Preprint ANL/MCS-P5077-0214

ESTIMATING GLOBAL ERRORS IN TIME STEPPING*

EMIL CONSTANTINESCU†

Abstract. This study introduces new strategies for global error estimation in time-stepping algorithms. The new methods propagate the defect along with the numerical solution much like the solving for the correction or Zadunaisky 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 is generally viewed as an expensive process, and therefore in practice only local error or the error from one step to the next is used to estimate the errors or control the step size. However, local error estimation is not always suitable, especially for problems with unstable modes. This aspect prompts us to revisit global error estimation in order to make it more practical.

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 [63]. 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 [26–30, 32, 45, 48–50, 53, 55–59, 68]. A posteriori global error estimation has been recently discussed in [1, 17, 33, 46]. Step-size control with multimethod Runge-Kutta (RK) is discussed in [18, 58, 60, 61]. Global error estimation for stiff problems is discussed in [25, 46, 47, 66, 67]. Adjoint methods for global error estimation for PDEs are analyzed in [21, 35]. These studies cover most of the types of strategy that have been proposed to address global error estimation. The Zadunaisky procedure [69] and the related solving for the correction procedure [63] are arguably the most popular global-error estimation strategy. The work of Dormand et al. [26,29] relies on this procedure and is extended to a composition of RK methods in [27]. Further extensions are introduced by Makazaga et al. [49]. Shampine [58] proposes using multiple methods to estimate global errors.

Our work builds on similar ideas as introduced by Shampine [58], Zadunaisky [69] 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. Previous strategies can be cast as particular cases of the one introduced

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

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

in this study when the overlapping part is omitted. Therefore, the new method automatically integrates the local truncation error or defect. This leads to new types of schemes that are naturally represented as general linear (GL) methods, which are perfectly suited for this strategy, as we demonstrate. Although these algorithms work well with variable time-steps, we do not address error control strategies in this study.

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

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

of size m with y_0 given. We will use the tensor notation denoting the components in (1.1) by $y^{\{j\}}$ and $f^{\{j\}}$, $j = 1, 2, \ldots, m$. 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 [51]; 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

$$\varepsilon(t_n) = y(t_n) - y_n\,,\ n = 1, 2, \ldots, T/\Delta t\,, \tag{1.2}$$

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 [39,42]. This study focuses on efficient a posteriori estimates of $\varepsilon(t_n)$.

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. 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 errors. Let us consider a one-step linear numerical discretization method for (1.1),

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

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

$$||y(t_n + \Delta t) - y_{n+1}|| \le C_1 \Delta t^{p+1}\,, \tag{2.2a}$$

for a constant C_1. The local error then satisfies

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

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

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

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

for a constant C_2.

This is proved in several treatises [39, 42, 64]. Under sufficient smoothness assumptions [38, 42], it follows that the *global error* satisfies

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

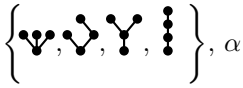
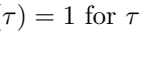
where $y_n := y_{\Delta t}(t)$ at $t = t_0 + n\Delta t$. These results are obtained by comparing the expansions of the exact and the numerical solutions. To alleviate the analysis difficulties that come with large p, we use the B-series representation of the derivatives.

DEFINITION 2.2 (Rooted trees and labeled trees [3,20]). *Let $\mathcal{T}$ be a set of ordered indexes $\mathcal{T}_q = \{j_1 < j_2 < j_3 < \cdots < j_q\}$ with cardinality q. A labeled tree of order q is a mapping $\tau : \mathcal{T}_q\backslash\{j_1\} \to \mathcal{T}_q$ such that $\tau(j) < j$, $\forall j \in \mathcal{T}_q\backslash\{j_1\}$. 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 $\sigma : \mathcal{T}_q \to \mathcal{T}_q$ 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 [3,20]). *For a labeled tree $\tau \in LT_q$ we call an elementary differential the expression*

$$F^{\{K_1\}}(\tau)(y) = \sum_{K_2,K_3,\ldots,K_q} \prod_{i=1}^{q} f^{\{K_i\}}_{\tau^{-1}(K_i)}\,, \tag{2.5}$$

where $K_1, K_2, \ldots, K_q = 1, 2, \ldots, m$, and $f^{\{J\}}_{K_1,K_2,\ldots,K_r} = \partial^r f^{\{J\}}/\partial y^{\{K_1\}} y^{\{K_2\}} \ldots y^{\{K_r\}}$. We denote by $F(\tau)(y) = [F^{\{1\}}(\tau)(y), F^{\{2\}}(\tau)(y), \ldots, F^{\{m\}}(\tau)(y)]^T$.
We use the graphical notation to represent derivatives discussed in [12, 39].

Example. The tree  corresponds to $f'f''(f,f)$. The trees of order 4 are $T_4 =$  , $\alpha(\tau) = 1$ for $\tau \in T_4\backslash$  , α  $= 3$.

DEFINITION 2.4 (B-series [41]). *Let $a : T \to \mathbb{R}$ be a mapping between the tree set and real numbers. The following is called a B-series:*

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

where $T = \{\emptyset\} \bigcup T_1 \bigcup T_2 \bigcup \cdots$.

The exact solution of an ODE system is a B-series [41]. Formally we have the following result.

THEOREM 2.5 (Exact solution as B-series [41]). *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* $\frac{\alpha(\tau)}{\rho(\tau)!}$, $\forall \tau \in T_k$, $k = 1, 2, \ldots, p$.

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

LEMMA 2.6 (Independence of elemetary differentials [12]). *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* τ_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 [39, 42]). *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*

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

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

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

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

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

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

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

It follows from Theorem 2.1 that the global error $e_p(t)$ satisfying (2.8) and

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

determines the asymptotic expansion. For more details see [39]. ∎

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. The result 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* Φ *and* $\tilde{\Phi}$ *satisfy*

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

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

where $d_{p+1}(t_n) = \frac{1}{(p+1)!}\sum_{\tau\in T_{p+1}} \alpha(\tau)a(\tau)F(\tau)(y_n)$ *with constant* $\gamma \neq 1$*, then the global error satisfies*

$$\varepsilon_p(t_n) = \frac{1}{1-\gamma}\left(\tilde{y}(t_n) - y(t_n)\right) = e_p(t_n)\Delta t^p + \mathcal{O}(\Delta t^{p+1})\,, \tag{2.13}$$

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:

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

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

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

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

for $n = 1, 2, \ldots$. ∎

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*

$$\widehat{y}_n = y_n + \varepsilon_n = \frac{1}{1-\gamma}\tilde{y}_n - \frac{\gamma}{1-\gamma}y_n\,. \tag{2.15}$$

We note that a related analysis has been carried out in [58] with an emphasis of reusing standard codes for solving ODEs with global error estimation.

The result presented above is the basis of the developments in this study. We introduce 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 [39], 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

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

and instead of (2.9) we obtain

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

Then (2.10) becomes

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

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

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

The results introduced in this study and summarized by Theorem 2.9 carry over to variable time stepping with Δt replaced by $\Delta t_{\max} = \max(\nu(t)\Delta t)$ and, therefore, allows 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 or methods used for global error estimation. Consider an asymptotic error expansion in (2.8) of

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

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

$$d(t) = \frac{d}{dt}\left[\sum_{\tau \in T_p} \alpha(\tau)\mathbf{e}(\tau)F(\tau)(y(t))\right] - \frac{\partial f}{\partial y}(y(t)) \cdot \sum_{\tau \in T_p} \alpha(\tau)\mathbf{e}(\tau)F(\tau)(y(t)) \\ = \sum_{\tau \in T_p} \alpha(\tau)\mathbf{e}(\tau)\left[\frac{d}{dt}F(\tau)(y(t)) - \frac{\partial f}{\partial y}(y(t))F(\tau)(y(t))\right]\,. \tag{2.18}$$

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 [6] in an attempt to break the order barriers of multistage methods under the alias "effective order." Stetter [65] 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 [65] 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 [6, 50, 53, 55–57, 65].

Algorithm [A:ExPrErEq]: Methods with exact principal error equation [65]
Solve

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

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

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 $p \le s$. 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 [8, 13–16, 37]. Effective order through method composition has recently been discussed in [19].

Although this concept is attractive in terms of efficiency, Prince and Wright [53] 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. 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 [69] and Skeel [63]. 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 [48, 69] and assume that there exists a solution $z(t)$ of a perturbed system

$$z(t)' = f(t, z(t)) - r(t)\,;\ z(t_0) = z_0\,,\ r(0) = y_0 - z_0\,,\ t_0 < t \le T\,, \tag{2.20}$$

close to $y(t)$. The error function (between the solutions of (1.1) and (2.20)) is given by [48]

$$e(t) = y(t) - z(t)\,, \tag{2.21}$$

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

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

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

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

$$D(t) = f(t, P(t)) - P'(t)\,. \tag{2.24}$$

Estimates of the local truncation errors can be obtained by using continuous output [31]. Lang and Verwer [48] 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, $D(t_{n+\frac{1}{2}}) = \frac{3}{2} d_{p+1}(t_n)\Delta t + \mathcal{O}(\Delta t^{p+1})$, $1 \le p \le 3$, 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

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

2.2.2. Solving the error equation. If the Jacobian of f is available, then (2.25) can be solved directly as in [48].

Algorithm [A:SoErEq]: Solving the error equation [48]
Solve

$$y' = f(t, y)\,, \qquad y(t_0) = y_0 \tag{2.26a}$$
$$\varepsilon'(t) = J\varepsilon(t) + [d_{p+1}(t_n)\Delta t]\,, \qquad \varepsilon(t_0) = 0 \tag{2.26b}$$
$$d_{p+1}(t_n) = y(t_{n+1}) - y_{n+1} + \mathcal{O}(\Delta t^{p+2})\,,\ J = \frac{\partial f}{\partial y}\,.$$

The authors of [48] 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 in presented in [52, 63, 69] and further refined in [26–29]. We start from (2.20) and denote by $P(t)$ its exact solution. Equation (2.21) becomes

$$e(t) = y(t) - P(t)\,, \text{ and} \tag{2.27a}$$
$$e'(t) = (y(t) - P(t))' = f(t, y(t)) - P(t)' = f(t, P(t) + e(t)) - P'(t)\,. \tag{2.27b}$$

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 solving for the correction procedure [63]

Algorithm [A:SolCor]: Solving for the correction [63]
Solve

$$y' = f(t, y)\,, \qquad y(t_0) = y_0 \tag{2.28a}$$
$$\varepsilon' = f(t, P(t) + \varepsilon) - P'(t)\,, \qquad \varepsilon(t_0) = 0 \tag{2.28b}$$
$$P(t) \approx y(t) - y_{\Delta t}(t)\,.$$

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 [69] 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 [69]
Solve

$$y' = f(t,y)\,, \qquad y(t_0) = y_0 \tag{2.29a}$$
$$z' = f(t,P(t)) - P'(t) - f(t,\varepsilon)\,, \qquad z(t_0) = y_0 \tag{2.29b}$$
$$\varepsilon_n = z_n - y_n \qquad P(t) = \mathcal{L}_p(y_{\Delta t}(t))\,.$$

2.3. Extrapolation approach. The global error estimation through extrapolation dates back to [54]. The procedure is the following. Propagate two solutions $y_{\Delta t,n}$ and $y_{\frac{\Delta t}{2},n}$, one with Δt 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 [42]

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

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

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

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 Δt and $\Delta t/2$

$$y' = f(t,y) \Rightarrow y_{\Delta t,n}\,,\ y_{\frac{\Delta t}{2},n}\,, \qquad y(t_0) = y_0 \tag{2.31a}$$
$$\varepsilon = \frac{2^p}{1-2^p}(y_{\Delta t,n} - y_{\frac{\Delta t}{2},n})\,. \tag{2.31b}$$

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

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

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 $\widehat{y}$ from

(2.15) and using the conditions imposed on P (see [26]). 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. General linear methods were introduced by Burrage and Butcher [2]; however, many GL-type schemes have been proposed to extend either Runge-Kutta methods [36] to linear multistep (LM) or vice versa [4, 34], as well as other extensions [5, 24, 40, 62]. 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 $\mathbb{y}^{[n-1]} = [\mathbb{y}^{[n-1]}_{(1)}\ \mathbb{y}^{[n-1]}_{(2)}\ \dots\ \mathbb{y}^{[n-1]}_{(r)}]^T$, 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 $\mathbf{f}_{(i)} = f\left(\mathbf{Y}_{(i)}\right)$, $i = 1, 2, \dots, s$, and can be compactly represented as $\mathbf{Y} = \left[\mathbf{Y}^T_{(1)}\ \mathbf{Y}^T_{(2)} \dots \mathbf{Y}^T_{(s)}\right]^T$ and $\mathbf{f} = \left[\mathbf{f}^T_{(1)}\ \mathbf{f}^T_{(2)} \dots \mathbf{f}^T_{(s)}\right]^T$.

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

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

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 $\mathbb{M}$:

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

Expression (3.1) is the most generic representation of GL methods [39, 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 $\mathbf{B}\mathbb{1} + \mathbf{V}q_1 = q_0 + q_1$ [11, Def. 3.2 and 3.3]) and stable ($\|\mathbf{V}^n\|$ remains bounded, $\forall n = 1, 2, \dots$ [11, Def. 3.1]), then the method (3.1) is convergent [11, Thm. 3.5], [12, 44]. In-depth descriptions and survey materials on GL methods can be found in [9, 11, 12, 39, 44]. In this study we use self-starting methods, and therefore $\mathbb{S} = I$. In general the initial input vector $\mathbb{y}^{[0]}$ can be generated through a "starting procedure," $\mathbb{S} = \{S_i : \mathbb{R}^m \to \mathbb{R}^m\}_{i=1\dots r}$, represented by generalized RK methods; see [12, Chap. 53] and [23]. The final solution is typically obtained by applying a "finishing procedure," $\mathbb{F} : \mathbb{R}^m \to \mathbb{R}^m$, 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 [12,22,23]. The derivatives of the numerical and exact solution are represented by rooted trees and expressed as a B-series [7,41] as delineated in Theorem 2.5 and order definition 2.7. We use an algebraic criterion characterize the order conditions for GL methods as follows. Let $\tau \in T$ and $E^{(\theta)} : T \to \mathbb{R}$, 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 $\xi_i : T \to \mathbb{R}$: $\xi_i(\emptyset) = b_0^{(i)}$, $\xi_i(\tau) = \Phi^{(i)}(\tau)$, where $\Phi^{(i)}(\tau)$, $i = 1, \ldots, r$, 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

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

where η, ηD 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 [12], 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 [22]. 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 [22, 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 $y^{[n+1]} = R(z)\, y^{[n]}$,

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

$$\Phi(w, z) = \det(wI_r - R(z))\,, \tag{3.4}$$

where $z = a\Delta t$ and $R(z)$ 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 [12].

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$\tilde{\text{y}}$ that evolves two solutions of the ODE problem y and $\tilde{y}$. Methods GLy$\tilde{\text{y}}$ take the following

form:

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

We will consider $V = I_r$, although more general forms can also be considered. The second form, denoted by GLyε, is given as a method that evolves the solution of the base method and the error explicitly, y and ε, 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* $\mathbb{y}^{[n]} = [(y^{[n]})^T, (\tilde{y}^{[n]})^T]^T$, *and* $(\mathbf{A}_{y\varepsilon}, \mathbf{U}_{y\varepsilon}, \mathbf{B}_{y\varepsilon}, \mathbf{V}_{y\varepsilon})$, *where* $\mathbb{y}^{[n]} = [(y^{[n]})^T, (\varepsilon^{[n]})^T]^T$ *are related by*

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

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

Proof. We start with a GLyε 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 $\frac{1}{1-\gamma}\left(\tilde{y}^{[n]} - y^{[n]}\right)$ as in Theorem 2.9, (2.13). The resulting expression can then be written as a GLy$\tilde{\text{y}}$ 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 $\gamma \neq 1$. ∎

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

$$
\begin{aligned}
&\{y^{[n]}, \varepsilon^{[n]}\} = \text{GLy}\varepsilon(\{y^{[n-1]}, \varepsilon^{[n-1]}\})\,, && \text{[solution, GEE]} && \text{(4.3a)}\\
&\varepsilon_{\text{loc}} = \varepsilon^{[n]} - \varepsilon^{[n-1]}\,, && \text{[local error]} && \text{(4.3b)}\\
&\widehat{y}^{[n]} = y^{[n]} + \varepsilon^{[n]} = \frac{1}{1-\gamma}\tilde{y}^{[n]} - \frac{\gamma}{1-\gamma} y^{[n]}\,. && \text{[high order]} && \text{(4.3c)}
\end{aligned}
$$

4.1. Consistency and preconsistency analysis. We now discuss consistency and preconsistency conditions in the case of a method with $r = 2$. Following [44], we require that

$$
\begin{aligned}
y_i^{[n-1]} &= q_{i,0} y(t_{n-1}) + \Delta t q_{i,1} y'(t_{n-1}) + \mathcal{O}(\Delta t^2)\,, \quad i = 1, 2 && \text{(4.4a)}\\
Y_i &= y(t_{n-1} + c_i \Delta t) + \mathcal{O}(\Delta t^2)\,, \quad i = 1, 2, \ldots, s && \text{(4.4b)}\\
y_i^{[n]} &= q_{i,0} y(t_n) + \Delta t q_{i,1} y'(t_n) + \mathcal{O}(\Delta t^2)\,, \quad i = 1, 2\,. && \text{(4.4c)}
\end{aligned}
$$

From (4.4b) we obtain

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

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. The order conditions are based on the algebraic representation of the propagation of the B-series through the GL process as discussed in §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 GLyỹ method by setting $E\xi_1(\tau) = \widehat{\xi}_1(\tau) = \widehat{\xi}_2(\tau)$, for all $\tau \in T_p$, and

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

assuming that the inputs of the GP process $\gamma\xi_1(\tau) = \xi_2(\tau)$, $\tau \in T_{p+1}$. Here $\widehat{\xi}$ 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

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

Expression (4.5a) is equivalent to imposing (2.12). We also impose stability order [10] $\tilde{p} = p + 3$: $\Phi(\exp(z), z) = \mathcal{O}(\Delta t^{\tilde{p}})$, 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 [53]. 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

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

where $\xi_{\{\ell\}}(\tau_j)$ is the coefficient of input ℓ corresponding to tree index j and $\widehat{\xi}_{\{i\}}(\tau_j)$ 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 the 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*

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

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

Proof. For the first tree $\tau_\emptyset$ we have $\eta D(\tau_\emptyset) = 0$. The next tree is $\tau_1 =$  , for which $\eta D($ $) = 1$. 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 $\eta D(\tau_2) = \eta D($ $) = \eta(\tau_1)$ 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 $\eta D(\tau_3) = \eta D($ $) = (\eta(\tau_1))^2$ and

$$\eta(\tau_3) = \mathbf{A}(\mathbf{A}\mathbb{1}_s + \mathbf{U}\xi(\tau_1))^2 + \mathbf{U}\xi(\tau_2)\,,$$

where the power is taken component-wise. The last third-order tree gives $\eta D(\tau_4) = \eta D($ $) = \eta(\tau_2))$ and

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

We then arrive at the following recurrence formula:

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

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

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

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

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

An inductive argument yields (4.8b). □

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

Proof. We use the results of Lemma 4.2 and 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

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

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

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

□

A similar calculation for $p+3$ reveals that matrices $\mathbf{BAU}$ and $\mathbf{B}\,\mathrm{diag}(\mathbf{A}\mathbb{1})\mathbf{U}$ need to have only diagonal entries.

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, it is important to 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 solving for the correction procedure (2.28) with method RK3(2)G1 (5.2) as introduced in [26]. 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 κ_i are some constants. 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, $\widehat{y}$. 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.

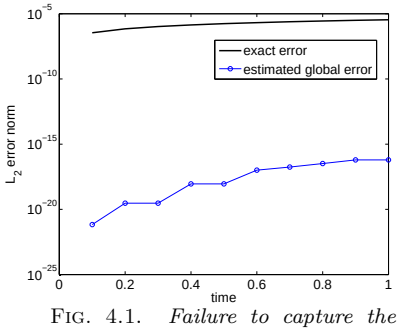


FIG. 4.1. *Failure to capture the global errors correctly for system* $y_1' = 1$, $y_2' = \kappa_2 y_1^3$, $y_3' = \kappa_3 y_1^4$ *solved with RK3(2)G1 (5.2) [26].*

4.4. Second-order explicit Runge-Kutta-type 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 via (2.15) with only two stages, which is a statement that is easy to disprove.

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

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

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

(4.9) can then be expressed as follows:

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

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

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

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

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

In (4.10) we note the Runge-Kutta structure; however, we see that the defect takes an active role in the stage calculations. By using (4.2), we obtain the GLyỹ form as

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

In particular, (4.11) is expressed as

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

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

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

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

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

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

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

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.

4.5. Third-order explicit Runge-Kutta-type 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 [23]. 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)

$$a_{2,1} = -\frac{2169604947363702313}{2431347499893714733 5}, \quad a_{3,1} = \frac{46526746497697123895}{941169174858564741 37}, \quad a_{3,2} = -\frac{10297879244026594958}{491994576037179882 19},$$

$$a_{4,1} = \frac{233647889358459824 99}{874253114447253894 46}, \quad a_{4,2} = -\frac{792051443374961166 38}{148994349441340815519}, \quad a_{4,3} = \frac{400511898593174437 82}{364876150180049843 09},$$

$$a_{5,1} = \frac{420895226640625392 05}{124911313006412840286}, \quad a_{5,2} = -\frac{150743847603427629 39}{137927286865289746282}, \quad a_{5,3} = -\frac{622746785222533710 16}{125918573676298591413},$$

$$a_{5,4} = \frac{137554757298524717 39}{792579270666516933 90}, \quad b_{1,1} = \frac{615466968374587037 23}{569825195237861608 13}, \quad b_{1,2} = -\frac{558108927928062933 55}{206957624151308356511},$$

$$b_{1,3} = \frac{240610489526763790 87}{158739347956038723465}, \quad b_{1,4} = \frac{35779720687435133 9}{75997333706771971 35}, \quad b_{1,5} = -\frac{594498329547805639 47}{137360038685338563670},$$

$$b_{2,1} = -\frac{9738262186984159168}{99299082461487742983}, \quad b_{2,2} = -\frac{327970979319486131 95}{615215656163621633 66}, \quad b_{2,3} = \frac{428955146064184206 31}{717142011885014373 36},$$

$$b_{2,4} = \frac{226085676331660650 68}{553719178056079570 03}, \quad b_{2,5} = \frac{946558094874764595 65}{151517167160302729021}, \quad u_{1,1} = \frac{708203091398346615 59}{808639235795094698 26},$$

$$u_{1,2} = \frac{100436144396748082 67}{808639235795094698 26}, \quad u_{2,1} = \frac{161694774978034105510}{106187653640211060371}, \quad u_{2,2} = -\frac{555071213378230451 39}{106187653640211060371},$$

$$u_{3,1} = \frac{784860946445662645 68}{881710308967338229 81}, \quad u_{3,2} = \frac{96849362521675584 13}{881710308967338229 81}, \quad u_{4,1} = \frac{653949221463348544 35}{845708538404054795 54},$$

$$u_{4,2} = \frac{191759316940706251 19}{845708538404054795 54}, \quad u_{5,1} = \frac{86072827701837541 08}{108658046436496925911}, \quad u_{5,2} = \frac{100050763666313171803}{108658046436496925911}.$$

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.

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 solving for the correction procedure and the extrapolation approach. Method [A:ExPrErEq] with exact principal error equation (2.19) can obviously be represented as a GL schemes. 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. Solving for the correction approach. Let us consider the Runge-Kutta methods that integrate the global errors introduced by [26,27,29,49]. The RK tableau is defined by the triplet $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and the interpolation operators by (B^*, D^*), where $B^* \cdot [\theta^0, \theta^1, \ldots, \theta^s]^T$ yields the interpolant weight vector and $D^* \cdot [\theta^0, \theta^1, \ldots, \theta^s]^T$ yields its derivative. In particular, $D^*_{ij} = B^*_{ij} \cdot j$, $j = 1, \ldots, s$. 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

$$P(t + \theta\Delta t) = y_n + \theta\Delta t \sum_{i=1}^{s} b^*_i f_i \text{ and } P'(t + \theta\Delta t) = \theta\Delta t \sum_{i=1}^{s} d^*_i f_i$$

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

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

Here we express the method for a scalar problem, in order to avoid the tensor products and represent the stacked stages in $Y_{\{1,2\}}$. For example, method RK3(2)G1 [26] is given by the following Butcher tableau:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ 1 & -1 & 2 & 0 & \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{array} , \qquad B^* = \begin{bmatrix} 1 & -\frac{3}{2} & \frac{2}{3} \\ 0 & 2 & -\frac{4}{3} \\ 0 & \frac{3}{6} & -\frac{3}{6} \\ 0 & -1 & 1 \end{bmatrix} . \tag{5.2}$$

$$D^*_{ij} = B^*_{ij} \cdot j$$

The equations to be solved when using the solving for the correction procedure [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). 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,

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

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 [53] (4.2) by

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

$$y(t) = \kappa * \exp(t) + \sin(t) . \tag{6.1b}$$

Here we take $\kappa = 0$. As a direct consequence of (6.1b), we see that 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) [53] in particular.

A similar problem [Kulikov2013I] is defined by Kulikov [47] by

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

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 in time.

Problem [Hull1972B4] is a nonlinear ODE defined in [43] (B4) by

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

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

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

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

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

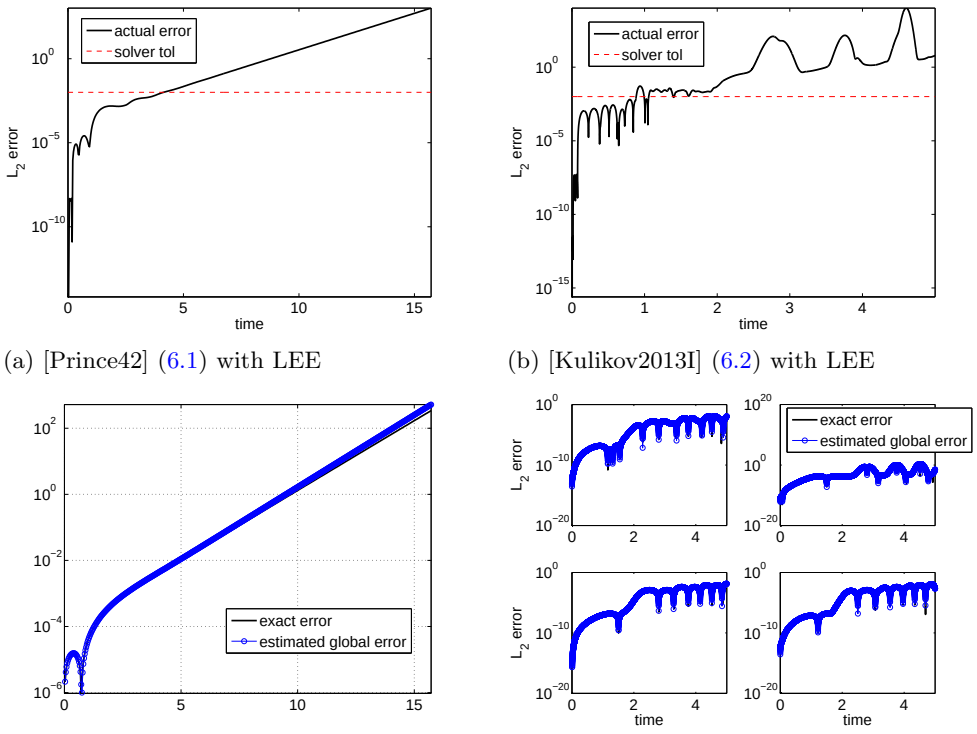
This problem allows one to choose the position of the eigenvalues of the Jacobian, $\Lambda(A)$, in order to simulate problems with different spectral properties.

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 Figure 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) – 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. In this case we observe an error drift to levels of 1e-03 over 1,000 time units. The method with GEE (4.11) can follow closely the error in time.

We next analyze the convergence properties of the methods discussed herein. 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 two (A.2) and three (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).



(a) [Prince42] (6.1) with LEE

(b) [Kulikov2013I] (6.2) with LEE

(c) problem (a) with GEE

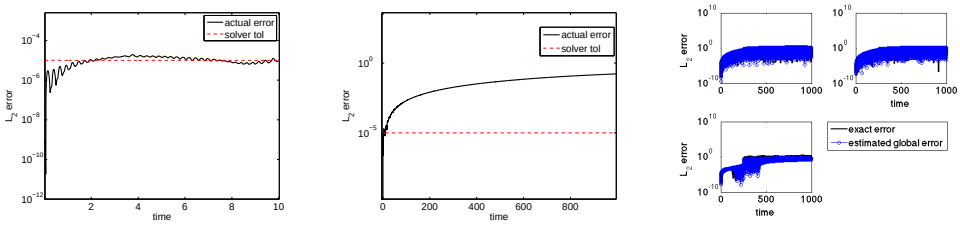
(d) problem (b) with GEE, by components

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 Fig. 6.4 we show the behavior of global error estimation when using [A:ExPrErEq], methods with exact principal error equation (2.19). Here we use method (C.1) [53, (3.11)], which fails to capture the error magnitude as discussed in [53] because the estimated error is several orders of magnitude smaller that the true global error.

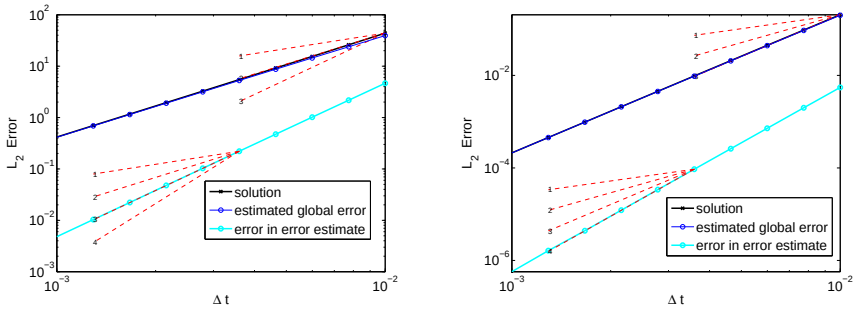
We next look at the linear stability properties of the methods introduced in this study. In Fig. 6.5(a) we delineate the stability regions according to (3.3). In Fig. 6.5(b) we show numerical results for problem [LStab2] (6.4) with $\lambda\Delta t = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\} \times (-1 \pm \sqrt{-1})$ 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).

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. We note that this strategy can be seen as a generalization of the solving for the correction procedure and of several others from the same class. We provide equivalent representation of these



(a) LEE for (6.3) (b) LEE for long times (c) GEE for long times

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 the error tolerance quite well.* (b) *However, for longer times we see an expected drift to error levels of 1e-03.* (c) *GEE method [A:GLMGEE] ((4.11) in this case) gives accurate error estimates even over long times.*



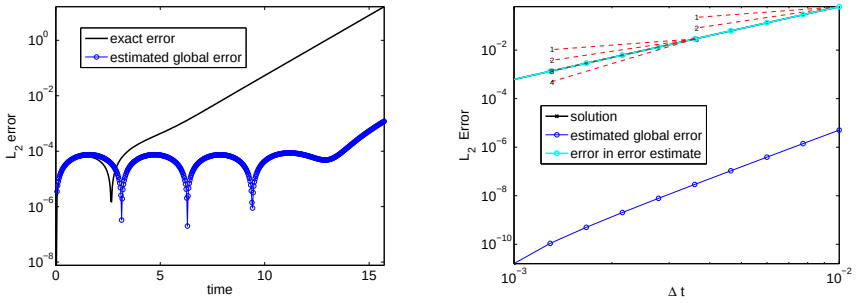
(a) 2nd order method (A.2) (b) third order method (4.14)

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 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).*

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) [65] are attractive because they offer global error estimates extremely cheaply; however, they were shown in [53] 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 [48] for low-order methods. The solving for the correction procedure [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].

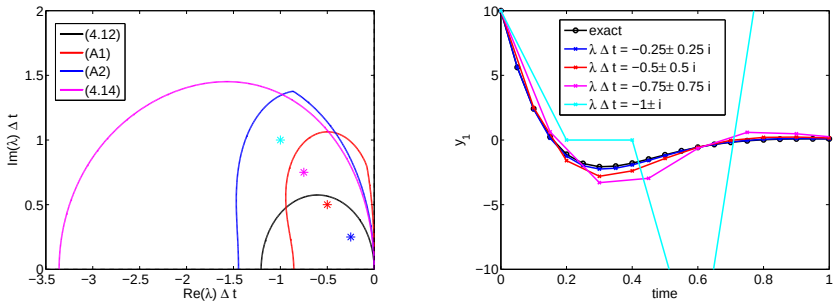
The methods introduced here are based on a general linear representation. The particular case under study is given by form (4.1); however, the analysis is not restricted to that situation. Particular instances of second and third order are presented throughout this study. The error estimates can be used for error control; however, in this study we do not address this issue.



(a) error estimation for methods [A:ExPrErEq] with exact principal error equation (2.19) for problem [Prince42]

(b) method (C.1) from [53, (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) *[53] 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).

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.

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 GLyε format is given by

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

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

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

Appendix B. RK3(2)G1 [26] in GL form (4.1). Method (5.1) corresponding to RK3(2)G1 (5.2) [26] results in the following tableau in GLyε form:

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

Appendix C. Second-order method with exact principal error equation. The following method is of type (2.19) and introduced in [53, (3.11)]:

(C.1)

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

REFERENCES

[1] 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.

[2] K. Burrage and J.C. Butcher, *Non-linear stability of a general class of differential equation methods*, BIT, 20 (1980), pp. 185–203.

[3] J.C. Butcher, *Coefficients for the study of Runge-Kutta integration processes*, Journal of the Australian Mathematical Society, 3 (1963), pp. 185–201.

[4] J.C. Butcher, *A modified multistep method for the numerical integration of ordinary differential equations*, J. ACM, 12 (1965), pp. 124–135.
[5] ———, *On the convergence of numerical solutions to ordinary differential equations*, Mathematics of Computation, 20 (1966), pp. 1–10.
[6] J.C. Butcher, *The effective order of Runge-Kutta methods*, in Conference on the numerical solution of differential equations, Springer, 1969, pp. 133–139.
[7] J.C. Butcher, *An algebraic theory of integration methods*, Mathematics of Computation, 26 (1972), pp. 79–106.
[8] J.C. Butcher, *Order and effective order*, Applied Numerical Mathematics, 28 (1998), pp. 179–191.
[9] J.C. Butcher, *General linear methods for stiff differential equations*, BIT, 41 (2001), pp. 240–264.
[10] J.C. Butcher, *The A-stability of methods with Padé and generalized Padé stability functions*, Numerical Algorithms, 31 (2002), pp. 47–58.
[11] J.C. Butcher, *General linear methods*, Acta Numerica, 15 (2006), pp. 157–256.
[12] ———, *Numerical Methods for Ordinary Differential Equations*, Wiley, second ed., 2008.
[13] J.C. Butcher and P. Chartier, *A generalization of singly-implicit Runge-Kutta methods*, Applied Numerical Mathematics, 24 (1997), pp. 343–350.
[14] ———, *The effective order of singly-implicit Runge-Kutta methods*, Numerical Algorithms, 20 (1999), pp. 269–284.
[15] J.C. Butcher and D.J.L. Chen, *ESIRK methods and variable stepsize*, Applied Numerical Mathematics, 28 (1998), pp. 193–207.
[16] J.C. Butcher and M.T. Diamantakis, *DESIRE: Diagonally extended singly implicit Runge–Kutta effective order methods*, Numerical Algorithms, 17 (1998), pp. 121–145.
[17] 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.
[18] 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.
[19] J.-C. Chang, T.M.H. Chan, and D.J.L. Chen, *Enhanced order composition methods*, Applied Numerical Mathematics, 58 (2008), pp. 223–235.
[20] P. Chartier, E. Hairer, and G. Vilmart, *Algebraic structures of B-series*, Foundations of Computational Mathematics, 10 (2010), pp. 407–427.
[21] J.M. Connors, J.W. Banks, J.A. Hittinger, and C.S. Woodward, *Adjoint error estimation for linear advection*, tech. report, Lawrence Livermore National Laboratory, Livermore, CA, 2011.
[22] E.M. Constantinescu, *On the order of general linear methods*, Applied Mathematics Letters, 22 (2009), pp. 1425–1428.
[23] E.M. Constantinescu and A. Sandu, *Optimal explicit strong-stability-preserving general linear methods*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3130–3150.
[24] 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.
[25] 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.
[26] 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.
[27] J.R. Dormand, J.P. Gilmore, and P.J. Prince, *Globally embedded Runge-Kutta schemes*, Ann. Numer. Math, 1 (1994), pp. 97–106.
[28] 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.
[29] 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.
[30] W.H. Enright, *Analysis of error control strategies for continuous Runge-Kutta methods*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 588–599.
[31] ———, *Continuous numerical methods for ODEs with defect control*, Journal of Computational and Applied Mathematics, 125 (2000), pp. 159–170.
[32] 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.
[33] 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.
[34] C.W. Gear, *Hybrid methods for initial value problems in ordinary differential equations*, SIAM Journal on Numerical Analysis, 2 (1965), pp. 69–86.
[35] 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.
[36] W.B. Gragg and H.J. Stetter, *Generalized multistep predictor-corrector methods*, J. ACM, 11 (1964), pp. 188–209.
[37] 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.
[38] E. Hairer and C. Lubich, *Asymptotic expansions of the global error of fixed-stepsize methods*, Numerische Mathematik, 45 (1984), pp. 345–360.
[39] E. Hairer, S.P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Non-stiff Problems*, Springer, 2008.
[40] E. Hairer and G. Wanner, *Multistep-multistage-multiderivative methods for ordinary differential equations*, Computing, 11 (1973), pp. 287–303.
[41] ———, *On the Butcher group and general multi-value methods*, Computing, 13 (1974), pp. 1–15.
[42] P. Henrici, *Discrete variable methods in ordinary differential equations*, New York: Wiley, 1962, 1962.
[43] 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.
[44] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, Wiley-Interscience, John Wiley & Sons, 2009.
[45] CP Jeannerod and J. Visconti, *Global error estimation for index-1 and-2 DAEs*, Numerical Algorithms, 19 (1998), pp. 111–125.
[46] G.Y. Kulikov, *Global error control in adaptive Nordsieck methods*, SIAM Journal on Scientific Computing, 34 (2012), p. A839.
[47] ———, *Cheap global error estimation in some Runge-Kutta pairs*, IMA Journal of Numerical Analysis, 33 (2013), pp. 136–163.
[48] 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.
[49] 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.
[50] P Merluzzi and C Brosilow, *Runge-Kutta integration algorithms with built-in estimates of the accumulated truncation error*, Computing, 20 (1978), pp. 1–16.
[51] J Oliver, *A curiosity of low-order explicit Runge-Kutta methods*, Mathematics of Computation, 29 (1975), pp. 1032–1036.
[52] 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.
[53] P.J. Prince and K. Wright, *Runge-Kutta processes with exact principal error equations*, IMA Journal of Applied Mathematics, 21 (1978), pp. 363–373.
[54] L.F. Richardson, *The deferred approach to the limit.*, Philosophical Transactions of the Royal Society of London, 226 (1927), pp. 299–349.
[55] WR Richert, *Implicit Runge-Kutta formulae with built-in estimates of the accumulated truncation error*, Computing, 39 (1987), pp. 353–362.
[56] D.L. Rive and F. Pasciutti, *Runge–Kutta methods with global error estimates*, IMA Journal of Applied Mathematics, 16 (1975), pp. 381–386.
[57] 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.
[58] L.F. Shampine, *Global error estimation with one-step methods*, Computers & Mathematics with Applications, 12 (1986), pp. 885–894.
[59] ———, *Error estimation and control for ODEs*, Journal of Scientific Computing, 25 (2005), pp. 3–16.
[60] L.F. Shampine and L.S. Baca, *Fixed versus variable order Runge-Kutta*, ACM Transactions on Mathematical Software (TOMS), 12 (1986), pp. 1–23.
[61] L.F. Shampine, M.K. Gordon, and J.A. Wisniewski, *Variable order Runge-Kutta codes*, tech. report, Sandia Labs., Albuquerque, NM, 1979.
[62] R. Skeel, *Analysis of fixed-stepsize methods*, SIAM Journal on Numerical Analysis, 13 (1976), pp. 664–685.

[63] R.D. Skeel, *Thirteen ways to estimate global error*, Numerische Mathematik, 48 (1986), pp. 1–20.
[64] M.N. Spijker, *On the structure of error estimates for finite-difference methods*, Numerische Mathematik, 18 (1971), pp. 73–100.
[65] H.J. Stetter, *Local estimation of the global discretization error*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 512–523.
[66] ———, *Economical global error estimation*, (1974), pp. 245–258.
[67] ———, *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.
[68] 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.
[69] P.E. Zadunaisky, *On the estimation of errors propagated in the numerical integration of ordinary differential equations*, Numerische Mathematik, 27 (1976), pp. 21–39.