$  and  $\tilde{\Phi}$  satisfy*

$$(2.12a) \quad y(t + \Delta t) - y(t) - \Delta t \Phi(t, y(t), \Delta t) = d_{p+1}(t_n)\Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}),$$

$$(2.12b) \quad y(t + \Delta t) - y(t) - \Delta t \tilde{\Phi}(t, y(t), \Delta t) = \gamma d_{p+1}(t_n)\Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}),$$

where  $d_{p+1}(t_n) = \frac{1}{(p+1)!} \sum_{\tau \in \mathcal{T}_{p+1}} \alpha(\tau) a(\tau) F(\tau)(y_n)$  with constant  $\gamma \neq 1$ , then the global error can be estimated as

$$(2.13) \quad \varepsilon_p(t_n) = \frac{1}{1 - \gamma} (\tilde{y}_n - y_n) = e_p(t_n)\Delta t^p + \mathcal{O}(\Delta t^{p+1}) = \varepsilon(t_n) + \mathcal{O}(\Delta t^{p+1}),$$

when  $y_0 = \tilde{y}_0 = y(t_0)$ ; hence,  $\varepsilon_n \asymp y(t_n) - y_n$ .

*Proof.* Use (2.7) and (2.8) to write the global error equations for the two methods with nearby solutions:

$$(2.14a) \quad e'_p(t) = \frac{\partial f}{\partial y}(t, y) \cdot e_p(t) + d_{p+1}(t), \quad e_p(t_0) = 0,$$

$$(2.14b) \quad \tilde{e}'_p(t) = \frac{\partial f}{\partial y}(t, y) \cdot \tilde{e}_p(t) + \gamma d_{p+1}(t), \quad \tilde{e}_p(t_0) = 0.$$

It follows that the solutions of the two ordinary differential equations satisfy  $\gamma e_p(t) = \tilde{e}_p(t)$ . We can then verify (2.13) by inserting (2.11):

$$\begin{aligned} \varepsilon_p(t_n) &= \frac{1}{1-\gamma} (\tilde{y}(t_n) - y(t_n)) \\ &= \frac{1}{1-\gamma} (y(t) - \tilde{e}_p(t)\Delta t^p - y(t) + e_p(t)\Delta t^p + \mathcal{O}(\Delta t^{p+1})) \\ &= e_p(t)\Delta t^p + \mathcal{O}(\Delta t^{p+1}) \end{aligned}$$

for  $n = 1, 2, \dots$   $\square$

A particular case is  $\gamma = 0$ . Moreover, under the assumptions of Theorem 2.9, one can always compute a higher-order approximation by combining the two solutions.

**COROLLARY 2.10.** *If  $\gamma = 0$  in Theorem 2.9, then we revert to the case of using two methods of different orders,  $p$  and  $p + 1$ , to estimate the global errors for the method of order  $p$ .*

**COROLLARY 2.11.** *A method of order  $p + 1$  can be obtained with conditions of Theorem 2.9 by*

$$(2.15) \quad \hat{y}_n = y_n + \varepsilon_p = \frac{1}{1-\gamma} \tilde{y}_n - \frac{\gamma}{1-\gamma} y_n.$$

We note that a related analysis has been carried out in [67] with an emphasis of reusing standard codes for solving ODEs with global error estimation.

The result presented above is the basis for the developments in this study. We introduce a new type of methods that provide *a posteriori* error estimates, and we show that this procedure generalizes all strategies that compute global errors by propagating multiple solutions or integrating related problems. The validity of this approach when variable time steps are used is discussed next.

**2.1.2. Global errors with variable time steps.** Following [41], for variable time stepping we consider  $t_{n+1} - t_n = \nu(t_n)\Delta t$ ,  $n = 1, 2, \dots$ .

Then the local error expansion (2.2b) becomes

$$\begin{aligned} y(t + \nu(t)\Delta t) - y(t) - \nu(t)\Delta t \Phi(t, y(t), \Delta t) = \\ d_{p+1}(t)\nu(t)^{p+1}\Delta t^{p+1} + \dots + d_{N+1}(t)\nu(t)^{N+1}\Delta t^{N+1} + \mathcal{O}(\Delta t^{N+2}), \end{aligned}$$

and instead of (2.9) we obtain

$$\hat{\Phi}(t, \hat{y}_n, \nu(t)\Delta t) = \Phi(t, \hat{y}_n - e_p(t)\Delta t^p, \nu(t)\Delta t) + (e_p(t + \nu(t)\Delta t) - e_p(t)) \frac{\Delta t^p}{\nu(t)\Delta t}.$$

Then (2.10) becomes

$$\begin{aligned} & y(t + \nu(t)\Delta t) - y(t) - \nu(t)\Delta t \widehat{\Phi}(t, y(t), \nu(t)\Delta t) \\ &= \nu(t) \left( d_{p+1}(t)\nu(t)^p + \frac{\partial f}{\partial y}(t, y)e_p(t) - e'_p(t) \right) \Delta t^{p+1} + \mathcal{O}(\Delta t^{p+2}). \end{aligned}$$

Instead of (2.8), the global error  $e_p(t)$  satisfies the following equation:

$$(2.16) \quad e'_p(t) = \frac{\partial f}{\partial y}(t, y) \cdot e_p(t) + \nu(t)^p d_{p+1}(t), \quad e_p(0) = 0.$$

The results introduced in this study and summarized by Theorem 2.9 carry over to variable time stepping with  $\Delta t$  replaced by  $\Delta t_{\max} = \max(\nu(t)\Delta t)$  and, therefore, allow the application of such strategies in practical contexts.

In this study we do not address the problem of time-step adaptivity based on global error estimates. In practice, the adaptivity can be based on asymptotically correct local error estimates that are provided directly by the methods proposed here.

**2.1.3. Methods satisfying the exact principal error equation.** We next review a class of methods used for global error estimation. Consider an asymptotic error expansion in (2.8) of

$$(2.17) \quad e(t) = \sum_{\tau \in T_p} \alpha(\tau) a(\tau) F(\tau)(y(t)), \quad t > t_0, \quad \text{and} \quad e(t_0) = \sum_{\tau \in T_p} \alpha(\tau) a(\tau) F(\tau)(y(t_0)),$$

for some constant  $a(\tau)$ . By inserting (2.17) in (2.8) we obtain

$$(2.18) \quad \begin{aligned} d_{p+1}(t) &= \frac{d}{dt} \left[ \sum_{\tau \in T_p} \alpha(\tau) \mathbf{e}(\tau) F(\tau)(y(t)) \right] - \frac{\partial f}{\partial y}(y(t)) \cdot \sum_{\tau \in T_p} \alpha(\tau) \mathbf{e}(\tau) F(\tau)(y(t)) \\ &= \sum_{\tau \in T_p} \alpha(\tau) \mathbf{e}(\tau) \left[ \frac{d}{dt} F(\tau)(y(t)) - \frac{\partial f}{\partial y}(y(t)) F(\tau)(y(t)) \right]. \end{aligned}$$

This expression implies that if the local error satisfies (2.18), then (2.17) is the exact solution of (2.8), and therefore the global errors can be estimated directly, as described below.

This strategy was indirectly introduced by Butcher [8] in an attempt to break the order barriers of multistage methods under the alias “effective order.” Stetter [74] observed the relationship between (2.18) and the global error (2.8). This strategy requires a starting procedure  $\mathbb{S}$  to enforce  $e(t_0)$ , a method  $\mathbb{M}$  that satisfies (2.18), and a finalizing procedure  $\mathbb{F}$  to extract the global error. We denote by  $\mathbb{S}(\circ)$ ,  $\mathbb{M}(\circ)$ ,  $\mathbb{F}(\circ)$  the application of each method on solution  $\circ$ . Stetter [74] found that  $\mathbb{S}$  and  $\mathbb{F}$  can be one order less than  $\mathbb{M}$ . Examples of such triplets can be found in many studies [8, 59, 62, 64–66, 74].

Algorithm [A:ExPrErEq]: Methods with exact principal error equation [74]

Solve

$$(2.19a) \quad y_1 = \mathbb{S}(y_0), \quad y(t_0) = y_0$$

$$(2.19b) \quad \begin{cases} y_n = \mathbb{M}(y_{n-1}) \\ \varepsilon_n = y_n - \mathbb{F}(y_{n-1}) \end{cases} \quad n = 2, 3, \dots, \quad \text{so that (2.18).}$$

One such scheme is provided in Appendix C. However, a caveat is that methods based on explicit Runge-Kutta schemes require as many nonzero stage coefficients as the order of the method because  $\mathbb{M}$  needs to have a nonzero tall tree of  $p + 1$ , hence, the effective order is limited by  $p \leq s$ . For instance, an order 5 method requires at least five stages. This requirement comes from the fact that tall trees need to be nonzero in (2.18). However, this strategy is still effective for high orders. Recently the effective order was discussed in [10, 15–18, 39]. Effective order through method composition has recently been discussed in [21].

Although this concept is attractive in terms of efficiency, Prince and Wright [62] noted a severe problem with using it for global error estimation: If the system has unstable components, then the error approximation becomes unreliable, as can be seen in Fig. 6.4 discussed later. This is a severe limitation because having unstable components makes the local error estimates unreliable, and this is precisely the case when one would need to use global error estimation.

**2.2. Differential correction.** The differential correction techniques for global error estimation are based on the work of Zadunaisky [80] and Skeel [72]. The discussion of these procedures is deferred to Sec. 2.2.3.

**2.2.1. Error equation and the defect.** We follow the exposition in [56, 80] and assume that there exists a solution  $z(t)$  of a perturbed system

$$(2.20) \quad z(t)' = f(t, z(t)) - r(t); \quad z(t_0) = z_0, \quad r(0) = y_0 - z_0, \quad t_0 < t \leq T,$$

close to  $y(t)$ . The error function (between the solutions of (1.1) and (2.20)) is given by [56]

$$(2.21) \quad e(t) = y(t) - z(t),$$

$$(2.22) \quad e'(t) = A(t)e(t) - r(t), \quad A = \int_0^1 f'(t, y(t) + se(t)) ds.$$

If  $e(t_0) = 0$  and we approximate  $A(t) = \frac{\partial f}{\partial t}(t, y) + \mathcal{O}(e(t))$  in (2.22), then we obtain

$$(2.23) \quad e'(t) = \frac{\partial f}{\partial t}(t, y)e(t) - r(t), \quad e(t_0) = 0, \quad t_0 < t \leq T,$$

with  $r(t) = -d_{p+1}(t)\Delta t^p$ . This is asymptotically equivalent to solving the first variation (leading term) of the global error equation for  $e_p$ ; i.e., (2.8). Consider now that the nearby solution,  $z(t)$ , is obtained through an interpolatory function  $P(t)$ , and define the defect  $D(t)$  as

$$(2.24) \quad D(t) = f(t, P(t)) - P'(t).$$

Estimates of the local truncation errors can be obtained by using continuous output [33]. Lang and Verwer [56] showed that if  $P(t)$  is obtained through Hermite interpolation, then

$$D(t) = [y'(t) - f(t, y(t))] - [f(t, P(t)) - P'(t)] = \mathcal{O}(\Delta t^3), \quad t \in (t_n, t_{n+1}),$$

and in particular  $D(t_n + \frac{\Delta t}{2}) = \mathcal{O}(\Delta t^4)$ . Furthermore, a relation between the defect at  $t_n + \frac{\Delta t}{2}$  and the leading term of the local truncation error,  $D(t_{n+\frac{1}{2}}) = \frac{3}{2}d_{p+1}(t_n)\Delta t + \mathcal{O}(\Delta t^{p+1})$ ,  $1 \leq p \leq 3$ , can be obtained. We can then set  $r(t) = \frac{2}{3}D(t_{n+\frac{1}{2}})$ ,  $t \in (t_n, t_{n+1})$ , and (2.8) and (2.23) become

$$(2.25) \quad e'(t) = f'(t_n, y_n)e(t) - r(t_{n+\frac{1}{2}}), \quad e(t_0) = 0, \quad t_n < t \leq t_{n+1}, \quad n = 0, 1, \dots, N.$$

**2.2.2. Solving the error equation.** If the Jacobian of  $f$  is available, then (2.25) can be solved directly as in [56].

Algorithm [A:SoErEq]: Solving the error equation [56]	
Solve	
(2.26a) $y' = f(t, y),$	$y(t_0) = y_0$
(2.26b) $\varepsilon'(t) = J\varepsilon(t) + [d_{p+1}(t_n)\Delta t],$	$\varepsilon(t_0) = 0$
$d_{p+1}(t_n) = y(t_{n+1}) - y_{n+1} + \mathcal{O}(\Delta t^{p+2}), J = \frac{\partial f}{\partial y}.$	

The authors of [56] argue that (2.26b) can be solved with a cheaper, lower-order method. In this case, however, the bulk of the work resides on determining  $d_{p+1}$ , which can be estimated by following the steps discussed in the previous section.

**2.2.3. Solving for the correction.** This approach follows the developments presented in [61, 72, 80] and further refined in [28–31]. We start from (2.20) and denote by  $P(t)$  its exact solution. Equation (2.21) becomes

$$(2.27a) \quad e(t) = y(t) - P(t), \text{ and}$$

$$(2.27b) \quad e'(t) = (y(t) - P(t))' = f(t, y(t)) - P'(t) = f(t, P(t) + e(t)) - P'(t).$$

We can see the connection between (2.27b) and (2.8) by starting with (2.8):

$$\begin{aligned} e'(t) &= f'(t, y(t))e(t) + D(t) = f(t, y(t)) - f(t, y(t) - e(t)) - f(t, P(t)) - P'(t) \\ &= f(t, y(t)) - P'(t) = f(t, P(t) + e(t)) - P'(t), \end{aligned}$$

where we neglected the higher-order terms and used (2.24) and (2.27a). The equations to be solved are known as the procedure for solving for the correction [72].

Algorithm [A:SolCor]: Solving for the correction [72]	
Solve	
(2.28a) $y' = f(t, y),$	$y(t_0) = y_0$
(2.28b) $\varepsilon' = f(t, P(t) + \varepsilon) - P'(t),$	$\varepsilon(t_0) = 0$
$P(t) \approx y(t) - y_{\Delta t}(t).$	

We will show that equations (2.28) (in [A:SolCor]) can be solved by using a general linear method representation (5.1) described in Sec. 5.1.

The related Zadunaisky procedure [80] is as follows. Calculate the polynomial of order  $p$ ,  $P(t)$ , by using Lagrange interpolation  $\mathcal{L}_p(y_{\Delta t}(t))$  over several steps and then apply a similar procedure as in (2.28) on a perturbed system.

Algorithm [A:ZaPr]: Zadunaisky procedure [80]	
Solve	
(2.29a) $y' = f(t, y),$	$y(t_0) = y_0$
(2.29b) $z' = f(t, P(t)) - P'(t) - f(t, z),$	$z(t_0) = y_0$
$\varepsilon_n = z_n - y_n$	$P(t) = \mathcal{L}_p(y_{\Delta t}(t)).$

**2.3. Extrapolation approach.** The global error estimation through extrapolation dates back to [63]. The procedure is the following. Propagate two solutions  $y_{\Delta t, n}$  and  $y_{\frac{\Delta t}{2}, n}$ , one with  $\Delta t$  and one with  $\Delta t/2$ , each with global errors  $\varepsilon_{\Delta t, n} = y(t_n) - y_{\Delta t, n}$ ,  $\varepsilon_{\frac{\Delta t}{2}, n} = y(t_n) - y_{\frac{\Delta t}{2}, n}$ , respectively.

Then it follows that by using a method of order  $p$  one obtains [44]

$$\begin{aligned}\varepsilon_{\Delta t,n} &= y(t_n) - y_{\Delta t,n} = e_p \Delta t^p + \mathcal{O}(\Delta t^{p+1}), \\ \varepsilon_{\frac{\Delta t}{2},n} &= y(t_n) - y_{\frac{\Delta t}{2},n} = e_p \left(\frac{\Delta t}{2}\right)^p + \mathcal{O}(\Delta t^{p+1}).\end{aligned}$$

The global error and a solution of one order higher can be obtained as

$$(2.30a) \quad \varepsilon_{\Delta t,n} = \frac{2^p}{1-2^p} (y_{\Delta t,n} - y_{\frac{\Delta t}{2},n}) + \mathcal{O}(\Delta t^{p+1}),$$

$$(2.30b) \quad \hat{y}_{\Delta t,n} = y_{\Delta t,n} + \varepsilon_{\Delta t,n} = \frac{1}{1-2^p} y_{\Delta t,n} - \frac{2^p}{1-2^p} y_{\frac{\Delta t}{2},n} = y(t_n) + \mathcal{O}(\Delta t^{p+1}).$$

These statements are a particular instantiation of (2.13) and (2.15) with  $\gamma = 1/2^p$ .

Algorithm [A:Ex]: Extrapolation  
Solve  $y' = f(t, y)$  by using a method of order  $p$  with two time steps  $\Delta t$  and  $\Delta t/2$

$$(2.31a) \quad y' = f(t, y) \Rightarrow y_{\Delta t,n}, y_{\frac{\Delta t}{2},n}, \quad y(t_0) = y_0$$

$$(2.31b) \quad \varepsilon = \frac{2^p}{1-2^p} (y_{\Delta t,n} - y_{\frac{\Delta t}{2},n}).$$

**2.4. Underlying higher-order method.** All the methods described in this study attempt to use an underlying higher-order method to estimate the global error. In the case of [A:ExPrErEq] the exact principal error algorithm (2.19) and of [A:SoErEq] solving the error equation (2.26), we find that the actual equation being solved is modified to include the truncation error term. By adding (2.26a) and (2.26b) one obtains

$$\begin{aligned}y' + \varepsilon' &= \hat{y}' = f(y) + J\varepsilon + D(y) \\ \hat{y}' &= f(\hat{y} - \varepsilon) + J\varepsilon + D(y) \\ \hat{y}' &= f(\hat{y}) + D(y).\end{aligned}$$

In the case of the Zadunaisky algorithm [A:SolCor] (2.28), one can recover the underlying higher-order method by replacing the error term in (2.28b) with  $\hat{y}$  from (2.15) and using the conditions imposed on  $P$  (see [28]). We show an example in Sec. 5.1. The extrapolation algorithm [A:Ex] (2.31) reveals the higher-order estimate directly in (2.30b).

**3. General linear methods.** The methods introduced in this study are represented by GL schemes. In this context, we take advantage of the existing theory underlying GL methods and augment it with global error estimation capabilities. Two key elements are required. The first is a consequence of Theorem 2.9, which imposes a restriction on the truncation error. The second has to do with restricting the interaction between the two outputs. Both will be addressed in Sec. 4. The advantage of representing existing global error strategies in a more general framework is that it allows for the development of more robust methods. For this we need GL methods that carry at least two quantities as discussed above. In this section we present the GL methods theory without built-in error estimates.

General linear methods were introduced by Burrage and Butcher [4]; however, many GL-type schemes have been proposed to extend either Runge-Kutta methods [38] to linear multistep (LM) or vice versa [6, 36], as well as other extensions [7, 26,

42, 71]. GL methods are thus a generalization of both RK and LM methods, and we use the GL formalism to introduce new methods that provide asymptotically correct global error estimates.

Denote the solution at the current step  $(n-1)$  by an  $r$ -component vector  $\mathbf{y}^{[n-1]} = [\mathbf{y}_{(1)}^{[n-1]} \ \mathbf{y}_{(2)}^{[n-1]} \ \dots \ \mathbf{y}_{(r)}^{[n-1]}]^T$ , which contains the available information in the form of numerical approximations to the ODE (1.1) solutions and their derivatives at different time indexes. To increase clarity, we henceforth denote the time index inside square brackets. The stage values (at step  $n$ ) are denoted by  $\mathbf{Y}_{(i)}$  and stage derivatives by  $\mathbf{f}_{(i)} = f(\mathbf{Y}_{(i)})$ ,  $i = 1, 2, \dots, s$ , and can be compactly represented as  $\mathbf{Y} = [\mathbf{Y}_{(1)}^T \ \mathbf{Y}_{(2)}^T \ \dots \ \mathbf{Y}_{(s)}^T]^T$  and  $\mathbf{f} = [\mathbf{f}_{(1)}^T \ \mathbf{f}_{(2)}^T \ \dots \ \mathbf{f}_{(s)}^T]^T$ .

The  $r$ -value  $s$ -stage GL method is described by

$$(3.1) \quad \begin{aligned} \mathbf{Y}_{(i)} &= \Delta t \sum_{j=1}^s \mathbf{A}_{ij} \mathbf{f}_{(j)} + \sum_{j=1}^r \mathbf{U}_{ij} \mathbf{y}_{(j)}^{[n-1]}, \quad i = 1, 2, \dots, s, \\ \mathbf{y}_{(i)}^{[n]} &= \Delta t \sum_{j=1}^s \mathbf{B}_{ij} \mathbf{f}_{(j)} + \sum_{j=1}^r \mathbf{V}_{ij} \mathbf{y}_{(j)}^{[n-1]}, \quad i = 1, 2, \dots, r, \end{aligned}$$

where  $(\mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$  are the coefficients that define each method and can be grouped further into the GL matrix  $\mathbb{M}$ :

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{y}^{[n]} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \otimes I_m & \mathbf{U} \otimes I_m \\ \mathbf{B} \otimes I_m & \mathbf{V} \otimes I_m \end{bmatrix} \begin{bmatrix} \Delta t \mathbf{f} \\ \mathbf{y}^{[n-1]} \end{bmatrix} = \mathbb{M} \begin{bmatrix} \Delta t \mathbf{f} \\ \mathbf{y}^{[n-1]} \end{bmatrix}.$$

Expression (3.1) is the most generic representation of GL methods [41, p. 434] and encompasses both RK methods ( $r = 1, s > 1$ ) and LM methods ( $r > 1, s = 1$ ) as particular cases. In this work we consider methods with  $r = 2$ , where the first component represents the primary solution of the problem (2.12a) and the second component can represent either the defect (2.13) or the secondary component (2.12b). Only multistage-like methods are considered; however, multistep-multistage methods ( $r > 2$ ) are also possible.

If method (3.1) is consistent (there exist vectors  $q_0, q_1$  such that  $\mathbf{V}q_0 = q_0, \mathbf{U}q_0 = \mathbb{1}$ , and  $\mathbf{B}\mathbb{1} + \mathbf{V}q_1 = q_0 + q_1$  [13, Def. 3.2 and 3.3]) and stable ( $\|\mathbf{V}^n\|$  remains bounded,  $\forall n = 1, 2, \dots$  [13, Def. 3.1]), then the method (3.1) is convergent [13, Thm. 3.5], [14, 49]. In-depth descriptions and survey materials on GL methods can be found in [11, 13, 14, 41, 49]. In this study we use self-starting methods, and therefore  $\mathbb{S} = I$ . In general the initial input vector  $\mathbf{y}^{[0]}$  can be generated through a ‘‘starting procedure,’’  $\mathbb{S} = \{S_i : \mathbb{R}^m \rightarrow \mathbb{R}^m\}_{i=1 \dots r}$ , represented by generalized RK methods; see [14, Chap. 53] and [25]. The final solution is typically obtained by applying a ‘‘finishing procedure,’’  $\mathbb{F} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ , to the last output vector; in our case this is also the identity. We denote by the GL process the GL method applied  $n$  times and described by  $\mathbb{S}\mathbb{M}^n\mathbb{F}$ ; that is,  $\mathbb{M}$  is applied  $n$  times on the vector provided by  $\mathbb{S}$ , and then  $\mathbb{F}$  is used to extract the final solution.

**3.1. Order conditions for GL methods.** The order conditions rely on the theory outlined by Butcher et al. [14, 24, 25]. The derivatives of the numerical and exact solution are represented by rooted trees and expressed as a B-series [9, 43] as delineated in Theorem 2.5 and order definition 2.7. We use an algebraic criterion to characterize the order conditions for GL methods as follows. Let  $\tau \in T$  and  $E^{(\theta)} : T \rightarrow \mathbb{R}$ , the ‘‘exact solution operator’’ of differential equation (1.1), which

represents the *elementary weights for the exact solution* at  $\theta\Delta t$ . If  $\theta = 1$ , then  $E^{(1)}(\tau) = E(\tau) = \rho(\tau)/(\sigma(\tau)\alpha(\tau))$ . The order can be analyzed algebraically by introducing a mapping  $\xi_i : T \rightarrow \mathbb{R}$ :  $\xi_i(\emptyset) = b_0^{(i)}$ ,  $\xi_i(\tau) = \Phi^{(i)}(\tau)$ , where  $\Phi^{(i)}(\tau)$ ,  $i = 1, \dots, r$ , results from the starting procedure and  $\emptyset$  represents the “empty tree.” Then for the general linear method  $(\mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$ , one has

$$(3.2) \quad \eta(\tau) = \mathbf{A}\eta D(\tau) + \mathbf{U}\xi(\tau), \quad \widehat{\xi}(\tau) = \mathbf{B}\eta D(\tau) + \mathbf{U}\xi(\tau), \quad \tau \in T,$$

where  $\eta, \eta D$  are mappings from  $T$  to scalars that correspond to the internal stages and stage derivatives and  $\widehat{\xi}$  represents the output vector. The exact weights are obtained from  $[E\xi](\tau)$ . The order of the GL method can be determined by a direct comparison between  $\widehat{\xi}(\tau)$  and  $[E\xi](\tau)$ . More details can be found in [14], where a criterion for order  $p$  is given for a GL method described by  $\mathbb{M}$  and  $\mathbb{S}$ . The criterion is simplified if  $\mathbb{S} = \mathbb{F} = I$  as discussed in [24]. Therefore, in general, an order  $p$  GL method results from the direct comparison of elementary wights of  $[\mathbb{M}^n](\tau_j) = [E^n\xi](\tau_j) \forall \tau_j, \rho(\tau_j) \leq p$ . This criterion is a direct consequence of [24, Def. 3 and Prop. 1]. In our particular case, methods satisfying Theorem 2.9 can be developed by enforcing (2.12) on the corresponding solution vector.

**3.2. Linear stability of GL methods.** The linear stability analysis of method (3.1) is performed on a linear scalar test problem:  $y'(t) = ay(t)$ ,  $a \in \mathbb{C}$ . Applying (3.1) to the test problem yields a solution of form  $y^{[n+1]} = R(z)y^{[n]}$ ,

$$(3.3) \quad R(z) = \mathbf{V} + z\mathbf{B}(I_s - z\mathbf{A})^{-1}\mathbf{U},$$

$$(3.4) \quad \Phi(w, z) = \det(wI_r - R(z)),$$

where  $z = a\Delta t$ ,  $R(z)$  is referred to as the stability matrix of the scheme, and  $\Phi(w, z)$  is the stability function.

For given  $z$ , method (3.1) is linearly stable if the spectral radius of  $R(z)$  is contained by the complex unit disk. The stability region is defined as the set  $\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \leq 1\}$ . The linear stability region provides valuable insight into the method’s behavior with nonlinear systems. Additional details can be found in [14].

**4. Methods with global error estimation (GEE).** We now introduce GL methods with global and local error estimation. We focus on Runge-Kutta-like schemes in the sense that the resulting GL methods are self-starting multistage schemes. We therefore restrict our exposition to methods that carry two solutions explicitly and where  $r = 2$ . Generalizations are possible but not addressed here. The methods are given in two forms that use different input and output quantities. The first form used for numerical analysis results in a scheme denoted by GLy $\tilde{y}$  that evolves two solutions of the ODE problem  $y$  and  $\tilde{y}$ . Methods GLy $\tilde{y}$  take the following form:

$$(4.1) \quad \begin{aligned} Y_{(i)} &= \Delta t \sum_{j=1}^s \mathbf{A}_{ij} f(Y_{(j)}) + \mathbf{U}_{i,1} y_{(1)}^{[n-1]} + \mathbf{U}_{i,2} y_{(2)}^{[n-1]}, \quad i = 1, 2, \dots, s, \\ y_{(1)}^{[n]} &= \Delta t \sum_{j=1}^s \mathbf{B}_{i,1} f(Y_{(j)}) + \mathbf{V}_{1,1} y_{(1)}^{[n-1]} + \mathbf{V}_{1,2} y_{(2)}^{[n-1]}, \\ y_{(2)}^{[n]} &= \Delta t \sum_{j=1}^s \mathbf{B}_{2,j} f(Y_{(j)}) + \mathbf{V}_{2,1} y_{(1)}^{[n-1]} + \mathbf{V}_{2,2} y_{(2)}^{[n-1]}. \end{aligned}$$

We will consider  $V = I_r$ , although more general forms can also be considered. The second form, denoted by  $\text{GLy}\varepsilon$ , is given as a method that evolves the solution of the base method and the error explicitly,  $y$  and  $\varepsilon$ , as  $\{y^{[n]}, \varepsilon^{[n]}\} = \text{GLy}\varepsilon(\{y^{[n-1]}, \varepsilon^{[n-1]}\})$ , and has a more practical flavor. Both forms can be expressed as GL methods with tableaux  $(\mathbf{A}_{y\tilde{y}}, \mathbf{U}_{y\tilde{y}}, \mathbf{B}_{y\tilde{y}}, \mathbf{V}_{y\tilde{y}})$  and  $(\mathbf{A}_{y\varepsilon}, \mathbf{U}_{y\varepsilon}, \mathbf{B}_{y\varepsilon}, \mathbf{V}_{y\varepsilon})$ , respectively; and one can switch between the forms as explained below.

LEMMA 4.1. *GL methods of form (4.1) that satisfy the conditions of Theorem 2.9 with coefficients  $(\mathbf{A}_{y\tilde{y}}, \mathbf{U}_{y\tilde{y}}, \mathbf{B}_{y\tilde{y}}, \mathbf{V}_{y\tilde{y}})$ , where  $y^{[n]} = [(y^{[n]})^T, (\tilde{y}^{[n]})^T]^T$ , and  $(\mathbf{A}_{y\varepsilon}, \mathbf{U}_{y\varepsilon}, \mathbf{B}_{y\varepsilon}, \mathbf{V}_{y\varepsilon})$ , where  $y^{[n]} = [(y^{[n]})^T, (\varepsilon^{[n]})^T]^T$ , are related by*

$$(4.2) \quad \mathbf{A}_{y\tilde{y}} = \mathbf{A}_{y\varepsilon}, \quad \mathbf{V}_{y\tilde{y}} = \mathbf{V}_{y\varepsilon}, \quad \mathbf{U}_{y\tilde{y}} = \mathbf{U}_{y\varepsilon} T_{y\varepsilon}^{-1}, \quad \mathbf{B}_{y\tilde{y}}(1, \cdot) = T_{y\varepsilon} \mathbf{B}_{y\varepsilon},$$

$$\text{where } T_{y\varepsilon} = \begin{bmatrix} 1 & 0 \\ 1 & 1 - \gamma \end{bmatrix}.$$

*Proof.* We start with a  $\text{GLy}\varepsilon$  method defined by  $(\mathbf{A}_{y\varepsilon}, \mathbf{U}_{y\varepsilon}, \mathbf{B}_{y\varepsilon}, \mathbf{V}_{y\varepsilon})$  and write the resulting expression by applying (4.1) with  $y_{(1)}^{[n]} = y^{[n]}$  and  $y_{(2)}^{[n]} = \varepsilon^{[n]}$ . We then replace  $\varepsilon^{[n]}$  with  $\frac{1}{1-\gamma}(\tilde{y}^{[n]} - y^{[n]})$  as in Theorem 2.9, (2.13). The resulting expression can then be written as a  $\text{GLy}\tilde{y}$  scheme with  $y_{(1)}^{[n]} = y^{[n]}$  and  $y_{(2)}^{[n]} = \tilde{y}^{[n]}$ . This calculation leads to (4.2). This transformation is unique as long as  $\gamma \neq 1$ .  $\square$

The following algorithm is proposed.

Algorithm [A:GLMGEE]: General linear methods with global error estimation	
Initialize: $y^{[0]} = y(t_0) = y_0$ , $\varepsilon^{[0]} = \varepsilon(t_0) = 0$ .	
Solve: $y' = f(t, y)$ using	
(4.3a)	$\{y^{[n]}, \varepsilon^{[n]}\} = \text{GLy}\varepsilon(\{y^{[n-1]}, \varepsilon^{[n-1]}\})$ , <span style="float: right;">[solution, GEE]</span>
(4.3b)	$\varepsilon_{\text{loc}} = \varepsilon^{[n]} - \varepsilon^{[n-1]}$ , <span style="float: right;">[local error]</span>
(4.3c)	$\hat{y}^{[n]} = y^{[n]} + \varepsilon^{[n]} = \frac{1}{1-\gamma}\tilde{y}^{[n]} - \frac{\gamma}{1-\gamma}y^{[n]}$ . <span style="float: right;">[high order]</span>

**4.1. Consistency and preconsistency analysis.** We now discuss consistency and preconsistency conditions in the case of a method with  $r = 2$ . Following [49], we require that

$$(4.4a) \quad y_i^{[n-1]} = q_{i,0}y(t_{n-1}) + \Delta t q_{i,1}y'(t_{n-1}) + \mathcal{O}(\Delta t^2), \quad i = 1, 2$$

$$(4.4b) \quad Y_i = y(t_{n-1} + c_i \Delta t) + \mathcal{O}(\Delta t^2), \quad i = 1, 2, \dots, s$$

$$(4.4c) \quad y_i^{[n]} = q_{i,0}y(t_n) + \Delta t q_{i,1}y'(t_n) + \mathcal{O}(\Delta t^2), \quad i = 1, 2.$$

From (4.4b) we obtain

$$\begin{aligned} y(t_{n-1}) &= (u_{i,1}q_{1,0} + u_{i,2}q_{2,0})y(t_{n-1}) \\ &+ c_i \Delta t y'(t_{n-1}) + \Delta t (u_{i,1}q_{1,1} + u_{i,2}q_{2,1})y'(t_{n-1}) \\ &+ \Delta t \sum_j a_{i,j}y'(t_{n-1}) + \mathcal{O}(\Delta t^2), \quad i = 1, 2, \dots, s, \end{aligned}$$

and therefore  $Uq_0 = 1$  and  $c_i = \sum_j a_{i,j} + Uq_1$ . We next combine (4.4a) and (4.4c):

$$\begin{aligned} q_{i,0}(y(t_{n-1}) + \Delta t y'(t_{n-1})) + \Delta t q_{i,1}y'(t_{n-1}) &= q_{i,0}y(t_{n-1}) + \Delta t q_{i,1}y'(t_{n-1}) \\ &+ \Delta t \sum_j b_{1,j}y(t_{n-1}) + \mathcal{O}(\Delta t^2), \end{aligned}$$

where we have considered that  $\mathbf{V} = I$ . The consistency condition  $\mathbf{B}\mathbb{1} = q_0$  follows.

**4.2. Order conditions for GEE methods.** The order conditions are based on the algebraic representation of the propagation of the B-series through the GL process as discussed in Sec. 3.1. Additional constraints are imposed so that Theorem 2.9 applies directly as a result of the GL process. To this end, we consider an order  $p$  GLy $\tilde{y}$  method by setting  $E\xi_1(\tau) = \hat{\xi}_1(\tau) = \hat{\xi}_2(\tau)$ , for all  $\tau \in T_p$ , and

$$(4.5a) \quad \gamma \left( E\xi_1(\tau) - \hat{\xi}_1(\tau) \right) = E\xi_2(\tau) - \hat{\xi}_2(\tau), \quad \tau \in T_{p+1}, \gamma \neq 1,$$

assuming that the inputs of the GL process  $\gamma\xi_1(\tau) = \xi_2(\tau)$ ,  $\tau \in T_{p+1}$ . Here  $\hat{\xi}$  represents the numerical output, and  $E\xi$  corresponds to the exact solution as introduced in (3.2). Then the error of the base method satisfies

$$(4.5b) \quad \varepsilon_p = \sum_{\tau \in T_{p+1}} (E\xi_1(\tau) - \hat{\xi}_1(\tau))F(\tau)(y) + \mathcal{O}(\Delta t^{p+1}).$$

Expression (4.5a) is equivalent to imposing (2.12). We also impose stability order [12]  $\tilde{p} = p + 3$ :  $\Phi(\exp(z), z) = \mathcal{O}(\Delta t^{\tilde{p}})$  defined in (3.4), to obtain robust methods.

The two solutions that evolve through the GL process are connected internally, and therefore the error estimation may be hindered in the case of unstable dynamics as discussed in [62]. In Fig. 6.4 we illustrate such a behavior. To this end, we require that the elementary differentials of the two methods resulting from applying the GL method be independent from each other's entries for all trees of order  $p+1$  and  $p+2$ . This requirement can be expressed as

$$(4.6) \quad \hat{\xi}_{\{1,2\}}(\tau_j)[\xi_{\{2,1\}}(\tau_k)] = 0, \quad \forall j, k, \quad \rho(\tau_k), \rho(\tau_j) \in \{p+1, p+2\},$$

where  $\xi_{\{\ell\}}(\tau_j)$  is the coefficient of input  $\ell$  corresponding to tree index  $j$  and  $\hat{\xi}_{\{i\}}(\tau_j)$  is the coefficient of GL output  $i$  corresponding to tree index  $k$ . In other words, output 1 that corresponds to tree index  $j$  does not depend on input 2 of tree index  $k$ , and the same for output 2 and input 1.

LEMMA 4.2. *The elementary differentials of a GL method (4.1) with  $\mathbf{V} = I$  satisfy*

$$(4.7) \quad \hat{\xi}_i(\tau_p) = K + \xi_i(\tau_p) + \mathbf{B}\mathbf{U}\xi(\tau_{p-1}) + G(\tau_{k \in \{1,2,\dots,p-2\}}), \quad i = 1, 2,$$

where  $K$  is a constant that depends on the tree index and  $G$  is a function of  $\tau$  of orders 1 to  $p-2$ .

*Proof.* For the first tree  $\tau_\emptyset$  we have  $\eta D(\tau_\emptyset) = 0$ . The next tree is  $\tau_1 = \bullet$ , for which  $\eta D(\bullet) = 1$ . Relation (3.2) gives

$$\eta(\tau_1) = \mathbf{A}\eta D(\tau_1) + \mathbf{U}\xi(\tau_1) = \mathbf{A}\mathbb{1}_s + \mathbf{U}\xi(\tau_1).$$

This is allowed by Lemma 2.6. Next we have  $\eta D(\tau_2) = \eta D(\bullet \bullet) = \eta(\tau_1)$  and

$$\eta(\tau_2) = \mathbf{A}\eta D(\tau_2) + \mathbf{U}\xi(\tau_2) = \mathbf{A} \cdot (\mathbf{A}\mathbb{1} + \mathbf{U}\xi(\tau_1)) + \mathbf{U}\xi(\tau_2).$$

For the next tree we have  $\eta D(\tau_3) = \eta D(\bullet \bullet \bullet) = (\eta(\tau_1))^2$  and

$$\eta(\tau_3) = \mathbf{A}(\mathbf{A}\mathbb{1}_s + \mathbf{U}\xi(\tau_1))^2 + \mathbf{U}\xi(\tau_3),$$

where the power is taken componentwise. The last third-order tree gives  $\eta D(\tau_4) = \eta D(\begin{smallmatrix} \bullet \\ \bullet \\ \bullet \end{smallmatrix}) = \eta(\tau_2)$  and

$$\begin{aligned} \eta(\tau_4) &= \mathbf{A}(\mathbf{A} \cdot (\mathbf{A}\eta D(\tau_1) + \mathbf{U}\xi(\tau_1)) + \mathbf{U}\xi(\tau_2)) + \mathbf{U}\xi(\tau_4) \\ &= \mathbf{A}^3 \mathbf{1} + \mathbf{A}^2 \mathbf{U}\xi(\tau_1) + \mathbf{A}\mathbf{U}\xi(\tau_2) + \mathbf{U}\xi(\tau_4). \end{aligned}$$

We then arrive at the following recurrence formula:

$$(4.8a) \quad \eta D(\tau_p) = \mathbf{A} \prod_{j \in \mathcal{I}_{p-1}} \eta(\tau_j) + \mathbf{U}\xi(\tau_{p-1}).$$

Similarly, one can verify that the recurrence for the output quantities satisfies

$$(4.8b) \quad \widehat{\xi}_i(\tau_p) = \mathbf{B}\mathbf{A} \prod_{j \in \mathcal{I}_{p-1}} \eta(\tau_j) + \mathbf{B}\mathbf{U}\xi(\tau_{p-1}) + \xi_i(\tau_p), \quad \mathcal{I}_{p-1} = \{1, 2, \dots, p-1\}.$$

This is a consequence of the fact that  $D(\tau_p) = \prod_{k \in \{1, 2, \dots, p-1\}} \tau_k$ . For trees with index 3 and 4 the output is obtained again from (3.2) and using the above derivations as

$$\begin{aligned} \widehat{\xi}_i(\tau_3) &= \mathbf{B}((\mathbf{A}\mathbf{1}_s)^2 + (\mathbf{U}\xi(\tau_1))^2) + \xi_i(\tau_3), \\ \widehat{\xi}_i(\tau_4) &= \mathbf{B}(\mathbf{A} \cdot (\mathbf{A}\mathbf{1} + \mathbf{U}\xi(\tau_1)) + \mathbf{U}\xi(\tau_2)) + \xi_i(\tau_4), \end{aligned}$$

An inductive argument yields (4.8b).  $\square$

**PROPOSITION 4.3** (Output independence of GL method). *A GL method for which the off-diagonal elements of matrix  $\mathbf{B}\mathbf{U}$  are zero satisfies the independence assumption (4.6).*

*Proof.* Using the results of Lemma 4.2, we compute the output  $i$  for trees of order  $p+1$  and assume that the input is consistent of order  $p$ , that is,  $\xi_i(\tau_{k \in \{1, 2, \dots, p\}}) = 0$ . We obtain

$$\widehat{\xi}_i(\tau_{p+1}) = K + \xi_i(\tau_{p+1}) + \mathbf{B}\mathbf{U}\xi(\tau_p) + G(\tau_{k \in \{1, 2, \dots, p-1\}}) = K + \xi_i(\tau_{p+1}), \quad i = 1, 2.$$

For  $p+2$  and together with the fact that  $\mathbf{B}\mathbf{U}$  is a diagonal matrix, we obtain

$$\begin{aligned} \widehat{\xi}_i(\tau_{p+2}) &= K + \xi_i(\tau_{p+2}) + \mathbf{B}\mathbf{U}\xi(\tau_{p+1}) + G(\tau_{k \in \{1, 2, \dots, p\}}) \\ &= K + \xi_i(\tau_{p+2}) + \mathbf{B}\mathbf{U}\xi(\tau_{p+1}) \\ &= K + \xi_i(\tau_{p+2}) + (\mathbf{B}\mathbf{U})_{ii}\xi_i(\tau_{p+1}), \quad i = 1, 2. \end{aligned}$$

$\square$

A similar calculation for  $p+3$  reveals that matrices  $\mathbf{B}\mathbf{A}\mathbf{U}$  and  $\mathbf{B} \operatorname{diag}(\mathbf{A}\mathbf{1})\mathbf{U}$  need to have only diagonal entries. These conditions are necessary when dealing with mildly stiff systems and for long-time simulations. Collectively, we call these decoupling conditions.

*Order barriers.* The order barriers of an  $(r, s)$ -GL method apply to the methods with built-in error estimates (GEE) of one order lower, and this is rather an upper bound. For example, a GEE method of order 2 has the same restrictions on  $r$  and  $s$  as does a GL method of order 3. These restrictions are reasonable because otherwise the underlying higher-order approximation discussed in Sec. 2.4 would violate the GL order barriers.

**4.3. Optimal methods.** We now discuss the need to balance the local truncation errors, which we would like to be as small as possible, with the ability to capture the global errors. Solving for the correction procedure is attractive because it allows the reuse of methods with well-established properties. In particular, one may consider methods that minimize the truncation errors. However, when such optimal methods are used in the context of global error estimation, one must verify that the errors are still quantifiable. For instance, if not all the truncation error terms are nonzero, then special care needs to be exercised because some problems may render the global error estimation “blind” to local error accumulation.

To illustrate this rather subtle point, we consider using the procedure for solving for the correction (2.28) with method RK3(2)G1 (5.2) as introduced in [28]. This is a

third-order scheme; however, it has no errors that correspond to fourth-order trees  and  but does not resolve exactly  and

; otherwise it would have been a fourth-order method. With the aid of Lemma 2.6 we construct a simple problem:  $y'_1 = 1$ ,  $y'_2 = \kappa_2 y_1^3$ ,  $y'_3 = \kappa_3 y_1^4$ , where  $\kappa_i$  are some constants (e.g.,  $\kappa_2 = 1/6$  and  $\kappa_3 = 4$ ). For this problem, the RK3(2)G1 is an order 4 method because the tall tree that would have affected the third component is matched exactly by this method. This means that the base method  $y$  has the same order as the higher-order companion,  $\hat{y}$ . Therefore, the third component can cause the results to be unreliable. In Fig. 4.1 we show the third component, which confirms the inadequacy in the error estimation procedure.

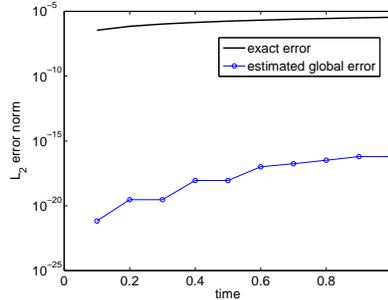


FIG. 4.1. Failure to capture the global errors correctly for system  $y'_1 = 1$ ,  $y'_2 = \kappa_2 y_1^3$ ,  $y'_3 = \kappa_3 y_1^4$  solved with RK3(2)G1 (5.2) [28].

**4.4. Second-order explicit Runge-Kutta-like methods.** We now introduce a few methods of type [A:GLMGEE] (4.3). We begin with a detailed inspection of second-order methods. Schemes with  $s = 2$  are not possible because that would imply that one can have an explicit third-order method by (2.15) with only two stages, which is a statement that is easy to disprove.

A method with  $s = 3$  and  $\gamma = 0$  in GLy $\epsilon$  form is given by the following tableaux,

$$(4.9) \quad \mathbb{M}_{y\epsilon} = \left[ \begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 10 \\ 1/4 & 1/4 & 0 & 1 & -1 \\ \hline 1/12 & 1/12 & 5/6 & 1 & 0 \\ 1/12 & 1/12 & -1/6 & 0 & 1 \end{array} \right],$$

where the four blocks represent  $(\mathbf{A}_{y\epsilon}, \mathbf{U}_{y\epsilon}, \mathbf{B}_{y\epsilon}, \mathbf{V}_{y\epsilon})$  as discussed above. Method

(4.9) can then be expressed as follows:

$$(4.10a) \quad Y_1 = y^{[n-1]},$$

$$(4.10b) \quad Y_2 = y^{[n-1]} + 10\varepsilon^{[n-1]} + \Delta t f(Y_1),$$

$$(4.10c) \quad Y_3 = y^{[n-1]} - \varepsilon^{[n-1]} + \Delta t \left( \frac{1}{4}f(Y_1) + \frac{1}{4}f(Y_2) \right),$$

$$(4.10d) \quad y^{[n]} = y^{[n-1]} + \Delta t \left( \frac{1}{12}f(Y_1) + \frac{1}{12}f(Y_2) + \frac{5}{6}f(Y_3) \right),$$

$$(4.10e) \quad \varepsilon^{[n]} = \varepsilon^{[n-1]} + \Delta t \left( \frac{1}{12}f(Y_1) + \frac{1}{12}f(Y_2) - \frac{1}{6}f(Y_3) \right).$$

In (4.10) we note the Runge-Kutta structure; however, we see that the defect takes an active role in the stage calculations. Using (4.2), we obtain the GLy $\tilde{y}$  form as

$$(4.11) \quad \mathbb{M}_{y\tilde{y}} = \left[ \begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -9 & 10 \\ 1/4 & 1/4 & 0 & 2 & -1 \\ \hline 1/12 & 1/12 & 5/6 & 1 & 0 \\ 1/6 & 1/6 & 2/3 & 0 & 1 \end{array} \right].$$

In particular, (4.11) is expressed as

$$(4.12a) \quad Y_1 = y^{[n-1]},$$

$$(4.12b) \quad Y_2 = -9y^{[n-1]} + 10\tilde{y}^{[n-1]} + \Delta t f(Y_1),$$

$$(4.12c) \quad Y_3 = 2y^{[n-1]} - \tilde{y}^{[n-1]} + \Delta t \left( \frac{1}{4}f(Y_1) + \frac{1}{4}f(Y_2) \right),$$

$$(4.12d) \quad y^{[n]} = y^{[n-1]} + \Delta t \left( \frac{1}{12}f(Y_1) + \frac{1}{12}f(Y_2) + \frac{5}{6}f(Y_3) \right),$$

$$(4.12e) \quad \tilde{y}^{[n]} = \tilde{y}^{[n-1]} + \Delta t \left( \frac{1}{6}f(Y_1) + \frac{1}{6}f(Y_2) + \frac{2}{3}f(Y_3) \right), \quad \varepsilon^{[n]} = \tilde{y}^{[n]} - y^{[n]}.$$

Here we note the explicit contribution of two solutions. A solution of order 3 is obtained according to (2.15) by  $\hat{y}^{[n]} = \tilde{y}^{[n]}$  because  $\gamma = 0$ . Moreover, a local error estimate for  $y^{[n]}$  in (4.12d) corresponds to

$$(4.13) \quad \varepsilon_{\text{loc}} = \varepsilon^{[n]} - \varepsilon^{[n-1]} = \Delta t \left( \frac{1}{12}f(Y_1) + \frac{1}{12}f(Y_2) - \frac{1}{6}f(Y_3) \right),$$

which is an obvious statement. This is also obtained by replacing  $\tilde{y}^{[n-1]}$  by  $y^{[n]}$  in the right-hand sides of (4.12) and taking the differences between the two solutions or setting  $\varepsilon^{[n-1]} = 0$  in (4.10). Additional second-order methods are given in Appendix A.1. We remark that methods using [A:SolCor] require at least four stages. While this seems like a marginal improvement, it is expected to reap more benefits when considering higher-order methods.

**4.5. Third-order explicit Runge-Kutta-like methods.** Closed-form solutions were difficult to obtain for methods of order 3. We therefore explored the space of such methods using a numerical optimization such as in [25]. One method of order

3 with  $\gamma = 0$ ,  $s = 5$  stages, and having significant negative real axis stability was found to have the following coefficients up to 40 digits accuracy:

(4.14)

$$\begin{aligned}
a_{2,1} &= -\frac{2169604947363702313}{24313474998937147335}, & a_{3,1} &= \frac{46526746497697123895}{94116917485856474137}, & a_{3,2} &= -\frac{10297879244026594958}{49199457603717988219}, \\
a_{4,1} &= \frac{23364788935845982499}{87425311444725389446}, & a_{4,2} &= -\frac{79205144337496116638}{148994349441340815519}, & a_{4,3} &= \frac{40051189859317443782}{36487615018004984309}, \\
a_{5,1} &= \frac{42089522664062539205}{124911313006412840286}, & a_{5,2} &= -\frac{15074384760342762939}{137927286865289746282}, & a_{5,3} &= -\frac{62274678522253371016}{125918573676298591413}, \\
a_{5,4} &= \frac{13755475729852471739}{79257927066651693390}, & b_{1,1} &= \frac{61546696837458703723}{56982519523786160813}, & b_{1,2} &= -\frac{55810892792806293355}{206957624151308356511}, \\
b_{1,3} &= \frac{24061048952676379087}{158739347956038723465}, & b_{1,4} &= \frac{3577972206874351339}{7599733370677197135}, & b_{1,5} &= -\frac{59449832954780563947}{137360038685338563670}, \\
b_{2,1} &= -\frac{9738262186984159168}{99299082461487742983}, & b_{2,2} &= -\frac{32797097931948613195}{61521565616362163366}, & b_{2,3} &= \frac{42895514606418420631}{71714201188501437336}, \\
b_{2,4} &= \frac{22608567633166065068}{55371917805607957003}, & b_{2,5} &= \frac{94655809487476459565}{151517167160302729021}, & u_{1,1} &= \frac{70820309139834661559}{80863923579509469826}, \\
u_{1,2} &= \frac{10043614439674808267}{80863923579509469826}, & u_{2,1} &= \frac{161694774978034105510}{106187653640211060371}, & u_{2,2} &= -\frac{55507121337823045139}{106187653640211060371}, \\
u_{3,1} &= \frac{78486094644566264568}{88171030896733822981}, & u_{3,2} &= \frac{9684936252167558413}{88171030896733822981}, & u_{4,1} &= \frac{65394922146334854435}{84570853840405479554}, \\
u_{4,2} &= \frac{19175931694070625119}{84570853840405479554}, & u_{5,1} &= \frac{8607282770183754108}{108658046436496925911}, & u_{5,2} &= \frac{100050763666313171803}{108658046436496925911}.
\end{aligned}$$

We note that this is not an optimal method. It is just an example that was relatively easy to obtain and will be used in the numerical experiments. We note that method RK3(2)G1 (5.2) as introduced in [28] requires 8 stages (that number reduces to 7 because of its FSAL property).

**5. Relationships with other global error estimation strategies.** Here we discuss the relationship between our approach and the existing strategies that we focus on in this study. We show how the latter are particular instantiations of the strategy introduced here. This inclusion is facilitated by the use of Lemma 4.1, which reveals a linear relationship between propagating two solutions and propagating one solution and its defect. We discuss below in some detail the procedure for solving for the correction and the extrapolation approach. Method [A:ZaPr] follows from the same discussions as those for [A:SolCor]. To express algorithm [A:SoErEq] as a GL method, we need more involved manipulations of these methods. Method [A:ExPrErEq] with the exact principal error equation (2.19) can obviously be represented as a GL scheme. Methods that implicitly solve the error equation can also be represented as GL schemes; however, in this study we will not expand on this point.

**5.1. Approach for solving for the correction.** Let us consider the Runge-Kutta methods that integrate the global errors introduced by [28, 29, 31, 57]. The RK tableau is defined by the triplet  $(\mathcal{A}, \mathcal{B}, \mathcal{C})$  and the interpolation operators by  $(B^*, D^*)$ , where  $B^* \cdot [\theta^0, \theta^1, \dots, \theta^s]^T$  yields the interpolant weight vector and  $D^* \cdot [\theta^0, \theta^1, \dots, \theta^s]^T$  yields its derivative. In particular,  $D_{ij}^* = B_{ij}^* \cdot j$ ,  $j = 1, \dots, s$ . Denote by  $b_i^*(\theta) = \sum_{j=1}^{p^*} B_{ij}^* \theta^j$ ,  $d_i^*(\theta) = \sum_{j=1}^{p^*} D_{ij}^* \theta^j$ , and consider the dense output formula given by

$$P(t + \theta \Delta t) = y_n + \theta \Delta t \sum_{i=1}^s b_i^* f_i \quad \text{and} \quad P'(t + \theta \Delta t) = \theta \Delta t \sum_{i=1}^s d_i^* f_i$$

and the error equation that is being solved is (2.28b) ( $\varepsilon'(t) = f(t, P(t) + \varepsilon(t)) - P'(t)$ ). We denote by  $\overline{B}^* = \text{diag}\{\mathcal{C}\} \cdot B^* \cdot W(\mathcal{C})^T$ , where  $W(\mathcal{C})$  is the Vandermonde matrix with entries  $\mathcal{C}$ ; that is,  $\{W(\mathcal{C})\}_{ij} = \mathcal{C}_i^{j-1}$ ; and  $\overline{D}^* = D^* \cdot W(\mathcal{C})^T$ . The resulting method cast in GL format (4.1) is

$$(5.1) \quad \begin{bmatrix} Y_1 \\ Y_2 \\ y_{n+1} \\ \varepsilon_{n+1} \end{bmatrix} = \left[ \begin{array}{cc|cc} \mathcal{A} & 0 & \mathbb{1}_s & 0 \\ \overline{B}^* - \mathcal{A}\overline{D}^* & \mathcal{A} & \mathbb{1}_s & \mathbb{1}_s \\ \hline \mathcal{B}^T & 0 & 1 & 0 \\ -\mathcal{B}^T\overline{D}^* & \mathcal{B}^T & 0 & 1 \end{array} \right] \begin{bmatrix} \Delta t f(Y_1) \\ \Delta t f(Y_2) \\ y_n \\ \varepsilon_n \end{bmatrix}.$$

Here we express the method for a scalar problem, in order to avoid the tensor products, and we represent the stacked stages in  $Y_{\{1,2\}}$ . For example, method RK3(2)G1 [28] is given by the following Butcher tableau:

$$(5.2) \quad \begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ 1 & -1 & 2 & 0 \\ \hline 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}, \quad B^* = \begin{bmatrix} 1 & -\frac{3}{2} & \frac{2}{3} \\ 0 & 2 & -\frac{4}{3} \\ 0 & \frac{3}{6} & -\frac{2}{6} \\ 0 & -1 & 1 \end{bmatrix},$$

$$D_{ij}^* = B_{ij}^* \cdot j$$

The equations to be solved when using the procedure for solving for the correction [A:SolCor] are then (2.28); however, one can show that they are equivalent to solving (5.1) and using the strategy [A:GLMGEE] (4.1) introduced here. The explicit coefficients are listed in tableau (B.1). This strategy of estimating the global errors is called Runge-Kutta triplets; a similar discussion can be extended for the peer-triplets strategy [78]. The Zadunaisky procedure can be shown to have a similar interpretation; however, it is a little more expensive, and the analysis has to be carried over several steps. We will draw conclusions about its behavior by using [A:SolCor] as a proxy.

**5.2. Global error extrapolation.** Let us consider again the Runge-Kutta methods defined by the triplet  $(\mathcal{A}, \mathcal{B}, \mathcal{C})$  of order  $p$ . By applying (2.30) we obtain the method in the GL format,

$$(5.3) \quad \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ y_{n+1} \\ \varepsilon_{n+1} \end{bmatrix} = \left[ \begin{array}{ccc|cc} \mathcal{A} & 0 & 0 & \mathbb{1}_s & 0 \\ 0 & \frac{1}{2}\mathcal{A} & 0 & \mathbb{1}_s & \beta^{-1}\mathbb{1}_s \\ 0 & \frac{1}{2}\mathcal{B}^T \otimes \mathbb{1}_s & \frac{1}{2}\mathcal{A} & \mathbb{1}_s & \beta^{-1}\mathbb{1}_s \\ \hline \mathcal{B}^T & 0 & 0 & 1 & 0 \\ -\beta\mathcal{B}^T & \frac{\beta}{2}\mathcal{B}^T & \frac{\beta}{2}\mathcal{B}^T & 0 & 1 \end{array} \right] \begin{bmatrix} \Delta t f(Y_1) \\ \Delta t f(Y_2) \\ \Delta t f(Y_3) \\ y_n \\ \varepsilon_n \end{bmatrix},$$

where  $\beta = \frac{1}{1-\gamma}$ ,  $\gamma = 1/2^p$ , and  $Y_{\{1,2,3\}}$  are the  $s$ -stage vectors corresponding to the original method stacked on top of each other. This is a method of type (4.1).

**6. Numerical results.** In this section we present numerical results with a detailed set of test problems.

**6.1. Test problems.** We consider a set of simple but comprehensive test problems.

Problem [Prince42] is defined in [62] (4.2) by

$$(6.1a) \quad y' = y - \sin(t) + \cos(t), \quad y(0) = \kappa$$

$$(6.1b) \quad y(t) = \kappa * \exp(t) + \sin(t).$$

Here we take  $\kappa = 0$ . As a direct consequence of (6.1b), we see that any perturbation of the solution  $y$ , such as numerical errors, leads to exponential growth. Therefore we have an unstable dynamical system; and even if we start with  $\kappa = 0$ , numerical errors will lead to an exponential solution growth. This is a classical example that is used to show the failure of local error estimation in general and of global error estimation by using Algorithm [A:ExPrErEq] (2.19) [62] in particular.

A similar problem [Kulikov2013I] is defined by Kulikov [53] by

$$(6.2) \quad y_1' = 2t y_2^{1/5} y_4, \quad y_2' = 10t \exp(5(y_3 - 1)) y_4, \quad y_3' = 2t y_4, \quad y_4' = -2t \ln(y_1),$$

so that  $y_1(t) = \exp(\sin(t^2))$ ,  $y_2(t) = \exp(5 \sin(t^2))$ ,  $y_3(t) = \sin(t^2) + 1$ ,  $y_4(t) = \cos(t^2)$ . This problem is nonautonomous and exhibits unstable modes later.

To illustrate the error behavior over long time integration, we consider problem [Hull1972B4], which is a nonlinear ODE defined in [48] (B4) by

$$(6.3) \quad y_1' = -y_2 - \frac{y_1 y_3}{\sqrt{y_1^2 + y_2^2}}, \quad y_2' = y_1 - \frac{y_2 y_3}{\sqrt{y_1^2 + y_2^2}}, \quad y_3' = \frac{y_1}{\sqrt{y_1^2 + y_2^2}},$$

with  $y_0 = [3, 0, 0]^T$ .

The last problem [LStab2] is used to assess linear stability properties of the proposed numerical methods.

$$(6.4a) \quad y' = Ay, \quad y(0) = [y_1(0), y_2(0)]^T, \quad A = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}, \quad \Lambda(A) = \{a + ib, a - ib\},$$

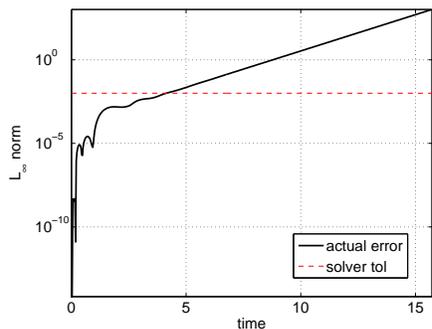
$$(6.4b) \quad \begin{cases} y_1(t) = \exp(at) (y_2(0) \cos(bt) - y_1(0) \sin(bt)) \\ y_2(t) = \exp(at) (y_1(0) \cos(bt) + y_2(0) \sin(bt)) \end{cases}.$$

This problem allows one to choose the position of the eigenvalues of the Jacobian,  $\Lambda(A)$ , in order to simulate problems with different spectral properties.

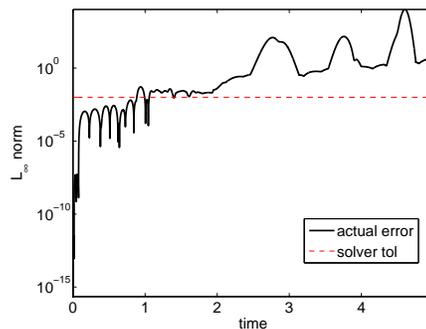
**6.2. Numerical experiments.** We begin with simple numerical experiments that show when local error estimation is not suitable. Local error estimation is typically used for error control; however, in this study we do not explore this aspect. We therefore compare the result of well-tuned numerical integrators that use local error control with the global error estimates for the same problem. The two contexts are different; however, the error estimation problem remains the same. We use Matlab's ode45 integrator with different tolerances whenever we refer to methods with local error estimation.

In Fig. 6.1 we show the errors over time for problems [Prince42] (6.1) and [Kulikov2013I] (6.2). These problems are solved by using local error estimation (LEE) – 6.1(a-b), and global error estimation (GEE) method (A.1) – 6.1(c-d). The absolute error tolerance for LEE control is set to 1e-02. The methods with LEE systematically underestimate the error levels as expected, whereas the methods with GEE capture the errors exactly. Moreover, the global errors are captured well across components, as shown in Fig. 6.1(d).

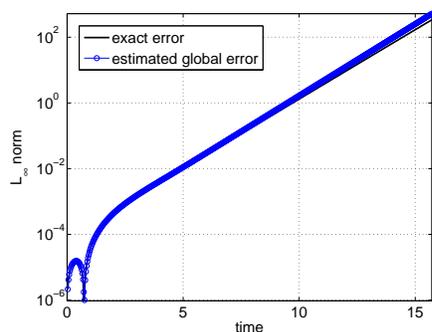
In Fig. 6.2 we show the error behavior for problem [Hull1972B4] (6.3) when long integration windows are considered. For LEE we set the absolute tolerance to 1e-05.



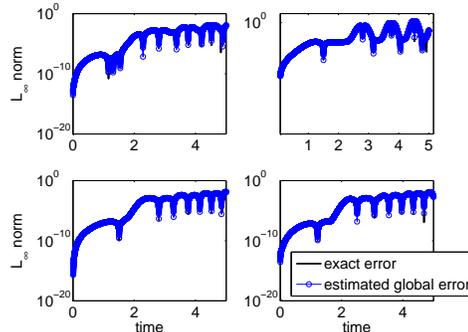
(a) [Prince42] (6.1) with LEE



(b) [Kulikov2013I] (6.2) with LEE



(c) problem (a) with GEE



(d) problem (b) with GEE, by components

FIG. 6.1. Errors when solving problems with unstable modes by using local error estimation (LEE) and global error estimation (GEE) [A:GLMGEE]. The absolute error tolerance for LEE control is set to  $1e-02$ . The GEE method used here is (A.1). GEE captures the errors exactly while LEE underestimates them.

In this case we observe an error drift to levels of  $1e-03$  over 1,000 time units. The method with GEE (A.3) can follow closely the error in time. For the built-in error estimator in the GEE method to maintain its accuracy over time, we need to decouple the two embedded schemes further by ensuring that  $\mathbf{BAU}$  is a diagonal matrix.

We next analyze the convergence properties of the methods discussed here. In Fig. 6.3 we show the convergence of the solution and of the error estimate. Here we illustrate the convergence of GEE methods of orders 2 (A.2) and 3 (4.14) for problem [Prince42] (6.1). The methods converge with their prescribed order; moreover, the error estimate also converges with order  $p + 1$ , as expected from (2.15). In Fig. 6.4 we show the behavior of global error estimation when using [A:ExPrErEq], methods with the exact principal error equation (2.19). Here we use method (C.1) [62, (3.11)], which fails to capture the error magnitude as discussed in [62] because the estimated error is several orders of magnitude smaller than the true global error.

We next look at the linear stability properties of the methods introduced in this study. In Fig. 6.5(a) we delineate the stability regions according to (3.3). In Fig. 6.5(b) we show numerical results for problem [LStab2] (6.4) with  $\lambda\Delta t = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\} \times (-1 \pm \sqrt{-1})$  when using method (A.1). As expected, all solutions except for the one corresponding to  $\lambda\Delta t = -1 \pm \sqrt{-1}$  are stable, as can be interpreted from Fig. 6.5(a).

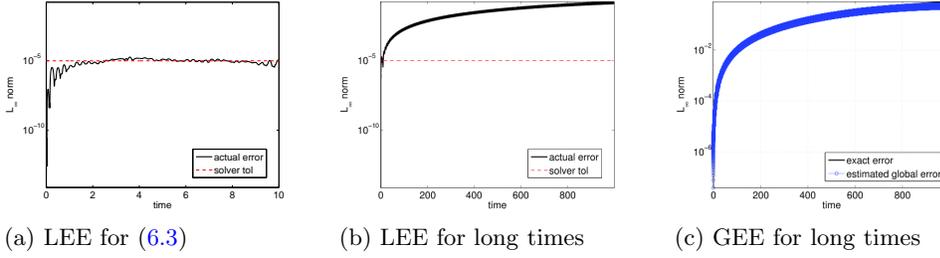


FIG. 6.2. Errors when solving [Hull1972B4] (6.3) with LEE and GEE. For LEE we set the absolute tolerance to  $1e-05$ . (a) During short integration times LEE satisfies the error tolerance well. (b) However, for longer times we see an expected drift to error levels of  $1e-03$ . (c) GEE method [A:GLMGEE] ((A.3) in this case) gives accurate error estimates even over long times.

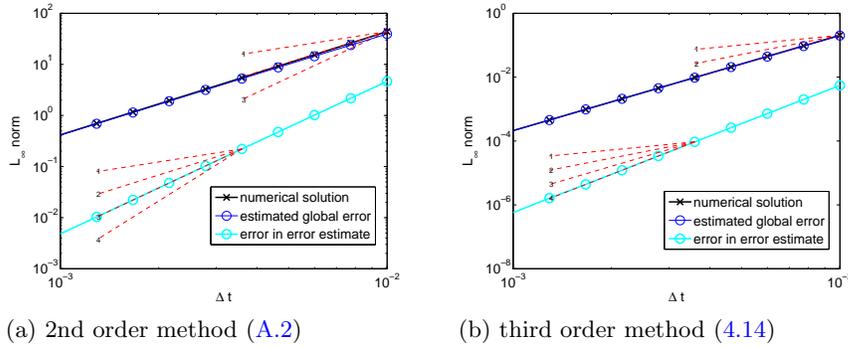
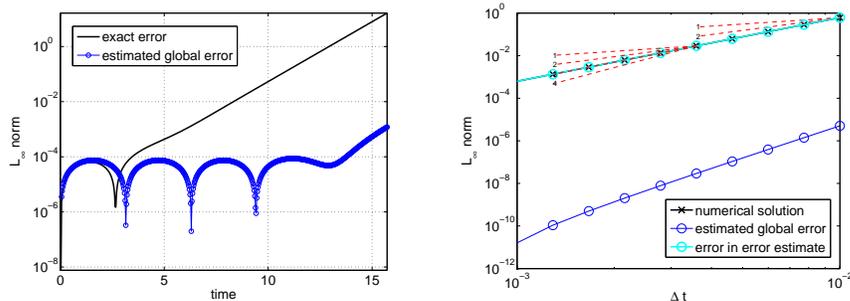


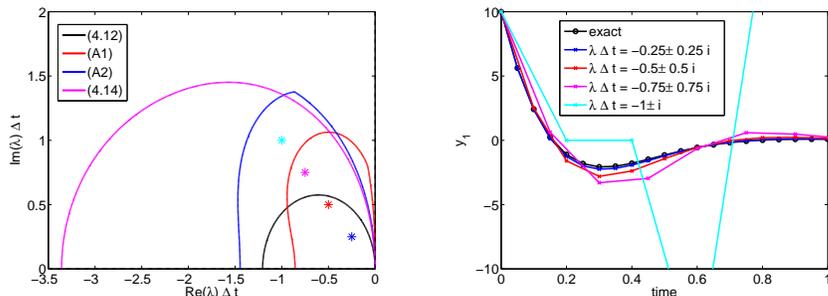
FIG. 6.3. Convergence of GEE methods [A:GLMGEE] for problem [Prince42] (6.1). In (a) we show the error in the solution obtained by the second-order GEE method (A.2) and the estimated global error, which follows it closely, as well as the difference between the true error and the error estimate. An asymptotic guide is provided by the red dashed lines. (b) This is the same as (a) but using the third-order method (4.14).

Two practical situations are illustrated next (*i*) a situation in which local error control is used to adapt the time steps and the global error estimate is used to validate the solution accuracy and (*ii*) a situation in which a prescribed level of accuracy is needed, which requires recomputing the solution with a different time step. In the first case we consider a simple local error control using local error estimates (4.3b) provided by GEE (4.14). In Fig. 6.6 we show the error evolution [Kulikov2013I] while attempting to restrict the local error to be around  $1e-05$ . The time-step length is allowed to vary between  $1e-03$  and  $1e-05$ . Note that the global error estimate remains faithful under the time-step change as expected based on the discussion in Sec. 2.1.2 following (2.16). Similar conditions are using in the second case in which the target is to satisfy a prescribed tolerance. In a first run the time step is fixed to  $0.0001$ , and an integration is carried out to the final time. The global error and the asymptotic are used to determine the time step that would guarantee a global error of  $1e-04$ . A simple asymptotic analysis indicates that a time step of  $0.00013$  needs to be used in order to achieve that target for GEE (4.14). The solution is recomputed by using this smaller fixed time-step, and the global errors shown in Fig. 6.7 satisfy the prescribed global accuracy. We note that this case is just an illustration of the theoretical properties



(a) error estimation for methods [A:ExPrErEq] with exact principal error equation (2.19) for problem [Prince42] (b) method (C.1) from [62, (3.11)]

FIG. 6.4. (a) Failure to capture the global errors correctly for problem [Prince42] (6.1) when using a [A:ExPrErEq] method (2.19) such as (C.1) [62] and (b) its convergence analysis.



(a) stability regions for second-order methods (4.12), (A.1), (A.2), and third order (4.14) (b) solution of [LStab2] (6.4) when using method (A.1)

FIG. 6.5. Linear stability regions for the [A:GLMGEE] methods introduced in this study (a) and solution of method (A.1) for problem [LStab2] (6.4) with parameters such that it matches the spectrum indicated in (a) with marker \*. Solutions are stable except the one for which the eigenvalues are outside the stability region (b).

of the methods introduced here. It is likely a suboptimal strategy for achieving a prescribed global tolerance.

**7. Discussion.** In this study we introduce a new strategy for global error estimation in time-stepping methods. This strategy is based on advancing in time the solution along with the defect or, equivalently, two solutions that have a fixed relation between their truncation errors. The main idea is summarized in Theorem 2.9, and practical considerations are brought up by Proposition 4.3. This translates in a particular relation among the truncation errors of GL outputs and in a decoupling constraint. We note that this strategy can be seen as a generalization of the procedure for solving for the correction and of several others from the same class. We provide equivalent representation of these methods in the proposed GL form (4.1).

We have explored several algorithms in this study. The methods [A:ExPrErEq] with exact principal error equation (2.19) [74] are attractive because they offer global

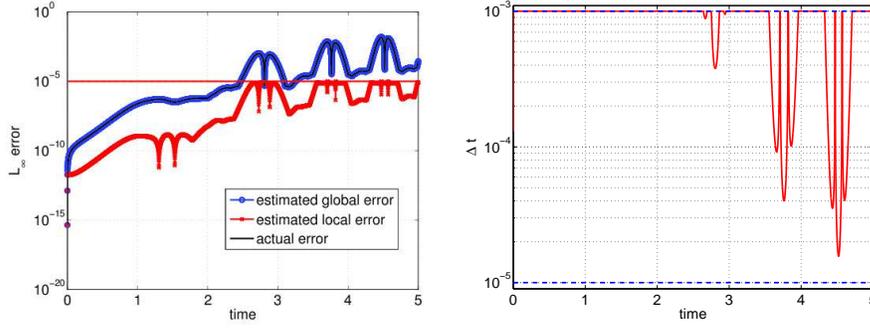


FIG. 6.6. Time-step adaptivity using a local error controller. The GEE method (4.14) is used to integrate problem [Kulikov2013I] by restricting local error to be less than  $1e-05$ . The error evolution is shown on the left, and the time-step length is shown on the right panel. The time-step length is allowed to vary between  $1e-03$  and  $1e-05$ .

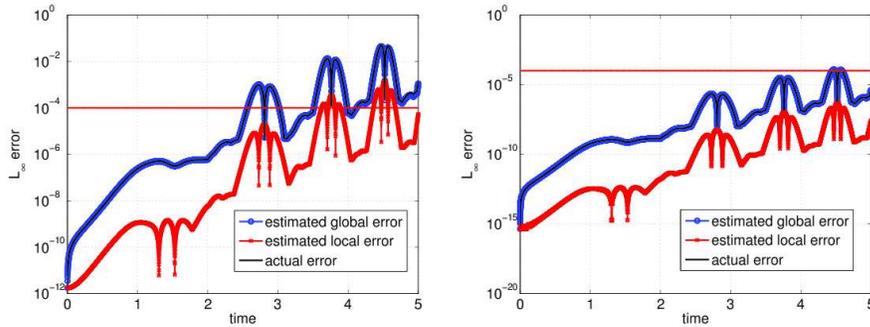


FIG. 6.7. Error evolution for [Kulikov2013I] when using the GEE method (4.14) with  $\Delta t = 0.001$  (left) and  $\Delta t = 0.00013$  (right). The smaller time step has been calculated based on the GEE global error estimate to keep the global errors below  $1e-04$ .

error estimates extremely cheaply; however, they were shown in [62] to be unreliable as illustrated in Fig. 6.4. Strategies that directly solve the error equation, such as [A:SoErEq] (2.26), need a reliable way of estimating the local errors and the availability of the Jacobian. We found these methods to be robust, especially the strategy proposed in [56] for low-order methods. The procedure for solving for the correction [A:SolCor] (2.28) is arguably one of the most popular approaches for global error estimation. It is related to [A:ZaPr] and [A:SoErEq], as discussed, and a particular case of this approach is introduced in this study. The extrapolation algorithm [A:Ex] (2.31) is the most robust; however, it is also the most expensive and also a particular case of [A:GLMGEE]. With respect to global error control, Kulikov and Weiner [53, 78] report very promising results.

The methods introduced here are based on a general linear representation. The particular case under study is given by form (4.1); however, the analysis is not restricted to that situation. Particular instances of second and third order are presented throughout this study. The error estimates can be used for error control; however, in this study we do not address this issue. The GL representation allows stages to be reused or shared among the solutions that are propagated within. This leads to methods having lower costs. Moreover, the stability analysis is simplified when compared

with global error strategies that use multiple formulas or equations. We consider only GL methods with two solutions, but this concept can be extended to having multiple values propagated in time (e.g., multistep-multistage or peer methods).

We provide several numerical experiments in which we illustrate the behavior of the global error estimation procedures introduced here, their convergence behavior, and their stability properties.

We investigate nonstiff differential equations. Additional constraints are necessary in order to preserve error estimation and avoid order reduction necessary for stiff ODEs.

Global error estimation is typically not used in practice because of its computational expense. This study targets strategies that would make it cheaper to estimate the global errors and therefore make them more practical.

## Appendix A. Second-order methods.

**A.1. Other GL second-order methods.** Here we provide two additional second-order methods that we used in our experiments. A second-order method with  $s = 3$  and  $\gamma = 0$  in GLy $\varepsilon$  format is given by

$$(A.1) \quad \mathbb{M}_{y\varepsilon} = \left[ \begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 4 \\ 1 & 0 & 0 & 1 & 0 \\ 4/9 & 2/9 & 0 & 1 & 0 \\ \hline 0 & -1/2 & 3/2 & 1 & 0 \\ 1/4 & 1/2 & -3/4 & 0 & 1 \end{array} \right].$$

Another second-order method with  $s = 3$  that is based on two second-order approximations ( $\gamma = 1/2$ ) in GLy $\varepsilon$  format is given by

$$(A.2) \quad \mathbb{M}_{y\varepsilon} = \left[ \begin{array}{ccc|cc} 0 & 0 & 0 & 1 & -11/10 \\ 1 & 0 & 0 & 1 & 13/30 \\ 4/9 & 2/9 & 0 & 1 & 5/3 \\ \hline 5/12 & 5/12 & 1/6 & 1 & 0 \\ -1/4 & -1/4 & 1/2 & 0 & 1 \end{array} \right].$$

A second order method that results in having both **BU** and **BAU** diagonal matrices is a four-stage method with the following coefficients in GLy $\tilde{y}$  form:

$$(A.3) \quad \begin{array}{lll} \mathbf{A}_{2,1} = 3/4, & \mathbf{A}_{3,1} = 1/4, & \mathbf{A}_{3,2} = 29/60, \\ \mathbf{A}_{4,1} = -21/44, & \mathbf{A}_{4,2} = 145/44, & \mathbf{A}_{4,3} = -20/11, \\ \mathbf{B}_{1,1} = 109/275, & \mathbf{B}_{1,2} = 58/75, & \mathbf{B}_{1,3} = -37/110, & \mathbf{B}_{1,4} = 1/6, \\ \mathbf{B}_{2,1} = 3/11, & & \mathbf{B}_{2,3} = 75/88, & \mathbf{B}_{2,4} = -1/8, \\ & \mathbf{U}_{2,1} = 75/58, & & \\ \mathbf{U}_{1,2} = 1, & \mathbf{U}_{2,2} = -17/58, & \mathbf{U}_{3,2} = 1, & \mathbf{U}_{4,2} = 1, \end{array}$$

where the rest of the coefficients are zero.

**Appendix B. RK3(2)G1 [28] in GL form (4.1).** Method (5.1) corresponding

to RK3(2)G1 (5.2) [28] results in the following tableau in  $GLy\epsilon$  form:

$$(B.1) \quad \mathbb{M}_{y\epsilon} = \left[ \begin{array}{cccccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/6 & 2/3 & 1/6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ -7/24 & 1/3 & 1/12 & -1/8 & 1/2 & 0 & 0 & 0 & 1 & 1 \\ 7/6 & -4/3 & -1/3 & 1/2 & -1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1/6 & 2/3 & 1/6 & 0 & 1 & 1 \\ \hline 1/6 & 2/3 & 1/6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1/6 & -2/3 & -1/6 & 0 & 1/6 & 2/3 & 1/6 & 0 & 0 & 1 \end{array} \right].$$

### Appendix C. Second-order method with exact principal error equation.

The following method is of type (2.19) and introduced in [62, (3.11)]:

(C.1)

$$\mathbb{S} := \left[ \begin{array}{ccc|ccc} 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & & \\ \frac{5}{8} & 0 & \frac{5}{8} & 0 & & \\ \hline & -\frac{1}{30} & \frac{1}{2} & \frac{8}{15} & & \end{array} \right], \quad \mathbb{M} := \left[ \begin{array}{cccc|ccc} 0 & & & & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & & & \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & & & \\ \hline & \frac{2}{3} & -1 & \frac{4}{3} & & & \end{array} \right], \quad \mathbb{F} := \left[ \begin{array}{ccc|ccc} 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & & \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & & \\ \hline & -\frac{29}{42} & -\frac{31}{42} & \frac{22}{21} & & \end{array} \right].$$

### REFERENCES

- [1] R. AÏD AND L. LEVACHER, *Numerical investigations on global error estimation for ordinary differential equations*, Journal of computational and applied mathematics, 82 (1997), pp. 21–39.
- [2] S. BALAY, S. ABHYANKAR, M.F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, V. EIJKHOUT, W.D. GROPP, D. KAUSHIK, M.G. KNEPLEY, L.C. MCINNES, K. RUPP, B.F. SMITH, AND H. ZHANG, *PETSc users manual*, Tech. Report ANL-95/11 - Revision 3.5, Argonne National Laboratory, 2014.
- [3] J.W. BANKS, J.A.F. HITTINGER, J.M. CONNORS, AND C.S. WOODWARD, *Numerical error estimation for nonlinear hyperbolic PDEs via nonlinear error transport*, Computer Methods in Applied Mechanics and Engineering, 213216 (2012), pp. 1–15.
- [4] K. BURRAGE AND J.C. BUTCHER, *Non-linear stability of a general class of differential equation methods*, BIT, 20 (1980), pp. 185–203.
- [5] J.C. BUTCHER, *Coefficients for the study of Runge-Kutta integration processes*, Journal of the Australian Mathematical Society, 3 (1963), pp. 185–201.
- [6] J.C. BUTCHER, *A modified multistep method for the numerical integration of ordinary differential equations*, J. ACM, 12 (1965), pp. 124–135.
- [7] ———, *On the convergence of numerical solutions to ordinary differential equations*, Mathematics of Computation, 20 (1966), pp. 1–10.
- [8] J.C. BUTCHER, *The effective order of Runge-Kutta methods*, in Conference on the numerical solution of differential equations, Springer, 1969, pp. 133–139.
- [9] J.C. BUTCHER, *An algebraic theory of integration methods*, Mathematics of Computation, 26 (1972), pp. 79–106.
- [10] J.C. BUTCHER, *Order and effective order*, Applied Numerical Mathematics, 28 (1998), pp. 179–191.
- [11] J.C. BUTCHER, *General linear methods for stiff differential equations*, BIT, 41 (2001), pp. 240–264.
- [12] J.C. BUTCHER, *The A-stability of methods with Padé and generalized Padé stability functions*, Numerical Algorithms, 31 (2002), pp. 47–58.
- [13] J.C. BUTCHER, *General linear methods*, Acta Numerica, 15 (2006), pp. 157–256.

- [14] ———, *Numerical Methods for Ordinary Differential Equations*, Wiley, second ed., 2008.
- [15] J.C. BUTCHER AND P. CHARTIER, *A generalization of singly-implicit Runge-Kutta methods*, Applied Numerical Mathematics, 24 (1997), pp. 343–350.
- [16] ———, *The effective order of singly-implicit Runge-Kutta methods*, Numerical Algorithms, 20 (1999), pp. 269–284.
- [17] J.C. BUTCHER AND D.J.L. CHEN, *ESIRK methods and variable stepsize*, Applied Numerical Mathematics, 28 (1998), pp. 193–207.
- [18] J.C. BUTCHER AND M.T. DIAMANTAKIS, *DESIRE: Diagonally extended singly implicit Runge-Kutta effective order methods*, Numerical Algorithms, 17 (1998), pp. 121–145.
- [19] Y. CAO AND L. PETZOLD, *A posteriori error estimation and global error control for ordinary differential equations by the adjoint method*, SIAM Journal on Scientific Computing, 26 (2005), pp. 359–374.
- [20] J.R. CASH AND A.H. KARP, *A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides*, ACM Transactions on Mathematical Software (TOMS), 16 (1990), pp. 201–222.
- [21] J.-C. CHANG, T.M.H. CHAN, AND D.J.L. CHEN, *Enhanced order composition methods*, Applied Numerical Mathematics, 58 (2008), pp. 223–235.
- [22] P. CHARTIER, E. HAIRER, AND G. VILMART, *Algebraic structures of B-series*, Foundations of Computational Mathematics, 10 (2010), pp. 407–427.
- [23] J. CONNORS, J. BANKS, J. HITTINGER, AND C. WOODWARD, *A method to calculate numerical errors using adjoint error estimation for linear advection*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 894–926.
- [24] E.M. CONSTANTINESCU, *On the order of general linear methods*, Applied Mathematics Letters, 22 (2009), pp. 1425–1428.
- [25] E.M. CONSTANTINESCU AND A. SANDU, *Optimal explicit strong-stability-preserving general linear methods*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3130–3150.
- [26] G.J. COOPER, *The order of convergence of general linear methods for ordinary differential equations*, SIAM Journal on Numerical Analysis, 15 (1978), pp. 643–661.
- [27] GERMUND DAHLQUIST, *On the control of the global error in stiff initial value problems*, in Numerical Analysis, G. Watson, ed., vol. 912 of Lecture Notes in Mathematics, Springer Berlin / Heidelberg, 1982, pp. 38–49. 10.1007/BFb0093147.
- [28] J.R. DORMAND, R.R. DUCKERS, AND P.J. PRINCE, *Global error estimation with Runge-Kutta methods*, IMA Journal of Numerical Analysis, 4 (1984), pp. 169–184.
- [29] J.R. DORMAND, J.P. GILMORE, AND P.J. PRINCE, *Globally embedded Runge-Kutta schemes*, Ann. Numer. Math, 1 (1994), pp. 97–106.
- [30] J.R. DORMAND, M.A. LOCKYER, N.E. MCGORRIGAN, AND P.J. PRINCE, *Global error estimation with Runge-Kutta triples*, Computers & Mathematics with Applications, 18 (1989), pp. 835–846.
- [31] J.R. DORMAND AND P.J. PRINCE, *Global error estimation with Runge-Kutta methods II*, IMA Journal of Numerical Analysis, 5 (1985), pp. 481–497.
- [32] W.H. ENRIGHT, *Analysis of error control strategies for continuous Runge-Kutta methods*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 588–599.
- [33] ———, *Continuous numerical methods for ODEs with defect control*, Journal of Computational and Applied Mathematics, 125 (2000), pp. 159–170.
- [34] D. ESTEP, *A posteriori error bounds and global error control for approximation of ordinary differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 1–48.
- [35] D. ESTEP, M. HOLST, AND M. LARSON, *Generalized Green’s functions and the effective domain of influence*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1314–1339.
- [36] C.W. GEAR, *Hybrid methods for initial value problems in ordinary differential equations*, SIAM Journal on Numerical Analysis, 2 (1965), pp. 69–86.
- [37] MICHAEL B GILES AND ENDRE SÜLI, *Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality*, Acta Numerica, 11 (2002), pp. 145–236.
- [38] W.B. GRAGG AND H.J. STETTER, *Generalized multistep predictor-corrector methods*, J. ACM, 11 (1964), pp. 188–209.
- [39] Y. HADJIMICHAEL, C. MACDONALD, D. KETCHESON, AND J. VERNER, *Strong stability preserving explicit Runge-Kutta methods of maximal effective order*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 2149–2165.
- [40] E. HAIRER AND C. LUBICH, *Asymptotic expansions of the global error of fixed-stepsize methods*, Numerische Mathematik, 45 (1984), pp. 345–360.
- [41] E. HAIRER, S.P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I: Non-stiff Problems*, Springer, 2008.
- [42] E. HAIRER AND G. WANNER, *Multistep-multistage-multiderivative methods for ordinary differ-*

- ential equations*, Computing, 11 (1973), pp. 287–303.
- [43] ———, *On the Butcher group and general multi-value methods*, Computing, 13 (1974), pp. 1–15.
- [44] P. HENRICI, *Discrete variable methods in ordinary differential equations*, New York: Wiley, 1962, 1962.
- [45] M. HEROUX, R. BARTLETT, V.H.R. HOEKSTRA, J. HU, T. KOLDA, R. LEHOUCQ, K. LONG, R. PAWLOWSKI, E. PHIPPS, A. SALINGER, H. THORNQUIST, R. TUMINARO, J. WILLENBRING, AND A. WILLIAMS, *An Overview of Trilinos*, Tech. Report SAND2003-2927, Sandia National Laboratories, 2003.
- [46] D.J. HIGHAM, *Global error versus tolerance for explicit Runge-Kutta methods*, IMA Journal of Numerical Analysis, 11 (1991), pp. 457–480.
- [47] A.C. HINDMARSH, P.N. BROWN, K.E. GRANT, S.L. LEE, R. SERBAN, D.E. SHUMAKER, AND C.S. WOODWARD, *SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 363–396.
- [48] T.E. HULL, W.H. ENRIGHT, B.M. FELLEN, AND A.E. SEDGWICK, *Comparing numerical methods for ordinary differential equations*, SIAM Journal on Numerical Analysis, 9 (1972), pp. 603–637.
- [49] Z. JACKIEWICZ, *General Linear Methods for Ordinary Differential Equations*, Wiley-Interscience, John Wiley & Sons, 2009.
- [50] C.P. JEANNEROD AND J. VISCONTI, *Global error estimation for index-1 and-2 DAEs*, Numerical Algorithms, 19 (1998), pp. 111–125.
- [51] G.Y. KULIKOV, *On quasi-consistent integration by Nordsieck methods*, Journal of Computational and Applied Mathematics, 225 (2009), pp. 268–287.
- [52] ———, *Global error control in adaptive Nordsieck methods*, SIAM Journal on Scientific Computing, 34 (2012), pp. A839–A860.
- [53] ———, *Cheap global error estimation in some Runge-Kutta pairs*, IMA Journal of Numerical Analysis, 33 (2013), pp. 136–163.
- [54] G.Y. KULIKOV AND R. WEINER, *Doubly quasi-consistent parallel explicit peer methods with built-in global error estimation*, Journal of Computational and Applied Mathematics, 233 (2010), pp. 2351–2364.
- [55] G.Y. KULIKOV AND R. WEINER, *Variable-stepsize interpolating explicit parallel peer methods with inherent global error control*, SIAM Journal on Scientific Computing, 32 (2010), pp. 1695–1723.
- [56] J. LANG AND J.G. VERWER, *On global error estimation and control for initial value problems*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1460–1475.
- [57] J. MAKAZAGA AND A. MURUA, *New Runge–Kutta based schemes for ODEs with cheap global error estimation*, BIT Numerical Mathematics, 43 (2003), pp. 595–610.
- [58] MATLAB, *version 8.1.0 (R2013a)*, The MathWorks Inc., Natick, Massachusetts, 2014.
- [59] P. MERLUZZI AND C. BROSILOW, *Runge-Kutta integration algorithms with built-in estimates of the accumulated truncation error*, Computing, 20 (1978), pp. 1–16.
- [60] J. OLIVER, *A curiosity of low-order explicit Runge-Kutta methods*, Mathematics of Computation, 29 (1975), pp. 1032–1036.
- [61] P.J. PETERSON, *Global Error Estimation Using Defect Correction Techniques for Explicit Runge-Kutta Methods*, Technical report (University of Toronto. Dept. of Computer Science), Univ. Department of Computer Science, 1986.
- [62] P.J. PRINCE AND K. WRIGHT, *Runge-Kutta processes with exact principal error equations*, IMA Journal of Applied Mathematics, 21 (1978), pp. 363–373.
- [63] L.F. RICHARDSON, *The deferred approach to the limit.*, Philosophical Transactions of the Royal Society of London, 226 (1927), pp. 299–349.
- [64] WR RICHERT, *Implicit Runge-Kutta formulae with built-in estimates of the accumulated truncation error*, Computing, 39 (1987), pp. 353–362.
- [65] D.L. RIVE AND F. PASCIUTTI, *Runge–Kutta methods with global error estimates*, IMA Journal of Applied Mathematics, 16 (1975), pp. 381–386.
- [66] S. SCHOLZ, *Implicit Runge-Kutta methods with a global error estimation for stiff differential equations*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 69 (1989), pp. 253–257.
- [67] L.F. SHAMPINE, *Global error estimation with one-step methods*, Computers & Mathematics with Applications, 12 (1986), pp. 885–894.
- [68] ———, *Error estimation and control for ODEs*, Journal of Scientific Computing, 25 (2005), pp. 3–16.
- [69] L.F. SHAMPINE AND L.S. BACA, *Fixed versus variable order Runge-Kutta*, ACM Transactions on Mathematical Software (TOMS), 12 (1986), pp. 1–23.
- [70] L.F. SHAMPINE, M.K. GORDON, AND J.A. WISNIEWSKI, *Variable order Runge-Kutta codes*,

- tech. report, Sandia Labs., Albuquerque, NM, 1979.
- [71] R. SKEEL, *Analysis of fixed-stepsize methods*, SIAM Journal on Numerical Analysis, 13 (1976), pp. 664–685.
- [72] R.D. SKEEL, *Thirteen ways to estimate global error*, Numerische Mathematik, 48 (1986), pp. 1–20.
- [73] M.N. SPIJKER, *On the structure of error estimates for finite-difference methods*, Numerische Mathematik, 18 (1971), pp. 73–100.
- [74] H.J. STETTER, *Local estimation of the global discretization error*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 512–523.
- [75] ———, *Economical global error estimation*, The IBM Research Symposium Series, (1974), pp. 245–258.
- [76] ———, *Global error estimation in ordinary initial value problems*, in Numerical Integration of Differential Equations and Large Linear Systems, Juergen Hinze, ed., vol. 968 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1982, pp. 269–279.
- [77] M. UTUMI, R. TAKAKI, AND T. KAWAI, *Optimal time step control for the numerical solution of ordinary differential equations*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 1644–1653.
- [78] R. WEINER AND G.Y. KULIKOV, *Local and global error estimation and control within explicit two-step peer triples*, Journal of Computational and Applied Mathematics, 262 (2014), pp. 261–270.
- [79] R. WEINER, G.Y. KULIKOV, AND H. PODHAISKY, *Variable-stepsize doubly quasi-consistent parallel explicit peer methods with global error control*, Applied Numerical Mathematics, 62 (2012), pp. 1591–1603.
- [80] P.E. ZADUNAIISKY, *On the estimation of errors propagated in the numerical integration of ordinary differential equations*, Numerische Mathematik, 27 (1976), pp. 21–39.

Government License The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.