Error estimation for numerical approximations of ODEs via composition techniques. Part II: BDF methods

Ahmad Deeb, Denys Dutykh, Maryam Al Zohbi

Highlights

Error estimation for numerical approximations of ODEs via composition techniques. Part II: BDF methods

Ahmad Deeb, Denys Dutykh, Maryam Al Zohbi

- The composition technique using complex coefficients is extended to the BDF scheme.
- Higher order of accuracy is reached within the real parts of outputs.
- Error estimation of order $p+1$ is provided in the imaginary parts.
- Improving the computational efficiency even with $p-1$ backward points compared to BDFp.
- Breaking the Dahlquist Barrier of order six with the composition technique.
- Adaptivity technique is analysed and lower bounds of two consecutive time steps is provided.

Error estimation for numerical approximations of ODEs via composition techniques. Part II: BDF methods

Ahmad Deeb[a], Denys Dutykh[a,b], Maryam Al Zohbi[c]

[a] *Khalifa University of Science and Technology, PO Box 127788, Abu Dhabi, UAE*
[b] *Causal Dynamics, Pty LTD, Perth, Australia*
[c] *University Cote D'Azur, LJAD, Nice, 06100, France*

Abstract

Integration of Ordinary Differential Equations (ODEs) using Backward Difference formula (BDF) methods with p backwards steps achieves order p accuracy if specific conditions are met. This work extends the composition technique with complex coefficients to the implicit BDF schemes, increasing the approximation order by one without additional backward points. The imaginary part of the composed flow provides an error estimate of order $p + 1$. Linear stability analysis reveals that the composed schemes break the Dahlquist barrier, achieving stability up to order eight. Computational performance of the composed flow outperforms BDF schemes when using the same number of backwards points, allowing reaching higher accuracy with lower CPU time. For non-uniform meshes, the ratio of consecutive time steps, which influences stability, appears as parameter in the roots of algebraic equations relative to the composed flow. Having a complex root with real positive part implies a lower bound to this ratio depending on the order. For example, the bound is 0.4506 for order three and 0.6806 for order four. Numerical tests demonstrate the effectiveness of this technique in improving the accuracy and stability compared to BDF methods.

Keywords: Backward Difference Formula, Composition technique, Stiff problems, Numerical integration, Error estimate
2008 MSC: 34E05, 65L04, 65L06, (primary) 65L50, 65L70 (secondary)

Email addresses: ahmad.deeb@ku.ac.ae (Ahmad Deeb), denys.dutykh@ku.ac.ae (Denys Dutykh), maryam.al-zohbi@univ-cotedazur.fr (Maryam Al Zohbi)

1. Introduction

In this two parts study, we are interested in providing a numerical solution and a local error estimate for the following Cauchy problem using the composition technique of BDF schemes:

$$\frac{\mathrm{dy}}{\mathrm{dt}} = f(\mathrm{t},\mathrm{y}), \quad \mathrm{t} \in \mathrm{I} =]\mathrm{t}_0, \mathbf{T}[, \qquad \mathrm{y}(\mathrm{t}_0) = \mathrm{y}_0, \tag{1}$$

with

$$\mathrm{y} : \left| \begin{array}{ccc} \mathrm{I} \subset \mathbb{R} & \to & \mathcal{U} \subset \mathbb{R}^d \\ \mathrm{t} & \mapsto & \mathrm{y}(\mathrm{t}), \end{array} \right. \qquad f : \left| \begin{array}{ccc} \mathrm{I} \times \mathcal{U} & \to & \mathbb{R}^d \\ (\mathrm{t},\mathrm{y}) & \mapsto & f(\mathrm{t},\mathrm{y}). \end{array} \right. \tag{2}$$

In real problems, the right hand side f is defined for real arguments and returns real values. In this manuscript, we consider the cases where f could also be extended to complex subsets and defined in $\mathcal{J} \times \mathcal{V}$, where $\mathrm{I} \subset \mathcal{J} \subset \mathbb{C}$ and $\mathcal{U} \subset \mathcal{V} \subset \mathbb{C}^d$. The extension, if possible, is done naturally.

Numerical approximations via different classes of schemes

Numerical integration of System (1) has been extensively investigated [1] and a wide range of numerical schemes has been proposed to approximate solutions for stiff and non-stiff problems [2, 3]. Classical numerical schemes are decomposed into two types: one-step methods as, for instance, the versatile explicit fourth-order Runge-Kutta (RK) scheme [4, 5] and Linear Multi-Step (LMS) methods such as the second order BDF2 resulting from approximating the first derivative by difference formulas. The topic of developing efficient and high order numerical methods for time integration continues to hold significance and interest in current research [6, 7, 8].

Accuracy and stability of the numerical methods are important for ensuring reliable numerical solutions [9]. Explicit RK schemes with s stages are easily available to produce numerical solutions for different orders of approximations [10, 11, 12]. The region of stability of these methods is small imposing a small time step when approximating solutions of highly stiff problems. Embedded Runge-Kutta (ERK) methods [13, 14] have been developed then to adapt the local time step and improve performance of simulation in the case of using explicit schemes. The idea consists of having two approximations of the solution with two different orders of approximation, thus an error estimate could be deduced. Bogacki and Shampine [15] developed a third-order ERK method with four stages, whilst a fifth-order ERK method was

presented by Dormand and Prince [16]. For more details about ERK methods, please refer to [17, 18]. Implicit Runge-Kutta (IRK) methods are then developed to produce numerical schemes with larger regions of stability, such as Lobatto methods [19] that are based on the trapezoidal quadrature rule, Radau [20, 21] and Gauß-RK methods [22, pp 34] which are collocation-type methods. Embedded techniques were also developed for collocation methods [23]. Other types of numerical methods exist such as Exponential-Time Differencing (ETD) methods [24, 25, 26, 27] that are based on integrating exactly, if possible, the linear part of the equation. Other classes of numerical integrators are based on Divergent Series Resumation (DSR) [28, 29, 30] such as the Borel-Padé-Laplace (BPL) that were applied to solving stiff and non-stiff problems [31, 32].

Another class of numerical methods that has been developed for solving the Cauchy problem is the so-called LMS schemes, see [2, Chap. 3] and [33, 34], as for instance the Adams-Bashforth or the Milne-Simpson methods. These methods were developed to improve stability of simulations for stiff problems [3, 35, 36]. They require additional starting points to provide numerical approximations on subsequent layers. Dahlquist [37, 38] explored and advanced the stability and convergence of these classes based on the generating polynomials of corresponding LMS methods.

Stability of BDF schemes

When having uniform mesh, explicit BDF schemes are stable up to the second order, while implicit ones are stable up to the sixth order [39]. When applied to PDEs, the second order implicit BDF scheme was studied and used for solving parabolic-type problems. Bokanowski [40] studied their stability, Wang [41] provided error estimation, Akrivis [42, 43] applied BDF schemes to the non-linear cases, Xu [44] provide a long error estimate for Delay differential equations, and Gragg and Stettar [45] used one or more off-grid points to increase the order of approximations.

BDF schemes were also developed for non-uniform mesh. Among others, Becker [46] proposed a variable time stepping technique for a parabolic problem. Stability is altered in non-uniform mesh, and was studied in the case of linear diffusion equations using a class of Discrete Orthogonal Convolution (DOC). An upper bound on the ratio between two consecutive time steps is established to maintain stability.

Increasing the order by composition techniques

Consider a situation where one has a basic one-step numerical scheme with low-order accuracy and wants to provide approximations with increased-orders. Composition techniques allow us to construct new numerical schemes with higher-orders. It was introduced in several works: Yoshida [47] used the composition to introduce symplectic integrators for Ordinary Differential Equations (ODE)s, Suzuki [48] improved the precision of the approximation of solution for two-body problems using the composition of basic numerical flows and McLachlan [49] considered a new type of differential equation $\dot{y} = A + B$, where vector fields A and B can be integrated exactly. Then, their solutions were composed to produce a general one of the initial problem. Blanes *et al.* [50, 51, 52] have developed numerical methods with high-order approximation by composing basic numerical integrators with the adjoint that is defined by the inverse flow with a negative time step, or by composing several times second or fourth-order symmetric methods. Casas *et al.* [53] have used composition of one-step methods with complex coefficients to develop pseudo-symplectic and pseudo-symmetric methods. Complex coefficients were used first by Castella *et al.* [54] for solving parabolic equations, while Blanes *et al.* [55] employed them in splitting techniques for enhancing the order of approximations [56], as negative real coefficients were inevitable for composing numerical methods with order higher than two [55].

However, composition of LMS methods has been proposed in a cyclic way, where several schemes are composed recursively. Hansen [57] established new predictor-corrector methods by the cyclic composition. Different works have been performed in the context of cyclic composition, see *e.g.* [58, 59, 60, 61], to enhance stability of numerical simulations for highly stiff differential equations. For example, a composition is obtained by computing first an approximation using the second-order Simpson rule, then marching in time using the third-order Adams-Moulton and finalize it with the Simpson rule again.

Error estimation and adaptivity

Error estimation is a must in numerical simulation provided via one-step [62, 63, 64, 65, 66, 67] or LMS methods [68, 69, 70]. It improves the stability and performance of simulations, as Adaptive Time Stepping (ATS) could be employed [71, 72]. Having a numerical scheme of order p that produces an approximation y_n, the exact local error $e(\mathrm{t}_n) \coloneqq \|\mathrm{y}(t_n) - \mathrm{y}_n\|$ could be approximated by $e_n \simeq \mathrm{C} \times \tau_n^{\mathrm{p}+1}$, where C is a positive constant

to be found. Finding this error constant for every numerical scheme is of capital importance. Hairer *et al.* [2] presented a list of error constants for RK and LMS methods. Other works have also studied the local error for RK [73, 74] and for LMS methods [75, 69]. Blanes *et al.* [76] established an error estimate of the approximation obtained by a composition technique of one-step methods when having real coefficients. This error estimates is used for adaptivity. When it comes to LMS, adaptivity affects the stability as mentioned above. Liao [77] establishes that the ratio should be smaller than 3.561 to maintain energy stability using BDF2 and smaller than 2.414 to satisfy the discrete maximum principle for the Allen-Cahn equations. Li [78] shows that the BDF2 scheme is stable for arbitrary time grids, while for BDF3, the ratio of adjacent time steps must be smaller than 2.553. Liao extends the ratio results for BDF2 and BDF3 to a class of diffusion equations [79, 80].

In recent works, Laadhari *et al.* [81, 82] have employed the composition technique with complex coefficients of (a) the second order Crank-Nicolson (CN) and (b) the implicit BDF2 both tailored for time marching of the membranes dynamic simulation of red blood cells in a quasi-Newtonian blood flow. The numerical outputs of this composition are complex values, whose real parts provide the fourth-order approximations for the CN and the third-order when developed for the BDF2. The imaginary parts provide the third-order error estimation for both resulting approximation, enabling dynamical simulation via ATS process. This technique of composition using complex coefficients has been generalized in the first part of this work [83] to any one-step method.

Novelties

In this manuscript, we extend the composition technique introduced for the implicit BDF2 scheme in [82] to any BDF of order p (BDFp) scheme with $\mathrm{p} \in \mathrm{S}_1^8$. The necessary condition for increasing the order of accuracy by one in the new (composed) scheme is the use of coefficients that are roots of associated algebraic equations. Among these roots, at least one is a complex number with a positive real part.

Performing the composition with the complex root requires computations in the complex plane, producing outputs with complex values. The real parts of these outputs represent approximations with an additional order of accuracy. While computations involving both real and imaginary parts increase the computational cost, numerical tests reveal that the composed

flow of order p, which uses $p-1$ backward points, competes with classical BDFp schemes (using p backward points) in achieving the same accuracy without additional CPU time for $p = 3$. Moreover, it outperforms classical BDFp schemes for $p > 3$, or for any order when the same number of backward points is used.

This composition technique also breaks the Dahlquist barrier of linear stability for implicit BDFp schemes, which is limited to order **six**. The new composed schemes are stable up to order **eight**. Additionally, the imaginary parts of the outputs are shown to serve as error estimates for the resulting approximations. These error estimates are then utilized in the ATS technique.

We further demonstrate that the root of the algebraic equations for compositions of order $p \in S_2^8$ will have a positive real part—necessary for stable time stepping—if the ratio of two consecutive time steps satisfies a lower bound. For instance, in the case of double composing BDF1 (a scheme of order two), the root will always have a positive real part, regardless of the ratio, aligning with the result in [78]. However, for the composition of BDF2 (a scheme of order three), the root will have a positive real part only if the ratio exceeds 0.4506. Similarly, for the fourth-order composed flow, the ratio must be greater than 0.6806. Other lower bounds for composed flows up to order eight are presented in Table 7.

The structure of the paper will be as follows. The next Section presents the mathematical framework of BDF schemes and tools we use to develop the process of composing them in Section 4. Section 3 headlines the outcomes of this work. Section 5 presents the proof of the main result. Section 6 demonstrates the convergence error of the composition. Section 7 presents the linear stability analysis of the composed flow, while Section 8 presents the variable time stepping strategy. Section 9 highlights some numerical tests that stress the usefulness of complex part in providing error estimates and discuss the computational efficiency after involving complex parts. We end with discussions, main conclusions and some perspectives in Section 10.

2. Notations and preliminaries

For every $t \in I$, we denote by φ_t the exact flow of the Cauchy problem (1). It is defined by the following map:

$$\varphi_t : \left| \begin{array}{ccc} \mathbb{R}^d & \longrightarrow & \mathbb{R}^d \\ y_0 & \mapsto & \varphi_t(y_0) = y(t), \end{array} \right. \tag{3}$$

such that y(t) is the solution to the Initial Value Problem (IVP) (1). However, it is not possible for almost all equations encountered in modeling problems to have the exact flow written explicitly with elementary or even special functions. Therefore, numerical methods are to be applied in order to provide numerical approximations to exact flows [18]. These approximations are sought on a discrete set of points $\{\mathrm{t}_0, \mathrm{t}_1, \ldots, \mathrm{t}_n\} \subset \mathrm{I}$ with $\mathrm{t}_{i-1} < \mathrm{t}_i$, $\forall i \in \mathrm{S}_0^n \equiv \{0, 1, \ldots, n\}$ where y_n denotes the approximation to $\mathrm{y}(\mathrm{t}_n)$ and $\tau_n := \mathrm{t}_n - \mathrm{t}_{n-1}$ denotes the n^{th} time step size after instant t_{n-1}. Having equidistant points t_{n-i} with fixed time step τ and approximations y_{n-i} of $\mathrm{y}(\mathrm{t}_{n-i})$ for $i \in \mathrm{S}_1^{\mathrm{p}} \equiv \{1, \ldots, \mathrm{p}\}$ (as illustrated in the below figure), we present the

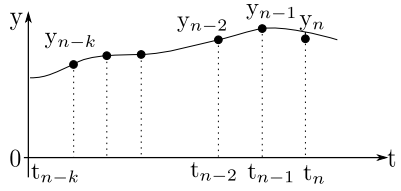

Figure 1: Illustration of the interpolation polynomial q.

general form of an LMS method that computes y_n by solving the following equation [84]:

$$\sum_{i=0}^{\mathrm{p}} \alpha_i \mathrm{y}_{n-i} = \tau \sum_{j=0}^{\mathrm{p}} \beta_j f(\mathrm{t}_{n-j}, \mathrm{y}_{n-j}), \tag{4}$$

where α_i and β_i are coefficients defining the LMS method. If $\beta_0 \neq 0$, the scheme is said to be implicit. The generating polynomials are defined as follows:

$$\varrho(\zeta) := \sum_{i=0}^{\mathrm{p}} \alpha_i \zeta^i, \quad \sigma(\zeta) := \sum_{i=0}^{\mathrm{p}} \beta_i \zeta^i. \tag{5}$$

We have the following

Theorem 1. *[2, Theorem 2.4, pp 370] An LMS method defined by Eq.* (4) *is of order* p*, if and only if one of the following conditions is satisfied:*

i) $\sum_{j=0}^{\mathrm{p}} \alpha_j = 0$ *and* $\sum_{j=0}^{\mathrm{p}} \alpha_j j^m = m \sum_{j=0}^{\mathrm{p}} \beta_j j^{m-1}$, $\forall\, m \in \mathrm{S}_1^{\mathrm{p}}$,

ii) $\varrho(\mathrm{e}^\tau) - \tau\sigma(\mathrm{e}^\tau) = \mathcal{O}(\tau^{\mathrm{p}+1})$ *for* $\tau \to 0$,

iii) $\dfrac{\varrho(\zeta)}{\log(\zeta)} - \sigma(\zeta) = \mathcal{O}\big((\zeta - 1)^{\mathrm{p}}\big)$ *for* $\zeta \to 1$.

The second point in this Theorem is used later to prove that the proposed composition of a scheme of order p is precisely of order $\mathrm{p}+1$.

2.1. BDFp

Assume we know the approximation of the solution on a set of points $\{\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}\}$, and we are looking to find an approximation y_n of $\mathrm{y}(\mathrm{t}_n)$. We consider the polynomial $q(\mathrm{t})$ interpolating the points $\left\{(\mathrm{t}_i, \mathrm{y}_i) \,|\, i \in \mathrm{S}_{n-\mathrm{p}}^{n}\right\}$.

2.1.1. Fixed time step

This polynomial could be expressed in terms of backward differences

$$\nabla^0 \mathrm{y}_n = \mathrm{y}_n, \quad \nabla^{j+1}\mathrm{y}_n = \nabla^j \mathrm{y}_n - \nabla^j \mathrm{y}_{n-1}, \tag{6}$$

for all $\mathrm{t} \in [\mathrm{t}_{n-1}, \mathrm{t}_n]$, such that $\mathrm{t}_n = \mathrm{t}_{n-1} + \tau$, as follows (see [2, pp 357]):

$$q(\mathrm{t}) \equiv q(\mathrm{t}_{n-1} + s\tau) = \sum_{j=0}^{\mathrm{p}} (-1)^j \binom{-s+1}{j} \nabla^j \mathrm{y}_n\,, \quad \forall s \in [0,1], \tag{7}$$

where $\binom{s}{j} \;\coloneqq\; \dfrac{s!}{j!(s-j)!}$ is the binomial coefficient. The implicit BDFp consists of determining the unknown y_n such that the classical interpolation polynomial $q(\mathrm{t})$ satisfies the following condition:

$$\frac{\mathrm{d}}{\mathrm{dt}} q(\mathrm{t})\bigg|_{\mathrm{t}=\mathrm{t}_n} = f(\mathrm{t}_n, \mathrm{y}_n). \tag{8}$$

Applying this condition to Eq. (7), we get the following relation:

$$\sum_{j=0}^{\mathrm{p}} \eta_j^* \nabla^j \mathrm{y}_n = \tau f(\mathrm{t}_n, \mathrm{y}_n), \quad \text{with } \eta_j^* \;\coloneqq\; (-1)^j \,\frac{\mathrm{d}}{\mathrm{d}s}\binom{-s+1}{j}\bigg|_{s=1}, \tag{9}$$

where $(.)|_{s=1}$ represents the evaluation operator at $s=1$. Though, we have $\eta_0^* = 0$ and $\eta_j^* = 1/j$ for $j \in \mathrm{S}_1^{\mathrm{p}}$. By replacing these coefficients and using difference formulas, we have:

$$\begin{aligned} \sum_{j=1}^{\mathrm{p}} \frac{1}{j} \nabla^j \mathrm{y}_n &= \sum_{j=1}^{\mathrm{p}} \frac{1}{j} \left[\sum_{i=0}^{j} (-1)^i \binom{j}{i} \mathrm{y}_{n-i}\right], \\ &\qquad \text{by permuting the } \sum \\ &= \sum_{i=0}^{\mathrm{p}} \left[(-1)^i \sum_{j=i+1}^{\mathrm{p}} \frac{1}{j}\binom{j}{i}\right] \mathrm{y}_{n-i} = \sum_{i=0}^{\mathrm{p}} \gamma_{(i)} \mathrm{y}_{n-i}. \end{aligned} \tag{10}$$

Table 1: Coefficients $\gamma_{(i)}$ of BDF with fixed time step.

p	$\gamma_{(0)}$	$\gamma_{(1)}$	$\gamma_{(2)}$	$\gamma_{(3)}$	$\gamma_{(4)}$	$\gamma_{(5)}$
1	1	-1				
2	$\frac{3}{2}$	-2	$\frac{1}{2}$			
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$		
4	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$	
5	$\frac{137}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$

We present in Table 1 coefficients $\{\gamma_{(i)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$ for $\mathrm{p} \in \mathrm{S}_1^5$. One has explicitly $\{\gamma_{(j)} \,|\, j \in \mathrm{S}_0^{\mathrm{p}}\}$, y_n is the solution of the following equation.

$$\sum_{j=0}^{\mathrm{p}} \gamma_{(j)} \mathrm{y}_{n-j} = \tau f(\mathrm{t}_n, \mathrm{y}_n). \tag{11}$$

We present another way to find coefficients $\gamma_{(i)}$. It is based on the point (ii) in Theorem 1. We consider a linear equation ($f(\mathrm{t}, \mathrm{y}) \equiv \mathrm{y}$). Though, all preceding solutions are given as a function of y_n such that: $\mathrm{y}_{n-j} = \mathrm{y}_n \mathrm{e}^{-j\tau}$. Substituting $\mathrm{y}_{n-j}, \forall\, j \in \mathrm{S}_0^{\mathrm{p}}$ in Eq. (11) allows us to write:

$$\begin{aligned} \cancel{\mathrm{y}_n} \sum_{j=0}^{\mathrm{p}} \gamma_{(j)} \mathrm{e}^{-j\tau} = \tau \cancel{\mathrm{y}_n} \quad \Longrightarrow \quad \sum_{j=0}^{\mathrm{p}} \gamma_{(j)} \left(1 + \sum_{m=1}^{\infty} \frac{(-j\tau)^m}{m!} \right) &= \tau, \\ \sum_{j=0}^{\mathrm{p}} \gamma_{(j)} + \tau \sum_{j=0}^{\mathrm{p}} -i\, \gamma_{(j)} + \sum_{m=2}^{\infty} \frac{1}{m!} \left(\sum_{j=0}^{\mathrm{p}} (-j)^m \gamma_{(j)} \right) \tau^m &= \tau. \end{aligned}$$

To obtain the p^{th} order approximation, coefficients $\{\gamma_{(j)} \,|\, j \in \mathrm{S}_0^{\mathrm{p}}\}$ must verify the following linear system:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 2 & \dots & \mathrm{p} \\ 0 & 1 & 2^2 & \dots & \mathrm{p}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2^{\mathrm{p}} & \dots & \mathrm{p}^{\mathrm{p}} \end{pmatrix} \cdot \begin{pmatrix} \gamma_{(0)} \\ \gamma_{(1)} \\ \gamma_{(2)} \\ \vdots \\ \gamma_{(\mathrm{p})} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{12}$$

Solving the linear system produces coefficients for any order p, including those were listed in Table 1 for $\mathrm{p} \in \mathrm{S}_1^5$. We end by presenting the difference operator, L, associated to a BDFp with a fixed time step τ:

$$L\big(\mathrm{y}, \mathrm{t}, \tau\big) \coloneqq \sum_{j=0}^{\mathrm{p}} \Big(\gamma_{(j)} \mathrm{y}(\mathrm{t} - j\tau) - \tau \frac{\mathrm{dy}}{\mathrm{dt}}(\mathrm{t})\Big). \tag{13}$$

The difference operator L is of great importance to determine the local error and order convergence. If one can write the difference operator L in its Taylor development as follows:

$$L\big(\mathrm{y}, \mathrm{t}, \tau\big) = \sum_{j=0}^{\infty} \mathrm{E}_j \tau^j \frac{\mathrm{d}^j \mathrm{y}}{\mathrm{dt}^j}(\mathrm{t}), \tag{14}$$

thus according to Theorem 1 the scheme is of order p if coefficients $\mathrm{E}_j = 0$ for $j \in \mathrm{S}_0^{\mathrm{p}}$. We can use Lemma 2 in [2, page 369] to find the following error using $\mathrm{E}_{\mathrm{p}+1}$:

$$\mathrm{y}(\mathrm{t}_{n+1}) - \mathrm{y}_{n+1} = \frac{1}{\gamma_{(0|n)}} \mathrm{E}_{\mathrm{p}+1} \tau^{\mathrm{p}+1} \frac{\mathrm{d}^{\mathrm{p}+1} \mathrm{y}}{\mathrm{dt}^{\mathrm{p}+1}}(\mathrm{t}_n) + O(\tau^{\mathrm{p}+2}). \tag{15}$$

This will be of use in Section 5. In the next subsection, we will present the procedure to compute coefficients $\gamma_{(j|n)}$ in the case of the variable time step.

2.1.2. Variable time stepping

When having a variable step size $\tau_n = \mathrm{t}_n - \mathrm{t}_{n-1}$, the polynomial $q(\mathrm{t})$ that interpolates the set $\big\{(\mathrm{t}_i, \mathrm{y}_i) |\, i \in \mathrm{S}_0^{\mathrm{p}}\big\}$ can be expressed as follows (see [2, pp 400]):

$$q(\mathrm{t}) \equiv \sum_{j=0}^{\mathrm{p}} \prod_{i=0}^{j-1} (\mathrm{t} - \mathrm{t}_{n-i}) \times \delta^j \mathrm{y}[\mathrm{t}_n, \mathrm{t}_{n-1}, \ldots, \mathrm{t}_{n-j}], \tag{16}$$

where the divided differences $\delta^j \mathrm{y}[\mathrm{t}_n, \ldots, \mathrm{t}_{n-j}]$ are defined recursively by:

$$\begin{aligned} \delta^0 \mathrm{y}[\mathrm{t}_n] &\coloneqq \mathrm{y}_n, \\ \delta^j \mathrm{y}[\mathrm{t}_n, .., \mathrm{t}_{n-j}] &\coloneqq \frac{\delta^{j-1} \mathrm{y}[\mathrm{t}_n, .., \mathrm{t}_{n-j+1}] - \delta^{j-1} \mathrm{y}[\mathrm{t}_{n-1}, .., \mathrm{t}_{n-j}]}{\mathrm{t}_n - \mathrm{t}_{n-j}} \equiv \sum_{i=0}^{j} c_{i,j} \mathrm{y}_{n-i}. \end{aligned}$$

Applying condition (8) on Eq. (16) leads us to the variable step size formula:

$$\sum_{j=1}^{\mathrm{p}} \tau_n \prod_{i=1}^{j-1} (\mathrm{t}_n - \mathrm{t}_{n-i}) \times \delta^j \mathrm{y}[\mathrm{t}_n, \mathrm{t}_{n-1}, \ldots, \mathrm{t}_{n-j}] = \tau_n f(\mathrm{t}_n, \mathrm{y}_n).$$

Replacing the formula for $\delta^j \mathrm{y}[\mathrm{t}_n, \ldots, \mathrm{t}_{n-j}]$ in the above equation and permuting the sum leads us to find y_n that is the solution of:

$$\sum_{j=0}^{\mathrm{p}} \gamma_{(j|n)} \mathrm{y}_{n-j} = \tau_n f(\mathrm{t}_n, \mathrm{y}_n), \tag{17}$$

where coefficients $\gamma_{(j|n)}$ are given below :

$$\left\{ \begin{array}{rcl} \gamma_{(j|n)} & = & \displaystyle\sum_{m=j}^{\mathrm{p}} b_m c_{j,m}, \quad \text{with} \\ b_m & = & \displaystyle\tau_n \prod_{\ell=1}^{m-1} (\mathrm{t}_n - \mathrm{t}_{n-\ell}), m \in \mathrm{S}_1^{\mathrm{p}}, \quad b_0 = 0, \quad \text{and} \\ c_{j,m} & = & \displaystyle\prod_{\substack{\ell=0 \\ \ell \neq j}}^{m} \frac{1}{\mathrm{t}_{n-j} - \mathrm{t}_{n-\ell}}, \quad j \in \mathrm{S}_0^{\mathrm{p}}, \quad m \in \mathrm{S}_{\max(1,j)}^{\mathrm{p}}. \end{array} \right. \tag{18}$$

Algorithm 4 in Appendix D, labeled Coeff, exhibits the computation of $\{\gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$ for any ordered set of points $\{\mathrm{t}_i \wedge \mathrm{t}_i < \mathrm{t}_{i+1} \,|\, i \in \mathrm{S}_{n-\mathrm{p}}^n \wedge n \geqslant \mathrm{p}\}$.

2.2. Fixed point algorithm

In nonlinear equations, y_n is obtained by solving an iterative fixed point problem $\mathrm{F}_{\mathrm{p}}\big([\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}], \mathrm{y}_n\big) = \mathrm{y}_n$, with:

$$\mathrm{F}_{\mathrm{p}}\left([\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}], \mathrm{y}_n\right) \;\coloneqq\; \frac{1}{\gamma_{(0|n)}}\Big(\tau_n f(\mathrm{t}_n, \mathrm{y}_n) - \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)} \mathrm{y}_{n-i}\Big). \tag{19}$$

The algorithm produces a series of approximations $\mathrm{y}_{n,\ell}$ to $\mathrm{y}(\mathrm{t}_n)$, though it requires the initialization $\mathrm{y}_{n,0}$, which is equal to y_{n-1} in most considered cases. This is represented by the following system:

$$\left\{ \begin{array}{rl} \mathrm{y}_{n,0} & = \mathrm{y}_{n-1}, \\ \mathrm{y}_{n,\ell+1} & = \mathrm{F}_{\mathrm{p}}\big([\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}], \mathrm{y}_{n,\ell}\big). \end{array} \right. \tag{20}$$

The series of iterations is stopped when two consecutive approximations are close in norm to to a pre-defined user tolerance tol parameter:

$$e_{n,\ell+1} = \|\mathrm{y}_{n,\ell+1} - \mathrm{y}_{n,\ell}\| < \mathsf{tol}.$$

The fixed point algorithm converges if the following inequality is verified when x belongs to a neighborhood of the exact solution $\mathrm{y}(\mathrm{t}_n)$:

$$\left|\frac{\partial \mathrm{F_p}}{\partial \mathrm{y}_n}(\mathrm{x})\right| < 1.$$

To accelerate the convergence rate, one can use the Aitken's delta square scalar acceleration process [85, 86].

2.3. Numerical flow

We define the numerical flow associated to the BDFp:

Definition 1. *We define* $\mathcal{T}_{n-1,\mathrm{p}} \coloneqq (\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}) \in \mathbb{R}^{\mathrm{p}}$ *and* $\mathrm{Y}_{n-1,\mathrm{p}} \coloneqq (\mathrm{y}_{n-\mathrm{p}}, \ldots, \mathrm{y}_{n-1}) \in (\mathbb{R}^d)^{\mathrm{p}}$. *The numerical flow* $\Phi_{\tau}^{[\mathrm{p}]}$ *of BDFp is defined below:*

$$\Phi_{\tau_n}^{[\mathrm{p}]} : \left| \begin{array}{ccc} \mathbb{R}^{\mathrm{p}} \times (\mathbb{R}^d)^{\mathrm{p}} & \longrightarrow & \mathbb{R}^{\mathrm{p}} \times (\mathbb{R}^d)^{\mathrm{p}}, \\ (\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}) & \longmapsto & (\mathcal{T}_{n,\mathrm{p}}, \mathrm{Y}_{n,\mathrm{p}}). \end{array} \right. \tag{21}$$

It has outputs $\mathcal{T}_{n,\mathrm{p}} \coloneqq (\mathrm{t}_{n-\mathrm{p}+1}, \ldots, \mathrm{t}_n) \in \mathbb{R}^{\mathrm{p}}$ *with* $\mathrm{t}_n = \mathrm{t}_{n-1} + \tau_n$ *and* $\mathrm{Y}_{n,\mathrm{p}} \coloneqq (\mathrm{y}_{n-\mathrm{p}+1}, \ldots, \mathrm{y}_n)$ *with* y_n *solution to Eq.* (17).

The numerical flow associated to BDFp scheme is implemented in Algorithm 3 in Appendix A, relying on Function Coeff schematized in Algorithm 4. The purpose of presenting these formulations here is to prepare the ground for the proposed algorithm.

3. Results/Findings

Consider the vector $\mathcal{T}_{n-1,\mathrm{p}}$ of p time instants where at every instant t_{n-i}, $i \in \mathrm{S}_1^{\mathrm{p}}$, there is an approximation of the solution of order p. Finding an approximation y_n of $\mathrm{y}(\mathrm{t}_n)$ at $\mathrm{t}_n \equiv \mathrm{t}_{n-1} + \tau_n$ by composing twice the numerical flow $\Phi^{[\mathrm{p}]}$ consists to advance first to an intermediate point between t_{n-1} and t_n, denoted by $\mathrm{t}_{n-1/2} \coloneqq \mathrm{t}_{n-1} + \kappa_1 \tau_n$. Here κ_1 is to be determined by solving an algebraic equation related to the set of points $\{t_{n-i} \,|\, i \in \mathrm{S}_1^{\mathrm{p}}\}$. We denote

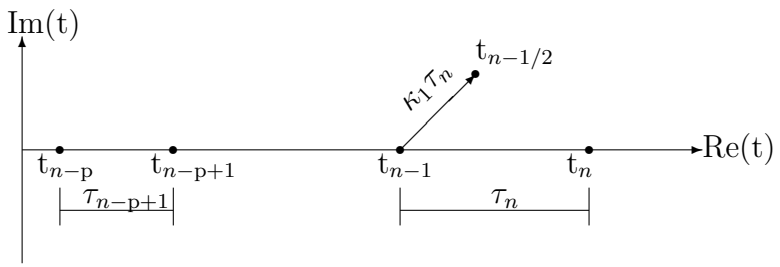

Figure 2: Illustration of the composition technique in the complex plane.

by $(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}})$ the outputs of the numerical flow such that $\mathrm{Y}_{n-1/2,\mathrm{p}}$ contains $\mathrm{y}_{n-1/2}$ to be an approximation of $\mathrm{y}(\mathrm{t}_{n-1/2})$.

$$\begin{cases} (\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}}) & \coloneqq \Phi^{[\mathrm{p}]}_{\kappa_1\tau_n}\big(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}\big), \\ \mathcal{T}_{n-1/2,\mathrm{p}} & \coloneqq (\mathrm{t}_{n-\mathrm{p}+1}, \ldots, \mathrm{t}_{n-1}, \mathrm{t}_{n-1/2}), \\ \mathrm{Y}_{n-1/2,\mathrm{p}} & \coloneqq (\mathrm{y}_{n-\mathrm{p}+1}, \ldots, \mathrm{y}_{n-1}, \mathrm{y}_{n-1/2}). \end{cases}$$

We apply again the numerical flow $\Phi^{[\mathrm{p}]}$ having $(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}})$ to be its inputs, and integrating with the time step $\kappa_2\tau_n \coloneqq (1-\kappa_1)\tau_n$. We denote by:

$$\begin{cases} \big(\mathcal{T}'_{n,\mathrm{p}}, \mathrm{Y}'_{n,\mathrm{p}}\big) & \coloneqq \Phi^{[\mathrm{p}]}_{\kappa_2\tau_n}\big(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}}\big), \\ \mathcal{T}'_{n,\mathrm{p}} & \coloneqq (\mathrm{t}_{n-\mathrm{p}+2}, \ldots, \mathrm{t}_{n-1}, \mathrm{t}_{n-1/2}, \mathrm{t}_n), \\ \mathrm{Y}'_{n,\mathrm{p}} & \coloneqq (\mathrm{y}_{n-\mathrm{p}+2}, \ldots, \mathrm{y}_{n-1}, \mathrm{y}_{n-1/2}, \hat{\mathrm{y}}_n). \end{cases}$$

We define also the following ratios:

$$\begin{cases} \varepsilon_{(j|n)} & \coloneqq \dfrac{\mathrm{t}_{n-1/2} - \mathrm{t}_{n-j}}{\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}}, \\ r_{(j|n)} & \coloneqq \dfrac{\mathrm{t}_{n-1} - \mathrm{t}_{n-j}}{\mathrm{t}_n - \mathrm{t}_{n-1}}, \end{cases} \quad \forall j \in \mathrm{S}_1^{\mathrm{p}}$$

and by using $\kappa_1 \coloneqq \dfrac{\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}}{\mathrm{t}_n - \mathrm{t}_{n-1}}$, we have:

$$\varepsilon_{(j|n)} \equiv 1 + \frac{r_{(j|n)}}{\kappa_1}, \quad \forall j \in \mathrm{S}_1^{\mathrm{p}}. \tag{22}$$

We enunciate the following

Theorem 2. *Consider having* $\big(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}\big)$ *and the numerical flow* $\Phi^{[\mathrm{p}]}$ *given in Definition 1 in Section 2.3. If* κ_1 *and* $\kappa_2 \in \mathbb{C}$ *verifying:*

$$\begin{cases} \kappa_1 + \kappa_2 = 1, \\ \varepsilon_{(\mathrm{p}|n)}\kappa_1^2 + \gamma_{(0|n)}\kappa_2^2 = 0 \end{cases} \tag{23}$$

with $\gamma_{(0|n)} \coloneqq \displaystyle\sum_{i=1}^{\mathrm{p}} \frac{1}{\varepsilon_{(i|n)}}$ *and if*

$$\mathrm{y}(\mathrm{t}_{n-i}) - \mathrm{y}_{n-i} = \mathcal{O}(\tau_{n-i}^{\mathrm{p}+1}) \quad \forall i \in \mathrm{S}_1^{\mathrm{p}},$$

then the following composition:

$$\Phi^{[\mathrm{p}]}_{\kappa_2\tau_n} \circ \Phi^{[\mathrm{p}]}_{\kappa_1\tau_n}\big(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}\big) \tag{24}$$

produces the couple $(\mathcal{T}'_{n,\mathrm{p}}, \mathrm{Y}'_{n,\mathrm{p}})$ *with* $\hat{\mathrm{y}}_n$*, in* $\mathrm{Y}'_{n,\mathrm{p}}$*, approximating* $\mathrm{y}(\mathrm{t}_n)$ *at* $\mathrm{t}_n = \mathrm{t}_{n-1} + \tau_n$ *such that:*

$$\mathrm{y}(\mathrm{t}_n) - \mathrm{Re}(\hat{\mathrm{y}}_n) = \mathcal{O}(\tau_n^{\mathrm{p}+2}). \tag{25}$$

In addition, we have the following error estimation:

$$|\mathrm{y}(\mathrm{t}_n) - \mathrm{Re}(\hat{\mathrm{y}}_n)| \sim \mathcal{C}_{\mathrm{p}} \times |\mathrm{Im}(\hat{\mathrm{y}}_n)| \tag{26}$$

with $\mathcal{C}_{\mathrm{p}}$ *is to be determined in Eq.* (59)*.*

This theorem shows that the composition employs complex arithmetic operations and **produces** $\hat{\mathrm{y}}_n$ **such that its real part approximates the solution with an additional order of accuracy to the basic numerical flow**. One chooses the next time step $\tau_n = \mathrm{t}_n - \mathrm{t}_{n-1}$, Theorem 2 shows that κ_1 is the solution of an algebraic equation depending on coefficients $r_{(j|n)}, j \in \mathrm{S}_1^{\mathrm{p}}$. It provides also in $\mathrm{Im}(\hat{\mathrm{y}}_n)$ an error estimate of order p+1 to the approximation $\mathrm{Re}(\hat{\mathrm{y}}_n)$. The proof of this theorem will be given in Section 5. We present in Definition 2 the composition, denoted by $\Psi^{[\mathrm{p}+1]}$, in a compact form.

Definition 2. *Consider the IVP defined by* (1) *and having the numerical flow* $\Phi^{[\mathrm{p}]}$ *associated to a BDFp scheme. Consider having the first* p *approximations* $\mathrm{y}_0, \ldots, \mathrm{y}_{\mathrm{p}+1}$ *to* $\mathrm{y}(\mathrm{t}_0), \ldots, \mathrm{y}(\mathrm{t}_{\mathrm{p}+1})$*. We define the new (composed) numerical flow by:*

$$\Psi^{[\mathrm{p}+1]}_{\tau_n} : \left| \begin{array}{ccc} \mathbb{R}^{\mathrm{p}} \times (\mathbb{R}^d)^{\mathrm{p}} & \to & \mathbb{R}^{\mathrm{p}} \times (\mathbb{R}^d)^{\mathrm{p}} \\ (\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}) & \mapsto & (\mathcal{T}_{n,\mathrm{p}}, \mathrm{Y}_{n,\mathrm{p}}), \end{array} \right. \tag{27}$$

where $\mathcal{T}_{n,\mathrm{p}} \coloneqq (\mathrm{t}_{n-\mathrm{p}+1}, \ldots, \mathrm{t}_n)$*,* $\mathrm{Y}_{n,\mathrm{p}} \coloneqq (\mathrm{y}_{n-\mathrm{p}+1}, \ldots, \mathrm{y}_{n-1}, \mathrm{y}_n)$ *such that* $\mathrm{y}_n = \mathrm{Re}(\hat{\mathrm{y}}_n)$ *with* $\hat{\mathrm{y}}_n$ *obtained by the composition in Theorem 2.*

We present the process of composition in Algorithm 2. After calculating coefficients $r_{(j|n)}$ and elaborating algebraic expressions $\varepsilon_{(\mathrm{p}|n)}$ and $\gamma_{(0|n)}$ that are functions of κ_1, roots κ_1 and κ_2 are to be determined by solving the algebraic system (23) with an appropriate software (Maple, Python, Matlab, Mathematica, ...). After choosing the complex root with a positive real part, the composition is established and the real part of $\hat{\mathrm{y}}_n$ is known to approximate $\mathrm{y}(\mathrm{t}_n)$. Once the approximation is obtained by the composition, we show that the imaginary part $\mathrm{Im}(\hat{\mathrm{y}}_n)$ is an error estimate of the produced approximation as stated in the last point of Theorem 2. Regarding the linear stability of the composition we have the following result.

Theorem 3. *The numerical flow* $\Psi^{[\mathrm{p}]}$ *is* $\mathrm{A}(\vartheta)$*-stable up to order* $\mathrm{p} = 8$ *with stability angles reported in Table 4.*

Algorithm 1 Adaptive numerical simulation for (1) using $\Psi^{[\mathrm{p}+1]}$

Require: f, t_0, y_0, $\mathbf{T}$, τ, tol, p
  for $i \leftarrow 1$ to $\mathrm{p}-1$ **do**
    $\tau_i \leftarrow \tau$
    $\mathrm{t}_i \leftarrow \mathrm{t}_{i-1} + \tau_i$
    Construct y_i, approximation to $\mathrm{y}(\mathrm{t}_i)$ of order p ▷ (See [83])
  end for
  $\tau_n \leftarrow \tau$
  $\mathrm{t}_n \leftarrow \mathrm{t}_{\mathrm{p}-1} + \tau_n$
  while $\mathrm{t}_n \leqslant \mathbf{T}$ **do**
    $\mathcal{T}_{n-1,\mathrm{p}} \leftarrow [\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}]$
    $\mathrm{Y}_{n-1,\mathrm{p}} \leftarrow [\mathrm{y}_{n-\mathrm{p}}, \ldots, \mathrm{y}_{n-1}]$
    $(\mathcal{T}_{n,\mathrm{p}}, \mathrm{Y}_{n,\mathrm{p}}) \leftarrow \Psi_{\tau_n}^{[\mathrm{p}+1]}(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}})$
    $e_n \leftarrow \mathcal{C}_{\mathrm{p}} \times \mathrm{Im}(\hat{\mathrm{y}}_n)$
    $\tau_{n+1} \leftarrow \tau_n \times \left(\dfrac{\mathsf{tol}}{|e_n|}\right)^{\frac{1}{\mathrm{p}+2}}$ ▷ (update the time step)
    $\tau_{n+1} \leftarrow \min\left(\max\left(\tau_{n+1}, \frac{\tau_n}{\ell_{\mathrm{p}}}\right), \tau_n \times \ell_{\mathrm{p}}\right)$ ▷ (ℓ_{p} in Eq. (69))
    $\mathrm{t}_{n+1} \leftarrow \mathrm{t}_n + \tau_{n+1}$
    $n \leftarrow n + 1$
  end while

The proof of Theorem 3 is done graphically in Section 7. To this end, we present the full outcome of this work in Algorithm 1. The Algorithm presents

the process of computing approximations using the composed flow with ATS. The loop **for** aims to initialize processing $\Psi^{[\mathrm{p}+1]}$ by finding approximations of $\mathrm{y}(\mathrm{t}_j)$ of order p with $\mathrm{t}_j = \mathrm{t}_0 + j\tau$, for $j \in \mathrm{S}_1^{\mathrm{p}-1}$. It could be done using the composition proposed in Part I [83]. The time step is updated regarding the rate between the user tolerance tol and the imaginary part representing an error estimate. A bound limit on the ratio between two consecutive time step is applied to ensure the validity of the time marching. We will show by numerical experiments in Section 6 the convergence error of the composition and in Section 9 that the imaginary part is a reliable estimation of the local error.

4. Procedure of composing BDFp

In this section, having the numerical flow $\Phi^{[\mathrm{p}]}$ of BDFp scheme, we establish the composition procedure generating $\Psi^{[\mathrm{p}+1]}$ to increase by one the order of approximation. We first generate coefficients $\{\gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$, using the last p points $\{\mathrm{t}_{n-i} \,|\, i \in \mathrm{S}_1^{\mathrm{p}}\}$ and $\mathrm{t}_{n-1/2} \;:=\; \mathrm{t}_{n-1} + \kappa_1\tau_n$ to integrate the equation by the numerical flow $\Phi^{[\mathrm{p}]}_{\kappa_1\tau_n}$, though finding approximation $\mathrm{y}_{n-1/2}$ to $\mathrm{y}(\mathrm{t}_{n-1/2})$. Then, we generate coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$ using the last $\mathrm{p}+1$ points $\{\mathrm{t}_{n-i} \wedge \mathrm{t}_{n-1/2} \,|\, i \in \mathrm{S}_1^{\mathrm{p}}\}$ and $\mathrm{t}_n \equiv \mathrm{t}_{n-1/2} + (1-\kappa_1)\tau_n$ to integrate with $\Phi^{[\mathrm{p}+1]}_{\kappa_2\tau_n}$ finding y_n. We select the step $\kappa_1\tau_n$ such that it cancels $\mathrm{y}_{n-\mathrm{p}}$ in the second integration, *i.e,* the coefficient $\Gamma_{(\mathrm{p}+1|n)}$ in front of $\mathrm{y}_{n-\mathrm{p}}$ must vanish.

4.1. Intermediate solution: first step

First, BDFp is used to compute an intermediate approximation $\mathrm{y}_{n-1/2}$ of $\mathrm{y}(\mathrm{t}_{n-1/2})$. For that, we solve the following equation:

$$\gamma_{(0|n)}\mathrm{y}_{n-1/2} + \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)}\mathrm{y}_{n-i} = (\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) f(\mathrm{t}_{n-1/2}, \mathrm{y}_{n-1/2}) \tag{28}$$

that is represented by the associated numerical flow:

$$\Big(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}}\Big) = \Phi^{[\mathrm{p}]}_{\kappa_1\tau_n}\Big(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}}\Big), \tag{29}$$

where $\mathcal{T}_{n-1/2,\mathrm{p}}$ and $\mathrm{Y}_{n-1/2,\mathrm{p}}$ are defined in Section 3. For every coefficient κ_1 and time step τ_n, coefficients $\{\gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$ could be obtained using the technique presented in [18, Appendix G.4]. Consider a linear differential equation

$f(\mathrm{t},\mathrm{y}) = \mathrm{y}$ with $\mathrm{y}(\mathrm{t}) = \mathrm{y}_{n-1/2}\,\mathrm{e}^{-(\mathrm{t}_{n-1/2}-t)}$ being its exact analytical solution. Consider also $\mathrm{y}_{n-i}, i \in \mathrm{S}_1^{\mathrm{p}}$ are known exactly; $\mathrm{y}_{n-i} = \mathrm{y}_{n-1/2}\,\mathrm{e}^{-(\mathrm{t}_{n-1/2}-t_{n-i})}, i \in \mathrm{S}_1^{\mathrm{p}}$, thus, they are to be substituted in Eq. (28). After simplifying by $\mathrm{y}_{n-1/2}$, we obtain:

$$\gamma_{(0|n)} + \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)} \mathrm{e}^{-(\mathrm{t}_{n-1/2}-\mathrm{t}_{n-i})} = \mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}.$$

Writing the Taylor series expansion of $\mathrm{e}^{-(\mathrm{t}_{n-1/2}-\mathrm{t}_{n-i})}$ in the above equation and equating terms up to order p, we obtain the following linear system generating the solution for unknowns $\gamma_{(i|n)}$:

$$\mathsf{A}_{n,\mathrm{p}} \cdot \begin{pmatrix} \gamma_{(0|n)} \\ \gamma_{(1|n)} \\ \gamma_{(2|n)} \\ \vdots \\ \gamma_{(\mathrm{p}|n)} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \; \mathsf{A}_{n,\mathrm{p}} \coloneqq \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \varepsilon_{(1|n)} & \varepsilon_{(2|n)} & \dots & \varepsilon_{(\mathrm{p}|n)} \\ 0 & \varepsilon_{(1|n)}^2 & \varepsilon_{(2|n)}^2 & \dots & \varepsilon_{(\mathrm{p}|n)}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \varepsilon_{(1|n)}^{\mathrm{p}} & \varepsilon_{(2|n)}^{\mathrm{p}} & \dots & \varepsilon_{(\mathrm{p}|n)}^{\mathrm{p}} \end{pmatrix}, \tag{30}$$

where $\{\varepsilon_{(i|n)} \,|\, i \in \mathrm{S}_1^{\mathrm{p}}\}$ are already defined in Eq. (22). System (30) can be solved symbolically using Cramer's rule and the determinant formulas of Vandermonde matrices. We have:

$$\gamma_{(i|n)} = \begin{cases} \displaystyle\sum_{j=1}^{\mathrm{p}} [\varepsilon_{(j|n)}]^{-1}, & \text{if } i = 0, \\ (-1)^{\mathrm{p}} [\varepsilon_{(i|n)}]^{-1} \displaystyle\prod_{\substack{j=1 \\ j\neq i}}^{\mathrm{p}} \varepsilon_{(j|n)} [\varepsilon_{(i|n)} - \varepsilon_{(j|n)}]^{-1}, & \text{if } i \in \mathrm{S}_1^{\mathrm{p}}. \end{cases} \tag{31}$$

Therefore, $\mathrm{y}_{n-1/2}$, in case of a linear equation is given by:

$$\mathrm{y}_{n-1/2} = \frac{1}{(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) - \gamma_{(0|n)}} \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)} \mathrm{y}_{n-i}, \tag{32}$$

For now, let us chose $\kappa_1 \in \mathbb{C}$ such that $0 < \mathrm{Re}(\kappa_1) < 1$ and $-1 < \mathrm{Im}(\kappa_1) < 1$. Before going to the second integration, we establish the following Lemma:

Lemma 4. *Taking $\varepsilon_{(j|n)}$ defined by Eq. (22) and $\gamma_{(j|n)}$ solution to System (30), we have the following identity:*

$$\sum_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}^{\mathrm{p}+1} \gamma_{(j|n)} = (-1)^{\mathrm{p}} \prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}. \tag{33}$$

Proof. First, we replace formulas of $\gamma_{(j|n)}$ from Eq. (31) in the above formula:

$$\sum_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}^{\mathrm{p}+1}\gamma_{(j|n)} \;=\; (-1)^{\mathrm{p}}\sum_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}^{\mathrm{p}+1}[\varepsilon_{(j|n)}]^{-1}\prod_{\substack{m=1\\m\neq j}}^{\mathrm{p}} \varepsilon_{(m|n)}[\varepsilon_{(j|n)}-\varepsilon_{(m|n)}]^{-1},$$

then, we factorize by $\prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}$:

$$= \; (-1)^{\mathrm{p}}\prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}\sum_{j=1}^{\mathrm{p}} \frac{\varepsilon_{(j|n)}^{\mathrm{p}-1}}{\prod\limits_{\substack{m=1\\m\neq j}}^{\mathrm{p}}[\varepsilon_{(j|n)}-\varepsilon_{(m|n)}]}.$$

We now take common denominator in the sum to get:

$$\begin{aligned}\sum_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}^{\mathrm{p}+1}\gamma_{(j|n)} \;&=\; (-1)^{\mathrm{p}}\prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}\frac{\displaystyle\sum_{j=1}^{\mathrm{p}}(-1)^{j-1}\varepsilon_{(j|n)}^{\mathrm{p}-1}\prod_{\substack{m=1\\m\neq j}}^{\mathrm{p}-1}\prod_{\substack{\ell=m+1\\\ell\neq j}}^{\mathrm{p}}[\varepsilon_{(m|n)}-\varepsilon_{(\ell|n)}]}{\displaystyle\prod_{m=1}^{\mathrm{p}-1}\prod_{\ell=m+1}^{\mathrm{p}}[\varepsilon_{(m|n)}-\varepsilon_{(\ell|n)}]}\\ &=\; (-1)^{\mathrm{p}}\prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)},\end{aligned}$$

which is true because of

$$\sum_{j=1}^{\mathrm{p}}(-1)^{j-1}\varepsilon_{(j|n)}^{\mathrm{p}-1}\prod_{\substack{m=1\\m\neq j}}^{\mathrm{p}-1}\prod_{\substack{\ell=m+1\\\ell\neq j}}^{\mathrm{p}}[\varepsilon_{(m|n)}-\varepsilon_{(\ell|n)}] = \prod_{m=1}^{\mathrm{p}-1}\prod_{\ell=m+1}^{\mathrm{p}}[\varepsilon_{(m|n)}-\varepsilon_{(\ell|n)}].$$

The last identity can be proven by induction. □

4.2. Second jump

After computing $\mathrm{y}_{n-1/2}$, we have $\mathrm{p}+1$ points with $\mathrm{y}_{n-\mathrm{p}},\ldots,\mathrm{y}_{n-1}$. We define:

$$\mathrm{t}_n = \mathrm{t}_{n-1/2} + (1-\kappa_1)\tau_n \equiv \mathrm{t}_{n-1} + \tau_n,$$

and having the scheme of BDF(p+1) as in Definition 1, we apply its numerical flow as follows:

$$\Big(\mathcal{T}'_{n,\mathrm{p}+1}, \mathrm{Y}'_{n,\mathrm{p}+1}\Big) = \Phi^{[\mathrm{p}+1]}_{(1-\kappa_1)\tau_n}\Big(\mathcal{T}_{n-1/2,\mathrm{p}+1}, \mathrm{Y}_{n-1/2,\mathrm{p}+1}\Big).$$

Here, $\mathcal{T}_{n-1/2,\mathrm{p}+1} = (\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}, \mathrm{t}_{n-1/2})$ and $\mathcal{T}'_{n,\mathrm{p}+1} = (\mathrm{t}_{n-\mathrm{p}+1}, \ldots, \mathrm{t}_{n-1}, \mathrm{t}_{n-1/2}, \mathrm{t}_n)$ $\in \mathbb{R}^{\mathrm{p}+1}$, $\mathrm{Y}_{n-1/2,\mathrm{p}+1} = (\mathrm{y}_{n-\mathrm{p}}, \ldots, \mathrm{y}_{n-1}, \mathrm{y}_{n-1/2})$ and $\mathrm{Y}'_{n,\mathrm{p}+1} = (\mathrm{y}_{n-\mathrm{p}+1}, \ldots, \mathrm{y}_{n-1}, \mathrm{y}_{n-1/2}, \hat{\mathrm{y}}_n)$ $\in (\mathbb{R}^d)^{\mathrm{p}+1}$, with $\hat{\mathrm{y}}_n$ solution to the following equation:

$$\Gamma_{(0|n)}\hat{\mathrm{y}}_n + \Gamma_{(1|n)}\mathrm{y}_{n-1/2} + \sum_{i=1}^{\mathrm{p}} \Gamma_{(i+1|n)}\mathrm{y}_{n-i} = (\mathrm{t}_n - \mathrm{t}_{n-1/2}) f(\mathrm{t}_n, \hat{\mathrm{y}}_n). \tag{34}$$

To find coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$ that fulfill the $(\mathrm{p}+1)^{\mathrm{th}}$ order conditions, we proceed in the same strategy as before and consider the linear equation $f(\mathrm{t}, \mathrm{y}) = \mathrm{y}$, where $\mathrm{y}_{n-1/2}$ and y_{n-i} for $i \in \mathrm{S}_1^{\mathrm{p}}$ are presented below:

$$\mathrm{y}_{n-1/2} = \frac{\hat{\mathrm{y}}_n \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)} \mathrm{e}^{-(\mathrm{t}_n - \mathrm{t}_{n-i})}}{(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) - \gamma_{(0|n)}}, \qquad \mathrm{y}_{n-i} = \hat{\mathrm{y}}_n \mathrm{e}^{-(\mathrm{t}_n - \mathrm{t}_{n-i})}, \quad i \in \mathrm{S}_1^{\mathrm{p}}.$$

Replacing all the above elements in Eq. (34) and simplifying by $\hat{\mathrm{y}}_n$, leads us to the following relation:

$$\Gamma_{(0|n)} + \Gamma_{(1|n)} \frac{\sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)} \mathrm{e}^{-(\mathrm{t}_n - \mathrm{t}_{n-i})}}{(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) - \gamma_{(0|n)}} + \sum_{i=1}^{\mathrm{p}} \Gamma_{(i+1|n)} \mathrm{e}^{-(\mathrm{t}_n - \mathrm{t}_{n-i})} = \mathrm{t}_n - \mathrm{t}_{n-1/2}.$$

Now, multiplying by $[(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) - \gamma_{(0|n)}]$ on both sides and writing the Taylor series expansion of $\mathrm{e}^{-(\mathrm{t}_n - \mathrm{t}_{n-i})}$, gives us the following equation:

$$\begin{aligned}
&-\Gamma_{(0|n)}\gamma_{(0|n)} + \Gamma_{(0|n)}(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) \\
&+ \left[\Gamma_{(1|n)}\gamma_{(1|n)} - \Gamma_{(2|n)}\gamma_{(0|n)} + \Gamma_{(2|n)}(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1})\right] \left[\sum_{i=0} \frac{(-1)^i}{i!} (\mathrm{t}_n - \mathrm{t}_{n-1})^i\right] \\
&\quad + \quad \cdots \\
&+ \left[\Gamma_{(1|n)}\gamma_{(\mathrm{p}|n)} - \Gamma_{(\mathrm{p}+1|n)}\gamma_{(0|n)} + \Gamma_{(\mathrm{p}+1|n)}(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1})\right] \left[\sum_{i=0} \frac{(-1)^i}{i!} (\mathrm{t}_n - \mathrm{t}_{n-\mathrm{p}})^i\right] \\
&= -\gamma_{(0|n)}(\mathrm{t}_n - \mathrm{t}_{n-1/2}) + (\mathrm{t}_n - \mathrm{t}_{n-1/2})(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}).
\end{aligned}$$

To simplify notations, we define:

$$\xi_{(i|n)} \;\coloneqq\; \frac{\mathrm{t}_n - \mathrm{t}_{n-i}}{\tau_n} = 1 - \kappa_1 + \kappa_1 \varepsilon_{(i|n)} = 1 + r_{(i|n)}, \quad i \in \mathrm{S}_1^{\mathrm{p}}. \tag{35}$$

Though, $(\mathrm{t}_n - \mathrm{t}_{n-i}) \equiv \xi_{(i|n)}\tau_n$ for $i \in \mathrm{S}_1^{\mathrm{p}}$. We recall that :

$$(\mathrm{t}_{n-1/2} - \mathrm{t}_{n-1}) = \kappa_1\tau_n, \qquad (\mathrm{t}_n - \mathrm{t}_{n-1/2}) = (1-\kappa_1)\tau_n$$

and we use them to rearrange the Taylor expansion as follows:

$$\begin{aligned}
&\Gamma_{(1|n)}\Big(\sum_{i=1}^{\mathrm{p}}\gamma_{(i|n)}\Big) - \Big(\Gamma_{(0|n)} + \sum_{i=2}^{\mathrm{p}+1}\Gamma_{(i|n)}\Big)\gamma_{(0|n)} \\
&+\Big(\kappa_1\Gamma_{(0|n)} - \Gamma_{(1|n)}\sum_{i=1}^{\mathrm{p}}\xi_{(i|n)}\gamma_{(i|n)} + \sum_{i=2}^{\mathrm{p}+1}\begin{bmatrix}\kappa_1\\ +\gamma_{(0|n)}\xi_{(i-1|n)}\end{bmatrix}\Gamma_{(i|n)}\Big)\tau_n \\
&+\sum_{j=2}^{\infty}(-1)^{j-1}\left(-\Gamma_{(1|n)}\sum_{i=1}^{\mathrm{p}}\xi_{(i|n)}^{j}\gamma_{(i|n)} + \sum_{i=2}^{\mathrm{p}+1}\begin{bmatrix}j\kappa_1\xi_{(i-1|n)}^{j-1}\\ +\gamma_{(0|n)}\xi_{(i-1|n)}^{j}\end{bmatrix}\Gamma_{(i|n)}\right)\frac{\tau_n^j}{j!} \\
&= -\gamma_{(0|n)}(1-\kappa_1)\,\tau_n + \kappa_1(1-\kappa_1)\,\tau_n^2. \qquad (36)
\end{aligned}$$

To simplify this equation, we state the following

Proposition 5. *Define* $\xi_{(0|n)} \coloneqq 1-\kappa_1 \equiv \kappa_2$ *and take* $\xi_{(i|n)}$ *that are defined in Eq.* (35) *and* $\{\gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$ *as a solution to System* (30)*. Then,*

$$-\sum_{i=1}^{\mathrm{p}}\xi_{(i|n)}^{j}\gamma_{(i|n)} = \xi_{(0|n)}^{j-1}\Big(j\kappa_1 + \xi_{(0|n)}\gamma_{(0|n)}\Big), \qquad \forall j \in \mathrm{S}_1^{\mathrm{p}}. \qquad (37)$$

In addition, the following identity holds:

$$-\sum_{i=1}^{\mathrm{p}}\xi_{(i|n)}^{\mathrm{p}+1}\gamma_{(i|n)} = \xi_{(0|n)}^{\mathrm{p}}\Big((\mathrm{p}+1)\kappa_1 + \xi_{(0|n)}\gamma_{(0|n)}\Big) + (-1)^{\mathrm{p}+1}\kappa_1^{\mathrm{p}+1}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}. \quad (38)$$

Proof. The proof of the first equality is obtained by replacing $\xi_{(i|n)}$ by Eq. (35) and using binomial formulas with System (30). The second equality is obtained also using System (30) and the use of Lemma 4. □

By using the relation $\sum_{i=1}^{\mathrm{p}}\gamma_{(i|n)} = -\gamma_{(0|n)}$, we simplify the first term relative to the 0^{th} rank (τ_n^0) in (36), we obtain:

$$\sum_{i=0}^{\mathrm{p}+1}\Gamma_{(i|n)} = 0. \qquad (39)$$

Now, we use Eq. (39) to simplify the term relative to τ_n in (36) to obtain:

$$\gamma_{(0|n)} \sum_{i=0}^{\mathrm{p}} \xi_{(i|n)} \Gamma_{(i+1|n)} = -\gamma_{(0|n)} \xi_{(0|n)}, \tag{40}$$

which will be used, with Eq. (37) in Proposition 5 for $j = 1$, to simplify the term associated to τ_n^2. We continue like this for all terms multiplying τ_n^j for $j \in \mathrm{S}_3^{\mathrm{p}}$ to get

$$\gamma_{(0|n)} \sum_{i=0}^{\mathrm{p}} \xi_{(i|n)}^{j} \Gamma_{(i+1|n)} = 0, \quad \forall j \in \mathrm{S}_2^{\mathrm{p}}. \tag{41}$$

We end by using Eq. (38) to simplify the term multiplying $\tau_n^{\mathrm{p}+1}$ and Eq. (41) for $j = \mathrm{p}$. The linear system generating coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$ is presented below:

$$\mathsf{A}'_{n,\mathrm{p}+1} \cdot \left(\Gamma_{(0|n)}, \ldots, \Gamma_{(\mathrm{p}+1|n)}\right)^{\top} = \left(0, -\xi_{(0|n)}, 0, \ldots, 0\right)^{\top}, \tag{42}$$

where $\mathsf{A}'_{n,\mathrm{p}+1} \in \mathsf{Mat}_{\mathrm{p}+2}(\mathbb{C})$ is the matrix given below:

$$\mathsf{A}'_{n,\mathrm{p}+1} \coloneqq \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & \xi_{(0|n)} & \xi_{(1|n)} & \ldots & \xi_{(\mathrm{p}|n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \xi_{(0|n)}^{\mathrm{p}} & \xi_{(1|n)}^{\mathrm{p}} & \ldots & \xi_{(\mathrm{p}|n)}^{\mathrm{p}} \\ 0 & \left[\begin{array}{c} \xi_{(0|n)}^{\mathrm{p}+1} + \\ \frac{(-\kappa_1)^{\mathrm{p}+1}}{\gamma_{(0|n)}} \prod_{i=1}^{\mathrm{p}} \varepsilon_{(i|n)} \end{array}\right] & \left[\xi_{(1|n)}^{\mathrm{p}+1}\right] & \ldots & \left[\xi_{(\mathrm{p}|n)}^{\mathrm{p}+1}\right] \end{pmatrix}. \tag{43}$$

After assembling the linear system relative to coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$, we proceed to establish its solution.

4.3. Solution to System (42)

To find the solution to the linear system associated with coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$, the Cramer's rule will be used. We have the following.

Proposition 6. *Having defined $\varepsilon_{(i|n)}$ for $i \in \mathrm{S}_1^{\mathrm{p}}$ and $\xi_{(i|n)}$ for $i \in \mathrm{S}_0^{\mathrm{p}}$, the expression of $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$, solution to System (42) are explicitly given by:*

$$
\begin{aligned}
\Gamma_{(0|n)} &= \frac{(\gamma_{(0|n)} - 2\kappa_1)^{\mathrm{p}-1}\Big((\gamma_{(0|n)} - 2\kappa_1)\xi_{(0|n)}\gamma_{(0|n)}\big(\sum_{i=0}^{\mathrm{p}} \frac{1}{\xi_{(i|n)}}\big) - \kappa_1\big(\sum_{i=1}^{\mathrm{p}} \frac{1}{\xi_{(i|n)}}\big)\Big)}{\Big(\xi_{(0|n)}\gamma_{(0|n)} - \kappa_1\Big)}, \\
\Gamma_{(1|n)} &= -\frac{\gamma_{(0|n)}\xi_{(0|n)}}{\big(\xi_{(0|n)}\gamma_{(0|n)} - \kappa_1\big)} \prod_{i=1}^{\mathrm{p}} \frac{1 + r_{(i|n)}}{\kappa_1 + r_{(i|n)}}, \\
\Gamma_{(j+1|n)} &= \frac{(-1)^{j+1} \prod\limits_{\substack{i=1 \\ i\neq j}}^{\mathrm{p}} \xi_{(i|n)}}{\prod\limits_{\ell=1}^{j-1}(\xi_{(j|n)} - \xi_{(\ell|n)})} \frac{(\kappa_1 - 1)\Big[\xi_{(0|n)}^2\gamma_{(0|n)} + \kappa_1^2\varepsilon_{(j|n)}\Big]}{\xi_{(j|n)}\big(r_{(j|n)} + \kappa_1\big)\big[\xi_{(0|n)}\gamma_{(0|n)} - \kappa_1\big]}, \quad j \in \mathrm{S}_1^{\mathrm{p}-1}, \\
\Gamma_{(\mathrm{p}+1|n)} &= (-1)^{\mathrm{p}+1} \prod_{i=1}^{\mathrm{p}-1} \left[\frac{\xi_{(i|n)}}{\xi_{(\mathrm{p}|n)} - \xi_{(i|n)}}\right] \frac{(\kappa_1 - 1)\Big[\xi_{(0|n)}^2\gamma_{(0|n)} + \kappa_1^2\varepsilon_{(\mathrm{p}|n)}\Big]}{\xi_{(\mathrm{p}|n)}\big(r_{(\mathrm{p}|n)} + \kappa_1\big)\big[\xi_{(0|n)}\gamma_{(0|n)} - \kappa_1\big]}.
\end{aligned}
\tag{44}
$$

Despite the fact that the proof should present the computation of all coefficients, we are interested in providing the procedure for $\Gamma_{(\mathrm{p}+1|n)}$ for what we want to cancel it by a suitable κ_1. The proof is presented in Appendix B. Although, the proof for other coefficients follows the same pattern as presented for $\Gamma_{(\mathrm{p}+1|n)}$. This proposition will allow us to generate, for any order p, the algebraic formulas of $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}+1}\}$. The explicit formula of $\Gamma_{(\mathrm{p}+1|n)}$ for $\mathrm{p} \in \mathrm{S}_1^3$ is presented in Appendix C.

5. Proof of Theorem 2

In this Section, we state the elements that give us the $(\mathrm{p}+2)^{\mathrm{th}}$ order of the approximation obtained by the twice composition of $\Phi^{[\mathrm{p}]}$ presented in Section 3. It is based on the use of the difference operator defined by Eq. (13) but in the adaptive version. As we start in the composition by finding $\mathrm{y}_{n-1/2}$ using Eq. (28), the associate difference operator L_1 is given below:

$$
L_1(\mathrm{y}, \mathrm{t}_n, \tau_n) = \gamma_{(0|n)}\mathrm{y}(\mathrm{t}_{n-1/2}) + \sum_{i=1}^{\mathrm{p}} \gamma_{(i|n)}\mathrm{y}(\mathrm{t}_{n-i}) - \kappa_1\tau_n \left.\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_{n-1/2}}. \tag{45}
$$

After writing the Taylor series development of $\mathrm{y}(\mathrm{t}_{n-1/2})$, its derivative and $\mathrm{y}(\mathrm{t}_{n-i})$ in the vicinity of t_n, as follows:

$$\begin{aligned}
\mathrm{y}(\mathrm{t}_{n-1/2}) &= \sum_{j=0}^{\infty} \frac{(-1)^j}{j!} \xi_{(0|n)}^j \tau_n^j \left.\frac{\mathrm{d}^j}{\mathrm{dt}^j}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n}, \\
\left.\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_{n-1/2}} &= \sum_{j=0}^{\infty} \frac{(-1)^j}{j!} \xi_{(0|n)}^j \tau_n^j \left.\frac{\mathrm{d}^{j+1}}{\mathrm{dt}^{j+1}}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n}, \\
\mathrm{y}(\mathrm{t}_{n-i}) &= \sum_{j=0}^{\infty} \frac{(-1)^j}{j!} \xi_{(i|n)}^j \tau_n^j \left.\frac{\mathrm{d}^j}{\mathrm{dt}^j}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n},
\end{aligned} \tag{46}$$

we can express the difference operator L_1 in its Taylor development in t_n:

$$L_1(\mathrm{y}, \mathrm{t}_n, \tau_n) = \sum_{j=0}^{\infty} \mathrm{E}_j \tau_n^j \left.\frac{\mathrm{d}^j}{\mathrm{dt}^j}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n}, \tag{47}$$

where

$$\mathrm{E}_j \coloneqq \begin{cases} \sum_{i=0}^{\mathrm{p}} \gamma_{(i|n)}, & j = 0 \\ \frac{(-1)^j}{j!} \left(\sum_{i=0}^{\mathrm{p}} \gamma_{(i|n)} \xi_{(i|n)}^j + j \xi_{(0|n)}^{j-1} \kappa_1 \right) & j \geqslant 1. \end{cases} \tag{48}$$

Using the relation between coefficients $\gamma_{(i|n)}$ in system (30), we have $\mathrm{E}_0 = 0$, and with Proposition 5 we have $\mathrm{E}_j = 0, \forall j \in \mathrm{S}_1^{\mathrm{p}}$, This is another way to see the p^{th} order of the scheme used in the first jump. It is also clear form Proposition 5 that:

$$\mathrm{E}_{\mathrm{p}+1} = -\frac{\kappa_1^{\mathrm{p}+1}}{(\mathrm{p}+1)!} \prod_{j=1}^{\mathrm{p}} \varepsilon_{(j|n)}, \tag{49}$$

which will be used to evaluate the error, according to formula (2.9) in [2, page 372], as follows:

$$\mathrm{y}(\mathrm{t}_{n-1/2}) - \mathrm{y}_{n-1/2} = \frac{1}{\gamma_{(0|n)}} \mathrm{E}_{\mathrm{p}+1} \tau_n^{\mathrm{p}+1} \left.\frac{\mathrm{d}^{\mathrm{p}+1}}{\mathrm{dt}^{\mathrm{p}+1}}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n} + O(\tau_n^{\mathrm{p}+2}). \tag{50}$$

After computing the approximation $\mathrm{y}_{n-1/2}$, the formula of the difference operator regarding the BDFp scheme used in the second jump is written as follows:

$$L_2(\mathrm{y}, \mathrm{t}_n, \tau_n) = \gamma'_{(0|n)} \mathrm{y}(\mathrm{t}_n) + \gamma'_{(1|n)} \mathrm{y}_{n-1/2} + \sum_{i=2}^{\mathrm{p}} \gamma'_{(i|n)} \mathrm{y}(\mathrm{t}_{n-i+1}) - \kappa_2 \tau_n \left.\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n}. \tag{51}$$

Using Eq. (50) we write $\mathrm{y}_{n-1/2}$ in function of $y(\mathrm{y}_{n-1/2})$ and using Eq. (46), we replace $\mathrm{y}(\mathrm{t}_{n-1/2})$ and $\mathrm{y}(\mathrm{t}_{n-i+1})$ in the above equation by their Taylor series expansion in the vicinity of t_n. We write the difference operator L_2 in its Taylor development in t_n as follows:

$$L_2(\mathrm{y},\mathrm{t},\tau_n) = \sum_{j=0}^{\infty} \mathcal{E}_j \tau_n^j \left.\frac{\mathrm{d}^j}{\mathrm{d}\mathrm{t}^j}\mathrm{y}(\mathrm{t})\right|_{\mathrm{t}=\mathrm{t}_n} \tag{52}$$

where the first $\mathrm{p}+1$ terms are given below:

$$\mathcal{E}_j \coloneqq \begin{cases} \displaystyle\sum_{i=0}^{\mathrm{p}} \gamma'_{(i|n)}, & j=0, \\ \displaystyle -\kappa_2 - \sum_{i=0}^{\mathrm{p}-1} \gamma'_{(i+1|n)}\xi_{(i|n)}, & j=1, \\ \displaystyle \frac{(-1)^j}{j!}\sum_{i=0}^{\mathrm{p}-1} \gamma'_{(i+1|n)}\xi^j_{(i|n)}, & j\in \mathrm{S}_2^{\mathrm{p}}, \\ \displaystyle \frac{(-1)^j}{j!}\sum_{i=0}^{\mathrm{p}-1} \gamma'_{(i+1|n)}\xi^j_{(i|n)} - \frac{\gamma'_{(1|n)}}{\gamma_{(0|n)}}\mathrm{E}_j, & j=\mathrm{p}+1, \end{cases} \tag{53}$$

By the fact that the second jump should be at least of order p, we should have $\mathcal{E}_j = 0$ for all $j \in \mathrm{S}_0^{\mathrm{p}}$. This will allow us to find the formulas of coefficients $\{\gamma'_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$. Having $\kappa_2 \equiv \xi_{(0|n)}$, coefficients $\{\gamma'_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$ verify the system defined by the matrix $\mathsf{A}'^{\mathrm{p}+2,\mathrm{p}+2}_{n,\mathrm{p}+1}$: the matrix resulting from deleting the last row and column from $\mathsf{A}'_{n,\mathrm{p}+1}$, with the right-hand side obtained after omitting the last row of that in System (42). Using Eq. (23) and the following lemma, we show that

$$\gamma'_{(i|n)} = \Gamma_{(i|n)}, \quad \forall i \in \mathrm{S}_0^{\mathrm{p}}. \tag{54}$$

Lemma 7. *Consider one having a linear system* $\mathsf{AX} = b$*, where* $\mathsf{X} \coloneqq (\mathsf{x}_1,\ldots,\mathsf{x}_{\mathrm{p}+1})^\top$ *and* $\mathsf{A} \in \mathrm{GL}_{\mathrm{p}+1}(\mathbb{C})$*. If one defines a new linear system* $\mathsf{A}^{j,j}\mathsf{Y} = b^j$ *where* $\mathsf{Y} \coloneqq (\mathsf{y}_1,\ldots,\mathsf{y}_{\mathrm{p}})^\top$ *and* b^j *results from deleting the* j^{th} *component from* b*. If* $\mathsf{x}_j = 0$*, then*

$$\mathsf{y}_i \equiv \begin{cases} \mathsf{x}_i, & i \in \mathrm{S}_1^{j-1}, \\ \mathsf{x}_{i+1}, & i \in \mathrm{S}_{j+1}^{\mathrm{p}}. \end{cases}$$

Lemma 7 is obtained after omitting the term relative to x_j in the original system and removing the j^{th} equation. Back to the linear system generating $\{\gamma'_{(i|n)} \,|\, i \in \mathrm{S}_0^{\mathrm{p}}\}$: it results after deleting the last row and last column from System (42). Having relation (23), thus $\Gamma_{(\mathrm{p}+1|n)} = 0$ and we conclude Eq. (54). To prove the order $\mathrm{p}+1$, we have to show that

$$\mathcal{E}_{\mathrm{p}+1} = 0,$$

which is verified in the last equation in system (42) after replacing $\mathrm{E}_{\mathrm{p}+1}$ by its formula in Eq. (49). To show the error constant, we have the following asymptotic formula:

$$\mathrm{y}(\mathrm{t}_n) - \hat{\mathrm{y}}_n = \frac{1}{\Gamma_{(0|n)}} \mathcal{E}_{\mathrm{p}+2} \tau_n^{\mathrm{p}+2} \left. \frac{\mathrm{d}^{\mathrm{p}+2}}{\mathrm{dt}^{\mathrm{p}+2}} \mathrm{y}(\mathrm{t}) \right|_{\mathrm{t}=\mathrm{t}_n} + \mathcal{O}(\tau_n^{\mathrm{p}+3}). \tag{55}$$

where:

$$\begin{aligned} \mathcal{E}_{\mathrm{p}+2} &= \frac{(-1)^{\mathrm{p}+2}}{(\mathrm{p}+2)!} \sum_{i=0}^{\mathrm{p}} \Gamma_{(i+1|n)} \xi_{(i|n)}^{j} - \frac{\Gamma_{(1|n)}}{\gamma_{(0|n)}} \mathrm{E}_{\mathrm{p}+2}, \\ &= \frac{(-1)^{\mathrm{p}+2}}{(\mathrm{p}+2)!} \left[\sum_{i=0}^{\mathrm{p}} \left(\Gamma_{(i+1|n)} - \frac{\Gamma_{(1|n)}}{\gamma_{(0|n)}} \gamma_{(i|n)} \right) \xi_{(i|n)}^{\mathrm{p}+2} + (\mathrm{p}+2) \kappa_1 \xi_{(0|n)}^{\mathrm{p}+1} \right]. \end{aligned} \tag{56}$$

In the composed flow, we consider that $\mathrm{y}_n = \mathrm{Re}(\hat{\mathrm{y}}_n)$, which means that the error is approximated by the real part of the right hand side of Eq. (55) with:

$$\mathrm{Re}\big(\mathrm{y}(\mathrm{t}_n) - \hat{\mathrm{y}}_n\big) = \mathrm{y}(\mathrm{t}_n) - \mathrm{y}_n \simeq \mathrm{Re}\left(\frac{\mathcal{E}_{\mathrm{p}+2}}{\Gamma_{(0|n)}} \right) \tau_n^{\mathrm{p}+2} \left. \frac{\mathrm{d}^{\mathrm{p}+2}}{\mathrm{dt}^{\mathrm{p}+2}} \mathrm{y}(\mathrm{t}) \right|_{\mathrm{t}=\mathrm{t}_n}. \tag{57}$$

Taking the imaginary part of Eq. (55), we reach the second conclusion of Theorem 2 showing that:

$$\mathrm{y}(\mathrm{t}_n) - \mathrm{y}_n \simeq \mathcal{C}_{\mathrm{p}} \times \mathrm{Im}(\hat{\mathrm{y}}_n), \tag{58}$$

with

$$\mathcal{C}_{\mathrm{p}} \coloneqq \mathrm{Re}\left(\frac{\mathcal{E}_{\mathrm{p}+2}}{\Gamma_{(0|n)}} \right) \times \left[\mathrm{Im}\left(\frac{\mathcal{E}_{\mathrm{p}+2}}{\Gamma_{(0|n)}} \right) \right]^{-1}. \tag{59}$$

We end it with presenting Algorithm 2 generating the composed flow $\Psi_{\tau_n}^{[\mathrm{p}+1]}$.

Algorithm 2 The composed numerical flow $\Psi_{\tau_n}^{[\mathrm{p}+1]}(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}})$

Require: τ_n
Require: $\mathrm{t}_n \leftarrow \mathrm{t}_{n-1} + \tau_n$
  for all $j \leftarrow 1$ to p **do**
    $r_{(j|n)} \leftarrow \dfrac{\mathrm{t}_{n-1} - \mathrm{t}_{n-j}}{\mathrm{t}_n - \mathrm{t}_{n-1}}$
  end for
  function $\varepsilon_{(\mathrm{p}|n)}(\kappa_1)$
    return $1 + \dfrac{r_{(\mathrm{p}|n)}}{\kappa_1}$
  end function
  function $\gamma_{(0|n)}(\kappa_1)$
    return $\displaystyle\sum_{j=1}^{\mathrm{p}} \frac{\kappa_1}{\kappa_1 + r_{(j|n)}}$
  end function
  $\kappa_1 \leftarrow$ Find Roots $[\kappa_1 \longmapsto (1-\kappa_1)^2\gamma_{(0|n)} + \kappa_1^2\varepsilon_{(\mathrm{p}|n)} = 0]$
Ensure: $\mathrm{Re}(\kappa_1) > 0$
  $(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}}) \leftarrow \Phi_{\kappa_1\tau_n}^{[\mathrm{p}]}(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}})$
  $(\mathcal{T}'_{n,\mathrm{p}}, \mathrm{Y}'_{n,\mathrm{p}}) \leftarrow \Phi_{(1-\kappa_1)\tau_n}^{[\mathrm{p}]}\big(\mathcal{T}_{n-1/2,\mathrm{p}}, \mathrm{Y}_{n-1/2,\mathrm{p}}\big)$
  return $(\mathcal{T}_{n,\mathrm{p}}, \mathrm{Y}_{n,\mathrm{p}}), \mathrm{Im}(\hat{\mathrm{y}}_n)$

6. Convergence error and CPU for fixed time steps

To show graphically the convergence order of $\Psi^{[\mathrm{p}+1]}$, we consider the case of a fixed time step ($\tau_n = \tau \Rightarrow r_{(j|n)} = j - 1$). Thus, the solution of $\Gamma_{(\mathrm{p}+1|n)}$, denoted here by $\Gamma_{(\mathrm{p}+1)}$, as it does not depend on instant t_n, is given for all $\mathrm{p} \in \mathrm{S}_1^4$:

$$\Gamma_{(1+1)} = \frac{(\kappa_1-1)(2\kappa_1^2 - 2\kappa_1 + 1)}{\kappa_1(2\kappa_1 - 1)},$$
$$\Gamma_{(2+1)} = -\frac{(\kappa_1-1)(3\kappa_1^3 - \kappa_1^2 + \kappa_1 + 1)}{2(\kappa_1+1)(3\kappa_1^2 - 1)},$$
$$\Gamma_{(3+1)} = \frac{(\kappa_1-1)(4\kappa_1^4 + 5\kappa_1^3 + \kappa_1^2 + 6\kappa_1 + 2)}{6(2\kappa_1+1)(\kappa_1+2)(\kappa_1^2 + \kappa_1 - 1)},$$
$$\Gamma_{(4+1)} = -\frac{(\kappa_1-1)(5\kappa_1^5 + 19\kappa_1^4 + 19\kappa_1^3 + 19\kappa_1^2 + 28\kappa_1 + 6)}{4(\kappa_1+3)(5\kappa_1^4 + 20\kappa_1^3 + 15\kappa_1^2 - 10\kappa_1 - 6)}.$$

When the function Find Roots comes, we use a computer algebra software. We exhibit below the explicit expression of κ_1, solution to $\Gamma_{(\mathrm{p}+1)} = 0$, when $\mathrm{p} \in \{1, 2, 3\}$, and a numerical approximation for its solution when $\mathrm{p} = 4$ (for the sake of simplicity):

$$\kappa_1 = \frac{1}{2} \pm \frac{\mathrm{i}}{2}; \qquad \mathrm{p} = 1$$

$$\kappa_1 = \left(\frac{-b}{18} + \frac{4}{9b} + \frac{1}{9}\right) \pm \mathrm{i}\sqrt{3}\left(\frac{b}{18} + \frac{4}{9b}\right); \qquad \mathrm{p} = 2$$
$$\approx 0.4013648789516588 \pm \mathrm{i} \times 0.7409710153124752$$

$$\kappa_1 = \frac{d}{48c} - \frac{5}{16} \pm \frac{\mathrm{i}}{2}\sqrt{\frac{2439c}{32d} + c^2 + \frac{7}{144c^2} - \frac{43}{96}}; \qquad \mathrm{p} = 3$$
$$\approx 0.3247753916537674 \pm \mathrm{i} \times 0.927940112670109$$

$$\kappa_1 \approx 0.2675589068337956 \pm \mathrm{i} \times 1.088573443182903; \qquad \mathrm{p} = 4$$

with

$$b \;\coloneqq\; \left(\frac{2\sqrt{19}}{\sqrt{3}} - 134\right)^{1/3},$$
$$c \;\coloneqq\; \frac{1}{2}\left(2\sqrt{\frac{9931}{6}} + \frac{2179}{27}\right)^{1/6}, \qquad d \;\coloneqq\; \sqrt{576c^4 + 129c^2 + 28}.$$

6.1. Global error

We solve numerically the IVP (1) with $f\big(\mathrm{t}, \mathrm{y}\big) = -\mathrm{y}^3$ over the interval $]0, 1[$, with $\mathrm{y}(0) = 1$ using a BDFp scheme and its two compositions when $\mathrm{p} \in \mathrm{S}_1^4$ for different time steps $\tau = 1/\mathrm{N}, \mathrm{N} \in \{10, 20, 40, 80, 160\}$. Knowing the exact solution $\mathrm{y}(\mathrm{t}) \equiv \sqrt{\frac{1}{1+2\mathrm{t}}}$, we use it to build approximations y_j on $\mathrm{t}_j = j\tau, j \in \mathrm{S}_1^{\mathrm{p}-1}$ as they are crucial for initializing $\Phi^{[\mathrm{p}]}$ and $\Psi_\tau^{[\mathrm{p}+1]}$. The part of Find Roots in Algorithm 2 is skipped at every time step. Algorithm 2 is then simplified by the last five lines: we apply first $\Phi^{[\mathrm{p}]}_{(1-\kappa_1)\tau}$ then $\Phi^{[\mathrm{p}]}_{\kappa_1\tau}$ to produce $\hat{\mathrm{y}}_n$ and take its real part to store in y_n at every time step. Here, the exact solution is also used to compute the exact error:

$$e(\mathrm{t}_n) \;\coloneqq\; \|\mathrm{y}(\mathrm{t}_n) - \mathrm{y}_n\|, \quad n \in \mathrm{S}_1^{\mathrm{N}}. \tag{60}$$

The global error is computed by using the trapezoidal rule of integration as follows:

$$\mathrm{E_N} \coloneqq \frac{1}{\mathrm{N}} \left(\sum_{n=\mathrm{p}}^{\mathrm{N}-1} e(\mathrm{t}_n) + \frac{e(\mathrm{t_N})}{2} \right). \tag{61}$$

For distinction, we denote by $\mathrm{E_N^{\Phi^{[p]}}}$ the global error associated to the approximation produced by the BDFp scheme while we denote by $\mathrm{E_N^{\Psi^{[p]}}}$ the one associated to the approximation produced by the composed flow of order p. We present in Fig. 3 the comparison between $\mathrm{E_N^{\Phi^{[p]}}}$ and $\mathrm{E_N^{\Psi^{[p+1]}}}$.

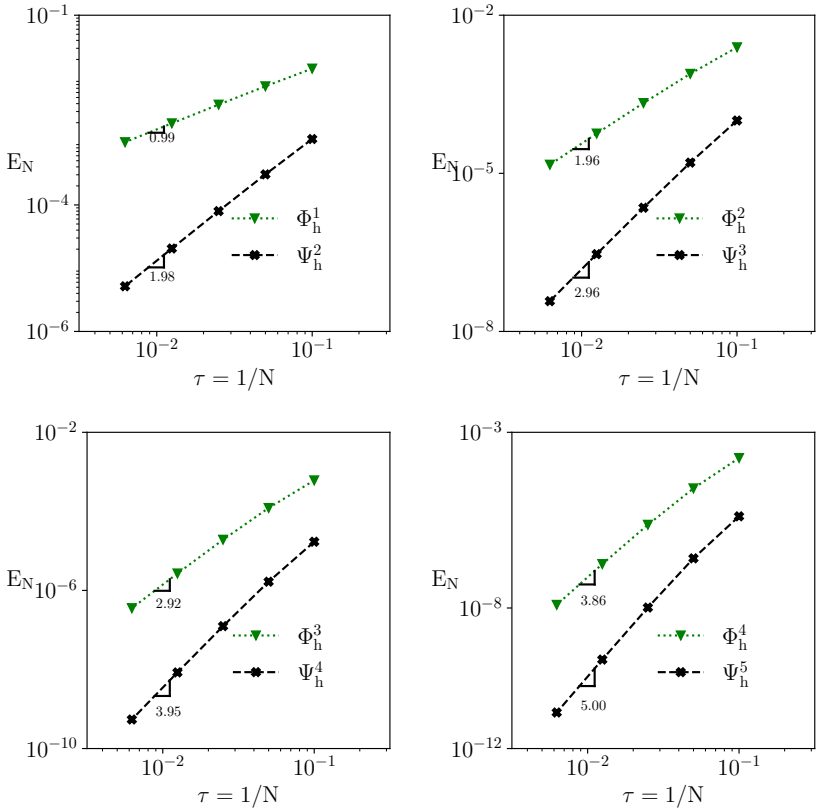

Figure 3: Convergence errors of $\Phi^{[\mathrm{p}]}$ and its composed flow $\Psi^{[\mathrm{p}+1]}$ for $\mathrm{p} \in \{1, 2, 3, 4\}$.

We mention that the slope for $\Phi^{[2]}$, the second order BDF scheme, is $\simeq 1.96$ while its twice composition $\Psi^{[3]}$ designed by Algorithm 2 has a slope of $\simeq 2.96$, showing an increasing by one the order of convergence. Other convergence rates for BDFp schemes and their twice compositions can be checked

in the same figure. We conclude that the composition designed in this work increases by one the accuracy order. To check the utility of this composition in using less number of backward points while producing approximation with the same order, we present in Table 2 the global error between the classical BDFp scheme using p backwards points and the proposed composed flow having the same order p while using $p-1$ backward points. The comparison is done for $p = 2, 3, 4, 5$.

Table 2: Comparison between the global error $E_N^{\Phi^{[p]}_\tau}$ and $E_N^{\Psi^{[p]}_\tau}$ for different values τ.

τ	10^{-1}	5×10^{-2}	2.5×10^{-2}	1.25×10^{-2}	6.25×10^{-3}
$E_N^{\Phi^{[2]}}$	2.46×10^{-3}	7.73×10^{-4}	2.15×10^{-4}	5.68×10^{-5}	1.45×10^{-5}
$E_N^{\Psi^{[2]}}$	1.10×10^{-3}	3.04×10^{-4}	7.99×10^{-5}	2.04×10^{-5}	5.18×10^{-6}
$E_N^{\Phi^{[3]}}$	6.08×10^{-4}	1.21×10^{-4}	1.91×10^{-5}	2.68×10^{-6}	3.56×10^{-7}
$E_N^{\Psi^{[3]}}$	1.00×10^{-4}	1.59×10^{-5}	2.22×10^{-6}	2.93×10^{-7}	3.75×10^{-8}
$E_N^{\Phi^{[4]}}$	1.85×10^{-4}	2.53×10^{-5}	2.33×10^{-6}	1.78×10^{-7}	1.22×10^{-8}
$E_N^{\Psi^{[4]}}$	1.70×10^{-5}	1.66×10^{-6}	1.24×10^{-7}	8.41×10^{-9}	5.43×10^{-10}
$E_N^{\Phi^{[5]}}$	6.41×10^{-5}	6.46×10^{-6}	3.60×10^{-7}	1.51×10^{-8}	5.52×10^{-10}
$E_N^{\Psi^{[5]}}$	4.06×10^{-6}	2.58×10^{-7}	1.02×10^{-8}	3.39×10^{-10}	1.06×10^{-11}

We conclude from Table 2 that the composed flow of order p, $\Psi^{[p]}$, has better accuracy than the one of BDFp, $\Phi^{[p]}$, with the same order, which means that it has a smaller error constants in the error estimation. This allows us to produce a better quality of approximation while using a smaller number of backward points. But how this increasing in order will affect the computational cost involving real and complex parts? Does the composed numerical flow $\Psi^{[p]}$ has a low performance compared to the numerical flow of the BDFp having the same order $\Phi^{[p]}$?

6.2. CPU comparison

To check the computational efficiency of the composition, whether or not involving computation in real and complex parts will significantly increase the computational cost, we compare in Table 3 the ratio of the global errors

and the ratio of CPU time between $\Phi^{[\mathrm{p}]}$ and $\Psi^{[\mathrm{p}]}$ for a set of time steps:

$$\mathrm{R}_{\mathrm{E_N}} = \frac{\mathrm{E_N}^{\Phi^{[\mathrm{p}]}}}{\mathrm{E_N}^{\Psi^{[\mathrm{p}]}}}, \qquad \mathrm{R}_{\mathrm{CPU}} = \frac{\mathrm{CPU}^{\Phi^{[\mathrm{p}]}}}{\mathrm{CPU}^{\Psi^{[\mathrm{p}]}}} \tag{62}$$

As for instance, the global error of $\Psi^{[\mathrm{p}]}$ is smaller with a factor of 2.234 compared to $\Phi^{[\mathrm{p}]}$ when $\mathrm{p} = 2$ and $\tau = 10^{-1}$ and the rate can go up to a factor of 52 when $\mathrm{p} = 5$ and $\tau = 6.25\times10^{-3}$, as presented in Table 3. This increasing in accuracy have a computational cost as it involves computation in real and complex parts. To check this, we present in the same table (Table 3) the CPU ratio $\mathrm{R}_{\mathrm{CPU}}$. We can see that, for $\mathrm{p} = 2$ and $\tau = 0.1$, the decreasing of the error by a factor of 2 using $\Psi^{[2]}$ implies an increasing in CPU time by almost the same factor. This factor decreases when τ decreases. For higher order p and smaller time steps, the increasing factor of CPU decreases and almost tends to 1, meaning that the computational time is the same for a given time step, while the error decreases faster allowing higher precision going up to a factor of 52 when $\mathrm{p} = 5$ and $\tau = 0.00625$.

Table 3: Ratio of the global error and the CPU time between $\Phi_\tau^{[\mathrm{p}]}$ and $\Psi_\tau^{[\mathrm{p}]}$.

	τ	10^{-1}	5×10^{-2}	2.5×10^{-2}	1.25×10^{-2}	6.25×10^{-3}
$\mathrm{p} = 2$	$\mathrm{R}_{\mathrm{E_N}}$	2.234	2.539	2.695	2.775	2.816
	$\mathrm{R}_{\mathrm{CPU}}$	1/2.343	1/2.295	1/2.291	1/2.242	1/1.625
$\mathrm{p} = 3$	$\mathrm{R}_{\mathrm{E_N}}$	6.0582	7.629	8.607	9.172	9.480
	$\mathrm{R}_{\mathrm{CPU}}$	1/2.225	1/2.037	1/1.915	1/2.103	1/1.968
$\mathrm{p} = 4$	$\mathrm{R}_{\mathrm{E_N}}$	10.890	15.266	18.734	21.150	22.626
	$\mathrm{R}_{\mathrm{CPU}}$	1/1.906	1/1.689	1/1.486	1/1.558	1/1.647
$\mathrm{p} = 5$	$\mathrm{R}_{\mathrm{E_N}}$	15.773	24.966	35.055	44.662	52.073
	$\mathrm{R}_{\mathrm{CPU}}$	1/1.680	1/1.372	1/1.321	1/1.328	1/1.168

we present in Fig. 4 the plot of CPU time versus the associated global precision for both, the classical BDFp scheme, $\Phi^{[\mathrm{p}]}$ and the proposed composed flow $\Psi^{[\mathrm{p}]}$ having the same order. Note that the composed flow with order p prescribe $\mathrm{p}-1$ backward points. We can see that the computational efficiency of the composed flow is the same when $\mathrm{p} = 3$ allowing reaching the

same accuracy with less number of backward points, and better for higher orders. In the next Section, we investigate the linear stability of $\Psi^{[p+1]}$ and

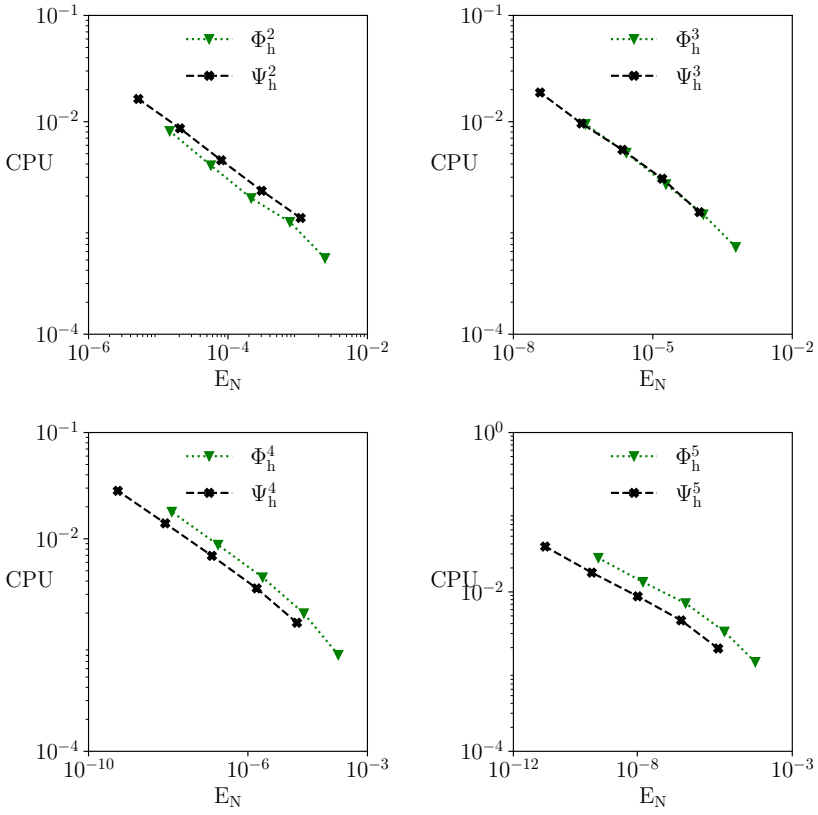

Figure 4: CPU versus global errors of $\Phi^{[p]}$ and the composed flow $\Psi^{[p]}$ for $p \in \{2, 3, 4, 5\}$.

see how the barrier of order six of BDF schemes is broken.

7. Linear stability analysis

The study of the stability of a linear multi-step method is done by its generating polynomials defined by Eq. (5). However, it is difficult to show the generating polynomials of the designed numerical method in this work. This is because the use of an iterative solver to obtain an intermediate solution. This is why we will focus solely on the A−stability. To do that, we consider the simplest linear scalar problem $\dot{y} = \lambda y$, where the numerical integration

using $\Psi_{\tau}^{[\mathrm{p}+1]}$ leads us to the following relation:

$$\sum_{i=0}^{\mathrm{p}} \Theta_i \mathrm{y}_{n-i} = 0$$

with

$$\Theta_i =: \begin{cases} (\kappa_1 z - \gamma_{(0|n)}).(\gamma'_{(0|n)} - (1-\kappa_1)z), & i = 0, \\ \gamma'_{(1|n)}\gamma_{(i|n)} + (\kappa_1 z - \gamma_{(0|n)})\gamma'_{(i+1|n)}, & i \in \mathrm{S}_1^{\mathrm{p}-1}, \\ \gamma'_{(1|n)}\gamma_{(\mathrm{p}|n)}, & i = \mathrm{p}, \end{cases}$$

where $z \coloneqq \tau\lambda$. We define the polynomial $p_{\mathrm{p}}(\omega, z)$ associated to $\Psi^{[\mathrm{p}+1]}$ by the following formula:

$$p_{\mathrm{p}}\big(\omega, z\big) = \sum_{i=0}^{\mathrm{p}} \Theta_{\mathrm{p}-i}\omega^i. \tag{63}$$

Consider $\omega_1(z), \ldots, \omega_{\mathrm{p}}(z)$ the p roots of $p_{\mathrm{p}}\big(\omega, z\big)$ that depend on z. The stability region of the numerical scheme $\Psi^{[\mathrm{p}+1]}$ is the following region defined in the complex plane:

$$\mathcal{D}_{\Psi^{[\mathrm{p}+1]}} = \left\{ z \in \mathbb{C} \;\middle|\; |\omega_i(z)| \leqslant 1, \forall\, i \in \mathrm{S}_1^{\mathrm{p}} \right\}. \tag{64}$$

According to Lemma 4.7 in [13], we have the following:

Definition 3. *The composed scheme $\Psi^{[\mathrm{p}+1]}$ is $\mathrm{A}(\vartheta)-$stable if:*

$$S_\vartheta \subseteq \mathcal{D}_{\Psi^{[\mathrm{p}+1]}},$$

where S_ϑ is a sector of angle 2ϑ bisected by the negative real axis. In addition, if $\vartheta = 90°$, the composed scheme is said to be $\mathrm{A}-$stable.

For every $\forall\, \mathrm{p} \in \mathrm{S}_1^8$, the polynomial $p_{\mathrm{p}}\big(\omega, z\big)$ is constructed, then a meshgrid of z values is built in the complex plane, where for every value $z_{i,j}$, roots $\omega_{i,j,s}, s \in \mathrm{S}_1^{\mathrm{p}}$ of $p_{\mathrm{p}}\big(w, z_{i,j}\big) = 0$ are obtained numerically using the NumPy package in Python in order to check those with absolute value that is smaller than one. The domain of stability is the set of all values $z_{i,j}$, where $|\omega_{i,j,s}| \leqslant 1$.

We plot the domain $\mathcal{D}_{\Psi^{[\mathrm{p}+1]}}$ for different $\mathrm{p} \in \mathrm{S}_1^4$ in Fig. 5 and $\mathrm{p} \in \mathrm{S}_5^8$ in Fig. 6. The black solid lines are the roots locus of the $\Phi^{[\mathrm{p}]}$ schemes and the red ones are the locus of roots to $\Psi^{[\mathrm{p}]}$. The outer domain of these roots locus represents the stable region of the scheme. We recall here that the numerical

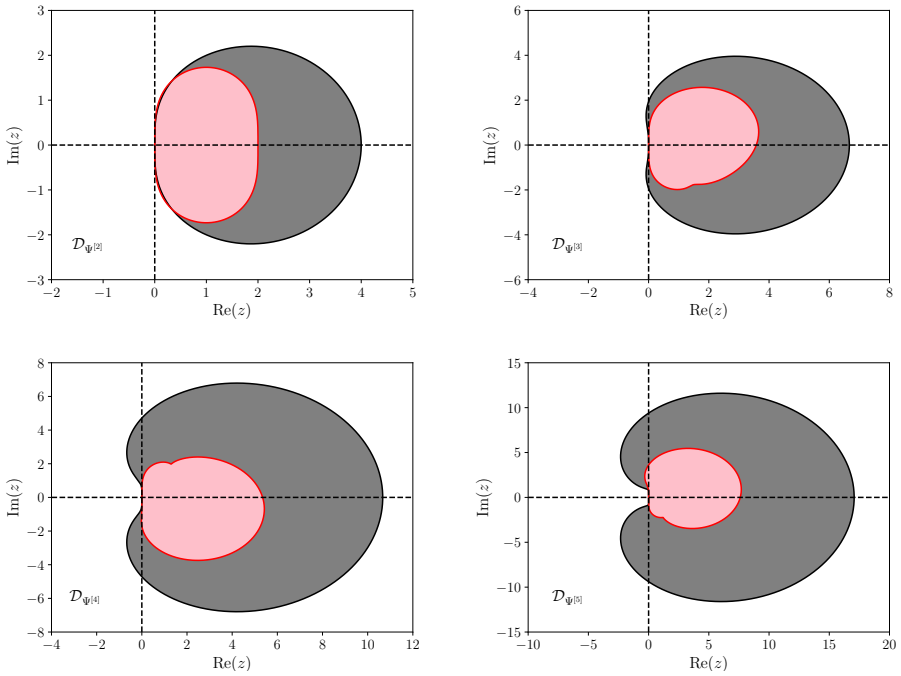

Figure 5: The grey colored region showed the unstable region of the BDFp scheme of order $p \in \{2, 3, 4, 5\}$, while the pink colored region shows the unstable region of the twice composition of the BDF$(p-1)$ scheme which is of order p.

scheme $\Psi^{[p+1]}$ is of order $p+1$. Though, the comparison of the stability region is done between two schemes of the same order: $\Phi^{[p]}$ and $\Psi^{[p]}$. We can see from these pictures that $\mathcal{D}_{\Phi^{[p]}} \subseteq \mathcal{D}_{\Psi^{[p]}}$ for $p \in S_3^9$.

In addition, the composed schemes of order $p = 2, 3, 4$ are stable, and those of order $p \in S_5^8$ are A(ϑ)-stable as it can be seen in the shaded region of stability and the zoom in the down left subplot in Fig. 6. This in not the case for $\Phi^{[p]}$ for $p > 6$. To check the angles of stability for the composed flow $\vartheta_{\Psi^{[p]}}$, we have followed the same strategy as in [87] to compute them. These angles of stability regions are presented in Table 4 with the ones of the $\vartheta_{\Phi^{[p]}}$ for the classical BDF schemes.

In conclusion of this section, the composition of BDFp schemes allows us to break the Dahlquist barrier, six, so we can use higher order BDF families of schemes.

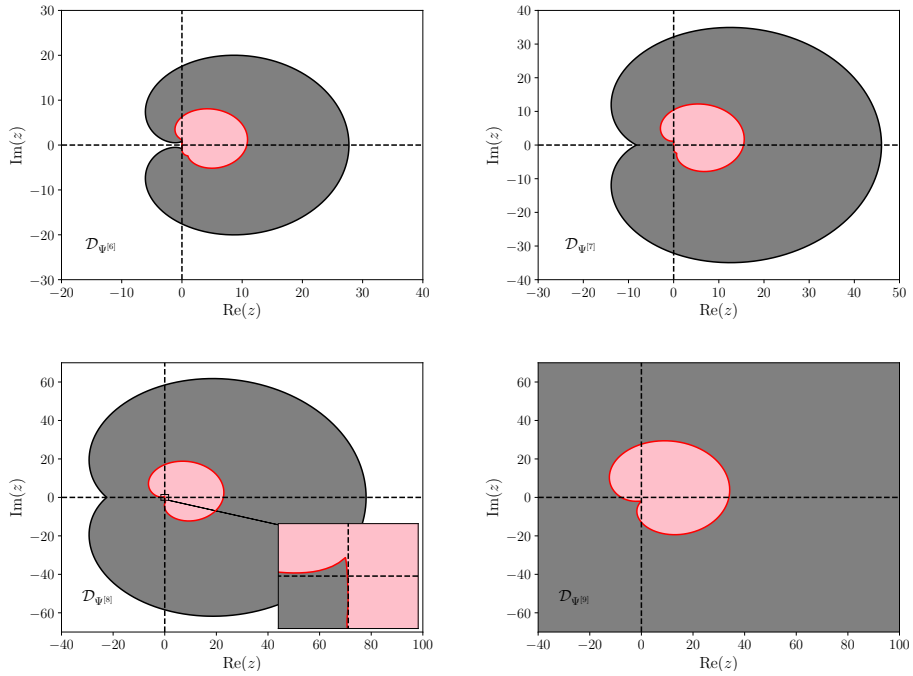

Figure 6: The grey colored region showed the unstable region of the BDFp, $\mathrm{p} \in \{6, 7, 8, 9\}$, while the pink colored region shows the unstable region of the twice composition of the BDF$(\mathrm{p}-1)$ which is of order p.

Table 4: Angles of the A$(\vartheta)-$stability domain

p	2	3	4	5	6	7	8	9
$\vartheta_{\Psi[\mathrm{p}]}$	90°	90	90	81.511	67.796	45	4.146	-
$\vartheta_{\Phi[\mathrm{p}]}$	90°	86.032	73.351	51.839	17.839	-	-	-

8. Variable time step

In this section, we explore the feasibility of implementing variable time stepping. Specifically, we examine whether certain time step values τ_n are constrained to ensure the existence of a single root of the equation $\Gamma_{(\mathrm{p}+1|n)} = 0$ with a real positive part. As System (23) depends only on $r_{(i|n)}, i \in \mathrm{S}_1^{\mathrm{p}}$, the question to ask: is there a limit, either upper or lower, on the values of $r_{(i|n)}$ that must not be surpassed to guarantee the time marching? We

will address this inquiry in the subsequent discussion. Contrary to the upper bounds restricted to the ratio of two adjacent time step found in [77, 78] for keeping the stability of the variable time stepping of the second and third BDF schemes, we find in our analysis lower bounds for adaptive time step when the composed flow of BDFp schemes ($\mathrm{p} \in \mathrm{S}_2^8$) is applied.

To begin, we will mention that $\kappa_1 = 1/2 \pm \mathrm{i}/2$ are the roots of $\Gamma_{(1+1|n)} = 2\kappa_1^2 - 2\kappa_1 + 1 = 0$ when $\mathrm{p} = 1$, which means that it does not depend at all of the ratio of time steps. Thus the composed flow of order 2 has no lower bound on the ratio. In the case of $\mathrm{p} = 2$, κ_1 is the root of

$$3\kappa_1^3 + \kappa_1^2(3r_{(2|n)} - 4) + \kappa_1(r_{(2|n)}^2 - 2r_{(2|n)} + 2) + r_{(2|n)} = 0,$$

where the algebraic expression of its roots as a function of $r_{(2|n)}$ is complicated. Therefore, finding the algebraic dependence of roots on $r_{(2|n)}$ to find where they have a positive real part becomes difficult to express. It is even more complicated for higher p. Therefore, we perform a numerical investigation to find whether or not there is a barrier on τ_n to ensure time marching-the positive real part of κ_1 for adaptive time steps.

8.1. The first step

Consider first that, for a given p, $\{\mathrm{t}_i \,|\, i \in \mathrm{S}_1^{\mathrm{p}-1}\}$ are equidistant ($\tau_i = \tau, \forall i \in \mathrm{S}_1^{\mathrm{p}-1}$). When finding the roots of $\Gamma_{(\mathrm{p}+1|n)} = 0$, we select among them the one with a positive real part to ensure marching in time. To check this, we select a range of values of $\tau_{\mathrm{p}} > 0$ where, for every value, we calculate $\{r_{(i|n)} \,|\, i \in \mathrm{S}_1^{\mathrm{p}}\}$ and then establish $\Gamma_{(\mathrm{p}+1|n)}$ to find its roots and check if there is one such that $\mathrm{Re}(\kappa_1) > 0$. Note that $r_{\{2|\mathrm{p}\}} \equiv \dfrac{\tau}{\tau_{\mathrm{p}}}$, which reflects the ratio of two adjacent time steps. This aims to find a bound on two consecutive time steps, for which at least one of the roots of $\Gamma_{(\mathrm{p}+1|n)} = 0$ has a positive real part. This is done numerically. We found that a lower bound on the ratio should be applied, as for instance when applying the double composition of BDF2 -which is of order 3- and having two approximations to backwards points t_0 and $\mathrm{t}_0 + \tau_1$, the time step τ_2 must verify $\tau_2 \geqslant 0.4506 \times \tau_1$ to ensure the time marching of the composition in having positive value in the real part of κ_1. Another example is illustrated when using the double composition of BDF3- which is of order 4- and having approximations of backwards points $\mathrm{t}_0, \mathrm{t}_0 + \tau, \mathrm{t}_0 + 2\tau$, the time step τ_3 must verify $\tau_3 \geqslant 0.6311\tau$. We note that the lower bound increase as p increases. The values of the lower bounds for the rest of the composed flow are reported in Table 5.

Table 5: Lower bounds of τ_{p}/τ.

p	2	3	4	5	6	7	8
$1/r_{\{2\|\mathrm{p}\}}$	0.4506	0.6311	0.7158	0.7717	0.8125	0.8454	0.8734
$1/\sqrt[2\mathrm{p}-3]{2}$	0.5	0.7937	0.8705	0.9057	0.9258	0.9389	0.9480

For security and simplicity, we can generate the following rule of thumbs to produce an lower bound of two adjoint time steps:

$$\frac{1}{r_{\{2|\mathrm{p}\}}} = \frac{\tau_{\mathrm{p}}}{\tau} \geqslant \frac{1}{\sqrt[2\mathrm{p}-3]{2}}, \quad \mathrm{p} \in \mathrm{S}_2^8.$$

For additional information, we plot in Fig. 7 the relation between τ_{p} and the real part of κ_1, a root of $\Gamma_{(\mathrm{p}+1|n)}$ for $\mathrm{p} \in \mathrm{S}_1^8$ in the first step.

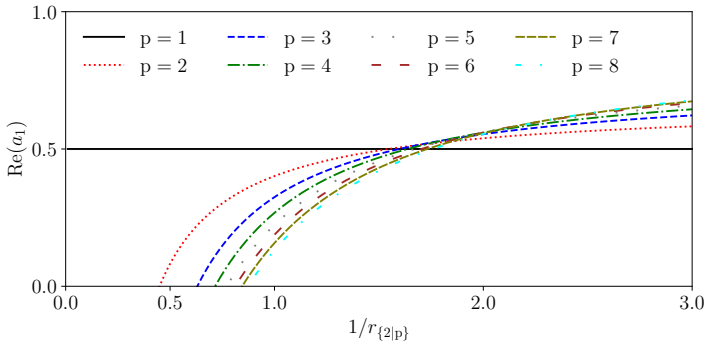

Figure 7: Plot of the real part of κ_1 solutions of system (23) for a range of values of $1/r_{\{2|\mathrm{p}\}}$.

Intermediate conclusions. Consider one has p starting points y_j approximating $\mathrm{y}(\mathrm{t}_j)$ for $\mathrm{t}_j = j\tau, \forall\, j \in \mathrm{S}_0^{\mathrm{p}-1}$ up order p and want to compose the numerical flow $\Phi^{[\mathrm{p}]}$. At the first step of computing an approximation y_{p} to $\mathrm{y}(\mathrm{t}_{\mathrm{p}} := \mathrm{t}_{\mathrm{p}-1} + \tau_{\mathrm{p}})$, the time step τ_{p} should be selected such that:

$$\tau_{\mathrm{p}} \geqslant \begin{cases} 0 & \mathrm{p} = 1, \\ \dfrac{\tau}{\sqrt[2\mathrm{p}-3]{2}} & \mathrm{p} \in \mathrm{S}_2^8. \end{cases} \tag{65}$$

8.2. Simulation at the n^{th} step

Once the solution is approximated on $\mathrm{t_p} = \mathrm{t_0} + (\mathrm{p}-1)\tau + \tau_{\mathrm{p}}$ with τ_{p} verifying inequality (65), selecting the next variable time step size is crucial. For instance, users may need to decrease it for stiff dynamical problems or increase it if the system is not stiff. In the case when $\mathrm{p} = 1$, and as it was shown before, there is no need to take care of the lower bound.

If $\mathrm{p} = 2$, we can conclude according to Table 5 and by induction, that the new time step should verify the following inequality to ensure a positive real part of the root:

$$\text{if } \mathrm{p} = 2 \quad \Longrightarrow \quad \tau_n \geqslant 0.4506 \times \tau_{n-1}, \quad \forall\, n \in \mathbb{Z}^+. \tag{66}$$

For the numerical flow $\Psi^{[\mathrm{p}]}$ with $\mathrm{p} \geqslant 3$, finding the limit bound of the time step is more complex. Thus, we follow the same procedure we applied in the first step. Results are reported in Table 6

Table 6: Lower bound of $\tau_{\mathrm{p}+1}/\tau_{\mathrm{p}}$ while $\tau_{\mathrm{p}} \geqslant \tau/\sqrt[2\mathrm{p}-3]{2}$.

p	2	3	4	5	6	7	8
$1/r_{\{2\mid \mathrm{p}+1\}}$	0.4506	0.6951	0.7763	0.8245	0.8613	0.0.8897	0.9131

We see that values of the lower bounds of the ratio between the new time step $\tau_{\mathrm{p}+1}$ and the last one τ_{p} increases, tending to unity, when p increases. This will reduce the range of possibilities to decrease the time step, if needed, when higher-order schemes are employed for stiff dynamical problems involving different time scales. However this may not limit their uses as high-order numerical schemes are able to catch the dynamics as with higher precision allowing higher values of time steps. We continue searching for the lower bound of the new time step τ_{n+1} according to the last one τ_n. The following rule of thumb in Eq. (67) could be applied, as reported in Table 7.

$$\tau_{n+1} \geqslant \frac{\tau_n}{\sqrt[2\mathrm{p}-3]{2}}, \quad \forall\, \mathrm{p} \in \mathrm{S}_3^8. \tag{67}$$

According to these values, we can see that the prescribed lower bound $\frac{1}{\sqrt[2\mathrm{p}-3]{2}}$ is a reliable solution for $\mathrm{p} \in \mathrm{S}_2^6$, but not when $\mathrm{p} \in \{7, 8\}$, where the following formula is well suited as reported in Table 8:

$$\tau_{n+1} \geqslant \frac{\tau_n}{\sqrt[\mathrm{p}(\mathrm{p}-1)]{\mathrm{p}}}. \tag{68}$$

Table 7: Lower bound of τ_{n+1}/τ_n, prescribed to $\Psi^{[\mathrm{p}+1]}$ while having $\tau_n \geqslant \tau_{n-1}/\sqrt[2\mathrm{p}-3]{2}$.

p	2	3	4	5	6	7	8
$1/r_{\{2\mid n+1\}}$	0.4506	0.6951	0.8063	0.8781	0.9193	0.9483	0.9709
$1/\sqrt[2\mathrm{p}-3]{2}$	0.5	0.7937	0.8705	0.9057	0.9258	0.9389	0.9480

Table 8: Lower bound of τ_{n+1}/τ_n, prescribed to $\Psi^{[\mathrm{p}+1]}$ while having $\tau_n \geqslant \frac{\tau_{n-1}}{\sqrt[\mathrm{p}(\mathrm{p}-1)]{\mathrm{p}}}$.

p	2	3	4	5	6	7	8
$1/r_{\{2\mid n+1\}}$	0.4501	0.6806	0.7900	0.8559	0.9019	0.9362	0.96351
$1/\sqrt[\mathrm{p}(\mathrm{p}-1)]{\mathrm{p}}$	0.7071	0.8326	0.8908	0.9226	0.9420	0.9547	0.96354

We mention here that there is no constraints in prescribing upper bound for $1/r_{\{2\mid n+1\}}$, *i.e.* the time step size could be taken greater than the last one with no limit. Nevertheless, studies [77, 78] shows that adaptivity of time steps for BDF schemes is constrained by having upper bounds to maintain stability. In this paper, we do not establish upper bounds ensuring the stability of the composed flow of BDFp for variable time for what it needs another study. Although, a restriction on the upper bound will be prescribed for robustness and which aligns with the results founds in [77, 78] for the second and third BDF schemes. We end by showing the conclusion of this section.

Intermediate conclusion. For stability of the computation, the upper and the lower bounds, denoted by ℓ_{p} and $\frac{1}{\ell_{\mathrm{p}}}$ respectively, on the new time step size of the associate numerical flow $\Psi_{\tau_n}^{[\mathrm{p}+1]}$ are given below:

$$\forall n : \begin{cases} 0 \leqslant \frac{\tau_{n+1}}{\tau_n} \leqslant 2, & \mathrm{p}=1, \\ \frac{1}{\sqrt[2\mathrm{p}-3]{2}} \leqslant \frac{\tau_{n+1}}{\tau_n} \leqslant \sqrt[2\mathrm{p}-3]{2}, & \mathrm{p} \in \mathrm{S}_2^6, \\ \frac{1}{\sqrt[\mathrm{p}(\mathrm{p}-1)]{\mathrm{p}}} \leqslant \frac{\tau_{n+1}}{\tau_n} \leqslant \sqrt[\mathrm{p}(\mathrm{p}-1)]{\mathrm{p}}, & \mathrm{p} \in \mathrm{S}_6^8. \end{cases} \tag{69}$$

9. Numerical tests

In the composition process, κ_1 is a complex value, though the approximation of the solution y_n in considered by the real part of $\hat{\mathrm{y}}_n$ produced by $\Psi_{\tau}^{[\mathrm{p}+1]}$.

We show in this section, throughout numerical tests, that the imaginary part could represent an error estimation of the approximation. This is a powerful tool that will let us adapt the time step to reinforce the stability in case of notoriously stiff problems. In this section we introduce some notations:

- The absolute value of the exact error relative to $\Phi_\tau^{[\mathrm{p}]}$:

$$e(\mathrm{t}_n) \;\coloneqq\; \|\mathrm{y}(\mathrm{t}_n) - \mathrm{y}_n\|, \tag{70}$$

- The exact error relative to the composed numerical flow $\Psi_\tau^{[\mathrm{p}]}$:

$$\hat{e}(\mathrm{t}_n) \;\coloneqq\; \|\mathrm{y}(\mathrm{t}_n) - \mathrm{Re}(\hat{\mathrm{y}}_n)\|. \tag{71}$$

- The imaginary part of $\hat{\mathrm{y}}_n$ in its absolute value

$$|\mathrm{Im}(\hat{\mathrm{y}}_n)|. \tag{72}$$

We denote by e_n and $\hat{e}_n$ approximations to $e(\mathrm{t}_n)$ and $\hat{e}(\mathrm{t}_n)$, respectively.

9.1. The first example

We consider the IVP defined over the open interval $]0, 5[$, having the initial condition $\mathrm{y}_0 \;\coloneqq\; \mathrm{y}(0)$, $f(\mathrm{t}, \mathrm{y}) = \lambda \mathrm{y} + g(\mathrm{t})$, and $g(\mathrm{t})$ a differentiable function. The exact solution is given below:

$$\mathrm{y}(\mathrm{t}) = \mathrm{e}^{\lambda \mathrm{t}} \cdot \left(\mathrm{y}_0 + \int_0^{\mathrm{t}} \mathrm{e}^{-\lambda \tau} \cdot g(\tau) d\tau \right).$$

First, we consider $\lambda = -1/10$ and $g(\mathrm{t}) = \sin(\omega \mathrm{t})$ with $\omega = 2\pi$ and $\mathrm{y}_0 = 2$, thus the exact solution is given as follows:

$$\mathrm{y}(\mathrm{t}) = \mathrm{e}^{\lambda \mathrm{t}} \cdot \left(\mathrm{y}_0 + \frac{\omega - \mathrm{e}^{-\lambda \mathrm{t}} \cdot \big(\lambda \sin(\omega \mathrm{t}) + \omega \cos(\omega \mathrm{t})\big)}{\omega^2 + \lambda^2} \right). \tag{73}$$

We will demonstrate that the imaginary part of the numerical approximation $\hat{\mathrm{y}}_n$, obtained by the composed numerical flow $\Psi_\tau^{[\mathrm{p}]}$, represents an error estimation. We compare the error of the numerical solution with the exact one for different orders p and we compare this error with the imaginary part. Fig. 8 presents for several orders p and a fixed time step $\tau = 0.01$ the plots

of $e_n, \hat{e}_n$ and $\mathrm{Im}(\hat{y}_n)$. We can see how the composed flow produces approximations with additional accuracy and how the imaginary part is of the same order as the exact error. It is worth to notice that when $p = 8$, the error increases with the simulation time in the case when numerical simulations are performed with BDF. This is because the scheme is unstable for $p \geqslant 6$, whereas the errors of approximations produced by the composed technique does not diverge, ensuring that the composed flow is stable by breaking the Dahlquist barrier.

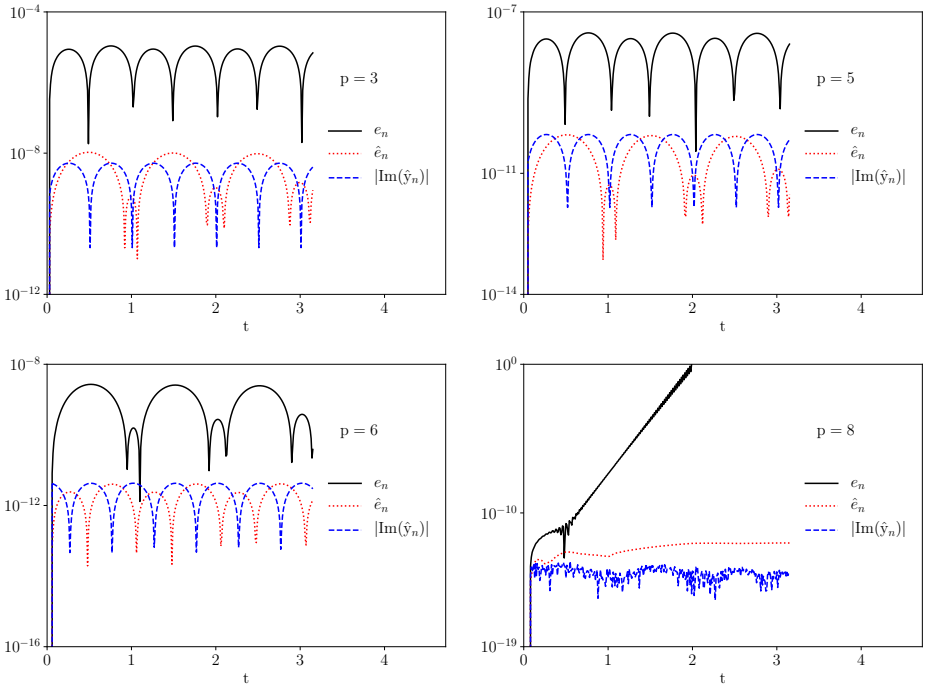

Figure 8: Plots of e_n and $\hat{e}_n$ relative of the approximation to solution (73) resulting from $\Phi^{[p]}$ and $\Psi^{[p+1]}$ respectively, with the imaginary part $\mathrm{Im}(\hat{y}_n)$ for $p \in \{3, 5, 6, 8\}$ and with $\tau = 0.01$.

9.2. ODEs with stiffness

In this section, we present two examples of ODEs with stiffness. The first one is a linear but non-homogeneous ODE while the second is a nonlinear equation. The solution of the latter is represented by the Lambert function. These examples will show the suitability of the imaginary part to follow the error and estimate it.

9.2.1. Linear equation

We consider here again the last ODE with $\lambda = -50$ and $g(\mathrm{t}) = -\lambda \arctan(20\mathrm{t})$. The simulation is spanning the segment $[0, 2\pi]$ with the initial condition $\mathrm{y}_0 \coloneqq 1$ and using the time step $\tau = 0.01$. The solution is plotted in Fig. 9. It is clear that the solution presents a steep variation around $\mathrm{t} = 0$ and $\mathrm{t} = 3$, involving a change in the time step to follow the variation and keep approximations stable.

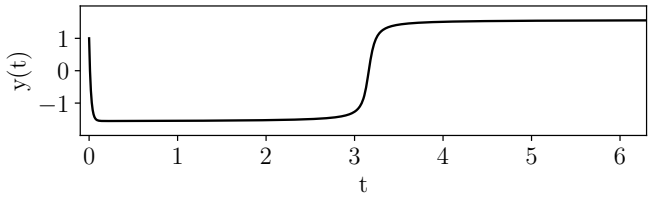

Figure 9: The exact solution of the problem $\dot{\mathrm{y}} = -50\mathrm{y} + 50\arctan(20\mathrm{t})$.

We add also the plot of the error between the exact solution and the one obtained by the composition of the BDFp for $\mathrm{p} \in \{4, 5\}$. Fig. 10 presents two evolutions of the exact errors obtained with $\Psi^{[4]}_{1/100}$ and $\Psi^{[5]}_{1/100}$. This figure represents also the imaginary part of $\hat{\mathrm{y}}_n$. First, we see that the accuracy of the solution increases when p increases. Second, we demonstrate that the evolution of the imaginary part follows that of the exact error while having the same order of magnitude.

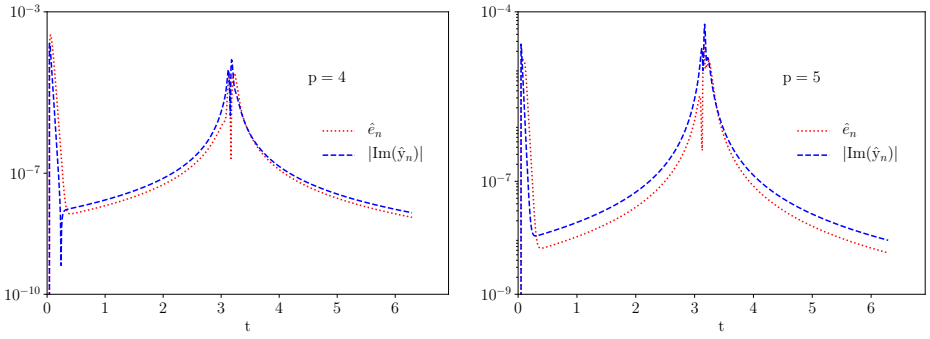

Figure 10: Evolution of the exact error $e(\mathrm{t}_n)$ and of the imaginary part of $\hat{\mathrm{y}}_n$ relative to the stiff problem defined by $g(\mathrm{t}) = -50\arctan(20\mathrm{t})$ and obtained by the composed flow $\Psi^{[\mathrm{p}+1]}$ for $\mathrm{p} \in \{4, 5\}$.

9.2.2. Lambert function

Consider the ODE defined by the right hand side $f(\mathrm{t},\mathrm{y}) \coloneqq \mathrm{y}^2 - \mathrm{y}^3$ with the initial condition $\mathrm{y}(0) = \delta$ over the segment $[0, 2/\delta]$, where its solution is given exactly by $\mathrm{y}(\mathrm{t}) \equiv \dfrac{1}{W(de^{\,d-\mathrm{t}}) + 1}$, with $d \coloneqq 1/\delta - 1$ and $W(z)$ being the Lambert function defined as the solution to the transcendent equation $We^W = z$. It is a classical example of testing the efficient of the numerical scheme in stiff case. The solution is plotted in Fig. 11 for $\delta = 0.01$.

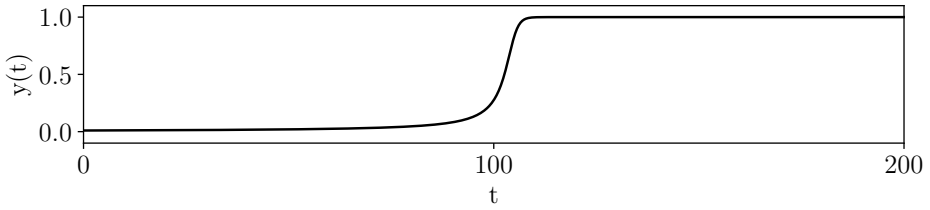

Figure 11: Exact solution of the Lambert equation when $\delta = 0.01$.

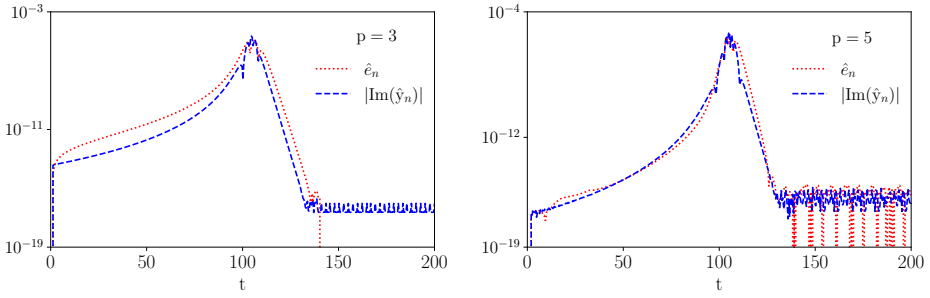

Figure 12: Evolution of the exact error $e(\mathrm{t}_n)$ and of the imaginary part of $\hat{\mathrm{y}}_n$ relative to the Lambert problem when $\delta = 0.01$. It is obtained with the composed flow $\Psi^{[\mathrm{p}+1]}$ for $\mathrm{p} \in \{3, 5\}$.

Fig. 12 presents the plots of the exact error evolution, for a fixed time step $\tau_n = 0.01$, compared with the imaginary part of the approximation obtained by the composed flow for two orders $\mathrm{p} \in \{3, 5\}$. It is clear that the latter follows the same pattern as the former. To check the efficiency of the rule of thumb formula in (69) for choosing the value of the new step size and ensure having a real positive part in the root κ_1, we proceed to update the time step with and without it. Fig. 13 presents this comparison between

the real parts during the simulation. The upper panel presents the case without it, where a negative real part occurs around t = 100, thus blocking the marching time. We can see that applying the rule of thumb presented in (69), ensures advancing in time by the composed flow as all roots of κ_1 during the simulation have positive real parts.

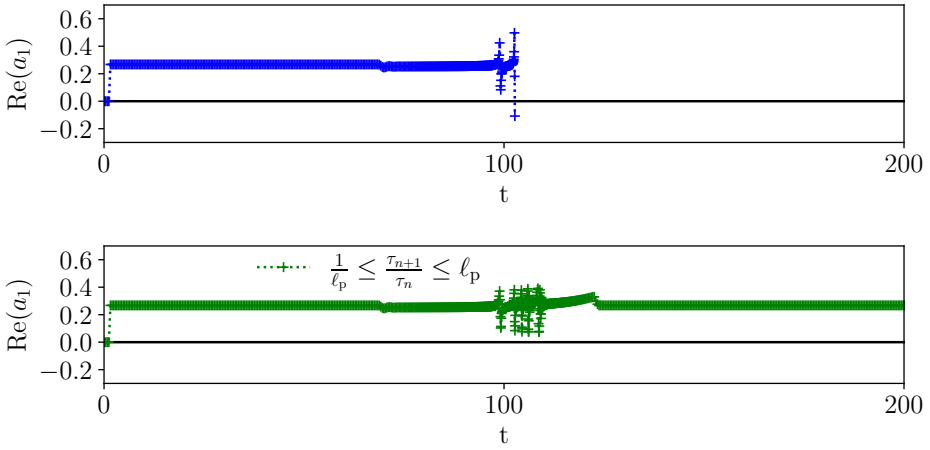

Figure 13: Evolution of the maximum of the the real part of roots κ_1 during the simulation when p = 4 and tol = 10^{-12} using the bounding strategy in Eq. (69) (lower panel) and without it (upper panel).

We end by plotting the evolution of the time step τ_n for different features of the simulation. Fig. 14 presents the evolution when applying the composed flow with different orders p $\in \{2, 3, 4\}$ and a fixed user tolerance tol. All simulations present at the beginning a steady time step as the solution shows no variation before reaching the neighborhood of t = 100, as Fig. 11 demonstrates it. We can see how the order of the scheme affect the time step adaptivity: the bigger the order is, the low the variation of the time step. However and for a fixed order p, the adaptivity of the time step size presents a high rate of variation when the tolerance is smaller. This is demonstrated in Fig. 15 for p = 2 and tol $\in \{10^{-7}, 10^{-9}, 10^{-12}\}$.

Intermediate conclusion. We conclude from these numerical experiments that the proposed numerical flow obtained by twice composing the BDFp numerical flow produces a numerical scheme where the output has two parts: the real part-an approximation to the solution with an additional accuracy, and the imaginary part- an error estimate of the resulting approximation.

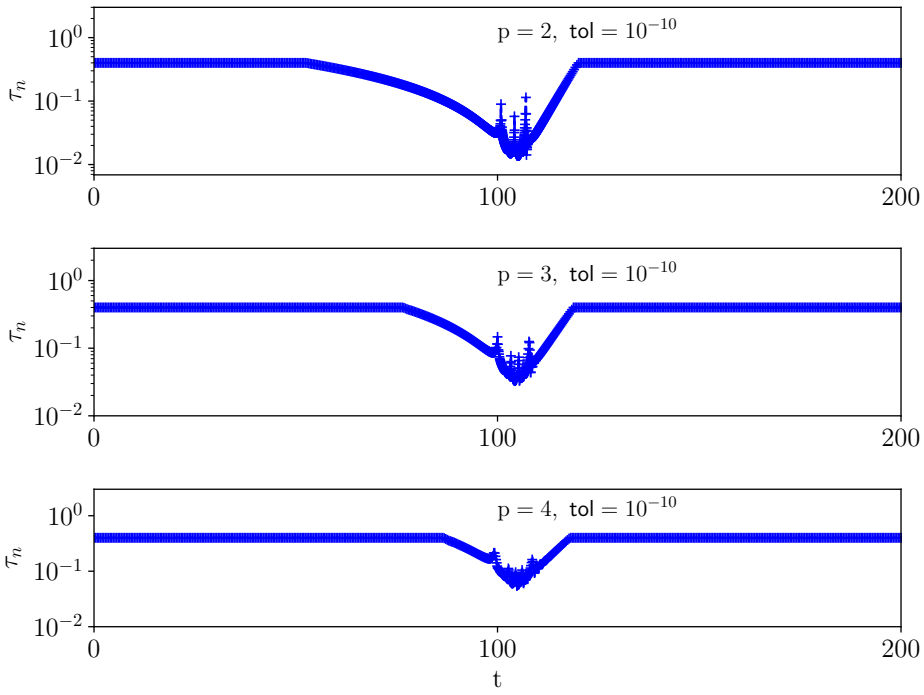

Figure 14: Evolution of the τ_n during the simulation with the adaptive Algorithm 1 for orders $\mathrm{p} = 2, 3, 4$ and $\mathsf{tol} = 10^{-10}$.

10. Conclusion and perspectives

In this paper, we presented a composition technique for implicit BDFp schemes. Unlike the cyclic composition methods proposed in the last century [57, 60], our technique is akin to those developed for one-step methods [50]. Specifically, it employs the same numerical flow with varying time step sizes.

We proved that two consecutive compositions of a BDF scheme of order p yield an approximation with an additional order of accuracy. Notably, this enhancement is achieved without requiring additional backward points. To attain the higher order, an algebraic equation must be solved once for all fixed time steps for a given order p, or at every time layer when variable time steps are used. The solution to this algebraic equation is a complex number, which can be computed numerically.

Furthermore, leveraging the difference operator for an LMS method, we demonstrated that the imaginary part of the outputs provides an error estimate for the approximation generated by the composed flow. The corre-

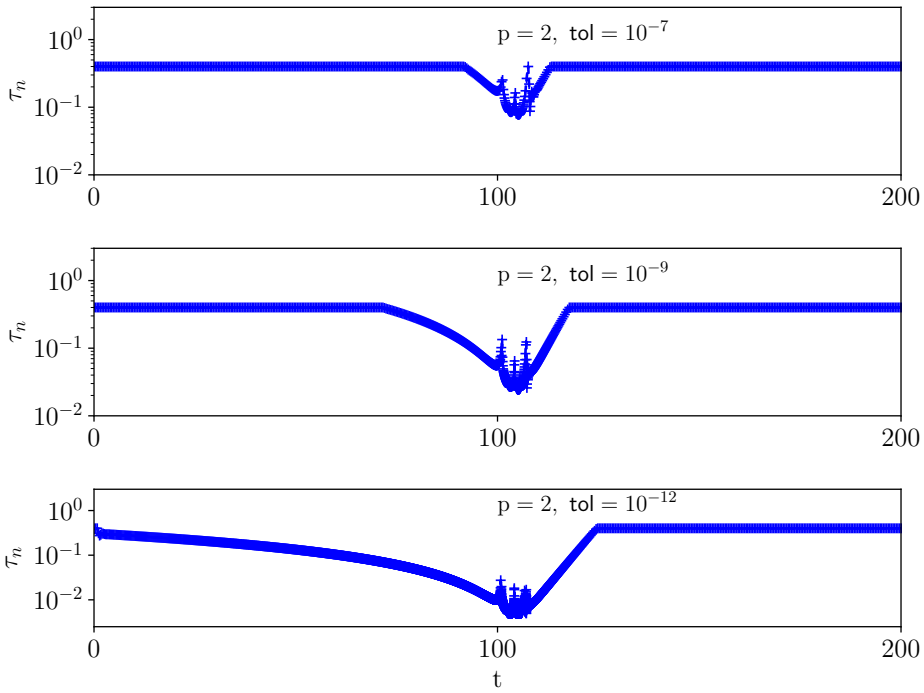

Figure 15: Evolution of the τ_n during the simulation with the adaptive Algorithm 1 for $\mathrm{p} = 2$ and various user tolerance $\mathsf{tol} \in \{10^{-7}, 10^{-9}, 10^{-12}\}$.

sponding error constant is also derived. This was illustrated through numerical experiments conducted on several classic stiff problems, both at the end of the paper and throughout the discussion.

Another advantage of the composed flow of order p^{th} is that it uses $\mathrm{p}-1$ backward points, compared to the BDFp method, which requires p points. This reduction in backward points helps minimize memory allocation.

Furthermore, numerical tests have demonstrated that the computational performance of the composed flow of order p is comparable to that of the classical BDF scheme of the same order for low orders (2 and 3) and surpasses it for higher orders. By surpassing, we mean that the composed flow achieves the same level of accuracy with lower CPU time. This improved performance is achieved with fewer backward points.

If the same number of backward points is maintained, and the computational efficiency of the composed flow is compared to the classical BDFp, the composed flow outperforms the classical BDFp for any order p.

The linear stability of the composed scheme was investigated. The generating polynomials were obtained in the case of linear differential equation and angles of stability were determined for every order. We have proved graphically that the new scheme $\Psi^{[p]}$ is $A(\vartheta)$ stable up to order $p = 8$, breaking the Dahlquist barrier, equal to six, in implicit BDF schemes.

Variable time stepping is analyzed, and conditions for ensuring a real positive part of the jumping step are presented. A lower bound formula for the ratio of two consecutive time steps is derived to guarantee stable time marching in simulations using the composed flow.

In our perspectives, the developed technique of the composition should be extended to Partial Differential Equations. To do that, we should develop the mathematical formulation to write the variational formulation of the problem in mixed form, where the real and imaginary parts are the two associated unknowns formulated in mix formulation strategy. In this case, the scalar field of the imaginary part associated to the variable in quest is of interest, not only to adapt the time step but also to use in mesh refinement. This should be investigated in future works.

Acknowledgments

This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. FSU-2023-014.

List of abbreviations

ODE Ordinary Differential Equations 4

IVP Initial Value Problem 7

LMS Linear Multi-Step 2

BDF Backward Difference formula 1

BDFp BDF of order p 5

RK Runge-Kutta 2

ERK Embedded Runge-Kutta 2

IRK Implicit Runge-Kutta 3

ETD Exponential-Time Differencing 3

DSR Divergent Series Resumation 3

Nomenclature

t	Time variable coordinate
T	The final time of simulation
y(t)	The unknown function
f	Right hand side of the differential system
e(t)	Local error
ϕ_t	Exact flow of the differential system
t_n	The n^{th} instant of time
τ_n	The n^{th} time step magnitude
y_n	The n^{th} approximation of y(t_n)
e_n	Error estimate of $e(\mathrm{t}_n)$
S_i^j	The set of positive integers numbers between i and j
$(.)\|_{\mathrm{t}=\mathrm{t}_n}$	Evaluation operator at $\mathrm{t} = \mathrm{t}_n$
$\nabla^j \mathrm{y}_n$	Backward Difference operator
$\gamma_{(i)}$	The i^{th} coefficients of BDF in case of fixed time step
$\gamma_{(i\|n)}$	The i^{th} coefficients of BDF(p) to find approximation at $\mathrm{t} = \mathrm{t}_n$
L	Difference operator
tol	User tolerance
$\mathcal{T}_{n,\mathrm{p}}$	Vector containing instants points set $\{\mathrm{t}_{n-i} \,\|\, i \in \mathrm{S}_0^{\mathrm{p}-1}\}$

$\mathrm{Y}_{n,\mathrm{p}}$ Vector containing elements of the set $\{\mathrm{y}_{n-i} \,|\, i \in \mathrm{S}_0^{\mathrm{p}-1}\}$

$\Phi_{\tau_n}^{[\mathrm{p}]}$ Numerical flow of the BDFp with a time step τ_n

κ_1 A complex number

$\mathrm{t}_{n-1/2}$ Intermediate instant between t_{n-1} and t_n

$\mathrm{y}_{n-1/2}$ Approximation of $\mathrm{y}(\mathrm{t}_{n-1/2})$

$\mathcal{T}_{n-1/2,\mathrm{p}}$ Vector of instants $\{\mathrm{t}_{n-i} \wedge \mathrm{t}_{n-1/2} \,|\, i \in \mathrm{S}_1^{\mathrm{p}-1}\}$

$\mathrm{Y}_{n-1/2,\mathrm{p}}$ Vector of approximations $\{\mathrm{y}_{n-i} \wedge \mathrm{y}_{n-1/2} \,|\, i \in \mathrm{S}_1^{\mathrm{p}-1}\}$

$\varepsilon_{(j|n)}$ The j^{th} ratio relative to the first integration in the composition

$\mathcal{T}'_{n,\mathrm{p}}$ Vector of instants $\{\mathrm{t}_{n-i} \wedge \mathrm{t}_{n-1/2} \,|\, i \in \mathrm{S}_2^{\mathrm{p}}\}$

$\mathrm{Y}'_{n,\mathrm{p}}$ Vector of approximations $\{\mathrm{y}_{n-i} \wedge \mathrm{y}_{n-1/2} \,|\, i \in \mathrm{S}_2^{\mathrm{p}}\}$

$\hat{\mathrm{y}}_n$ Approximation of $\mathrm{y}(\mathrm{t}_n)$ using double composition

$\mathrm{Re}(\hat{\mathrm{y}}_n)$ The real part of $\hat{\mathrm{y}}_n$

$\mathrm{Im}(\hat{\mathrm{y}}_n)$ The imaginary part of $\hat{\mathrm{y}}_n$

$\Psi_{\tau_n}^{[\mathrm{p}+1]}$ The numerical flow of the double composition of $\Phi^{[\mathrm{p}]}$

ℓ_{p} Bound element in the adaptive tomestep choosing process

$\xi_{(j|n)}$ j^{th} ratio relative to the second integration in the composition

$\Gamma_{(i|n)}$ i^{th} coefficients of $\mathrm{BDF}(\mathrm{p}+1)$ to find approximation at $\mathrm{t} = \mathrm{t}_n$

L Difference operator of the first integration in $\Psi^{[\mathrm{p}+1]}$

E_j j^{th} term of the Taylor development of L_1

L_2 Difference operator of the second integration in $\Psi^{[\mathrm{p}+1]}$

$\mathcal{E}_j$ j^{th} term of the Taylor development of L_2

z A complex variable

$p_{\mathrm{p}}(\omega, z)$ Polynomial associated to an LMS method

$\omega_i(z)$ i^{th} root of p_{p} as a function of z

$\mathcal{D}_{\Psi^{[\mathrm{p}]}}$ A-stability domain of the numerical flow $\Psi^{[\mathrm{p}]}$

$\vartheta_{\Psi^{[\mathrm{p}]}}$ Stability angle of the numerical flow $\Psi^{[\mathrm{p}]}$

Appendix A. Algorithm of flow $\boldsymbol{\Phi}^{[\mathrm{p}]}_{\boldsymbol{\tau_n}}$

In this section, we present the numerical flow associated with the BDFp in Algorithm 3.

Algorithm 3 The numerical flow $\Phi^{[\mathrm{p}]}_{\tau_n}(\mathcal{T}_{n-1,\mathrm{p}}, \mathrm{Y}_{n-1,\mathrm{p}})$ of BDFp

Require: τ_n, tol
  $[\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}] \leftarrow \mathsf{Coeff}\big([\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}, \tau_n]\big)$ (see Algorithm 4)
  $\ell \leftarrow 0$
  $\mathrm{y}_{n,\ell} \leftarrow \mathrm{y}_{n-1}$
  $e_{n,\ell} \leftarrow \mathsf{tol} \times 2$
  while $e_{n,\ell} > \mathsf{tol}$ **do**
    $\mathrm{y}_{n,\ell+1} \leftarrow \mathrm{F}_{\mathrm{p}}\big([\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}], \mathrm{y}_{n,\ell}\big)$
    $e_{n,\ell+1} \leftarrow \|\mathrm{y}_{n,\ell+1} - \mathrm{y}_{n,\ell}\|$
    $\mathrm{y}_{n,\ell+1} \leftarrow \mathrm{y}_{n,\ell}$
    $\ell \leftarrow \ell + 1$
  end while
  $\mathrm{y}_n \leftarrow \mathrm{y}_{n,\ell+1}$
  return $(\mathcal{T}_{n,\mathrm{p}}, \mathrm{Y}_{n,\mathrm{p}})$

Appendix B. Proof of Proposition 6

In this section, the proof of Proposition 6 is done only for $\Gamma_{(\mathrm{p}+1|n)}$. The proofs for other coefficients $\{\Gamma_{(i|n)} \,|\, i \in \mathrm{S}^{\mathrm{p}}_0\}$ can be obtained using the same strategy. Mainly, we use the Cramer rule to compute the solution of $\Gamma_{(\mathrm{p}+1|n)}$:

$$\Gamma_{(\mathrm{p}+1|n)} = \frac{\det(\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2})}{\det(\mathsf{A}'_{n,\mathrm{p}+1})}, \tag{B.1}$$

where $\det : \mathsf{Mat}_n(\mathbb{C}) \to \mathbb{C}$ is the application returning the determinant of a given matrix and $\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2}$ is the matrix obtained by replacing the right hand side in the $(\mathrm{p}+2)^{\mathrm{th}}$ column in matrix $\mathsf{A}'_{n,\mathrm{p}+1}$. We can see that the last case in the second column contains only two elements: $\xi_{(0|n)}^{\mathrm{p}+1}$ and $(-\kappa_1)^{\mathrm{p}+1}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}$ leading us to decompose the determinant into two parts as follows:

$$\det\left(\mathsf{A}'_{n,\mathrm{p}+1}\right) = \det\left(\mathsf{B}_{\mathrm{p}}\right) - \frac{\kappa_1^{\mathrm{p}+1}}{\gamma_{(0|n)}}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}\det\left(\mathsf{B}_{\mathrm{p}}^{1,\mathrm{p}+1}\right). \tag{B.2}$$

In the same context and after taking factor of $-\xi_{(0|n)}$ in the last column, we get:

$$\det\left(\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2}\right) = (-1)^{\mathrm{p}+1}\xi_{(0|n)}\left(\det\left(\mathsf{B}_{\mathrm{p}+1}\right) + \frac{\kappa_1^{\mathrm{p}+1}}{\gamma_{(0|n)}}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}\det\left(\mathsf{B}_{\mathrm{p}+1}^{1,\mathrm{p}}\right)\right), \tag{B.3}$$

where $\mathsf{B}_{\mathrm{p}-i} \in \mathsf{Mat}_{\mathrm{p}-i}(\mathbb{C})$ is the following matrix defined in Eq. (B.4) and $\mathsf{B}_{\mathrm{p}-i}^{j,\ell}$ results from B_{p} after deleting the j^{th} column and the ℓ^{th} row.

$$\mathsf{B}_{\mathrm{p}-j} := \begin{pmatrix} \xi_{(0|n)}^{1+j} & \cdots & \xi_{(\mathrm{p}-j|n)}^{1+j} \\ \vdots & \ddots & \vdots \\ \xi_{(0|n)}^{\mathrm{p}} & \cdots & \xi_{(\mathrm{p}-j|n)}^{\mathrm{p}} \end{pmatrix}. \tag{B.4}$$

After manipulating columns and rows in both B_{p} and $\mathsf{B}_{\mathrm{p}}^{1,\mathrm{p}}$, and using elementary algebraic operations, we can show that:

$$\det(\mathsf{B}_{\mathrm{p}-j}) = \prod_{i=0}^{\mathrm{p}-j}\xi_{(i|n)}^{j+1}\prod_{i=0}^{\mathrm{p}-j-1}\prod_{\ell=i+1}^{\mathrm{p}-j}(\xi_{(\ell|n)} - \xi_{(i|n)}), \tag{B.5}$$

$$\det(\mathsf{B}_{\mathrm{p}-j}^{1,\mathrm{p}-j+1}) = \prod_{i=1}^{\mathrm{p}-j}\xi_{(i|n)}^{j+1}\prod_{i=1}^{\mathrm{p}-j-1}\prod_{\ell=i+1}^{\mathrm{p}-j}(\xi_{(\ell|n)} - \xi_{(i|n)}). \tag{B.6}$$

After using formulas (B.5) and (B.6) in (B.3), and taking common factors, we get:

$$\begin{aligned}\det\left(\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2}\right) = (-1)^{\mathrm{p}+1}\xi_{(0|n)}\Bigg(&\prod_{i=0}^{\mathrm{p}-1}\xi_{(i|n)}^2\prod_{i=0}^{\mathrm{p}-2}\prod_{\ell=i+1}^{\mathrm{p}-1}(\xi_{(\ell|n)} - \xi_{(i|n)}) \\ &+ \frac{\kappa_1^{\mathrm{p}+1}}{\gamma_{(0|n)}}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}\prod_{i=1}^{\mathrm{p}-1}\xi_{(i|n)}^2\prod_{i=1}^{\mathrm{p}-2}\prod_{\ell=i+1}^{\mathrm{p}-1}(\xi_{(\ell|n)} - \xi_{(i|n)})\Bigg)\end{aligned}$$

Taking common factors, we have:

$$\begin{aligned}\det\left(\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2}\right) &= (-1)^{\mathrm{p}+1}\xi_{(0|n)}\prod_{i=1}^{\mathrm{p}+1}\xi^2_{(i|n)}\prod_{i=1}^{\mathrm{p}-2}\prod_{\ell=i+1}^{\mathrm{p}+1}(\xi_{(\ell|n)}-\xi_{(i|n)})\times \\ &\left(\xi^2_{(0|n)}\prod_{\ell=1}^{\mathrm{p}+1}(\xi_{(\ell|n)}-\xi_{(0|n)})+\frac{\kappa_1^{\mathrm{p}+1}}{\gamma_{(0|n)}}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)},\right)\end{aligned}$$

which equal to:

$$\begin{aligned}\det\left(\mathsf{A}'_{n,\mathrm{p}+1|\mathrm{p}+2}\right) &= \frac{(-1)^{\mathrm{p}+1}}{\gamma_{(0|n)}}\kappa_1^{\mathrm{p}+1}\xi_{(0|n)}\prod_{i=1}^{\mathrm{p}-1}\varepsilon_{(i|n)}\xi^2_{(i|n)}\times \\ &\prod_{i=1}^{\mathrm{p}-2}\prod_{\ell=i+1}^{\mathrm{p}-1}(\xi_{(\ell|n)}-\xi_{(i|n)})\times\left(\gamma_{(0|n)}\xi^2_{(0|n)}+\kappa_1^2\varepsilon_{(\mathrm{p}|n)}\right)\end{aligned} \tag{B.7}$$

We show also that:

$$\det(\mathsf{A}'_{n,\mathrm{p}+1}) = \frac{\kappa_1^{\mathrm{p}}}{\gamma_{(0|n)}}\prod_{i=1}^{\mathrm{p}}\varepsilon_{(i|n)}\xi_{(i|n)}\prod_{i=1}^{\mathrm{p}-1}\prod_{\ell=i+1}^{\mathrm{p}}(\xi_{(\ell|n)}-\xi_{(i|n)})\Big(\gamma_{(0|n)}\xi_{(0|n)}-\kappa_1\Big). \tag{B.8}$$

To this end, we replace Formulas (B.7) and (B.8) in (B.1). After simplifying, we obtain the formula of $\Gamma_{(\mathrm{p}+1|n)}$ in (44), which completes proof for $\Gamma_{(\mathrm{p}+1|n)}$:

$$\Gamma_{(\mathrm{p}+1|n)} = (-1)^{\mathrm{p}+1}\prod_{i=1}^{\mathrm{p}-1}\left[\frac{\xi_{(i|n)}}{\xi_{(\mathrm{p}|n)}-\xi_{(i|n)}}\right]\frac{(\kappa_1-1)\left[\xi^2_{(0|n)}\gamma_{(0|n)}+\kappa_1^2\varepsilon_{(\mathrm{p}|n)}\right]}{\xi_{(\mathrm{p}|n)}\big(r_{(\mathrm{p}|n)}+\kappa_1\big)\left[\xi_{(0|n)}\gamma_{(0|n)}-\kappa_1\right]} \tag{B.9}$$

Appendix C. Closed form of $\Gamma_{(\mathrm{p}+1|n)}$

We present in this Appendix the explicit expression of $\Gamma_{(\mathrm{p}+1|n)}$ when $\mathrm{p}\in \mathrm{S}_1^4$

$$\begin{aligned}\Gamma_{(1+1|n)} &= \frac{(\kappa_1-1)(2\kappa_1^2-2\kappa_1+1)}{\kappa_1(2\kappa_1-1)}, \\ \Gamma_{(2+1|n)} &= -\frac{(\kappa_1-1)(3\kappa_1^3+\kappa_1^2(3r_{(2|n)}-4)+\kappa_1(r^2_{(2|n)}-2r_{(2|n)}+2)+r_{(2|n)})}{r_{(2|n)}(r_{(2|n)}+\kappa_1)(r_{(2|n)}+1)(3\kappa_1^2+2\kappa_1(r_{(2|n)}-1)-r_{(2|n)})},\end{aligned}$$

$$\Gamma_{(3+1|n)} = \frac{\left[\begin{array}{l}(\kappa_1-1)(r_{(2|n)}+1)\times\\ \Big(4\kappa_1^4+\kappa_1^3(3r_{(2|n)}+4r_{(3|n)}-6)\\ +\kappa_1^2\big((r_{(2|n)}(3r_{(3|n)}-4)+\big(r_{(3|n)}^2-4r_{(3|n)}+3\big)\big)\\ +\kappa_1\big(r_{(2|n)}(r_{(3|n)}^2-2r_{(3|n)}+2)+2r_{(3|n)}\big)+r_{(2|n)}r_{(3|n)}\Big)\end{array}\right]}{\left[\begin{array}{l}r_{(3|n)}(r_{(3|n)}+\kappa_1)(r_{(3|n)}+1)(r_{(3|n)}-r_{(2|n)})\times\\ \Big(4\kappa_1^3+3\kappa_1^2(r_{(2|n)}+r_{(3|n)}-1)\\ +2\kappa_1(r_{(2|n)}r_{(3|n)}-r_{(2|n)}-r_{(3|n)})-r_{(2|n)}r_{(3|n)}\Big)\end{array}\right]},$$

Appendix D. Algorithm computing $\gamma_{(i|n)}$

We present below steps in Algorithm 4 computing $\{\gamma_{(i|n)} \,|\, i \in \mathrm{S^p}\}$ related to Eq. (17) and Eq. (18) approximating $\mathrm{y}(\mathrm{t}_n)$ using last approximations on $\mathcal{T}_{n-1,\mathrm{p}}$ and with a time step τ_n [2, page 419]. We denote by Coeff the function computing these coefficients.

Competing interests

The research leading to these results received funding from Khalifa University under Grant Agreement No FSU-2023-014. All authors have approved the submission and have no conflicts of interest to disclose.

References

[1] J. C. Butcher, General linear methods for ordinary differential equations, Mathematics and Computers in Simulation 79 (6) (2009) 1834–1845. doi:https://doi.org/10.1016/j.matcom.2007.02.006.

[2] E. Hairer, S. P. Norsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd Edition, Vol. 1 of Springer Series in Computational Mathematics, Springer, Heidelberg, 2009.

[3] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, second revised Edition, Springer Series in Computational Mathematics 14, Springer-Verlag, Berlin, 1996.

Algorithm 4 Steps to compute coefficients $\gamma_{(i|n)}$ in Eq. (17): $\mathsf{Coeff}\big([\mathrm{t}_{n-\mathrm{p}}, \ldots, \mathrm{t}_{n-1}, \tau_n]\big)$

Require: $\mathrm{t}_n \leftarrow \mathrm{t}_{n-1} + \tau_n$
Require: $\mathcal{T} \leftarrow [\mathrm{t}_n, \mathrm{t}_{n-1}, \ldots, \mathrm{t}_{n-\mathrm{p}}]$
  $\mathrm{p} \leftarrow \mathsf{size}(\mathcal{T})$
  for $i \leftarrow 0$ to p **do**
    $\gamma_{(i|n)} \leftarrow 0$
    $b \leftarrow \tau_n$
    $c \leftarrow 1$
    for $m \leftarrow 1$ to i **do**
      if $m < i - 2$ **then**
        $b \leftarrow b \times (\mathcal{T}_0 - \mathcal{T}_m)$
      end if
      if $m \neq i$ **then**
        $c \leftarrow c \times \dfrac{1}{\mathcal{T}_i - \mathcal{T}_m}$
      end if
    end for
    for $j \leftarrow \max(1, i)$ to p **do**
      if $j \geqslant 2$ **then**
        $b \leftarrow b \times (\mathcal{T}_0 - \mathcal{T}_{j-1})$
      end if
      if $j \neq i$ **then**
        $c \leftarrow c \times \dfrac{1}{\mathcal{T}_i - \mathcal{T}_j}$
      end if
      $\gamma_{(i|n)} \leftarrow \gamma_{(i|n)} + (b \times c)$
    end for
  end for
  return $[\gamma_{(0|n)}, \ldots, \gamma_{(\mathrm{p}|n)}]$

[4] C. Runge, Ueber die numerische Auflösung von differentialgleichungen, Math. Ann. (1895) 167–178.

[5] J. C. Butcher, On Runge-Kutta processes of high order, Journal of the Australian Mathematical Society 4 (2) (1964) 179–194. doi:10.1017/S1446788700023387.

[6] R. Díaz-Adame, S. Jerez, Convergence of time-splitting approximations for degenerate convection–diffusion equations with a random source, Journal of Numerical Mathematics 29 (1) (2021) 23–38. doi:10.1515/jnma-2020-0012.

[7] A. Vu, L. Cappanera, Stability and error analysis of a semi-implicit scheme for incompressible flows with variable density and viscosity, Journal of Numerical Mathematics (2024). doi:10.1515/jnma-2024-0033.

[8] K. Kropielnicka, K. Lademann, K. Schratz, Effective highly accurate time integrators for linear Klein–Gordon equations across the scales, Journal of Numerical Mathematics (2024). doi:10.1515/jnma-2023-0070.

[9] O. Axelsson, A class of A-stable methods, BIT Numerical Mathematics 9 (1969) 185–199. doi:10.1007/BF01946812.

[10] J. C. Butcher, Coefficients for the study of Runge-Kutta integration processes, Journal of Australian Mathematical Society 3 (2) (1963) 185–201.

[11] J. C. Butcher, A history of Runge-Kutta methods, App. Num. Math. 20 (3) (1996) 247–260.

[12] J. Verwer, Explicit Runge-Kutta methods for parabolic Partial Differential Equations, App. Num. Math. 22 (1) (1996) 359–379.

[13] A. Iserles, A First Course in the Numerical Analysis of Differential Equations, 2nd Edition, Cambridge University Press, Cambridge, 2008.

[14] B. C. Vermeire, Embedded paired explicit Runge-Kutta schemes, Journal of Computational Physics 487 (2023) 112–159. doi:https://doi.org/10.1016/j.jcp.2023.112159.

[15] P. Bogacki, L. F. Shampine, A 3(2) pair of Runge-Kutta formulas, App. Math. Lett. 2 (4) (1989) 321 – 325.

[16] J. Dormand, P. Prince, A family of embedded Runge-Kutta formulae, J. Comp. App. Math. 6 (1) (1980) 19 – 26. doi:http://dx.doi.org/10.1016/0771-050X(80)90013-3.

[17] J. C. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd Edition, John Wiley & Sons, West Sussex, 2008.

[18] T. B. Co, Methods of Applied Mathematics for Engineers Scientists, Michigan Technology University, Cambridge University Press, New York, 2013.

[19] L. O. Jay, Lobatto Methods, Springer, Berlin, Heidelberg, 2015, pp. 817–826. doi:10.1007/978-3-540-70529-1_123.

[20] R. Radau, Étude sur les formules d'approximation qui servent à calculer la valeur numérique d'une intégrale définie, Journal de Mathématiques Pures et Appliquées 6 (3) (1880) 283–336.

[21] E. Hairer, G. Wanner, Stiff differential equations solved by Radau methods, Journal of Computational and Applied Mathematics 111 (1) (1999) 93–111. doi:https://doi.org/10.1016/S0377-0427(99)00134-X.

[22] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving algorithms for Ordinary Differential Equations, 2nd Edition, Springer series in computational mathematics, Springer, Berlin, 2002.

[23] X. Piao, S. Bu, D. Kim, P. Kim, An embedded formula of the Chebyshev collocation method for stiff problems, Journal of Computational Physics 351 (2017) 376–391. doi:https://doi.org/10.1016/j.jcp.2017.09.046.

[24] D. A. Pope, An exponential method of numerical integration of ordinary differential equations, Commun. ACM 6 (8) (1963) 491–493. doi:10.1145/366707.367592.

[25] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numerica 19 (2010) 209–286.

[26] S. Cox, P. Matthews, Exponential Time Differencing for stiff systems, Journal of Computational Physics 176 (2) (2002) 430 – 455. doi:http://dx.doi.org/10.1006/jcph.2002.6995.

[27] S. Maset, Relative error analysis of matrix exponential approximations for numerical integration, Journal of Numerical Mathematics 29 (2) (2021) 119–158. doi:10.1515/jnma-2020-0019.

[28] A. Deeb, A. Hamdouni, E. Liberge, D. Razafindralandy, Borel-Laplace summation method used as time integration

scheme, ESAIM: Procedings and Surveys 45 (2014) 318–327. doi:https://doi.org/10.1051/proc/201445033.

[29] A. Deeb, D. Razafindralandy, A. Hamdouni, Comparison between Borel-Padé summation and factorial series, as time integration methods, Discrete and Continuous Dynamical Systems - Series S 9 (2) (2016) 393–408. doi:https://doi.org/10.3934/dcdss.2016003.

[30] D. Razafindralandy, A. Hamdouni, A. Deeb, Considering factorial series as time integration method, in: 11th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences, Vol. 1798 of AIP Conference Proceedings, American Institute of Physics, La Rochelle-France, 2017. doi:https://doi.org/10.1063/1.4972721.

[31] A. Deeb, A. Hamdouni, D. Razafindralandy, Performance of Borel-Padé-Laplace integrator for the solution of stiff and non-stiff problems, Applied Mathematics and Computation 426 (2022). doi:https://doi.org/10.1016/j.amc.2022.127118.

[32] A. Deeb, O. Kalaoun, R. Belarbi, Proper Generalized Decomposition using Taylor expansion for non-linear diffusion equations, Mathematics and Computers in Simulation 208 (2023) 71–94. doi:https://doi.org/10.1016/j.matcom.2023.01.008.

[33] G. Kirlinger, Linear multistep methods applied to stiff initial value problems–a survey, Mathematical and Computer Modelling 40 (11) (2004) 1181–1192. doi:https://doi.org/10.1016/j.mcm.2005.01.012.

[34] J. Zhan, R. LeiDu, Z. Cui, Towards preserving geometric properties of landau-lifshitz-gilbert equation using multistep methods, Communications in Computational Physics 35 (5) (2024) 1327–1351. doi:https://doi.org/10.4208/cicp.OA-2023-0201.

[35] R. I. Okounghae, M. N. O. Ikhile, On the construction of high order A(α)-stable hybrid linear multistep methods for stiff IVPs in ODEs, Numerical Analysis and Applications 5 231–241. doi:https://doi.org/10.1134/S1995423912030056.

[36] A. Xiao, G. Zhang, X. Yi, Two classes of implicit–explicit multistep methods for nonlinear stiff initial-value problems,

Applied Mathematics and Computation 247 (2014) 47–60. doi:https://doi.org/10.1016/j.amc.2014.08.066.

[37] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations, Mathematica Scandinavica 4 (1956) 33–53.

[38] G. G. Dahlquist, A special stability problem for linear multistep methods, BIT Numerical Mathematics 3 (1963) 27–43. doi:https://doi.org/10.1007/BF01963532.

[39] E. Hairer, G. Wanner, On the instability of the BDF formulas, SIAM Journal on Numerical Analysis 20 (6) (1983) 1206–1209. doi:10.1137/0720090.

[40] O. Bokanowski, A. Picarelli, C. Reisinger, Stability and convergence of second order Backward Differentiation schemes for parabolic Hamilton-Jacobi-Bellman equations, Numerische Mathematik 148 (2021) 187–222. doi:https://doi.org/10.1007/s00211-021-01202-x.

[41] W. Wang, M. Mao, Z. Wang, Stability and error estimates for the variable step-size BDF2 method for linear and semilinear parabolic equations, Advances in Computational Mathematics 47 (2021) 187–222. doi:https://doi.org/10.1007/s10444-020-09839-2.

[42] G. Akrivis, Implicit–explicit multistep methods for nonlinear parabolic equations, Mathematics of Computation 82 (2013) 45–68. doi:10.1090/S0025-5718-2012-02628-7.

[43] G. Akrivis, E. Katsoprinakis, An analogue to the A(ϑ)-stability concept for implicit-explicit BDF methods, SIAM Journal on Numerical Analysis 58 (6) (2020) 3475–3503. doi:10.1137/19M1275103.

[44] D. Xu, The long time error estimates for the second order backward difference approximation to sub-diffusion equations with boundary time delay and feedback gain, Mathematics and Computers in Simulation 208 (2023) 186–206. doi:10.1016/j.matcom.2023.01.027.

[45] W. Gragg, H. Stetter, Generalized multistep predictor-corrector methods, Journal of ACM 11 (1964) 188–209. doi:https://doi.org/10.1145/321217.321223.

[46] J. Becker, A second order Backward Difference method with variable steps for a parabolic problem, BIT Numerical Mathematics 38 (1998) 644–662. doi:https://doi.org/10.1007/BF02510406.

[47] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A 150 (5) (1990) 262–268. doi:10.1016/0375-9601(90)90092-3.

[48] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Physics Letters A 146 (6) (1990) 319–323. doi:10.1016/0375-9601(90)90962-N.

[49] R. I. McLachlan, On the numerical integration of ordinary differential equations by symmetric composition methods, SIAM Journal on Scientific Computing 16 (1) (1995) 151–168. doi:10.1137/0916010.

[50] S. Blanes, High order numerical integrators for differential equations using composition and processing of low order methods, Applied Numerical Mathematics 37 (3) (2001) 289–306. doi:https://doi.org/10.1016/S0168-9274(00)00044-1.

[51] S. Blanes, F. Casas, A. Murua, Composition methods for differential equations with processing, SIAM Journal on Scientific Computing 27 (6) (2006) 1817–1843. doi:10.1137/030601223.

[52] S. Blanes, F. Casas, A. Murua, Splitting and composition methods in the numerical integration of differential equations, Bol. Scc. Esp. Mat. Apl. 45 (2008) 89–145.

[53] F. Casas, P. Chartier, A. Escorihuela-Tomàs, Y. Zhang, Compositions of pseudo-symmetric integrators with complex coefficients for the numerical integration of differential equations, J. Comput. Appl. Math. 381 (2021) 113006.

[54] F. Castella, P. Chartier, S. Descombes, G. Vilmart, Splitting methods with complex times for parabolic equations, BIT Numerical Mathematics 49 (2009) 487–508.

[55] S. Blanes, F. Casas, A. Murua, Splitting methods with complex coefficients, SeMA Journal: Bulletin of the Spanish Society of Applied Mathematics 50 (2010) 47–60. doi:https://doi.org/10.1007/BF03322541.

[56] S. Blanes, F. Casas, J. Ros, Symplectic integration with processing: A general study, SIAM Journal on Scientific Computing 21 (2) (1999) 711–727. doi:10.1137/S1064827598332497.

[57] E. Hansen, Cyclic composite multistep predictor-corrector methods, 1969, pp. 135–139. doi:10.1145/800195.805925.

[58] J. Donelson III, E. Hansen, Cyclic composite multistep predictor-corrector methods, SIAM Journal on Numerical Analysis 8 (1) (1971) 137–157. doi:10.1137/0708018.

[59] T. Bickart, Z. Picel, High order stiffly stable composite multistep methods for numerical integration of stiff differential equations, BIT Numerical Mathematics 13 (1973) 272–286. doi:https://doi.org/10.1007/BF01951938.

[60] J. Cash, On a class of cyclic methods for the numerical integration of stiff systems of O.D.E.s, BIT Numerical Mathematics 17 (1977) 270–280. doi:https://doi.org/10.1007/BF01932147.

[61] J. Tendler, T. Bickart, Z. Picel, A stiffly stable integration process using cyclic composite methods, ACM Transactions on Mathematical Software 4 (1977) 339–368. doi:10.1137/0708018.

[62] C. Johnson, Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations, SIAM Journal on Numerical Analysis 25 (4) (1988) 908–926. doi:10.1137/0725051.

[63] C. Chang-Koon, C. Heung-Jin, Error estimates and adaptive time stepping for various direct time integration methods, Computers & Structures 60 (6) (1996) 923–944. doi:10.1016/0045-7949(95)00452-1.

[64] G. Söderlind, Automatic control and adaptive time-stepping, Numerical Algorithms 31 (2002) 281–310. doi:10.1023/A:1021160023092.

[65] D. Yan, M. Pugh, F. Dawson, Adaptive time-stepping schemes for the solution of the Poisson-Nernst-Planck equations, Applied Numerical Mathematics 163 (2021) 254–269. doi:10.1016/j.apnum.2021.01.018.

[66] M. K. Jaradat, W. Sik Yang, An adaptive time-stepping control algorithm for molten salt reactor transient analyses, Annals of Nuclear Energy 190 (2023) 109880. doi:10.1016/j.anucene.2023.109880.

[67] Y. Wang, W. T. Leung, An adaptive space and time method in partially explicit splitting scheme for multiscale flow problems, Computers & Mathematics with Applications 144 (2023) 100–123. doi:10.1016/j.camwa.2023.05.034.

[68] V. Baron, Y. Coudière, P. Sochala, Adaptive multistep time discretization and linearization based on a posteriori error estimates for the Richards equation, Applied Numerical Mathematics 112 (2017) 104–125. doi:10.1016/j.apnum.2016.10.005.

[69] C. Arévalo, G. Söderlind, H. Yiannis, I. Feket, Local error estimation and step size control in adaptive linear multistep methods, Numerical Algorithms 86 (2021) 537–563. doi:10.1007/s11075-020-00900-1.

[70] X. Meng, Z. Zhang, An adaptive BDF2 implicit time-stepping method for the no-slope-selection epitaxial thin film model, Computational and Applied Mathematics 42 (124) (2021). doi:10.1007/s40314-023-02250-9.

[71] G. Söderlind, L. Wang, Adaptive time-stepping and computational stability, J. Comput. Appl. Math. 185 (2) (2006) 225–243.

[72] N. Ganesh, N. Balakrishnan, A h-adaptive algorithm using residual error estimates for fluid flows, Communications in Computational Physics 13 (2) (2013) 461–478. doi:https://doi.org/10.4208/cicp.170811.210212a.

[73] S. Khashin, Estimating the error in the classical Runge-Kutta methods, Computational Mathematics and Mathematical Physics 54 (2014) 767–774. doi:10.1134/S0965542514050145.

[74] M. Hochbruck, T. Pazur, R. Schnaubelt, Error analysis of implicit Runge-Kutta methods for quasilinear hyperbolic evolution equations, Numerische Mathematik 138 (2018) 557–579. doi:10.1007/s00211-017-0914-6.

[75] J. D. Lambert, On the local error and the local truncation error of linear multistep methods, BIT Numerical Mathematics 30 (1990) 637–681. doi:10.1007/BF01933215.

[76] S. Blanes, F. Casas, M. Thalhammer, Splitting and composition methods with embedded error estimators, Applied Numerical Mathematics 146 (2019) 400–415. doi:https://doi.org/10.1016/j.apnum.2019.07.022.

[77] H.-l. Liao, T. Tang, T. Zhou, On energy stable, maximum-principle preserving, second-order BDF scheme with variable steps for the Allen–Cahn equation, SIAM Journal on Numerical Analysis 58 (4) (2020) 2294–2314. doi:10.1137/19M1289157.

[78] Z. Li, H.-l. Liao, Stability of variable-step BDF2 and BDF3 methods, SIAM Journal on Numerical Analysis 60 (4) (2022) 2253–2272. doi:10.1137/21M1462398.

[79] H.-l. Liao, Z. Zhang, Analysis of adaptive BDF2 scheme for diffusion equations, Mathematics of Compution 90 (2021) 1207–1226. doi:10.1090/mcom/3585.

[80] H.-l. Liao, T. Tang, T. Zhou, Discrete energy analysis of the third-order variable-step BDF time-stepping for diffusion equations, Journal of Computational Mathematics 41 (2) (2023) 325–344. doi:10.4208/jcm.2207-m2022-0020.

[81] A. Laadhari, A. Deeb, B. Kawi, Hydrodynamics simulation of red blood cells: Employing a penalty method with double jump composition of lower order time integrator, Mathematical Methods in the Applied Sciences (August 2023). doi:https://doi.org/10.1002/mma.9607.

[82] A. Laadhari, A. Deeb, Computational modeling of individual Red Blood Cell dynamics using discrete flow composition and adaptive time-stepping strategies, Symmetry 15 (2023) 1138. doi:https://doi.org/10.3390/sym15061138.

[83] A. Deeb, D. Dutykh, Error estimation for numerical approximations of ODEs via composition techniques. Part I: One-step methods, submitted (Juin 2024). doi:https://doi.org/10.48550/arXiv.2409.10548.

[84] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962.

[85] I. Ramière, T. Helfer, Iterative residual-based vector methods to accelerate fixed point iterations, Computers &

Mathematics with Applications 70 (9) (2015) 2210–2226. doi:https://doi.org/10.1016/j.camwa.2015.08.025.

[86] R. Martinez-Guerra, J. P. Flores-Flores, A fixed-point state observer with Steffensen-Aitken accelerated convergence, Journal of the Franklin Institute 360 (10) (2023) 6757–6782. doi:https://doi.org/10.1016/j.jfranklin.2023.04.023.

[87] M. Gander, G. Wanner, Exact BDF stability angles with Maple, BIT Numerical Mathematics 60 (2020) 615–617. doi:10.1007/s10543-019-00796-x.