YIC GACM 2015
3^{rd} ECCOMAS Young Investigators Conference
6^{th} GACM Colloquium
July 20–23, 2015, Aachen, Germany

Multiderivative time-integrators for the hybridized discontinuous Galerkin method

A. Jaust [a,*], J. Schütz [b], D.C. Seal [c]

[a] MathCCES, RWTH Aachen University
Schinkelstraße 2, 52062 Aachen, Germany

[b] IGPM, RWTH Aachen University
Templergraben 55, 52056 Aachen, Germany

[c] Department of Mathematics, Michigan State University
619 Red Cedar Road, East Lansing, MI 48824

*jaust@mathcces.rwth-aachen.de

Abstract. *We present a combination of multiderivative Runge-Kutta methods and a hybridized discontinuous Galerkin (HDG) method. Multiderivative Runge-Kutta methods employ additional time derivatives of the unknown to achieve, with the same number of stages, a higher order of temporal accuracy than standard Runge-Kutta methods. A way how to incorporate these derivatives into the discretization is described. In order to validate the method we show numerical results for the linear advection equation.*

Keywords: hybridized discontinuous Galerkin; multiderivative Runge-Kutta method; high-order CFD

1 INTRODUCTION

The discontinuous Galerkin (DG) method is a popular scheme for high-order CFD. If implicit time-stepping methods are used, a large number of globally coupled unknowns arises making the method rather costly. However, the cost can be reduced by using hybridized DG methods (HDG) [1, 2, 3]. These methods introduce an additional *hybrid* variable on cell interfaces, allowing the system to be rewritten in such a way that it is coupled globally only through these new unknowns [1]. HDG can significantly reduce the number of globally coupled unknowns especially for polynomials of high degree p. Due to the implicit nature of the HDG method it is required to use implicit time-stepping methods. In previous works diagonally implicit Runge-Kutta (DIRK) and backward differentiation formulae (BDF) have been used [4, 5]. However, BDF schemes suffer from stability degradation when the order of accuracy is increased. Stable DIRK methods can be constructed for arbitrary orders, but this comes at the cost of additional stages, increasing the computational complexity of DIRK methods.

An interesting class of time integrators are multiderivative Runge-Kutta schemes [6, 7]. In contrast to classical methods, higher time-derivatives are incorporated in the formula to increase its temporal accuracy. Explicit multiderivative methods have shown to be a feasible alternative to classical time integrators for high-order methods such as DG [6]. We present an approach to couple the HDG method to implicit multiderivative Runge-Kutta methods for the linear advection equation. Afterwards, numerical results are presented to show stability and accuracy of the method.

2 NUMERICAL METHOD

In this section, we describe two different two-derivative Runge-Kutta (TDRK) methods. Afterwards we introduce an HDG method employing TDRK methods for the linear advection equation

$$u_t + \nabla \cdot (\vec{c}u) = 0, \qquad \forall (x,t) \in \Omega \times [0,T] \tag{1}$$

$$u(x,0) = u_0(x) \qquad \forall x \in \Omega \tag{2}$$

on an open bounded domain $\Omega \subset \mathbb{R}^2$. Here, u is a scalar unknown, $\vec{c}$ is a given vector and $u_0(x)$ is an initial datum. We assume that the equation is equipped with appropriate boundary conditions.

2.1 Two-derivative Runge-Kutta methods

As a starting point, we consider an ordinary differential equation (ODE)

$$\frac{\partial}{\partial t} y(t) = f(y), \quad y(0) = y_0, \quad t > 0. \tag{3}$$

Common integrators approximate $y(t)$ only using $f(y)$. For Runge-Kutta methods, the order of accuracy is increased by introducing additional stages. This causes a substantial growth in terms of computational cost since each stage requires solving a system of (nonlinear) equations.

Instead of adding stages, it is possible to consider additional derivatives of y (or f, respectively). In case of the ODE (3) the second time derivative can be expressed by

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial t} y(t)\right) = \frac{\partial}{\partial t} f(y) = \frac{\partial}{\partial t} y(t) \cdot f'(y) = f(y) \cdot f'(y) =: g(y). \tag{4}$$

The actual time derivative is replaced by the derivative of the right hand side of the ODE. Runge-Kutta methods employing one additional derivative are called two-derivative Runge-Kutta methods. For an r-stage TDRK method the solution at each time step is given by

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{r} b_i^{(1)} f(y^{(i)}) + \Delta t^2 \sum_{i=1}^{r} b_i^{(2)} g(y^{(i)}), \tag{5}$$

where the solution at each stage i is given by

$$y^{(i)} = y^n + \Delta t \sum_{j=1}^{r} a_{ij}^{(1)} f(y^{(j)}) + \Delta t^2 \sum_{j=1}^{r} a_{ij}^{(2)} g(y^{(j)}). \tag{6}$$

The index n indicates the current time $t^n = n\Delta t$ where Δt is the time step size. Only two-stage TDRK methods are used in this work, i.e. $r = 2$.

In the setting of the linear advection equation the functions f and g are given by

$$f(u) := -\nabla \cdot (\vec{c}u)\,, \quad g(u) := \nabla \cdot (\mathbf{C}\nabla u)\,. \tag{7}$$

The additional time derivative has been replaced using Cauchy-Kovalevskaja's procedure [6]. The unknown is called u here to indicate that it is the solution of a partial differential equation (PDE) and not of an ODE. The matrix $\mathbf{C}$ depends on the vector $\vec{c}$ through $\mathbf{C} = \vec{c}\,\vec{c}^T$.

The coefficients can be presented in an extended Butcher tableau (cf. Table 1) similar as for standard Runge-Kutta methods. The nonzero structure determines whether a two-derivative method is explicit or implicit. We focus on two different two-stage two-derivative methods that are third (TDRK3) and fourth (TDRK4) [7] order accurate. The coefficients of both methods are given in Table 1. For both schemes the first stage is explicit and only the second stage is implicit. The TDRK3 method is A- and L-stable while the TDRK4 method is A- but not L-stable. A more detailed description of two-derivative Runge-Kutta methods can be found in [6, 7], for example.

2.2 Hybridized discontinuous Galerkin method

For the spatial discretization the domain Ω has to be partitioned into N disjoint elements

$$\Omega = \bigcup_{k=1}^{N} \Omega_k. \tag{8}$$

Table 1: Extended Butcher tableau and coefficients of the third (TDRK3) and fourth (TDRK4) order two-derivative Runge-Kutta methods (from left to right).

$$
\begin{array}{c|cc|cc}
c_1 & a_{11}^{(1)} & a_{12}^{(1)} & a_{11}^{(2)} & a_{12}^{(2)} \\
c_2 & a_{21}^{(1)} & a_{22}^{(1)} & a_{21}^{(2)} & a_{22}^{(2)} \\
\hline
 & b_1^{(1)} & b_2^{(1)} & b_1^{(2)} & b_2^{(2)}
\end{array}
\qquad
\begin{array}{c|cc|cc}
0 & 0 & 0 & 0 & 0 \\
1 & \frac{1}{3} & \frac{2}{3} & 0 & -\frac{1}{6} \\
\hline
 & \frac{1}{3} & \frac{2}{3} & 0 & -\frac{1}{6}
\end{array}
\qquad
\begin{array}{c|cc|cc}
0 & 0 & 0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{12} & -\frac{1}{12} \\
\hline
 & \frac{1}{2} & \frac{1}{2} & \frac{1}{12} & -\frac{1}{12}
\end{array}
$$

We refer to edges of two intersecting elements and elements intersecting the domain boundary $\partial\Omega$ with e_k. The set of all edges is Γ and the number of edges is given by $\widehat{N} := |\Gamma|$. On these edges we define a new hybrid unknown λ. As u_{tt} contains the expression $\nabla \cdot (\mathbf{C}\nabla u)$, we treat the ocurring derivative as an additional unknown $\sigma := \nabla u$.
For the description of the method we need the following function spaces

$$H_h := \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^p(\Omega_k)\ \forall k = 1, \dots, N\}^2 \tag{9}$$

$$V_h := \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^p(\Omega_k)\ \forall k = 1, \dots, N\} \tag{10}$$

$$M_h := \{f \in L^2(\Gamma) \mid f_{|e_k} \in \Pi^p(e_k) \quad \forall k = 1, \dots, \widehat{N},\ e_k \in \Gamma\} \tag{11}$$

where Π^p is the space of polynomials up to degree p. Then, to solve equation (6) for TDRK3 or TDRK4 one seeks for functions $(\sigma_h, u_h, \lambda_h) \in H_h \times V_h \times M_h$ such that

$$\left(\sigma_h^{(i)} - \nabla u_h^{(i)}, \tau_h\right) - \left\langle \lambda_h^{(i)} - u_h^{(i),-}, \tau_h^- \cdot n \right\rangle_{\partial\Omega_k} = 0 \quad \forall \tau_h \in H_h \tag{12}$$

$$\begin{aligned}\left((u_h^{(i)} - u_h^n)_t, \varphi_h\right) + \sum_{j=1}^{i} \Big[&- \left(\Delta t a_{ij}^{(1)} f(u_h^{(j)}) + \Delta t^2 a_{ij}^{(2)} g(u_h^{(j)}, \sigma_h^{(j)}), \nabla\varphi_h\right) \\ &+ \left\langle \left(\Delta t a_{ij}^{(1)} \hat{f}^{(j)} + \Delta t^2 a_{ij}^{(2)} \hat{g}^{(j)}\right) \cdot n, \varphi_h^- \right\rangle_{\partial\Omega_k} \Big] = 0 \quad \forall \varphi_h \in V_h \end{aligned} \tag{13}$$

$$\left\langle [\![\Delta t a_{ii}^{(1)} \hat{f}^{(i)}]\!] \cdot n, \mu_h \right\rangle_\Gamma = 0 \quad \forall \mu_h \in M_h \tag{14}$$

holds with f and g as described in equation (7). Here, $(\cdot, \cdot)$ is the inner product on elements and $\langle \cdot, \cdot \rangle_{\partial\Omega_k}$ and $\langle \cdot, \cdot \rangle_\Gamma$ refer to the inner product on edges. Fluxes over edges have been replaced by numerical fluxes

$$\hat{f}^{(j)} := f(\lambda_h^{(j)}) - \alpha(\lambda_h^{(j)} - u_h^{(j),-})n, \quad \hat{g}^{(j)} := g(\lambda_h^{(j)}, \sigma_h^{(j),-}) + \beta(\lambda_h^{(j)} - u_h^{(j),-})n. \tag{15}$$

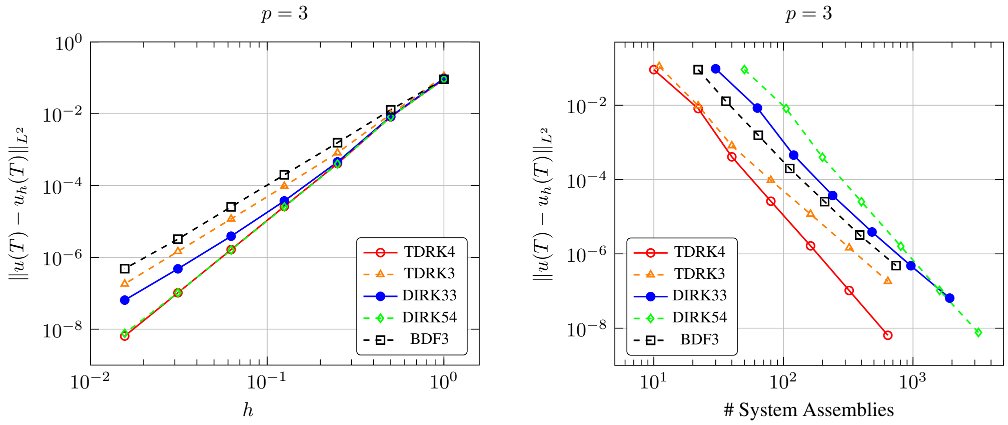
with positive parameters α and β that are chosen to ensure stability. A minus superscript indicates that the variable is evaluated at the element's interior.
The discretization in equation (12)–(14) looks similar to standard DG discretizations. However, u_h and σ_h are only evaluated locally on each element. The coupling between elements is solely realized by the hybrid variable λ_h. This allows to rewrite the discrete system, such that it is only globally coupled in λ_h, using static condensation [1]. A more elaborate description of the HDG method is given in previous papers, e.g. see [1, 2, 3].

3 NUMERICAL RESULTS FOR THE LINEAR ADVECTION EQUATION

We present some results of the method applied to the linear advection equation (1). The vector $\vec{c}$ is set to $\vec{c} = (1, 1)^T$. Then, the matrix $\mathbf{C}$ has entries equal to one everywhere. The linear advection equation is solved on $\Omega = (-1, 1)^2$ for a final time $T = 0.5$. The domain is discretized using a triangular mesh and the time step on the coarsest mesh is $\Delta t = 0.1$. The coarsest mesh consists of 8 elements. Initial data and boundary conditions are chosen such that the exact solution is

$$u(x, y, t) = \sin(\pi(x + y - 2t)). \tag{16}$$

In Figure 1 the error over the element size h for polynomials of degree $p = 3$ and the number of system assemblies is shown. Both TDRK methods show good approximation properties and achieve the expected order of convergence in time. The errors of the TDRK methods are comparable to DIRK and BDF methods. The error of the third order TDRK methods is between the errors of BDF3 and a third order DIRK method with three stages. The errors of the fourth order methods almost coincide as can be seen in Figure 1a.
We measure the costs of the methods as number of assemblies of the system arising from Newton's method because this is the most expensive step. The BDF3 method is less expensive than the DIRK methods as can be seen in Figure



(a) Error at the end of the simulation.

(b) Number of system assemblies.

Figure 1: Errors for different time integrators on different meshes.

1b. It is a drawback of the DIRK methods the system has to be solved in each stage. The TDRK methods have the fewest system assemblies since the first stage is explicit and no startup phase with another method is needed. During the start-up phase of BDF3 we use BDF2 with a smaller time step size.

4 CONCLUSION AND OUTLOOK

We have presented a hybridized discontinuous Galerkin method for time-dependent problems. In contrast to earlier publications [4, 5] a two-derivative Runge-Kutta method is applied for time integration. The arising system of equations requires the approximation of additional spatial derivatives. This can be incorporated in the HDG approximation in a stable manner. The numerical results reflect the stability and accuracy of the method. The number of system assemblies is similar to BDF methods and much lower than for DIRK methods. Thus, the multiderivative methods are promising candidates for high-order time integration.

Future work will extend the formulation to nonlinear equations such as the Euler or Navier-Stokes equations. Moreover, the efficiency of two-derivative Runge-Kutta methods compared to multistep and common Runge-Kutta methods has to be evaluated in more detail.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the computing time granted by the IT Center (ITC) of RWTH Aachen University on the RWTH Compute Cluster.

REFERENCES

[1] B. Cockburn, J. Gopalakrishnan, R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis* **47**:1319–1365, 2009.

[2] N.C. Nguyen, J. Peraire, B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics* **228**:8841–8855, 2009.

[3] J. Schütz, G. May. A hybrid mixed method for the compressible Navier-Stokes equations. *Journal of Computational Physics* **240**:58–75, 2013.

[4] A. Jaust, J. Schütz. A temporally adaptive hybridized discontinuous Galerkin method for time-dependent flows. *Computers and Fluids* **92**:177–15, 2014.

[5] N.C. Nguyen, J. Peraire, B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics* **230**:3695–3718, 2011.

[6] D.C. Seal, Y. Güçlü, A.J. Christlieb. High-order multiderivative time integrators for hyperbolic conservation laws. *Journal of Scientific Computing* **60**:101–140, 2014.

[7] A.Y.J. Tsai, R.P.K. Chan, S. Wang. Two-derivative Runge-Kutta methods for PDEs using a novel discretization approach. *Numerical Algorithms* **65**:687–703, 2014.