Variational time discretizations of higher order and higher regularity

Simon Becher¹ · Gunar Matthies¹

Received: 10 March 2020 / Accepted: 4 February 2021 / Published online: 24 March 2021
© The Author(s) 2021

Abstract

We consider a family of variational time discretizations that are generalizations of discontinuous Galerkin (dG) and continuous Galerkin–Petrov (cGP) methods. The family is characterized by two parameters. One describes the polynomial ansatz order while the other one is associated with the global smoothness that is ensured by higher order collocation conditions at both ends of the subintervals. Applied to Dahlquist’s stability problem, the presented methods provide the same stability properties as dG or cGP methods. Provided that suitable quadrature rules of Hermite type are used to evaluate the integrals in the variational conditions, the variational time discretization methods are connected to special collocation methods. For this case, we present error estimates, numerical experiments, and a computationally cheap postprocessing that allows to increase both the accuracy and the global smoothness by one order.

Keywords Discontinuous Galerkin · Continuous Galerkin–Petrov · Stability · Collocation method · Postprocessing · Superconvergence

Mathematics Subject Classification 65L05 · 65L20 · 65L60

1 Introduction

This paper deals with a family of variational time discretizations that generalizes discontinuous Galerkin (dG) and continuous Galerkin–Petrov (cGP) methods. For its definition, in addition to variational equations, collocation conditions are used.

Communicated by Rolf Stenberg.

✉ Simon Becher
Simon.Becher@tu-dresden.de

Gunar Matthies
Gunar.Matthies@tu-dresden.de

¹ Institute of Numerical Mathematics, Technical University of Dresden, 01062 Dresden, Germany

The considered family is characterized by two parameters: the first one is the local polynomial ansatz order and the second parameter is related to the global smoothness of the numerical solution that is ensured by higher order collocation conditions at both ends of the subintervals. Note that a related approach, which also combines variational conditions and a certain number of other conditions, was considered in [12]. Since the test space for each family member is allowed to be discontinuous with respect to the time mesh, the discrete problem can be solved in a time-marching process, i.e., by a sequence of local problems on the subintervals.

With respect to their behavior when applied to Dahlquist's stability problem, the family of variational time discretizations can be divided into two groups. While the first group shares its stability properties with the cGP method, the second group behaves like the dG method. These observations, presented in [10], imply that the considered methods are at least *A*-stable. This suggests that the whole family of methods is appropriate to handle stiff problems such as those arising, for example, from semi-discretizations of parabolic partial differential equations in space.

Various aspects of dG and cGP time discretization methods have already been studied in the literature. The a priori and a posteriori analysis of dG methods is well understood, see e.g. [15], [28, Chap. 12], and references therein. Moreover, dG methods are known to be strongly *A*-stable. Under certain conditions, dG schemes can preserve various versions of dissipativity, see [17] for details. For the cGP method a rigorous a priori and a posteriori error analysis in the context of ordinary differential equations is presented in [16]. Optimal error estimates and superconvergence results for the fully discretized heat equation, based on a finite element method in space and a cGP method in time, are given in [7]. For further investigations of the cGP method we refer to [27]. Therein, using energy arguments, also the *A*-stability of cGP methods of arbitrary order is shown. In addition, cGP methods provide an energy decreasing property for the gradient flow equation of an energy functional, see [27]. As shown in [18], the cGP schemes are also able to preserve certain energy properties such as Lyapunov functionals.

Postprocessing techniques for dG and cGP methods applied to systems of ordinary differential equations have been given in [24]. They allow to obtain an improved convergence by one order in integral-based norms. Furthermore, the postprocessed dG solution is continuous while the postprocessing of cGP solutions leads to continuously differentiable trajectories. This paper transfers and generalizes the postprocessing ideas to the whole family of variational time discretizations. The postprocessing technique applied to a numerical solution increases the global smoothness by one differentiation order and the local polynomial degree by one. This results in improved accuracy in integral-based norms, see [1,3,4,6,8,14,24]. Note that the postprocessing comes with almost no computational costs since just jumps of derivatives of the discrete solution are needed. Beside the improvements of accuracy and global smoothness, the postprocessing can be used to drive an efficient adaptive time step control, see [2].

As mentioned above, the variational time discretizations analyzed in this paper use collocation conditions at the end points of the time subintervals. We describe connections between pure collocation methods and numerically integrated variational time discretizations provided a suitable quadrature rule of Hermite type is applied. Based

on these connections the existence and uniqueness of discrete solutions are shown. Moreover, optimal error estimates follow. The connection between collocation methods and postprocessed numerically integrated discontinuous Galerkin methods using the right-sided Gauss–Radau quadrature formulas was considered in [29]. Moreover, connections between collocation methods and the numerically integrated continuous Galerkin–Petrov methods (using interpolatory quadrature formulas with as many quadrature points as number of independent variational conditions) were shown in [21,22].

For affine linear systems of ordinary differential equations with time-independent coefficients, an interpolation cascade is presented that allows multiple postprocessing steps leading to very accurate solutions with low computational costs. Moreover, temporal derivatives of discrete solutions to affine linear systems with time-independent coefficients form also solutions of variational time discretization schemes. This relation was used to prove optimal error estimates of stabilized finite element methods for linear first-order partial differential equations [14] and for parabolic wave equations [6,8].

The paper is organized as follows. In Sect. 2 some notation and the family of variational time discretizations are introduced. The general postprocessing technique is considered in Sect. 3. The connection between collocation methods and numerically integrated variational time discretizations is given in Sect. 4. This connection is exploited to provide results on the existence of unique solutions and to obtain error estimates. Section 5 is devoted to affine linear ODE systems for which an interpolation cascade that enables multiple postprocessing steps is presented. Moreover, properties of derivatives of solutions to variational time discretization methods are discussed. Numerical experiments supporting the theoretical results are presented in Sect. 6.

2 Notation and formulation of the methods

We consider the initial value problem

$$Mu'(t) = F(t, u(t)), \quad u(t_0) = u_0 \in \mathbb{R}^d, \quad (2.1)$$

where $M \in \mathbb{R}^{d \times d}$ is a regular matrix and F , sufficiently smooth, satisfies a Lipschitz-condition with respect to the second variable. Furthermore, let $I = (t_0, t_0 + T]$ be an arbitrary but fixed time interval with positive length T . The value u_0 at $t = t_0$ is called the initial value in the following.

If the ODE system (2.1) originates from a finite element semi-discretization in space of a parabolic partial differential equation then M is the mass matrix. Since in this context the computation of M^{-1} is costly, usually a linear system with M is solved instead. By the explicit occurrence of M we can investigate where this is necessary.

To describe the vector-valued case ($d > 1$) in an easy way, let $(\cdot, \cdot)$ be the standard inner product and $\|\cdot\|$ the Euclidean norm on $\mathbb{R}^d$, $d \in \mathbb{N}$. Also, let e_j be the j th standard unit vector in $\mathbb{R}^d$, $1 \leq j \leq d$.

For an arbitrary interval J and $q \in \mathbb{N}$, the spaces of continuous and p times continuously differentiable $\mathbb{R}^q$ -valued functions on J are written as $C(J, \mathbb{R}^q)$ and

$C^p(J, \mathbb{R}^q)$, respectively. Furthermore, the space of square-integrable $\mathbb{R}^q$ -valued functions is denoted by $L^2(J, \mathbb{R}^q)$ or, for convenience, sometimes also by $C^{-1}(J, \mathbb{R}^q)$. For $s \in \mathbb{N}_0$ we write $P_s(J, \mathbb{R}^q)$ for the space of $\mathbb{R}^q$ -valued polynomials on J of degree less than or equal to s . Further notation is introduced later at the beginning of the sections where it is needed.

In order to describe the methods, we need a time mesh. Therefore, the interval I is decomposed by

$$t_0 < t_1 < \cdots < t_{N-1} < t_N = t_0 + T$$

into N disjoint subintervals $I_n := (t_{n-1}, t_n]$, $n = 1, \dots, N$. Furthermore, we set

$$\tau_n := t_n - t_{n-1}, \quad \tau := \max_{1 \leq n \leq N} \tau_n.$$

For any piecewise continuous function v we define by

$$v(t_n^+) := \lim_{t \rightarrow t_n+0} v(t), \quad v(t_n^-) := \lim_{t \rightarrow t_n-0} v(t), \quad [v]_n := v(t_n^+) - v(t_n^-)$$

the one-sided limits and the jump of v at t_n . Moreover, standard notation for the floor function is used, i.e., for any real number x let $\lfloor x \rfloor$ be the greatest integer less than or equal to x .

In this paper C denotes a generic constant independent of the time mesh parameter τ .

2.1 Local formulation

Let $r, k \in \mathbb{N}_0$ with $0 \leq k \leq r$. In order to numerically solve the initial value problem (2.1), we introduce the family of variational time discretization methods $\mathbf{VTD}_k^r$ with parameters r and k .

Using a standard time-marching strategy, the discrete solution is successively determined on I_n , $n = 1, \dots, N$, by local problems of the form

Find $U \in P_r(I_n, \mathbb{R}^d)$ such that

$$U(t_{n-1}^+) = U(t_{n-1}^-), \quad \text{if } k \geq 1, \quad (2.2a)$$

$$MU^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, U(t)) \right) \Big|_{t=t_n^-}, \quad \text{if } k \geq 2, \ i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1, \quad (2.2b)$$

$$MU^{(i+1)}(t_{n-1}^+) = \frac{d^i}{dt^i} \left(F(t, U(t)) \right) \Big|_{t=t_{n-1}^+}, \quad \text{if } k \geq 3, \ i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1, \quad (2.2c)$$

and

$$\mathcal{J}_n[(MU', \varphi)] + \delta_{0,k}(M[U]_{n-1}, \varphi(t_{n-1}^+)) = \mathcal{J}_n[(F(\cdot, U(\cdot)), \varphi)] \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d) \quad (2.2d)$$

where $U(t_0^-) = u_0$ and $\delta_{i,j}$ denotes the Kronecker delta. Moreover, the integrator $\mathcal{J}_n$ represents either the integral over I_n or the application of a quadrature formula for approximate integration over I_n . Details will be described later.

Note that the formulation can easily be extended to the case $k = r + 1$. Then the variational condition (2.2d) must formally hold for all $\varphi \in P_{-1}(I_n, \mathbb{R}^d)$. This should be interpreted as “there is no variational condition”. Hence, only conditions at both ends of the interval I_n are used.

The method $\mathbf{VTD}_k^r$ can shortly be described by

$$\begin{array}{l|l} \text{trial space: } P_r, & \text{if } k \geq 1 : \text{initial condition,} \\ \text{test space: } P_{r-k}, & \text{if } k \geq 2 : \text{ODE}^{(i)} \text{ in } t_n^-, \quad i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1, \\ & \text{if } k \geq 3 : \text{ODE}^{(i)} \text{ in } t_{n-1}^+, \quad i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1. \end{array}$$

The notation $\text{ODE}^{(i)}$ means that the discrete solution fulfills the i th derivative of the system of ordinary differential equations. Obviously, the reduction of the test space for $k \geq 1$ is compensated by other conditions. For a somewhat related approach see [12, (3.3)].

Counting the number of conditions leads for $k \geq 1$ to

$$\begin{aligned} \dim P_{r-k} + 1 + \lfloor \frac{k}{2} \rfloor + \lfloor \frac{k-1}{2} \rfloor &= r - k + 1 + 1 + \frac{k}{2} + \frac{k-1}{2} - \frac{1}{2} \\ &= r - k + 2 + k - 1 = r + 1 \end{aligned}$$

while we have $\dim P_r = r + 1$ conditions if $k = 0$. The number of degrees of freedom equals for all k to $\dim P_r = r + 1$. Hence, in any case the number of conditions coincides with the number of degrees of freedom.

Remark 2.1 The $\mathbf{VTD}_k^r$ framework generalizes two well-known types of variational time discretization methods. The method $\mathbf{VTD}_0^r$ is the discontinuous Galerkin method dG(r) whereas the method $\mathbf{VTD}_1^r$ equates to the continuous Galerkin–Petrov method cGP(r).

On closer considerations we see that methods $\mathbf{VTD}_k^r$ with even k are dG-like since there are point conditions on the $\lfloor \frac{k}{2} \rfloor$ th derivative of the discrete solution but this derivative might be discontinuous. The methods $\mathbf{VTD}_k^r$ with odd k are cGP-like since there are point conditions up to the $\lfloor \frac{k}{2} \rfloor$ th derivative of the discrete solution and this derivative is continuous. We have in detail

$$\mathbf{VTD}_k^r \hat{=} \begin{cases} \text{dG}(r), & k = 0, \\ \text{cGP}(r), & k = 1, \\ \text{dG-C}^{\lfloor \frac{k-1}{2} \rfloor}(r), & k \geq 2, k \text{ even}, \\ \text{cGP-C}^{\lfloor \frac{k-1}{2} \rfloor}(r), & k \geq 3, k \text{ odd}, \end{cases}$$

where we use and generalize the definitions and notation of [24]. Note that there is also another reason to name the methods this way. All methods with odd k share their

A-stability with the cGP method while methods with even k are strongly A-stable as the dG method, for more details see Remark 5.3.

In order to obtain a fully computable discrete problem usually a quadrature formula Q_n is chosen as integrator, i.e., $\mathcal{I}_n = Q_n$. To indicate this choice, we simply write Q_n -VTD $_k^r$. Moreover, we agree that integration over I_n is used if no quadrature rule is specified. We mostly use quadrature rules that are exact for polynomials of degree up to $2r - k$. This ensures in the case of an affine linear right-hand side $F(t, u) = f(t) - Au$ with time-independent A that all u depending terms in (2.2d) are integrated exactly.

The special structure of the method (2.2) motivates to use an associated interpolation operator that conserves derivatives at the end points of the interval up to a certain order. In detail, we define on the reference interval $[-1, 1]$ the interpolation operator $\widehat{\mathcal{I}}_k^r : C^{\lfloor \frac{k}{2} \rfloor}([-1, 1]) \rightarrow P_r([-1, 1])$ that uses the interpolation points

- at the left end: derivatives up to order $\lfloor \frac{k-1}{2} \rfloor$ in -1^+ ,
- at the right end: derivatives up to order $\lfloor \frac{k}{2} \rfloor$ in 1^- ,
- in the interior: zeros $\hat{t}_i \in (-1, 1)$ of the $(r - k)$ th Jacobi-polynomial
with respect to the weight $(1 + \hat{t})^{\lfloor \frac{k-1}{2} \rfloor + 1} (1 - \hat{t})^{\lfloor \frac{k}{2} \rfloor + 1}$. (2.3)

Note that there is no point evaluation at the left end for $k = 0$. In any case, the number of interpolation conditions is

$$r - k + \lfloor \frac{k}{2} \rfloor + 1 + \lfloor \frac{k-1}{2} \rfloor + 1 = r - k + k - 1 + 2 = r + 1$$

and, thus, coincides with the dimension of P_r . The interpolation operator $\widehat{\mathcal{I}}_k^r$ is of Hermite-type and provides the standard error estimates for Hermite interpolation.

In addition, we define by

$$\widehat{Q}_k^r[f] := \int_{-1}^1 (\widehat{\mathcal{I}}_k^r f)(\hat{t}) d\hat{t}$$

a quadrature rule on $[-1, 1]$ that is in a natural way assigned to the method VTD $_k^r$. The quadrature rules $\widehat{Q}_k^r$ are known in the literature as generalized Gauss–Radau and Gauss–Lobatto formulas, respectively, see e.g. [19, 23]. The weights of the quadrature rule $\widehat{Q}_k^r$ can be calculated by integrating the appropriate Hermite basis functions on $[-1, 1]$. Finally, we obtain

$$\begin{aligned} \int_{-1}^1 \widehat{\varphi}(\hat{t}) d\hat{t} &\approx \widehat{Q}_k^r[\widehat{\varphi}] = \int_{-1}^1 (\widehat{\mathcal{I}}_k^r \widehat{\varphi})(\hat{t}) d\hat{t} \\ &= \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} w_i^L \widehat{\varphi}^{(i)}(-1^+) + \sum_{i=1}^{r-k} w_i^I \widehat{\varphi}(\hat{t}_i) + \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} w_i^R \widehat{\varphi}^{(i)}(+1^-). \end{aligned}$$

The quadrature rule $\widehat{Q}_k^r$ is exact for polynomials up to degree $2r - k$. It can be shown that all quadrature weights are different from zero, see [23]. More precisely, we have

$$w_j^I > 0, \quad w_j^L > 0, \quad (-1)^j w_j^R > 0,$$

so even the sign of the weights is known. Note that for $k \geq 2$ not all weights are positive. Semi-explicit and recursive formulas for the weights of these methods can be found in [26].

Transferring the quadrature rule $\widehat{Q}_k^r$ and the interpolation operator $\widehat{\mathcal{I}}_k^r$ from $[-1, 1]$ to the interval I_n , we obtain Q_k^r and $\mathcal{I}_k^r$ where we skip the interval I_n in the notation because this will be clear from the context. Hence, we have

$$\begin{aligned} \int_{I_n} \varphi(t) dt &\approx Q_k^r[\varphi] \\ &= \frac{\tau_n}{2} \left[\sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} w_i^L \left(\frac{\tau_n}{2}\right)^i \varphi^{(i)}(t_{n-1}^+) + \sum_{i=1}^{r-k} w_i^I \varphi(t_{n,i}) + \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} w_i^R \left(\frac{\tau_n}{2}\right)^i \varphi^{(i)}(t_n^-) \right] \end{aligned}$$

where $t_{n,i} = \frac{1}{2}(t_{n-1} + t_n + \tau_n \hat{t}_i) \in I_n$, $i = 1, \dots, r - k$. The factors $\left(\frac{\tau_n}{2}\right)^i$ originate from the chain rule.

Remark 2.2 The quadrature rule Q_0^r is the well-known right-sided Gauss–Radau quadrature formula with $r + 1$ points that is typically used for the discontinuous Galerkin method dG(r). Q_1^r is the Gauss–Lobatto quadrature rule with $r + 1$ points that is often used together with the continuous Galerkin–Petrov method cGP(r).

Remark 2.3 For $1 \leq k \leq r$ the **VTD** $_k^r$ method with exact integration could be analyzed also in a generalization of the unified framework of [5] as we shall show below, see (2.5). Note that the dG method ($k = 0$) has already been fitted in this framework, cf. [5].

We define for $k \geq 1$ a projection operator $\mathcal{P}_n : C^{\lfloor \frac{k}{2} \rfloor - 1}(\bar{I}_n, \mathbb{R}^d) \rightarrow P_{r-1}(\bar{I}_n, \mathbb{R}^d)$ by

$$(\mathcal{P}_n v)^{(i)}(t_n^-) = v^{(i)}(t_n^-), \quad \text{if } k \geq 2, i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1, \quad (2.4a)$$

$$(\mathcal{P}_n v)^{(i)}(t_{n-1}^+) = v^{(i)}(t_{n-1}^+), \quad \text{if } k \geq 3, i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1, \quad (2.4b)$$

and

$$\int_{I_n} (\mathcal{P}_n v(t), \varphi(t)) dt = \int_{I_n} (v(t), \varphi(t)) dt \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d). \quad (2.4c)$$

Then an equivalent formulation of (2.2) with $1 \leq k \leq r$ and exact integration reads

Find $U \in P_r(I_n, \mathbb{R}^d)$ with given $U(t_{n-1}) \in \mathbb{R}^d$ such that

$$MU'(t) = \mathcal{P}_n F(t, U(t)) \quad \forall t \in I_n \quad (2.5)$$

where $U(t_0) = u_0$.

Indeed, if U solves (2.2) then $MU' \in P_{r-1}(I_n, \mathbb{R}^d)$ obviously satisfies all conditions of (2.4) with $v = F(\cdot, U(\cdot))$. Since $\mathcal{P}_n v$ is uniquely defined we directly get (2.5).

Otherwise let U solve (2.5). Since there are polynomials on both sides, we can differentiate the equation by any order. With (2.4a) and (2.4b) we have

$$MU^{(i+1)}(\tilde{t}) = \frac{d^i}{dt^i} \left(\mathcal{P}_n F(t, U(t)) \right) \Big|_{t=\tilde{t}} = \frac{d^i}{dt^i} \left(F(t, U(t)) \right) \Big|_{t=\tilde{t}}$$

for $\tilde{t} = t_n^-$ and all $i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1$ if $k \geq 2$, as well as for $\tilde{t} = t_{n-1}^+$ and all $i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1$ if $k \geq 3$, respectively. Hence, the conditions (2.2b) and (2.2c) hold. Taking the inner product of (2.5) with an arbitrary $\varphi \in P_{r-k}(I_n, \mathbb{R}^d)$ and integrating over I_n yield together with (2.4c)

$$\int_{I_n} (MU'(t), \varphi(t)) dt = \int_{I_n} (\mathcal{P}_n F(t, U(t)), \varphi(t)) dt = \int_{I_n} (F(t, U(t)), \varphi(t)) dt$$

which is (2.2d) with exact integration.

Note that certain numerically integrated versions of $\mathbf{VTD}_k^r$ can also be written in the form (2.5) if appropriate projections are applied.

2.2 Global formulation

For $s \in \mathbb{N}_0$ we define the space Y_s of vector-valued piecewise polynomials of maximal degree s by

$$Y_s := \left\{ \varphi \in L^2(I, \mathbb{R}^d) : \varphi|_{I_n} \in P_s(I_n, \mathbb{R}^d), n = 1, \dots, N \right\}.$$

Studying the conditions (2.2a), (2.2b), and (2.2c) we see that the solution U of $\mathcal{J}_n\text{-}\mathbf{VTD}_k^r$ is $\lfloor \frac{k-1}{2} \rfloor$ times continuously differentiable on I if F is sufficiently smooth.

Furthermore, the condition (2.2b) for $U \in C^{\lfloor \frac{k-1}{2} \rfloor}(I)$ already implies (2.2c) for $n \geq 2$. Consequently, the method could be reformulated as follows

Find $U \in Y_r \cap C^{\lfloor \frac{k-1}{2} \rfloor}(I, \mathbb{R}^d)$ such that

$$U^{(i)}(t_0^+) = U^{(i)}(t_0^-), \quad \text{if } k \geq 1, i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor, \quad (2.6a)$$

$$MU^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, U(t)) \right) \Big|_{t=t_n^-}, \quad \text{if } k \geq 2, i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1, \quad (2.6b)$$

for all $n = 1, \dots, N$, and

$$\sum_{n=1}^N \left\{ \mathcal{J}_n[(MU' - F(\cdot, U(\cdot)), \varphi)] + \delta_{0,k}(M[U]_{n-1}, \varphi(t_{n-1}^+)) \right\} = 0 \quad \forall \varphi \in Y_{r-k} \quad (2.6c)$$

where $U^{(i)}(t_0^-) = u^{(i)}(t_0)$, $0 \leq i \leq \lfloor \frac{k}{2} \rfloor$, which includes the initial value u_0 in the problem formulation. We agree on defining $u^{(j)}(t_0)$ recursively using the differential equation, i.e.,

$$\begin{aligned} u^{(0)}(t_0) &:= u_0, & Mu^{(2)}(t_0) &:= \partial_t F(t_0, u(t_0)) + \partial_u F(t_0, u(t_0))u^{(1)}(t_0), \\ Mu^{(1)}(t_0) &:= F(t_0, u(t_0)), & Mu^{(j)}(t_0) &:= \frac{d^{j-1}}{dt^{j-1}} F(t, u(t)) \Big|_{t=t_0}, \quad j \geq 3. \end{aligned} \quad (2.7)$$

The term $\frac{d^{j-1}}{dt^{j-1}} F(t, u(t)) \Big|_{t=t_0}$ depends only on $u(t_0), \dots, u^{(j-1)}(t_0)$ and can be calculated using some generalization of Faà di Bruno's formula, see e.g. [13, 25]. If F is affine linear in u , i.e., $F(t, u(t)) = f(t) - A(t)u(t)$, then we simply have

$$Mu^{(j)}(t_0) := \frac{d^{j-1}}{dt^{j-1}} F(t, u(t)) \Big|_{t=t_0} = f^{(j-1)}(t_0) - \sum_{l=0}^{j-1} \binom{j-1}{l} A^{(j-1-l)}(t_0) u^{(l)}(t_0),$$

for $j \geq 1$, by Leibniz' rule for the $(j-1)$ th derivative.

Note that since the test space Y_{r-k} in (2.6c) allows discontinuities at the boundaries of subintervals, the problem can be decoupled by choosing test functions φ supported on a single time interval I_n only. Moreover, exploiting for $k \geq 1$ that $U \in C^{\lfloor \frac{k-1}{2} \rfloor}$ as well as (2.6a) and (2.6b), we also obtain (2.2a) and (2.2c). Therefore, the global problem (2.6) can be converted back into a sequence of local problems (2.2) in time on the different subintervals I_n , $n = 1, \dots, N$.

3 Postprocessing

We present in this section a simple postprocessing that makes considerable use of the properties of the quadrature rule Q_k^r associated to $\mathbf{VTD}_k^r$.

Theorem 3.1 (Postprocessing Q_k^r - $\mathbf{VTD}_k^r \rightsquigarrow Q_k^r$ - $\mathbf{VTD}_{k+2}^{r+1}$) *Let $r, k \in \mathbb{N}_0$ with $0 \leq k \leq r$ and suppose that $U \in Y_r$ solves Q_k^r - $\mathbf{VTD}_k^r$. For every $n = 1, \dots, N$ set*

$$\tilde{U}|_{I_n} = U|_{I_n} + a_n \vartheta_n, \quad \vartheta_n \in P_{r+1}(I_n, \mathbb{R}),$$

where ϑ_n vanishes in all $(r+1)$ quadrature points of Q_k^r and satisfies $\vartheta_n(\lfloor \frac{k}{2} \rfloor + 1)(t_n^-) = 1$ while the vector $a_n \in \mathbb{R}^d$ is defined by

$$a_n = M^{-1} \left(\frac{d^{\lfloor \frac{k}{2} \rfloor}}{dt^{\lfloor \frac{k}{2} \rfloor}} F(t, U(t)) \Big|_{t=t_n^-} - MU(\lfloor \frac{k}{2} \rfloor + 1)(t_n^-) \right). \quad (3.1)$$

Moreover, let $\tilde{U}(t_0^-) = U(t_0^-)$. Then $\tilde{U} \in Y_{r+1}$ solves Q_k^r - $\mathbf{VTD}_{k+2}^{r+1}$.

Proof We have to verify that $\tilde{U}$ satisfies all conditions for $Q_k^r\text{-VTD}_{k+2}^{r+1}$ where Q_k^r is the quadrature rule associated to VTD_k^r which is exact for polynomials up to degree $2r - k$.

First of all we show an identity needed later. The special form of ϑ_n , the exactness of Q_k^r , and integration by parts yield

$$\begin{aligned} Q_k^r[\vartheta_n' \varphi] &= \int_{I_n} \vartheta_n'(t) \varphi(t) \, dt = - \int_{I_n} \vartheta_n(t) \varphi'(t) \, dt + (\vartheta_n \varphi) \Big|_{t_{n-1}^+}^{t_n^-} \\ &= - \underbrace{Q_k^r[\vartheta_n \varphi']}_{=0} - \delta_{0,k} (\vartheta_n \varphi)(t_{n-1}^+) = -\delta_{0,k} (\vartheta_n \varphi)(t_{n-1}^+) \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}). \end{aligned} \quad (3.2)$$

Precisely, we used that both $\vartheta_n' \varphi$ and $\vartheta_n \varphi'$ are polynomials of maximal degree $2r - k$ and that ϑ_n vanishes in all quadrature points, especially in t_n^- and for $k \geq 1$ also in t_{n-1}^+ .

For $k \geq 1$ we have $\vartheta_n(t_{n-1}^+) = \vartheta_n(t_n^-) = 0$. Therefore, the initial condition holds due to $\tilde{U}(t_{n-1}^+) = U(t_{n-1}^+) = U(t_{n-1}^-) = \tilde{U}(t_{n-1}^-)$. For $k = 0$ it is somewhat more complicated to prove $\tilde{U}(t_{n-1}^+) = \tilde{U}(t_{n-1}^-)$, for details see (iii) below. The remaining conditions can be verified as follows. $\square$

- (i) Conditions at t_n^- for $0 \leq i \leq \lfloor \frac{k+2}{2} \rfloor - 2 = \lfloor \frac{k}{2} \rfloor - 1$:

From the definitions of $\tilde{U}$ and U we obtain

$$\begin{aligned} M\tilde{U}^{(i+1)}(t_n^-) &= MU^{(i+1)}(t_n^-) + Ma_n \underbrace{\vartheta_n^{(i+1)}(t_n^-)}_{=0} \\ &= \frac{d^i}{dt^i} F(t, U(t)) \Big|_{t=t_n^-} = \frac{d^i}{dt^i} F(t, \tilde{U}(t)) \Big|_{t=t_n^-} \end{aligned}$$

since the derivatives of U and $\tilde{U}$ in t_n^- coincide up to order $\lfloor \frac{k}{2} \rfloor$ due to the definition of ϑ_n .

- (ii) Condition at t_n^- for $i = \lfloor \frac{k+2}{2} \rfloor - 1 = \lfloor \frac{k}{2} \rfloor$:

Just like above we get, additionally using the definition of a_n ,

$$\begin{aligned} M\tilde{U}^{(\lfloor \frac{k}{2} \rfloor + 1)}(t_n^-) &= MU^{(\lfloor \frac{k}{2} \rfloor + 1)}(t_n^-) + Ma_n \underbrace{\vartheta_n^{(\lfloor \frac{k}{2} \rfloor + 1)}(t_n^-)}_{=1} \\ &= MU^{(\lfloor \frac{k}{2} \rfloor + 1)}(t_n^-) + \frac{d^{\lfloor \frac{k}{2} \rfloor}}{dt^{\lfloor \frac{k}{2} \rfloor}} F(t, U(t)) \Big|_{t=t_n^-} - MU^{(\lfloor \frac{k}{2} \rfloor + 1)}(t_n^-) \\ &= \frac{d^{\lfloor \frac{k}{2} \rfloor}}{dt^{\lfloor \frac{k}{2} \rfloor}} F(t, U(t)) \Big|_{t=t_n^-} = \frac{d^{\lfloor \frac{k}{2} \rfloor}}{dt^{\lfloor \frac{k}{2} \rfloor}} F(t, \tilde{U}(t)) \Big|_{t=t_n^-}. \end{aligned}$$

(iii) Variational condition:

We have to prove that $Q_k^r[(M\tilde{U}', \varphi)] = Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)]$ for all $\varphi \in P_{(r+1)-(k+2)}(I_n, \mathbb{R}^d)$. Actually, we can even test with functions $\varphi \in P_{r-k}(I_n, \mathbb{R}^d)$. We first study the case $k \geq 1$. By the definitions of $\tilde{U}$ and U , the identity (3.2), and the fact that U and $\tilde{U}$ coincide at all quadrature points we have

$$\begin{aligned} Q_k^r[(M\tilde{U}', \varphi)] &= Q_k^r[(MU', \varphi)] + Q_k^r[(Ma_n \vartheta'_n, \varphi)] \\ &= Q_k^r[(F(\cdot, U(\cdot)), \varphi)] + \underbrace{Q_k^r[\vartheta'_n(Ma_n, \varphi)]}_{=0, \text{ since } (Ma_n, \varphi) \in P_{r-k}(I_n, \mathbb{R})} \\ &= Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d). \end{aligned}$$

Now let $k = 0$. The same arguments as for $k \geq 1$ yield for all $\varphi \in P_r(I_n, \mathbb{R}^d)$

$$\begin{aligned} Q_0^r[(M\tilde{U}', \varphi)] &= Q_0^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] - (M[U]_{n-1}, \varphi(t_{n-1}^+)) \\ &\quad - \vartheta_n(t_{n-1}^+)(Ma_n, \varphi(t_{n-1}^+)). \end{aligned}$$

We study the last two terms. Using the definitions of the jump $[U]_{n-1}$ and of $\tilde{U}$, we find

$$[U]_{n-1} + a_n \vartheta_n(t_{n-1}^+) = \tilde{U}(t_{n-1}^+) - U(t_{n-1}^-) = [\tilde{U}]_{n-1} \quad (3.3)$$

where we also exploited that $\vartheta_{n-1}(t_{n-1}^-) = 0$. Hence, we have

$$Q_0^r[(M\tilde{U}', \varphi)] + (M[\tilde{U}]_{n-1}, \varphi(t_{n-1}^+)) = Q_0^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] \quad \forall \varphi \in P_r(I_n, \mathbb{R}^d). \quad (3.4)$$

Choosing the special test functions $\varphi_j \in P_r(I_n, \mathbb{R}^d)$, $j = 1, \dots, d$, that vanish in the r inner quadrature points of Q_0^r and satisfy $\varphi_j(t_{n-1}^+) = e_j$ as well as having in mind (ii), we find $[\tilde{U}]_{n-1} = 0$ component by component. Thereby, at once we have proven the initial condition and verified the needed variational condition since now also the jump term in (3.4) can be dropped.

(iv) Conditions at t_{n-1}^+ for $0 \leq i \leq \lfloor \frac{k+2-1}{2} \rfloor - 2 = \lfloor \frac{k-1}{2} \rfloor - 1$:

With an argumentation similar to that in (i) we gain

$$\begin{aligned} M\tilde{U}^{(i+1)}(t_{n-1}^+) &= MU^{(i+1)}(t_{n-1}^+) + \underbrace{Ma_n \vartheta_n^{(i+1)}(t_{n-1}^+)}_{=0} \\ &= \frac{d^i}{dt^i} F(t, U(t)) \Big|_{t=t_{n-1}^+} = \frac{d^i}{dt^i} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+}. \end{aligned}$$

- (v) Condition at t_{n-1}^+ for $i = \lfloor \frac{k+2-1}{2} \rfloor - 1 = \lfloor \frac{k-1}{2} \rfloor$ if $k \geq 1$:

It remains to prove that

$$M\tilde{U}(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^+) = \frac{d\lfloor \frac{k-1}{2} \rfloor}{dt\lfloor \frac{k-1}{2} \rfloor} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+}.$$

We use the variational condition for $\tilde{U}$ with specially chosen test functions $\varphi_j \in P_{r-k}(I_n, \mathbb{R}^d)$, $j = 1, \dots, d$, that vanish at all inner quadrature points of Q_k^r , i.e.,

$$\varphi_j(t_{n,i}) = 0, \quad i = 1, \dots, r-k, \quad \text{and satisfy} \quad \varphi_j(t_{n-1}^+) = e_j.$$

As shown in (iii) we have

$$Q_k^r[(M\tilde{U}', \varphi_j)] = Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi_j)], \quad j = 1, \dots, d,$$

since $k \geq 1$. The special choices of φ_j , the definition of the quadrature rule, and the already known identities from (i), (ii), and (iv) yield after a short calculation using Leibniz' rule for the i th derivative that

$$\begin{aligned} Q_k^r[(M\tilde{U}', \varphi_j)] &= Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi_j)], \quad j = 1, \dots, d, \\ &\Leftrightarrow w_{\lfloor \frac{k-1}{2} \rfloor}^L M\tilde{U}(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^+) \cdot \underbrace{\varphi_j(t_{n-1}^+)}_{=e_j} \\ &= w_{\lfloor \frac{k-1}{2} \rfloor}^L \frac{d\lfloor \frac{k-1}{2} \rfloor}{dt\lfloor \frac{k-1}{2} \rfloor} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+} \cdot \underbrace{\varphi_j(t_{n-1}^+)}_{=e_j}, \quad j = 1, \dots, d, \\ &\Leftrightarrow M\tilde{U}(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^+) = \frac{d\lfloor \frac{k-1}{2} \rfloor}{dt\lfloor \frac{k-1}{2} \rfloor} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+}. \end{aligned}$$

Note that we also used that $w_{\lfloor \frac{k-1}{2} \rfloor}^L \neq 0$.

Collecting the above arguments, we see that $\tilde{U}$ solves $Q_k^r\text{-VTD}_{k+2}^{r+1}$. $\square$

From the definition (3.1), it seems that a linear system with the mass matrix M has to be solved in every time step in order to obtain the correction vector a_n . However, the computational costs for calculating a_n can be reduced significantly as we show now.

Proposition 3.2 *The correction vectors $a_n \in \mathbb{R}^d$ defined in (3.1) for the postprocessing presented in Theorem 3.1 can alternatively be calculated by*

$$a_n = \frac{-1}{\vartheta_n(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^+)} \left(U(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^+) - \tilde{U}(\lfloor \frac{k-1}{2} \rfloor + 1)(t_{n-1}^-) \right) \quad \text{for } n > 1,$$

and

$$a_1 = \frac{-1}{\vartheta_1\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_0^+)} \left(U\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_0^+) - u\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_0) \right)$$

where $u\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_0)$ is defined in (2.7).

Proof For $k = 0$, we get from (3.3) combined with $[\tilde{U}]_{n-1} = 0$, which was shown just below (3.4), that $a_n = \frac{-1}{\vartheta_n(t_{n-1}^+)}[U]_{n-1} = \frac{-1}{\vartheta_n(t_{n-1}^+)}(U(t_{n-1}^+) - \tilde{U}(t_{n-1}^-))$. Taking $\tilde{U}(t_0^-) = U(t_0^-) = u(t_0) = u_0$ into account, we are done in this case.

Otherwise, for $k \geq 1$, using the definition of the postprocessing and (v) of the proof of Theorem 3.1, we obtain that

$$\begin{aligned} MU\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^+) + Ma_n\vartheta_n\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^+) &= M\tilde{U}\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^+) \\ &= \frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, \tilde{U}(t))\Big|_{t=t_{n-1}^+}. \end{aligned}$$

Furthermore, we have $\vartheta_n^{(i)}(t_{n-1}^+) = 0$ for $i = 0, \dots, \left\lfloor \frac{k-1}{2} \right\rfloor$ and therefore

$$\frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, \tilde{U}(t))\Big|_{t=t_{n-1}^+} = \frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, U(t))\Big|_{t=t_{n-1}^+}. \quad (3.5)$$

Since F is sufficiently smooth and U is $\left\lfloor \frac{k-1}{2} \right\rfloor$ times continuously differentiable we get for $n > 1$

$$\begin{aligned} \frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, U(t))\Big|_{t=t_{n-1}^+} &= \frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, U(t))\Big|_{t=t_{n-1}^-} \\ &= \frac{d\left\lfloor \frac{k-1}{2} \right\rfloor}{dt\left\lfloor \frac{k-1}{2} \right\rfloor} F(t, \tilde{U}(t))\Big|_{t=t_{n-1}^-} = M\tilde{U}\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^-) \end{aligned}$$

where also $\vartheta_{n-1}^{(i)}(t_{n-1}^-) = 0$ for $i = 0, \dots, \left\lfloor \frac{k}{2} \right\rfloor$ and (i) or (ii) of the proof of Theorem 3.1 were used. Altogether exploiting that M is regular an easy manipulation of the identities yields

$$a_n = \frac{-1}{\vartheta_n\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^+)} \left(U\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^+) - \tilde{U}\left(\left\lfloor \frac{k-1}{2} \right\rfloor + 1\right)(t_{n-1}^-) \right), \quad n > 1.$$

A similar formula can also be established for $n = 1$. Since U satisfies (2.2c) and $U(t_0) := u_0$, we obviously obtain, recalling the definition (2.7) of $u^{(i)}(t_0)$, that

$$U^{(i)}(t_0^+) = u^{(i)}(t_0) \quad \text{for } i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor.$$

Therefore, we have in (3.5) for $n = 1$

$$\begin{aligned} \frac{d \lfloor \frac{k-1}{2} \rfloor}{dt \lfloor \frac{k-1}{2} \rfloor} F(t, \tilde{U}(t)) \Big|_{t=t_0^+} &= \frac{d \lfloor \frac{k-1}{2} \rfloor}{dt \lfloor \frac{k-1}{2} \rfloor} F(t, U(t)) \Big|_{t=t_0^+} \\ &= \frac{d \lfloor \frac{k-1}{2} \rfloor}{dt \lfloor \frac{k-1}{2} \rfloor} F(t, u(t)) \Big|_{t=t_0^+} = Mu \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0). \end{aligned}$$

This results in

$$a_1 = \frac{-1}{\vartheta_1 \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0^+)} \left(U \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0^+) - u \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0) \right).$$

Hence, the alternative calculation provides the same correction vector. $\square$

Note that a_n can be calculated in this way without solving a system of linear equations. From the structure of a_n we see that the postprocessing can be interpreted as a correction of the jump in the lowest order derivative of the discrete solution that is not continuous by construction.

Since the division by $\vartheta_n \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_{n-1}^+)$ changes the normalization of ϑ_n only, we conclude the following.

Corollary 3.3 (Alternative postprocessing $Q_k^r\text{-VTD}_k^r \rightsquigarrow Q_k^r\text{-VTD}_{k+2}^{r+1}$) *Let $r, k \in \mathbb{N}_0$ with $0 \leq k \leq r$ and suppose that $U \in Y_r$ solves $Q_k^r\text{-VTD}_k^r$. For every $n = 1, \dots, N$ set*

$$\tilde{U}|_{I_n} = U|_{I_n} - \tilde{a}_n \tilde{\vartheta}_n, \quad \tilde{\vartheta}_n \in P_{r+1}(I_n, \mathbb{R}),$$

where $\tilde{\vartheta}_n(t) = \vartheta_n(t) / \vartheta_n \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_{n-1}^+)$ with ϑ_n from Theorem 3.1, i.e., $\tilde{\vartheta}_n$ vanishes in all $(r+1)$ quadrature points of Q_k^r and satisfies $\tilde{\vartheta}_n \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_{n-1}^+) = 1$. The vector $\tilde{a}_n \in \mathbb{R}^d$ is defined by

$$\tilde{a}_n := \begin{cases} U \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0^+) - u \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_0), & n = 1, \\ U \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_{n-1}^+) - \tilde{U} \left(\lfloor \frac{k-1}{2} \rfloor + 1 \right) (t_{n-1}^-), & n > 1, \end{cases}$$

where $u(\lfloor \frac{k-1}{2} \rfloor + 1)_{(t_0)}$ is given by (2.7). Moreover, let $\tilde{U}(t_0^-) = U(t_0^-)$. Then $\tilde{U} \in Y_{r+1}$ solves $Q_k^r\text{-VTD}_{k+2}^{r+1}$.

A direct proof for the alternative postprocessing is given in [9, App. A].

4 Connections between numerically integrated variational time discretization methods and collocation methods

In this section we prove that the (local) solution of $Q_k^r\text{-VTD}_l^{r+1}$ with $1 \leq l \leq k+2$ (which obviously includes $Q_k^r\text{-VTD}_{k+2}^{r+1}$) can be characterized as the solution of the (local) collocation problem with respect to the quadrature points of Q_k^r , i.e.,

Find $\tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ with given $\tilde{U}(t_{n-1}) \in \mathbb{R}^d$ such that

$$M\tilde{U}^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_n^-}, \quad i = 0, \dots, \lfloor \frac{k}{2} \rfloor, \quad (4.1a)$$

$$M\tilde{U}^{(i+1)}(t_{n-1}^+) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_{n-1}^+}, \quad \text{if } k \geq 1, i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor, \quad (4.1b)$$

$$M\tilde{U}'(t_{n,i}) = F(t_{n,i}, \tilde{U}(t_{n,i})), \quad i = 1, \dots, r-k, \quad (4.1c)$$

where $\tilde{U}(t_0) = u_0$. Here, $t_{n,i}$ are the zeros of the $(r-k)$ th Jacobi-polynomial with respect to the weight $(1+\hat{t})^{\lfloor \frac{k-1}{2} \rfloor + 1} (1-\hat{t})^{\lfloor \frac{k}{2} \rfloor + 1}$ transformed to the interval $[t_{n-1}, t_n]$, see also (2.3).

Methods similar to (4.1) are known as collocation methods with multiple nodes, see e.g. [20, p. 275]. Unfortunately, existing results in the literature often neglect to study the unique solvability or conditions on F are not explicitly given. However, the connections mentioned above directly imply that all these methods are equivalent as we prove now.

Theorem 4.1 (Equivalence to collocation methods) *Let $r, k, l \in \mathbb{N}_0$, $0 \leq k \leq r$, and $1 \leq l \leq k+2$. Then $\tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ solves $Q_k^r\text{-VTD}_l^{r+1}$ if and only if $\tilde{U}$ solves the collocation method (4.1) with respect to the quadrature points of Q_k^r .*

Proof For clarity and convenience, we recall the conditions of the $Q_k^r\text{-VTD}_l^{r+1}$ method with $0 \leq k \leq r$ and $1 \leq l \leq k+2$.

Given $\tilde{U}(t_{n-1}^-)$, find $\tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ such that

$$\tilde{U}(t_{n-1}^+) = \tilde{U}(t_{n-1}^-),$$

$$M\tilde{U}^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_n^-}, \quad \text{if } l \geq 2, i = 0, \dots, \lfloor \frac{l}{2} \rfloor - 1, \quad (4.2a)$$

$$M\tilde{U}^{(i+1)}(t_{n-1}^+) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_{n-1}^+}, \quad \text{if } l \geq 3, i = 0, \dots, \lfloor \frac{l-1}{2} \rfloor - 1, \quad (4.2b)$$

$$Q_k^r[(M\tilde{U}', \varphi)] = Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] \quad \forall \varphi \in P_{r+1-l}(I_n, \mathbb{R}^d). \quad (4.2c)$$

First of all, assume that $\tilde{U}$ solves (4.1). Then because of (4.1a) and (4.1b) obviously $\tilde{U}$ satisfies the conditions (4.2a) and (4.2b) since $l \leq k + 2$. In order to gain a better understanding of the numerically integrated variational condition, we have a look at its detailed definition. For $\varphi \in C^{\lfloor \frac{k}{2} \rfloor}(\bar{I}_n, \mathbb{R}^d)$ we have by definition of the quadrature rule

$$\begin{aligned} Q_k^r[(M\tilde{U}' - F(\cdot, \tilde{U}(\cdot)), \varphi)] &= \frac{\tau_n}{2} \left[\sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} w_i^L \left(\frac{\tau_n}{2} \right)^i \frac{d^i}{dt^i} (M\tilde{U}'(t) - F(t, \tilde{U}(t)), \varphi(t)) \Big|_{t=t_{n-1}^+} \right. \\ &\quad + \sum_{i=1}^{r-k} w_i^I (M\tilde{U}'(t_{n,i}) - F(t_{n,i}, \tilde{U}(t_{n,i})), \varphi(t_{n,i})) \\ &\quad \left. + \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} w_i^R \left(\frac{\tau_n}{2} \right)^i \frac{d^i}{dt^i} (M\tilde{U}'(t) - F(t, \tilde{U}(t)), \varphi(t)) \Big|_{t=t_n^-} \right]. \end{aligned}$$

Applying Leibniz' rule for the i th derivative, the right-hand side above can be rewritten as

$$\begin{aligned} &\frac{\tau_n}{2} \left[\sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} w_i^L \left(\frac{\tau_n}{2} \right)^i \sum_{j=0}^i \binom{i}{j} (M\tilde{U}^{(j+1)}(t_{n-1}^+) - \frac{d^j}{dt^j} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+}, \varphi^{(i-j)}(t_{n-1}^+)) \right. \\ &\quad + \sum_{i=1}^{r-k} w_i^I (M\tilde{U}'(t_{n,i}) - F(t_{n,i}, \tilde{U}(t_{n,i})), \varphi(t_{n,i})) \\ &\quad \left. + \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} w_i^R \left(\frac{\tau_n}{2} \right)^i \sum_{j=0}^i \binom{i}{j} (M\tilde{U}^{(j+1)}(t_n^-) - \frac{d^j}{dt^j} F(t, \tilde{U}(t)) \Big|_{t=t_n^-}, \varphi^{(i-j)}(t_n^-)) \right]. \end{aligned}$$

Using the collocation conditions (4.1b), (4.1c), and (4.1a), we see that all three sums equal to 0. Hence, we obtain

$$Q_k^r[(M\tilde{U}' - F(\cdot, \tilde{U}(\cdot)), \varphi)] = 0 \quad \forall \varphi \in C^{\lfloor \frac{k}{2} \rfloor}(\bar{I}_n, \mathbb{R}^d) \quad (4.3)$$

which immediately gives (4.2c).

Now, we study the other direction and assume that $\tilde{U}$ solves (4.2). In order to verify (4.1a), for example, we need to prove that (4.2a) also holds for $i = \lfloor \frac{l}{2} \rfloor, \dots, \lfloor \frac{k}{2} \rfloor$.

For this purpose (in the case that $\lfloor \frac{l}{2} \rfloor \leq \lfloor \frac{k}{2} \rfloor$) we use the special test function

$$\psi = c(t - t_{n-1})^{\lfloor \frac{k-1}{2} \rfloor - \lfloor \frac{l-1}{2} \rfloor} + 1(t - t_n)^{\lfloor \frac{k}{2} \rfloor - \lfloor \frac{l}{2} \rfloor} \prod_{i=1}^{r-k} (t - t_{n,i})$$

$$\in P_{r+1-l}(I_n, \mathbb{R}^d)$$

with an arbitrary vector $c \in \mathbb{R}^d$ where $t_{n,i}, i = 1, \dots, r-k$, denote the inner quadrature points of Q_k^r . Then (4.2c), the special construction of ψ , (4.2a), and (4.2b) yield

$$0 = Q_k^r \left[(M\tilde{U}' - F(\cdot, \tilde{U}(\cdot)), \psi) \right]$$

$$= \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} w_i^L \left(\frac{\tau_n}{2} \right)^{i+1} \sum_{j=\lfloor \frac{l-1}{2} \rfloor}^i \binom{i}{j} \left(M\tilde{U}^{(j+1)}(t_{n-1}^+) - \frac{d^j}{dt^j} F(t, \tilde{U}(t)) \Big|_{t=t_{n-1}^+}, \psi^{(i-j)}(t_{n-1}^+) \right)$$

$$+ \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} w_i^R \left(\frac{\tau_n}{2} \right)^{i+1} \sum_{j=\lfloor \frac{l}{2} \rfloor}^i \binom{i}{j} \left(M\tilde{U}^{(j+1)}(t_n^-) - \frac{d^j}{dt^j} F(t, \tilde{U}(t)) \Big|_{t=t_n^-}, \psi^{(i-j)}(t_n^-) \right).$$

Furthermore, we have that $\psi^{(i)}(t_{n-1}^+) = 0$ for $i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - \lfloor \frac{l-1}{2} \rfloor$ and $\psi^{(i)}(t_n^-) = 0$ for $i = 0, \dots, \lfloor \frac{k}{2} \rfloor - \lfloor \frac{l}{2} \rfloor - 1$. Thus, the above identity simplifies to

$$0 = w_i^R \left(\frac{\tau_n}{2} \right)^{i+1} \binom{i}{j} \left(M\tilde{U}^{(j+1)}(t_n^-) - \frac{d^j}{dt^j} F(t, \tilde{U}(t)) \Big|_{t=t_n^-}, \psi^{(i-j)}(t_n^-) \right)$$

with $i = \lfloor \frac{k}{2} \rfloor$ and $j = \lfloor \frac{l}{2} \rfloor$. Since $w_i^R \neq 0$, $\psi^{(\lfloor \frac{k}{2} \rfloor - \lfloor \frac{l}{2} \rfloor)}(t_n^-) \neq 0$, and the vector $c \in \mathbb{R}^d$ can be chosen arbitrarily, it follows

$$M\tilde{U}^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_n^-} \quad \text{for } i = \lfloor \frac{l}{2} \rfloor.$$

Using the test functions

$$\psi = c(t - t_{n-1})^{\lfloor \frac{k-1}{2} \rfloor - \lfloor \frac{l-1}{2} \rfloor} + 1(t - t_n)^{\lfloor \frac{k}{2} \rfloor - \lfloor \frac{l}{2} \rfloor} - j \prod_{i=1}^{r-k} (t - t_{n,i})$$

$$\in P_{r+1-l-j}(I_n, \mathbb{R}^d)$$

with an arbitrary vector $c \in \mathbb{R}^d$ and $j = 1, \dots, \lfloor \frac{k}{2} \rfloor - \lfloor \frac{l}{2} \rfloor$ we iteratively also prove

$$M\tilde{U}^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} \left(F(t, \tilde{U}(t)) \right) \Big|_{t=t_n^-} \quad \text{for } i = \lfloor \frac{l}{2} \rfloor + 1, \dots, \lfloor \frac{k}{2} \rfloor.$$

A similar argument can also be used for the missing point conditions at t_{n-1}^+ . For the above implications we needed that the weights of Q_k^r do not vanish which is known from [23].

It remains to verify (4.1c). Since we already know that $\tilde{U}$ satisfies the collocation conditions (4.1a) and (4.1b), the variational condition (4.2c) reduces to

$$\sum_{i=1}^{r-k} w_i^I (M\tilde{U}'(t_{n,i}), \varphi(t_{n,i})) = \sum_{i=1}^{r-k} w_i^I (F(t_{n,i}, \tilde{U}(t_{n,i})), \varphi(t_{n,i})) \quad \forall \varphi \in P_{r+1-l}(I_n, \mathbb{R}^d). \quad (4.4)$$

Recalling that $w_j^I > 0$ for all $1 \leq j \leq r-k$ and choosing in (4.4) test functions of the form

$$\varphi_j(t) = c \prod_{\substack{i=1 \\ i \neq j}}^{r-k} (t - t_{n,i}) \in P_{r-k-1}(I_n, \mathbb{R}^d) \subseteq P_{r+1-l}(I_n, \mathbb{R}^d)$$

with an arbitrary vector $c \in \mathbb{R}^d$, we get the collocation condition (4.1c) in $t_{n,j}$.

Hence, the stated equivalence has been proven. $\square$

Summarizing, we have shown that a solution of $Q_k^r\text{-VTD}_{k+2}^{r+1}$ also solves $Q_k^r\text{-VTD}_l^{r+1}$ with $1 \leq l \leq k+2$ as well as a collocation with respect to the quadrature points of Q_k^r and vice versa. Shortly, we have

$$\begin{aligned} Q_k^r\text{-VTD}_{k+2}^{r+1} &\cong Q_k^r\text{-VTD}_l^{r+1} \quad \text{with } 1 \leq l \leq k+2 \\ &\cong \text{collocation with respect to the quadrature points of } Q_k^r. \end{aligned}$$

Remark 4.2 Independent of the above findings, the connection between collocation methods and (postprocessed) numerically integrated discontinuous Galerkin methods (using the right-sided Gauss–Radau quadrature), i.e., Theorem 4.1 for $k = 0 \leq r$ and $l = 2$, was already observed in [29]. Moreover, connections between collocation methods and the numerically integrated continuous Galerkin–Petrov methods (using interpolatory quadrature formulas with as many quadrature points as number of independent variational conditions) were shown in [21, 22].

4.1 Error estimates for collocation methods

The method defined in (4.1) is a collocation method with multiple nodes as considered for example in [20, p. 275]. Thus, the usual error analysis for collocation methods also applies here, provided that F and u are sufficiently smooth. According to [20, p. 276], we have the following error estimate.

Proposition 4.3 *The collocation method (4.1) possesses the same order $2r - k + 1$ as the underlying quadrature formula Q_k^r . Hence, it holds*

$$\max_{1 \leq n \leq N} \|(u - \tilde{U})(t_n^-)\| \leq C(F, u) \tau^{2r-k+1}$$

where $\tilde{U}$ and u denote the solutions of the collocation method (4.1) and the initial value problem (2.1), respectively.

Moreover, global error estimates can be shown by adapting techniques presented in [21, Theorem 2]. Together with [20, p. 276, pp. 212–214], we obtain the following.

Proposition 4.4 *Let $\tilde{U}$ denote the solution of the collocation method (4.1) and u the exact solution of (2.1). Then we have*

$$\sup_{t \in I} \|(u - \tilde{U})(t)\| \leq C(F, u) \tau^{\min\{2r-k+1, (r+1)+1\}}$$

and

$$\sup_{t \in I} \|(u - \tilde{U})^{(l)}(t)\| \leq C(F, u) \tau^{(r+1)+1-l}, \quad 1 \leq l \leq r+1,$$

where the derivatives are understood in an interval-wise sense.

The term $2r - k + 1$ inside the minimum is due to the fact that the convergence order of these collocation methods is limited by the accuracy of the underlying quadrature formula Q_k^r that is exactly $2r - k + 1$. Note that the limitation is active for $r = k$ only.

4.2 Reversed postprocessing

In Sect. 3 we studied a postprocessing for $Q_k^r\text{-VTD}_k^r$. We have seen that starting from a solution U of $Q_k^r\text{-VTD}_k^r$ we can easily construct a solution $\tilde{U}$ of $Q_k^r\text{-VTD}_{k+2}^{r+1}$. This already implies uniqueness of U provided that solutions of $Q_k^r\text{-VTD}_{k+2}^{r+1}$ or (4.1) are unique. Indeed, if U^1 and U^2 solve $Q_k^r\text{-VTD}_k^r$ and their postprocessed solutions are identical, then by construction of the postprocessing U^1 and U^2 coincide in the $(r+1)$ quadrature points of Q_k^r . Thus, since both are polynomials of degree r , it follows $U^1 \equiv U^2$.

We now ask whether or not this postprocessing step can be reversed for $0 \leq k \leq r$. Then the solvability of $Q_k^r\text{-VTD}_k^r$ follows provided that $Q_k^r\text{-VTD}_{k+2}^{r+1}$ or (4.1) has a solution.

Proposition 4.5 (Reversed postprocessing) *Let $r, k \in \mathbb{N}_0$, $0 \leq k \leq r$, and suppose that $\tilde{U} \in Y_{r+1}$ solves $Q_k^r\text{-VTD}_{k+2}^{r+1}$. Then $\mathcal{I}_k^r \tilde{U} \in Y_r$ solves $Q_k^r\text{-VTD}_k^r$.*

Proof Let $\tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ solve $Q_k^r\text{-VTD}_{k+2}^{r+1}$. We shall prove that then $\mathcal{I}_k^r \tilde{U} \in P_r(I_n, \mathbb{R}^d)$ is a solution of $Q_k^r\text{-VTD}_k^r$. Since $\mathcal{I}_k^r$ conserves the derivatives up to order

$\lfloor \frac{k}{2} \rfloor$ at t_n^- and up to order $\lfloor \frac{k-1}{2} \rfloor$ at t_{n-1}^+ , respectively, we have

$$M(\mathcal{I}_k^r \tilde{U})^{(i+1)}(t_n^-) = M\tilde{U}^{(i+1)}(t_n^-) = \frac{d^i}{dt^i} F(t, \tilde{U}(t)) \Big|_{t=t_n^-} = \frac{d^i}{dt^i} F(t, \mathcal{I}_k^r \tilde{U}(t)) \Big|_{t=t_n^-},$$

if $k \geq 2$, $i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1$,

and analogously

$$M(\mathcal{I}_k^r \tilde{U})^{(i+1)}(t_{n-1}^+) = \frac{d^i}{dt^i} F(t, \mathcal{I}_k^r \tilde{U}(t)) \Big|_{t=t_{n-1}^+}, \quad \text{if } k \geq 3, i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1.$$

These are in fact (2.2b) and (2.2c).

It remains to prove that $\mathcal{I}_k^r \tilde{U}$ also satisfies the variational condition (2.2d) with $\mathcal{J}_n = Q_k^r$. According to (4.3) we have for $\tilde{U}$ that

$$Q_k^r[(M\tilde{U}', \varphi)] = Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d).$$

Note that originally from the definition of Q_k^r -**VTD** $_{k+2}^{r+1}$ the variational condition is postulated for $\varphi \in P_{r-k-1}(I_n, \mathbb{R}^d)$ only. Since $\mathcal{I}_k^r \tilde{U} - \tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ vanishes in the quadrature points of Q_k^r it holds

$$\mathcal{I}_k^r \tilde{U} - \tilde{U} = c_n \tilde{\vartheta}_n \quad \text{with} \quad c_n = (\mathcal{I}_k^r \tilde{U} - \tilde{U}) \Big|_{\lfloor \frac{k-1}{2} \rfloor + 1} (t_{n-1}^+) \in \mathbb{R}^d$$

and $\tilde{\vartheta}_n \in P_{r+1}(I_n, \mathbb{R})$ as defined in Corollary 3.3. Hence, using (3.2) we conclude

$$\begin{aligned} Q_k^r[(M(\mathcal{I}_k^r \tilde{U})', \varphi)] &= Q_k^r[(M\tilde{U}', \varphi)] + Q_k^r[(Mc_n \tilde{\vartheta}_n', \varphi)] \\ &= Q_k^r[(F(\cdot, \tilde{U}(\cdot)), \varphi)] + Q_k^r[\tilde{\vartheta}_n'(Mc_n, \varphi)] \\ &= Q_k^r[(F(\cdot, \mathcal{I}_k^r \tilde{U}(\cdot)), \varphi)] - \delta_{0,k} \tilde{\vartheta}_n(t_{n-1}^+) (Mc_n, \varphi(t_{n-1}^+)) \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d) \end{aligned}$$

which, because of

$$\tilde{\vartheta}_n(t_{n-1}^+) Mc_n = M(\mathcal{I}_0^r \tilde{U} - \tilde{U})(t_{n-1}^+) = M(\mathcal{I}_0^r \tilde{U}(t_{n-1}^+) - \tilde{U}(t_{n-1}^-)) = M[\mathcal{I}_0^r \tilde{U}]_{n-1}$$

for $k = 0$, completes the argument. $\square$

4.3 Consequences for existence, uniqueness, and error estimates

The connections between numerically integrated variational time discretization methods and collocation methods with multiple nodes observed in the previous subsections can now be used to obtain results on the existence and uniqueness of solutions of Q_k^r -**VTD** $_k^r$ as well as give rise to global error estimates and superconvergence estimates in the time mesh points.

Corollary 4.6 (Existence and uniqueness) *If there is a solution $\tilde{U} \in P_{r+1}(I_n, \mathbb{R}^d)$ of the collocation method with multiple nodes defined by (4.1) then $U = \mathcal{I}_k^r \tilde{U} \in P_r(I_n, \mathbb{R}^d)$ solves Q_k^r -VTD $_k^r$. Furthermore, if $\tilde{U}$ is uniquely defined as solution of (4.1) then so is U as solution of Q_k^r -VTD $_k^r$.*

Corollary 4.7 (Global error estimates) *Let the error estimates of Proposition 4.4 hold for the solution $\tilde{U}$ of (4.1) and the exact solution u of (2.1). Then we have for the solution U of Q_k^r -VTD $_k^r$ that*

$$\sup_{t \in I} \|(u - U)(t)\| \leq C(F, u) \tau^{r+1}$$

and

$$\sup_{t \in I} \|(u - U)^{(l)}(t)\| \leq C(F, u) \tau^{r+1-l}, \quad 1 \leq l \leq r,$$

where the derivatives are understood in an interval-wise sense.

Proof Since $U = \mathcal{I}_k^r \tilde{U}$, Proposition 4.4 and standard error estimates yield

$$\begin{aligned} \sup_{t \in I} \|(u - U)(t)\| &\leq \sup_{t \in I} \|(u - \tilde{U})(t)\| + \sup_{t \in I} \|(\tilde{U} - \mathcal{I}_k^r \tilde{U})(t)\| \\ &\leq C(F, u) \tau^{\min\{2r-k+1, (r+1)+1\}} + C \tau^{r+1} \sup_{t \in I} \|\tilde{U}^{(r+1)}(t)\|. \end{aligned}$$

In order to estimate $\sup_{t \in I} \|\tilde{U}^{(r+1)}(t)\|$ we again use Proposition 4.4 as follows

$$\sup_{t \in I} \|\tilde{U}^{(r+1)}(t)\| \leq \sup_{t \in I} \|(\tilde{U} - u)^{(r+1)}(t)\| + \sup_{t \in I} \|u^{(r+1)}(t)\| \leq C(F, u) \tau + C(u).$$

Because of $0 \leq k \leq r$ this completes the proof of the first statement. The second estimate can be proven similarly. $\square$

Corollary 4.8 (Superconvergence in time mesh points) *Let the error estimates of Proposition 4.3 hold for the solution $\tilde{U}$ of (4.1) and the exact solution u of (2.1). Then we have*

$$\max_{1 \leq n \leq N} \|(u - U)(t_n^-)\| = \max_{1 \leq n \leq N} \|(u - \tilde{U})(t_n^-)\| \leq C(F, u) \tau^{2r-k+1}$$

for the solution U of Q_k^r -VTD $_k^r$.

Proof Recall that $U = \mathcal{I}_k^r \tilde{U}$ and that $\mathcal{I}_k^r$ especially preserves the function value in t_n^- . Hence, $U(t_n^-) = \mathcal{I}_k^r \tilde{U}(t_n^-) = \tilde{U}(t_n^-)$ and the estimate follows immediately from Proposition 4.3. $\square$

Remark 4.9 (Superconvergence in quadrature points) We obtain under the assumptions of Corollary 4.7 also a (lower order) superconvergence estimate for the solution U of $Q_k^r\text{-VTD}_k^r$ in the quadrature points of Q_k^r if $0 \leq k < r$. In fact, let $t_{n,i}, i = 1, \dots, r-k$, denote the local quadrature points of Q_k^r in the interior of I_n . Then, we have

$$\|(u - U)(t_{n,i})\| = \|(u - \tilde{U})(t_{n,i})\| \leq C(F, u)\tau^{(r+1)+1}.$$

In addition, we obtain

$$\|(u - U)^{(l)}(t_n^-)\| = \|(u - \tilde{U})^{(l)}(t_n^-)\| \leq C(F, u)\tau^{(r+1)+1-l}, \quad 0 \leq l \leq \lfloor \frac{k}{2} \rfloor,$$

and

$$\|(u - U)^{(l)}(t_{n-1}^+)\| = \|(u - \tilde{U})^{(l)}(t_{n-1}^+)\| \leq C(F, u)\tau^{(r+1)+1-l}, \quad 0 \leq l \leq \lfloor \frac{k-1}{2} \rfloor,$$

provided $k \geq 1$.

These superconvergence estimates especially imply

$$\begin{aligned} \left(\sum_{n=1}^N Q_k^r [\|u - U\|^2]_{I_n} \right)^{1/2} &= \left(\sum_{n=1}^N Q_k^r [\|u - \tilde{U}\|^2]_{I_n} \right)^{1/2} \\ &\leq (t_N - t_0)^{1/2} C(F, u)\tau^{(r+1)+1} \end{aligned}$$

which compared to

$$\left(\sum_{n=1}^N \int_{I_n} \|(u - U)(x)\|^2 dx \right)^{1/2} \leq (t_N - t_0)^{1/2} C(F, u)\tau^{r+1}$$

gives an extra order of convergence.

Remark 4.10 (Superconvergence of derivative(s) in time mesh points) From the point conditions (2.2b) and the bound of Corollary 4.8 we also gain superconvergence estimates up to the $\lfloor \frac{k}{2} \rfloor$ th derivative of the solution U of $Q_k^r\text{-VTD}_k^r$ in t_n^- , provided that F satisfies certain Lipschitz-conditions. Indeed, supposing that

$$\left\| \frac{d^i}{dt^i} (F(t, v(t)) - F(t, w(t))) \Big|_{t=t_n^-} \right\| \leq C \sum_{j=0}^i \|(v - w)^{(j)}(t_n^-)\|,$$

holds for all $0 \leq i \leq \lfloor \frac{k}{2} \rfloor - 1$, we obtain

$$\begin{aligned} \|(u - U)^{(i+1)}(t_n^-)\| &= \left\| \frac{d^i}{dt^i} (F(t, u(t)) - F(t, U(t))) \Big|_{t=t_n^-} \right\| \leq C \sum_{j=0}^i \|(u - U)^{(j)}(t_n^-)\| \\ &\leq C(F, u)\tau^{2r-k+1} \end{aligned}$$

by iteration over $i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1$.

Remark 4.11 In order to assess the considered methods, we look at their costs, measured in the size of the local systems to be solved, and their benefit in terms of reachable convergence orders. From the $r+1$ vector coefficients needed to describe the local solution $U|_{I_n}$ of Q_k^r -VTD $_k^r$, already $\lfloor \frac{k-1}{2} \rfloor + 1 = \lfloor \frac{k+1}{2} \rfloor$ coefficients are fixed by the initial conditions or data inherited from the previous subinterval I_{n-1} since $U \in C[\frac{k-1}{2}]$. Hence, the local system size is $r - \lfloor \frac{k-1}{2} \rfloor$. The extreme case $k = r$ leads to a system size of $\lfloor \frac{r}{2} \rfloor + 1$ that is roughly half of the size $r+1$ obtained for $k = 0$.

Although the system size decreases with increasing k , the global supremum error converges always with order $r+1$, see Corollary 4.7. This suggests to choose large values for k , also cf. [10]. However, the larger k the more derivatives of the function F have to be provided. Depending on F , this may result in more complex systems on I_n that might be more difficult to solve. The suggestion how to choose k changes if one is interested in the value at the time mesh points only. Given a required superconvergence order, the size of the local systems is independent of k . Thus, small values for k are preferable since no derivatives of F are needed then.

5 Special features for affine linear problems

This section is restricted to the study of affine linear problems of the form

Find $u : \bar{I} \rightarrow \mathbb{R}^d$ such that

$$Mu'(t) = f(t) - Au(t), \quad u(t_0) = u_0 \in \mathbb{R}^d, \quad (5.1)$$

where $M, A \in \mathbb{R}^{d \times d}$ are time-independent matrices and M is regular, i.e., in the general setting we have $F(t, u) = f(t) - Au$.

5.1 A slight modification of the method

Let $0 \leq k \leq r$. In order to solve (5.1) numerically, we define the $\mathcal{J}_n$ -VTD $_k^r(g)$ problem by

Given $U(t_{n-1}^-)$, find $U \in P_r(I_n, \mathbb{R}^d)$ such that

$$U(t_{n-1}^+) = U(t_{n-1}^-), \quad \text{if } k \geq 1, \quad (5.2a)$$

$$MU^{(i+1)}(t_n^-) = g^{(i)}(t_n^-) - AU^{(i)}(t_n^-), \quad \text{if } k \geq 2, i = 0, \dots, \lfloor \frac{k}{2} \rfloor - 1, \quad (5.2b)$$

$$MU^{(i+1)}(t_{n-1}^+) = g^{(i)}(t_{n-1}^+) - AU^{(i)}(t_{n-1}^+), \quad \text{if } k \geq 3, i = 0, \dots, \lfloor \frac{k-1}{2} \rfloor - 1, \quad (5.2c)$$

and

$$\mathcal{I}_n[(MU', \varphi)] + \delta_{0,k}(M[U]_{n-1}, \varphi(t_{n-1}^+)) = \mathcal{I}_n[(g - AU, \varphi)] \quad \forall \varphi \in P_{r-k}(I_n, \mathbb{R}^d) \quad (5.2d)$$

where $U(t_0^-) = u_0$ and g will be chosen later depending on f . As before $\mathcal{I}_n$ denotes an integrator, typically the integral over I_n or a quadrature formula.

In a first step, we consider $Q_k^r\text{-VTD}_k^r(\mathcal{I}_{k+2}^{r+1}f)$ for $0 \leq k \leq r-1$. Here, the case $r = k$ is excluded since otherwise $\mathcal{I}_{k+2}^{r+1}f$ would not be well defined, cf. (2.3). In view of the postprocessing of Sect. 3 this modified method has some interesting properties.

Theorem 5.1 *Let $r, k \in \mathbb{N}_0$, $0 \leq k \leq r-1$. Suppose that $U \in Y_r$ solves $Q_k^r\text{-VTD}_k^r(\mathcal{I}_{k+2}^{r+1}f)$. Determine $\tilde{U} \in Y_{r+1}$ by the postprocessing of Theorem 3.1 or Corollary 3.3, respectively. Then $\tilde{U}$ solves $Q_{k+2}^{r+1}\text{-VTD}_{k+2}^{r+1}(f)$.*

Proof First of all, note that $\mathcal{I}_{k+2}^{r+1}f$ could be used instead of f in the definition of the correction vector $a_n \in \mathbb{R}^d$, see (3.1), since $\mathcal{I}_{k+2}^{r+1}$ preserves all occurring point values. Hence, postprocessing U as in Sect. 3 gives a function $\tilde{U} \in Y_{r+1}$ that solves $Q_k^r\text{-VTD}_{k+2}^{r+1}(\mathcal{I}_{k+2}^{r+1}f)$.

Since the interpolation operator $\mathcal{I}_{k+2}^{r+1}$ preserves all point values occurring in the collocation conditions of the method $Q_k^r\text{-VTD}_{k+2}^{r+1}(\mathcal{I}_{k+2}^{r+1}f)$, it could be dropped there. Moreover, only polynomials of maximal degree $2r-k$ appear in the variational condition of this method. Since both quadrature rules Q_k^r and Q_{k+2}^{r+1} are exact in this case and since $\mathcal{I}_{k+2}^{r+1}$ uses the quadrature points of Q_{k+2}^{r+1} , we obtain that $\tilde{U}$ also solves $Q_{k+2}^{r+1}\text{-VTD}_{k+2}^{r+1}(f)$. $\square$

Remark 5.2 Within the above argument we proved that the method $Q_{k+2}^{r+1}\text{-VTD}_{k+2}^{r+1}(f)$ and the method $Q_k^r\text{-VTD}_{k+2}^{r+1}(\mathcal{I}_{k+2}^{r+1}f)$ are equivalent for $0 \leq k \leq r-1$. Similarly, one can show that $Q_k^r\text{-VTD}_{k+2}^{r+1}(f)$ and $Q_{k+2}^{r+1}\text{-VTD}_{k+2}^{r+1}(\mathcal{I}_k^r f)$ are equivalent for $0 \leq k \leq r-1$. Note that also $\mathcal{I}_k^r$ preserves all derivatives that appear in the point conditions at both ends of the interval.

5.2 Interpolation cascade

Having a closer look at the result of Theorem 5.1, we see that the postprocessed solution of the modified discrete problem also solves a numerically integrated variational time discretization method but with the “right” associated quadrature rule. This enables one further postprocessing step.

For $1 \leq \ell \leq r-k$, using an interpolation cascade we even could enable up to $\ell+1$ additional postprocessing steps. More concretely we have

$$\begin{aligned} Q_k^r\text{-VTD}_k^r(\mathcal{I}_{k+2}^{r+1} \circ \mathcal{I}_{k+4}^{r+2} \circ \dots \circ \mathcal{I}_{k+2\ell}^{r+\ell} f) &\rightsquigarrow Q_{k+2}^{r+1}\text{-VTD}_{k+2}^{r+1}(\mathcal{I}_{k+4}^{r+2} \circ \dots \circ \mathcal{I}_{k+2\ell}^{r+\ell} f) \\ &\rightsquigarrow \dots \rightsquigarrow Q_{k+2(\ell-1)}^{r+\ell-1}\text{-VTD}_{k+2(\ell-1)}^{r+\ell-1}(\mathcal{I}_{k+2\ell}^{r+\ell} f) \rightsquigarrow Q_{k+2\ell}^{r+\ell}\text{-VTD}_{k+2\ell}^{r+\ell}(f) \\ &\rightsquigarrow Q_{k+2\ell}^{r+\ell}\text{-VTD}_{k+2(\ell+1)}^{r+\ell+1}(f) \end{aligned}$$

where $\rightsquigarrow$ denotes the postprocessing steps. Note that f itself can be used in each postprocessing step to calculate the correction vector $a_n \in \mathbb{R}^d$ (cf. Theorem 3.1) since in each step the occurring derivative of f at t_n^- is preserved by the respective interpolation cascade.

Remark 5.3 For Dahlquist's stability equation, i.e., $d = 1$, $M = 1$, $A = -\lambda$, and $f = 0$ in (5.1), we easily see that

$$\begin{aligned} \mathbf{VTD}_{k-2\ell}^{r-\ell}(f) &\cong Q_{k-2\ell}^{r-\ell} \mathbf{VTD}_{k-2\ell}^{r-\ell}(f) \\ &\cong Q_{k-2\ell}^{r-\ell} \mathbf{VTD}_{k-2\ell}^{r-\ell}(\mathcal{I}_{k-2\ell+2}^{r-\ell+1} \circ \mathcal{I}_{k-2\ell+4}^{r-\ell+2} \circ \dots \circ \mathcal{I}_k^r f) \end{aligned}$$

for all $\ell = 0, \dots, \lfloor \frac{k}{2} \rfloor$. Thus, ℓ postprocessing steps can be applied for this equation. Since the postprocessing does not change the function value in the end points of the intervals, the stability function does not change either. Therefore, $\mathbf{VTD}_k^r$ as well as $Q_k^r \mathbf{VTD}_k^r$ provide the same stability function as $\mathbf{VTD}_{k-2\ell}^{r-\ell}$. With the special choice $\ell = \lfloor \frac{k}{2} \rfloor$, we immediately find that $\mathbf{VTD}_k^r$ shares its stability properties with

$$\mathbf{VTD}_{k-2\lfloor \frac{k}{2} \rfloor}^{r-\lfloor \frac{k}{2} \rfloor} \cong \begin{cases} \mathbf{VTD}_0^{r-\lfloor \frac{k}{2} \rfloor} \cong \text{dG}(r - \lfloor \frac{k}{2} \rfloor), & \text{if } k \text{ is even,} \\ \mathbf{VTD}_1^{r-\lfloor \frac{k}{2} \rfloor} \cong \text{cGP}(r - \lfloor \frac{k}{2} \rfloor), & \text{if } k \text{ is odd.} \end{cases}$$

Hence, all methods with even k share their strong A-stability with the dG method while methods with odd k are A-stable as the cGP method, cf. Remark 2.1 and [6].

Remark 5.4 Analogously to Theorem 3.1 also a postprocessing from $Q_k^r \mathbf{VTD}_k^r(g)$ to $Q_k^r \mathbf{VTD}_{k+2}^{r+1}(g)$ can be proven when instead of (3.1) the correction vector is determined by

$$\check{a}_n = M^{-1} \left(g(\lfloor \frac{k}{2} \rfloor)(t_n^-) - AU(\lfloor \frac{k}{2} \rfloor)(t_n^-) - MU(\lfloor \frac{k}{2} \rfloor + 1)(t_n^-) \right).$$

However, when g and its derivatives are not globally continuous up to a sufficiently high order, then usually the discrete solution of $Q_k^r \mathbf{VTD}_k^r(g)$ is not $\lfloor \frac{k-1}{2} \rfloor$ -times continuously differentiable. Therefore, in general the postprocessing by jumps and the postprocessing by (modified) residuals do not provide the same correction anymore, also cf. Tables 6 and 7.

5.3 Derivatives of solutions

Since the quadrature formula Q_k^r is exact for polynomials up to degree $2r - k$ and the associated interpolation operator $\mathcal{I}_k^r$ yields polynomials of degree r , we easily find that

$$Q_k^r\text{-}\mathbf{VTD}_k^r(f) \cong Q_k^r\text{-}\mathbf{VTD}_k^r(\mathcal{I}_k^r f) \cong \mathbf{VTD}_k^r(\mathcal{I}_k^r f)$$

for all $r, k \in \mathbb{N}_0$, $0 \leq k \leq r$. Therefore, $\mathbf{VTD}_k^r(f)$ and $Q_k^r\text{-}\mathbf{VTD}_k^r(f)$ can nicely be written as one method of the form $\mathbf{VTD}_r^k(\mathcal{I}f)$ with $\mathcal{I} = \text{Id}$ and $\mathcal{I} = \mathcal{I}_k^r$, respectively.

Theorem 5.5 *Let $r, k \in \mathbb{N}_0$, $0 \leq k \leq r$, and suppose that $U \in Y_r$ solves $\mathbf{VTD}_r^k(\mathcal{I}f)$ with $\mathcal{I} \in \{\text{Id}, \mathcal{I}_k^r\}$. Then it holds*

$$\int_{I_n} (M(U^{(j)})' + AU^{(j)}, \varphi) dt + \delta_{0,k-2j} (M[U^{(j)}]_{n-1}, \varphi(t_{n-1}^+)) = \int_{I_n} ((\mathcal{I}f)^{(j)}, \varphi) dt$$

for all $0 \leq j \leq \lfloor \frac{k}{2} \rfloor$ and all $\varphi \in P_{r-k+j}(I_n, \mathbb{R}^d)$. Note that the integrals can be replaced for $\mathcal{I} = \mathcal{I}_k^r$ by any quadrature rule which is exact for polynomials of degree less than or equal to $2r - k$, for example by Q_{k-2j}^{r-j} .

Proof Let us start with the case $0 \leq j \leq \lfloor \frac{k-1}{2} \rfloor$. Integrating by parts several times and using (5.2) with $g = \mathcal{I}f$, we gain for any $\varphi \in P_{r-k+j}(I_n, \mathbb{R}^d)$

$$\begin{aligned} \int_{I_n} (M(U^{(j)})' + AU^{(j)}, \varphi) dt &= \int_{I_n} (\partial_t^j (MU' + AU), \varphi) dt \\ &= (-1)^j \int_{I_n} (MU' + AU, \underbrace{\varphi^{(j)}}_{\in P_{r-k}(I_n, \mathbb{R}^d)}) dt + \sum_{l=0}^{j-1} (-1)^l [(\partial_t^{j-1-l} (MU' + AU), \varphi^{(l)})]_{t_{n-1}^+}^{t_n^-} \\ &= (-1)^j \int_{I_n} (\mathcal{I}f, \varphi^{(j)}) dt + \sum_{l=0}^{j-1} (-1)^l [(\mathcal{I}f)^{(j-1-l)}, \varphi^{(l)}]_{t_{n-1}^+}^{t_n^-} = \int_{I_n} ((\mathcal{I}f)^{(j)}, \varphi) dt \end{aligned}$$

which is the desired statement. So, for odd k we are done due to $\lfloor \frac{k-1}{2} \rfloor = \lfloor \frac{k}{2} \rfloor$ in this case.

Hence, it only remains to study the case $j = \lfloor \frac{k}{2} \rfloor$ for even $k \geq 2$. Similar as above we conclude from (5.2) with $g = \mathcal{I}f$ for any $\varphi \in P_{r-k+j}(I_n, \mathbb{R}^d)$ that

$$\begin{aligned} \int_{I_n} (M(U^{(j)})' + AU^{(j)}, \varphi) dt \\ &= \int_{I_n} ((\mathcal{I}f)^{(j)}, \varphi) dt - (MU^{(j)}(t_{n-1}^+) + AU^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)) \\ &\quad + ((\mathcal{I}f)^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)). \end{aligned}$$

Since point values of $\mathcal{I}f$ and f only appear at t_n^- and t_{n-1}^+ up to the $(\lfloor \frac{k}{2} \rfloor - 1)$ th derivative which are preserved by the operator $\mathcal{I} \in \{\text{Id}, \mathcal{I}_k^r\}$, we obtain, for $n \geq 2$

$$\begin{aligned} ((\mathcal{I}f)^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)) &= (f^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)) = (f^{(j-1)}(t_{n-1}^-), \varphi(t_{n-1}^+)) \\ &= ((\mathcal{I}f)^{(j-1)}(t_{n-1}^-), \varphi(t_{n-1}^+)) = (MU^{(j)}(t_{n-1}^-) + AU^{(j-1)}(t_{n-1}^-), \varphi(t_{n-1}^+)) \end{aligned}$$

where the continuity of $f^{(j-1)}$ and (5.2b) with $g = \mathcal{I}f$ is used. Since also $U^{(j-1)}$ is continuous, we get

$$\begin{aligned} & (MU^{(j)}(t_{n-1}^+) + AU^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)) - ((\mathcal{I}f)^{(j-1)}(t_{n-1}^+), \varphi(t_{n-1}^+)) \\ &= (M[U^{(j)}]_{n-1}, \varphi(t_{n-1}^+)) \end{aligned}$$

with $[U^{(j)}]_{n-1} = U^{(j)}(t_{n-1}^+) - U^{(j)}(t_{n-1}^-)$. Thus, for $n \geq 2$ the wanted identity is proven. Moreover, recalling the definitions of $U^{(j)}(t_0^-)$ and $u^{(j)}(t_0)$, we find

$$\begin{aligned} (\mathcal{I}f)^{(j-1)}(t_0^+) - AU^{(j-1)}(t_0^+) &= f^{(j-1)}(t_0^+) - AU^{(j-1)}(t_0^-) \\ &= f^{(j-1)}(t_0^+) - Au^{(j-1)}(t_0) = Mu^{(j)}(t_0) = MU^{(j)}(t_0^-) \end{aligned}$$

which yields the wanted statement also for $n = 1$. So, the proof is completed. $\square$

Using the appropriate initial condition, derivatives of **VTD** solutions are themselves solutions of **VTD** methods.

Corollary 5.6 *Let $r, k \in \mathbb{N}_0$, $0 \leq k \leq r$, and suppose that $U \in Y_r$ solves $\mathbf{VTD}_k^r(\mathcal{I}f)$ where $\mathcal{I} \in \{\text{Id}, \mathcal{I}_k^r\}$. Then $U^{(j)} \in Y_{r-j}$, $0 \leq j \leq \lfloor \frac{k}{2} \rfloor$, solves $\mathbf{VTD}_{k-2j}^{r-j}((\mathcal{I}f)^{(j)})$ if $u^{(j)}(t_0)$ is used as initial condition.*

Proof Because of Theorem 5.5 it only remains to prove the needed conditions at t_{n-1}^+ and t_n^- . Since by construction U is $\lfloor \frac{k-1}{2} \rfloor$ -times continuously differentiable, the desired identities follow easily from (5.2a), (5.2b), and (5.2c) with $g = \mathcal{I}f$. $\square$

6 Numerical experiments

In this section we present some numerical tests supporting the theoretical results. All calculations were carried out using the software Julia [11] using the floating point data type `BigFloat` with 512 bits.

Example 6.1 We consider for $t \in (0, 15)$ the initial value problem

$$\begin{aligned} u_1'(t) &= u_3(t), \quad u_2'(t) = u_4(t), \quad u_3'(t) = \frac{-u_1(t)}{(u_1^2(t) + u_2^2(t))^{3/2}}, \\ u_4'(t) &= \frac{-u_2(t)}{(u_1^2(t) + u_2^2(t))^{3/2}} \end{aligned}$$

with the initial condition

$$u_1(0) = \frac{3}{5}, \quad u_2(0) = 0, \quad u_3(0) = 0, \quad u_4(0) = 2.$$

Its solution is

Table 1 Example 6.1: Results for $Q_0^6\text{-VTD}_0^6 = \text{dG}(6)$

N	$\ e\ _{L^2}$	$\ e\ _{\ell^\infty}$	$\ \tilde{e}\ _{L^2}$	$\ e'\ _{L^2}$	$\ e'\ _{\ell^\infty}$	$\ \tilde{e}'\ _{L^2}$	$\ \tilde{e}'\ _{\ell^\infty}$
128	8.5971-08	1.5479-10	1.2502-08	2.6835-05	3.7480-05	2.0549-06	7.4871-10
256	7.0038-10	2.2722-14	5.1387-11	4.4062-07	6.2083-07	1.6951-08	1.1009-13
512	5.5331-12	2.8719-18	2.0336-13	6.9747-09	9.8440-09	1.3425-10	1.3941-17
1024	4.3360-14	3.5308-22	7.9710-16	1.0936-10	1.5421-10	1.0525-12	1.7143-21
2048	3.3906-16	4.3181-26	3.1166-18	1.7104-12	2.4142-12	8.2298-15	2.0968-25
4096	2.6497-18	5.2739-30	1.2177-20	2.6732-14	3.7734-14	6.4310-17	2.5611-29
8192	2.0703-20	6.4388-34	4.7572-23	4.1773-16	5.8962-16	5.0245-19	3.1269-33
eoc	7.00	13.00	8.00	6.00	6.00	7.00	13.00
theo	7	13	8	6	6	7	13

$$u_1(t) = \cos(\theta) - \frac{3}{5}, \quad u_2(t) = \frac{4}{5} \sin(\theta), \quad u_3(t) = -\frac{5 \sin(\theta)}{5 - 3 \cos(\theta)},$$

$$u_4(t) = \frac{4 \cos(\theta)}{5 - 3 \cos(\theta)}$$

where θ is for any given t the unique solution of

$$\theta - \frac{3}{5} \sin(\theta) = t.$$

The problem is taken from [16, Example 4.4].

The appearing nonlinear systems within each time step were solved by Newton's method where we applied a Taylor expansion of the inherited data from the previous time interval to calculate an initial guess for all unknowns on the current interval. If higher order derivatives were needed at initial time $t = 0$, the ODE system and its temporal derivatives were used, see (2.7). The postprocessing used the jumps of the derivatives, as given in Corollary 3.3.

We denote by

$$e := u - U, \quad \tilde{e} := u - \tilde{U}$$

the error of the solution U and the error of the postprocessed solution $\tilde{U}$, respectively. Errors were measured in the norms

$$\|\varphi\|_{L^2} := \left(\int_{t_0}^{t_N} \|\varphi(t)\|^2 dt \right)^{1/2}, \quad \|\varphi\|_{\ell^\infty} := \max_{1 \leq n \leq N} \|\varphi(t_n^-)\|$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^d$.

Table 1 presents the results for $Q_0^6\text{-VTD}_0^6$ which is just dG(6) with numerical quadrature by the right-sided Gauss–Radau formula with 7 points. We show norms of the error between the solution u and the discrete solution U as well as the error

Table 2 Example 6.1: Results for $Q_5^6\text{-VTD}_5^6$

N	$\ e\ _{L^2}$	$\ e\ _{\ell^\infty}$	$\ \tilde{e}\ _{L^2}$	$\ e'\ _{L^2}$	$\ e'\ _{\ell^\infty}$	$\ \tilde{e}'\ _{L^2}$	$\ \tilde{e}'\ _{\ell^\infty}$
128	1.6008-05	1.8461-05	1.5985-05	8.7029-05	8.9590-05	6.7230-05	8.9590-05
256	8.7256-08	9.6357-08	8.6947-08	1.0063-06	4.6950-07	3.8814-07	4.6950-07
512	3.6241-10	3.9738-10	3.5761-10	1.4916-08	1.9468-09	2.0110-09	1.9468-09
1024	1.4859-12	1.5695-12	1.4122-12	2.3236-10	7.6864-12	1.2536-11	7.6864-12
2048	6.6085-15	6.1488-15	5.5307-15	3.6307-12	3.0102-14	9.0475-14	3.0102-14
4096	3.5588-17	2.4034-17	2.1618-17	5.6732-14	1.1766-16	6.9139-16	1.1766-16
8192	2.3648-19	9.3898-20	8.4461-20	8.8646-16	4.5970-19	5.3708-18	4.5970-19
eoc	7.23	8.00	8.00	6.00	8.00	7.01	8.00
theo	7	8	8	6	8	7	8

between the solution u and the postprocessed discrete solution $\tilde{U}$ in different norms. Using the results for $N = 4096$ and $N = 8192$, the experimental order of convergence (eoc) is calculated. In addition, the theoretically predicted convergence orders (theo) are given. We see clearly from Table 1 that the experimental orders of convergence coincide with the theoretical predictions. This holds for the function itself and its time derivative. Moreover, the order of convergence increases by 1 if one postprocessing step is applied. It is noteworthy that the error norm $\|\tilde{e}'\|_{\ell^\infty}$ shows the same high order superconvergence order as $\|e\|_{\ell^\infty}$. This behavior is due to the collocation conditions satisfied by the postprocessed solution $\tilde{U}$.

The results of our calculations using the variational time discretization $Q_5^6\text{-VTD}_5^6$ are collected in Table 2. Again we present the results in different norms for both the error itself and the error obtained after postprocessing the discrete solution. Also for this temporal discretization, all theoretically predicted orders of convergence are met by our numerical experiments. Compared to the results of $Q_0^6\text{-VTD}_0^6$ the superconvergence order measured in $\|\cdot\|_{\ell^\infty}$ is much smaller which is in agreement with our theory. In addition, the order of convergence of $\|\tilde{e}'\|_{\ell^\infty}$ is the same as the order of convergence of $\|e'\|_{\ell^\infty}$ since collocation conditions are fulfilled already by the discrete solution U . Hence, an improvement of this quantity by applying the postprocessing is not possible.

Table 3 shows the results for calculations using $Q_6^6\text{-VTD}_6^6$ as discretization in time. The presented error norms indicate that the experimental order of convergence are in agreement with our theory. Note that the postprocessing does not lead to an improvement of the error itself. However, there is an improvement if we look at the L^2 norm of the time derivative. We clearly see that the order of convergence is increased from 6 to 7 which is in agreement with Proposition 4.4. Moreover, there is no superconvergence at the discrete time points, as predicted by our theory.

Example 6.2 We consider the affine linear initial value problem

$$\begin{pmatrix} 10 & -20 \\ -10 & 10 \end{pmatrix} \begin{pmatrix} u_1'(t) \\ u_2'(t) \end{pmatrix} = \begin{pmatrix} -10e^{-10t} \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & -101 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \quad t \in (0, 40),$$

Table 3 Example 6.1: Results for $Q_6^6\text{-VTD}_6^6$

N	$\ e\ _{L^2}$	$\ e\ _{\ell^\infty}$	$\ \tilde{e}\ _{L^2}$	$\ e'\ _{L^2}$	$\ e'\ _{\ell^\infty}$	$\ \tilde{e}'\ _{L^2}$	$\ \tilde{e}'\ _{\ell^\infty}$
128	5.0614-04	4.6808-04	5.0626-04	1.9070-03	2.2689-03	1.9097-03	2.2689-03
256	5.1681-07	5.6697-07	5.1589-07	2.8381-06	2.7120-06	1.9073-06	2.7120-06
512	7.4117-09	9.3406-09	7.3967-09	4.5695-08	4.5324-08	3.0105-08	4.5324-08
1024	8.5890-11	1.0625-10	8.5784-11	6.3800-10	5.1584-10	3.4402-10	5.1584-10
2048	7.8180-13	9.5722-13	7.8101-13	8.9479-12	4.6479-12	3.1175-12	4.6479-12
4096	6.5416-15	7.9718-15	6.5355-15	1.3358-13	3.8713-14	2.6038-14	3.8713-14
8192	5.2802-17	6.4204-17	5.2755-17	2.0578-15	3.1179-16	2.1001-16	3.1179-16
eoc	6.95	6.96	6.95	6.02	6.96	6.95	6.96
theo	7	7	7	6	7	7	7

where the initial condition is given by

$$u_1(0) = 2, \quad u_2(0) = 1.$$

This results in

$$u_1(t) = e^{-t/10} + (1+t)e^{-10t}, \quad u_2(t) = (1+t)e^{-10t}$$

as the solution components. This problem is a slight modification of [15, Example 7.3]. In particular, a non-trivial mass matrix was introduced. Moreover, in order to see effects caused by the interpolation cascade, we added a non-vanishing right-hand side function $f = (-10e^{-10t}, 0)^T$.

Table 4 presents the results for $Q_0^9\text{-VTD}_0^9$ where the cascadic interpolation has been applied to the function f on the right-hand side, see Sect. 5. Let $\text{PP}_s U$ denote the discrete solution obtained after applying s postprocessing steps starting from U . We show norms of the error after s postprocessing steps using 2048 time steps. The given experimental orders of convergence were calculated from the results with 1024 and 2048 time steps. Looking at the convergence orders in the L^2 -like norms, we clearly see that each postprocessing step increases the experimental order of convergence by 1 if at most 9 postprocessing steps are applied. The postprocessing step 10 leads to an improvement of the convergence order only for the temporal derivative since the function itself already converges with the optimal order 19. The postprocessing has no influence to the ℓ_∞ norm of the error itself while the very first postprocessing step improves the results for the derivative of the error in the ℓ_∞ norm. This is caused by the fact that the postprocessed solution fulfills a collocation condition at the discrete time points.

Table 5 presents the experimental orders of convergence of $\|(u - \text{PP}_s U)'\|_{L^2}$ for $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, after s postprocessing steps where at most $r + 1 - k = 10 - k$ steps have been applied. The cascadic interpolation of the right-hand function f is used for all considered methods. It can clearly be seen that each additional

Table 4 Example 6.2: Results for $Q_0^9\text{-VTD}_0^9 = \text{dG}(9)$ with cascadic interpolation of f and s postprocessing steps

s	$\ u - \text{PP}_s U\ _{L^2}$		$\ (u - \text{PP}_s U)'\ _{L^2}$		$\ u - \text{PP}_s U\ _{\ell^\infty}$		$\ (u - \text{PP}_s U)'\ _{\ell^\infty}$	
	Error	Order	Error	Order	Error	Order	Error	Order
0	8.5293-22	10.000	2.7583-18	9.000	7.2828-38	18.962	6.3983-18	9.003
1	5.8267-24	10.933	8.3426-21	9.933	7.2828-38	18.962	7.2828-37	18.962
2	1.1459-25	11.957	1.3119-22	10.957	7.2828-38	18.962	7.2828-37	18.962
3	1.7574-27	12.966	1.8529-24	11.966	7.2828-38	18.962	7.2828-37	18.962
4	2.5135-29	13.970	2.4402-26	12.970	7.2828-38	18.962	7.2828-37	18.962
5	3.5521-31	14.972	3.2211-28	13.972	7.2828-38	18.962	7.2828-37	18.962
6	5.3505-33	15.974	4.3347-30	14.973	7.2828-38	18.962	7.2828-37	18.962
7	8.9769-35	16.974	6.3180-32	15.974	7.2828-38	18.962	7.2828-37	18.962
8	1.8583-36	17.974	1.0353-33	16.975	7.2828-38	18.962	7.2828-37	18.962
9	5.8032-38	18.972	2.1056-35	17.975	7.2828-38	18.962	7.2828-37	18.962
10	3.0646-38	18.997	6.4645-37	18.972	7.2828-38	18.962	7.2828-37	18.962

Table 5 Example 6.2: Experimental orders of convergence for $\|(u - \text{PP}_s U)'\|_{L^2}$ using $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, with cascadic interpolation of f , after s postprocessing steps

k	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$	$s = 8$	$s = 9$	$s = 10$
0	9.000	9.933	10.957	11.966	12.970	13.972	14.973	15.974	16.975	17.975	18.972
1	9.000	9.948	10.969	11.977	12.980	13.981	14.982	15.983	16.982	17.980	
2	9.000	9.932	10.956	11.965	12.969	13.971	14.972	15.972	16.970		
3	9.000	9.949	10.970	11.977	12.980	13.981	14.981	15.979			
4	9.000	9.930	10.954	11.963	12.967	13.969	14.967				
5	9.000	9.949	10.970	11.976	12.979	13.977					
6	9.001	9.927	10.952	11.961	12.962						
7	9.001	9.950	10.969	11.972							
8	9.003	9.926	10.946								
9	9.001	9.944									

postprocessing step increases the convergence by one order. Using the same number of postprocessing steps, the obtained convergence orders do not depend on the particular methods. Since each postprocessing step is covered by our theory and postprocessing by jumps and postprocessing by residual are equivalent for every single step, both types of postprocessing lead to identical results if the cascadic interpolation of the right-hand side function f is used.

The behavior changes substantially if just f and not its cascadic interpolation is used. Table 6 shows for the methods $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, the experimental convergence order of the error $\|(u - \text{PP}_s U)'\|_{L^2}$ after s postprocessing steps based on jumps where at most $r + 1 - k = 10 - k$ steps have been carried out. The column $s = 1$ shows, as predicted by our theory, that the convergence order increases by 1

Table 6 Example 6.2: Experimental orders of convergence for $\|(u - \text{PP}_s U)'\|_{L^2}$ using $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, and s postprocessing steps based on jumps, cf. Corollary 3.3

k	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$	$s = 8$	$s = 9$	$s = 10$
0	9.000	9.940	10.974	11.004	10.979	10.945	10.915	10.903	10.904	10.897	10.888
1	9.000	9.948	10.980	9.980	8.980	7.980	6.980	5.980	4.980	3.980	
2	9.000	9.935	10.971	10.970	10.981	11.038	11.250	11.319	11.209		
3	9.000	9.948	9.994	8.994	7.994	6.994	5.994	4.994			
4	9.000	9.932	10.966	11.059	10.971	10.934	10.905				
5	9.000	9.949	10.981	9.981	8.980	7.980					
6	9.000	9.929	10.957	10.954	10.991						
7	9.000	9.949	9.995	8.994							
8	9.001	9.924	10.936								
9	9.001	9.944									

Table 7 Example 6.2: Experimental orders of convergence for $\|(u - \text{PP}_s U)'\|_{L^2}$ using $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, and s postprocessing steps based on residuals, cf. Theorem 3.1

k	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$	$s = 8$	$s = 9$	$s = 10$
0	9.000	9.940	10.972	10.976	10.980	10.981	10.981	10.981	10.981	10.981	10.981
1	9.000	9.948	10.968	10.973	10.984	10.986	10.986	10.987	10.986	10.987	
2	9.000	9.935	10.962	10.967	10.977	10.979	10.979	10.980	10.979		
3	9.000	9.948	10.962	10.966	10.977	10.986	10.982	10.988			
4	9.000	9.932	10.952	10.958	10.968	10.978	10.973				
5	9.000	9.949	10.956	10.958	10.963	10.987					
6	9.000	9.929	10.940	10.945	10.945						
7	9.000	9.949	10.946	10.947							
8	9.001	9.924	10.915								
9	9.001	9.944									

for all methods. The behavior after using at least two postprocessing steps depends strongly on the parameter k of the variational time discretizations. For dG-like methods (characterized by even k), an additional improvement by one order is obtained independent of the number of postprocessing steps. The situation is completely different for cGP-like method (corresponding to odd k). For $k \equiv 3 \pmod{4}$, the second postprocessing step does not lead to an improvement of the convergence order compared to a single postprocessing step. If $k \equiv 1 \pmod{4}$ then the second postprocessing step provides an increased convergence order. However, for all cGP-like methods, the obtained convergence rates start to decrease with increasing numbers of postprocessing steps. This is in complete contrast to dG-like methods. Calculations for the methods $Q_k^{10}\text{-VTD}_k^{10}$, $k = 0, \dots, 10$, show for dG-like methods the same behavior as in the case $r = 9$. However, the roles of $k \equiv 1 \pmod{4}$ and $k \equiv 3 \pmod{4}$ for cGP-like methods are switched compared to the case $r = 9$.

Table 8 Example 6.3: Errors and experimental orders of convergence for $Q_0^6\text{-VTD}_0^6 = \text{dG}(6)$ without and with interpolation cascade of the right-hand side

N	Without cascade				With cascade			
	$\ e\ _{L^2}$		$\ e\ _{\ell^\infty}$		$\ e\ _{L^2}$		$\ e\ _{\ell^\infty}$	
	Error	eoc	Error	eoc	Error	eoc	Error	eoc
128	2.6606-19		7.2305-25		2.6606-19		1.3099-35	
256	2.0787-21	7.000	1.1858-27	9.252	2.0787-21	7.000	1.6049-39	12.995
512	1.6241-23	7.000	1.6219-30	9.514	1.6241-23	7.000	1.9624-43	12.998
1024	1.2688-25	7.000	1.8287-33	9.793	1.2688-25	7.000	2.3975-47	12.999
2048	9.9128-28	7.000	1.8326-36	9.963	9.9128-28	7.000	2.9277-51	12.999
4096	7.7444-30	7.000	1.0066-39	10.830	7.7444-30	7.000	3.5746-55	13.000
8192	6.0504-32	7.000	2.6935-43	11.868	6.0504-32	7.000	4.3639-59	13.000

Our theory provides that postprocessing based on jumps and postprocessing based on residuals are equivalent if a single postprocessing step is applied. The situation changes if at least two postprocessing steps are used, also cf. Remark 5.4. Table 7 shows the experimental orders of convergence of $\|(u - \text{PP}_s U)'\|_{L^2}$ after s postprocessing steps based on residuals for the methods $Q_k^9\text{-VTD}_k^9$, $k = 0, \dots, 9$, that are the same ones as used for obtaining the results in Table 6. Independent of k , the application of at least two postprocessing steps leads always to an improvement of the convergence order by two compared to the results without postprocessing. Moreover, the orders of convergence do not decrease even if more than two postprocessing steps based on residuals are applied. The same behavior is observed for the methods $Q_k^{10}\text{-VTD}_k^{10}$, $k = 0, \dots, 10$.

Example 6.3 We consider the one-dimensional heat equation $u_t(t, x) - u_{xx}(t, x) = f(t, x)$ for $(t, x) \in (0, 2) \times (0, 1)$ with homogeneous Dirichlet boundary conditions. The initial condition at $t = 0$ and the right-hand side function f are chosen such that the solution is given by

$$u(t, x) = 4x(1 - x)(\cos(t) + x \sin(2t)).$$

For semi-discretization in space we use a finite element method with continuous, piecewise cubic elements on an equidistant decomposition of the spatial interval $(0, 1)$ into 10 subintervals. This results in an ODE system of dimension 29 due to the homogeneous Dirichlet boundary conditions at $x = 0$ and $x = 1$. Note that by the proper choice of the trial space for the semi-discretization in space the spatial error is negligible.

On the basis of Example 6.3 we want to take another view on the interpolation cascade. To this end, the results for $Q_0^6\text{-VTD}_0^6$ which is just dG(6) are presented in Table 8. The errors in the L^2 norm and the ℓ^∞ norm are compared for the standard method and the method with cascadic interpolation. While measured in $\|\cdot\|_{L^2}$ the error is almost equal for both methods and the expected order of convergence is clearly reached, there are significant differences with regard to the error in the time mesh points. For the

standard method without cascade the expected superconvergence order (which is 13) clearly is not obtained, even for quite small time steps. This is probably due to the well-known order reduction phenomenon. In contrast, the wanted superconvergence behavior is achieved already for large time steps when the interpolation cascade is used. Moreover, the ℓ^∞ norm error is also considerably smaller. This suggests that the cascadic interpolation could be quite advantageous in some cases.

Acknowledgements We would like to thank the anonymous referees for their useful comments.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ahmed, N., Becher, S., Matthies, G.: Higher-order discontinuous Galerkin time stepping and local projection stabilization techniques for the transient Stokes problem. *Comput. Methods Appl. Mech. Eng.* **313**(1), 28–52 (2017)
2. Ahmed, N., John, V.: Adaptive time step control for higher order variational time discretizations applied to convection–diffusion–reaction equations. *Comput. Methods Appl. Mech. Eng.* **285**, 83–101 (2015)
3. Ahmed, N., Matthies, G.: Higher order continuous Galerkin–Petrov time stepping schemes for transient convection–diffusion–reaction equations. *ESAIM: M2AN* **49**(5), 1429–1450 (2015)
4. Ahmed, N., Matthies, G.: Numerical study of SUPG and LPS methods combined with higher order variational time discretization schemes applied to time-dependent linear convection–diffusion–reaction equations. *J. Sci. Comput.* **67**(1), 988–1018 (2016)
5. Akrivis, G., Makridakis, C., Nohetto, R.H.: Galerkin and Runge–Kutta methods: unified formulation, a posteriori error estimates and nodal superconvergence. *Numer. Math.* **118**, 429–456 (2011)
6. Anselmann, M., Bause, M., Becher, S., Matthies, G.: Galerkin-collocation approximation in time for the wave equation and its post-processing. *ESAIM: M2AN* **54**(6), 2099–2123 (2020)
7. Aziz, A.K., Monk, P.: Continuous finite elements in space and time for the heat equation. *Math. Comput.* **52**(186), 255–274 (1989)
8. Bause, M., Köcher, U., Radu, F.A., Schieweck, F.: Post-processed Galerkin approximation of improved order for wave equations. *Math. Comput.* **89**(322), 595–627 (2020)
9. Becher, S., Matthies, G.: Variational time discretizations of higher order and higher regularity. Preprint. [arXiv: 2003.04056](https://arxiv.org/abs/2003.04056)
10. Becher, S., Matthies, G., Wenzel, D.: Variational methods for stable time discretization of first-order differential equations. In: Georgiev, K., Todorov, M.I.G. (eds.) *Advanced Computing in Industrial Mathematics: BGSIAM 2017, Studies in Computational Intelligence*, vol. 793, pp. 63–75. Springer International Publishing, Cham (2019)
11. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. *SIAM Rev.* **59**(1), 65–98 (2017)
12. Delfour, M.C., Dubeau, F.: Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Math. Comput.* **47**(175), 169–189 (1986)
13. Encinas, L.H., Masqué, J.M.: A short proof of the generalized Faà di Bruno's formula. *Appl. Math. Lett.* **16**, 975–979 (2003)

14. Ern, A., Schieweck, F.: Discontinuous Galerkin method in time combined with a stabilized finite element method in space for linear first-order PDEs. *Math. Comput.* **85**(301), 2099–2129 (2016)
15. Estep, D.: A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM J. Numer. Anal.* **32**(1), 1–48 (1995)
16. Estep, D., French, D.: Global error control for the continuous Galerkin finite element method for ordinary differential equations. *ESAIM: M2AN* **28**(7), 815–852 (1994)
17. Estep, D., Stuart, A.: The dynamical behavior of the discontinuous Galerkin method and related difference schemes. *Math. Comput.* **71**(239), 1075–1103 (2002)
18. French, D., Schaeffer, J.W.: Continuous finite element methods which preserve energy properties for nonlinear problems. *Appl. Math. Comput.* **39**(3), 271–295 (1990)
19. Gautschi, W.: *Orthogonal Polynomials: Computation and Approximation*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
20. Hairer, E., Nørsett, S.P., Wanner, G.: *Solving Ordinary Differential Equations I*, 2nd edn. Springer-Verlag (1993). Corrected 3rd printing (2008)
21. Hulme, B.L.: Discrete Galerkin and related one-step methods for ordinary differential equations. *Math. Comput.* **26**(120), 881–891 (1972)
22. Hulme, B.L.: One-step piecewise polynomial Galerkin methods for initial value problems. *Math. Comput.* **26**(118), 415–426 (1972)
23. Joulak, H., Beckermann, B.: On Gautschi's conjecture for generalized Gauss–Radau and Gauss–Lobatto formulae. *J. Comput. Appl. Math.* **233**(3), 768–774 (2009)
24. Matthies, G., Schieweck, F.: Higher order variational time discretizations for nonlinear systems of ordinary differential equations. Preprint 23/2011, Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg (2011)
25. Mishkov, R.L.: Generalization of the formula of Faà di Bruno for a composite function with a vector argument. *Int. J. Math. Math. Sci.* **24**(7), 481–491 (2000)
26. Petrova, G.: Generalized Gauss–Radau and Gauss–Lobatto formulas with Jacobi weight functions. *BIT Numer. Math.* **57**(1), 191–206 (2017)
27. Schieweck, F.: A-stable discontinuous Galerkin–Petrov time discretization of higher order. *J. Numer. Math.* **18**(1), 25–57 (2010)
28. Thomée, V.: *Galerkin Finite Element Methods for Parabolic Problems*, 2nd edn. Springer, Berlin (2006)
29. Vlasák, M., Roskovec, F.: On Runge–Kutta, collocation and discontinuous Galerkin methods: mutual connections and resulting consequences to the analysis. In: *Programs and Algorithms of Numerical Mathematics 17*, pp. 231–236. Institute of Mathematics AS CR, Prague (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.