A temporally adaptive hybridized discontinuous Galerkin method for instationary compressible flows

Alexander Jaust and Jochen Schütz

Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Aachen, Germany

Abstract

The potential of the hybridized discontinuous Galerkin (HDG) method has been recognized for the computation of stationary flows. Extending the method to instationary problems can, e.g., be done by backward difference formulae (BDF) or diagonally implicit Runge-Kutta (DIRK) methods. In this publication, we investigate the use of *embedded* DIRK methods in an HDG solver, including the use of adaptive timestep control. Numerical results are shown that demonstrate the performance of the method for both linear and nonlinear (systems of) instationary convection-diffusion equations.

Keywords: hybridized discontinuous Galerkin, embedded diagonally implicit Runge-Kutta methods, instationary convection-diffusion equations

1. Introduction

The last few years have seen a tremendous increase in the use and development of high order methods for aerodynamic applications, see [1, 2, 3] to only mention a few. These methods represent the unknown function by a (piecewise) polynomial of degree larger than two, and so exceed the design order of Finite-Volume schemes that are nowadays standard tools in the aerospace industry. One particular example of high order methods are the discontinuous Galerkin (DG) methods introduced by Reed and Hill [4] and subsequently extended to all sorts of equations by many authors, see, e.g., [5, 6, 7, 8, 9, 10, 11, 12]. DG offers a lot of inherent advantages, such as the flexibility concerning the local degree of polynomial, the easy incorporation of boundary conditions on complicated domains, local conservativity and many more.

However, in the context of stationary problems, where usually the Jacobian of the method is needed, DG methods suffer from large memory requirements. This is due to the fact that the number of unknowns increases as $O((p+1)^d N)$, where p is the order of the local polynomial, d the spatial dimension and N the number of elements in the triangulation. Especially for the combination of high p and $d > 1$, this poses a severe restriction. One way of tackling this problem is to use *hybridized* DG methods, see [13, 14, 15, 16, 17, 18, 19]. The globally coupled unknowns in this type of method are the degrees of freedom belonging to the unknown function w on the *edges* instead of on the elements. Obviously, this yields a reduction in dimension, and the number of globally coupled unknowns behaves as $O((p+1)^{d-1}\widehat{N})$, where $\widehat{N}$ is the number of edges in the triangulation.

During the second high-order workshop at DLR in Cologne (see [20] for a summary of the first workshop), we have presented our method for 'easy' stationary problems in the context

of two-dimensional Euler and Navier-Stokes equations. It could be seen that hybridized DG methods have a potential of outperforming more traditional schemes. Still, there are a few things missing, among them the efficient implementation of time-integration routines. One potential problem of the hybrid method is that there is no straightforward method of lines approach as for DG methods, where one can easily use standard SSP schemes [21, 22]. However, using a dual timestepping approach [23], one can incorporate *implicit* time integration methods.

Extension to time-dependent problems has been made using BDF methods [13, 14, 24] and DIRK schemes [16]. In this publication, we investigate the use of different embedded DIRK schemes [25, 26, 27] for the temporal discretization of the hybridized DG method including timestep control. We show numerical results, demonstrating the performance of the method.

The paper is organized as follows: In Sec. 2, we shortly introduce the underlying equations. In Sec. 3, we first introduce the HDG method for the semi-discrete case, and then extend it via embedded DIRK methods to the fully discrete case. In addition, timestep control is discussed in this section. Sec. 4 shows numerical results, and Sec. 5 offers conclusions and an outlook. In the appendix section we give the Butcher tableaus of the embedded DIRK methods we use.

2. Underlying Equations

In this publication, we consider for a two-dimensional domain $\Omega \subset \mathbb{R}^2$ the general unsteady convection-diffusion equation, given as

$$w_t + \nabla \cdot (f(w) - f_v(w, \nabla w)) = g \qquad \forall (x,t) \in \Omega \times (0,T), \tag{1}$$

$$w(x,0) = w_0(x) \quad \forall x \in \Omega, \tag{2}$$

where w_0 and g are given functions, and $f : \mathbb{R}^m \to \mathbb{R}^{m\times 2}$, $f_v : \mathbb{R}^m \times \mathbb{R}^{m\times 2} \to \mathbb{R}^{m\times 2}$ are convective and diffusive flux, respectively. $T > 0$ denotes a final time; while m is the dimension of the system. The equations are equipped with appropriate boundary conditions, which depend on the particular choice of the fluxes f and f_v.

Note that both unsteady Euler and Navier-Stokes equations fall into this framework, where the unknown is $w = (\rho, \rho u, \rho v, E)$, i.e., density, momentum, and total energy. Corresponding fluxes are defined by

$$f_1 = (\rho u, p + \rho u^2, \rho uv, u(E+p))^T, \qquad f_2 = (\rho v, \rho uv, p + \rho v^2, v(E+p))^T, \tag{3}$$

$$f_v^1 = (0, \tau_{11}, \tau_{21}, \tau_{11}u + \tau_{12}v + k\,\mathfrak{T}_{x_1})^T, \quad f_v^2 = (0, \tau_{12}, \tau_{22}, \tau_{21}u + \tau_{22}v + k\,\mathfrak{T}_{x_2})^T, \tag{4}$$

and the right-hand side $g \equiv 0$. As usual, p denotes pressure, τ stress tensor, $\mathfrak{T}$ temperature and k thermal conductivity coefficient. p is coupled to the conservative variables using the ideal gas law in the form

$$p = (\gamma - 1)\left(E - \frac{\rho(u^2 + v^2)}{2}\right). \tag{5}$$

In aerodynamic applications the ratio of specific heats γ is usually taken to be 1.4.

As it is frequently done when considering diffusion equations [12], we formulate (1) as a first-order system by introducing the unknown function $\sigma := \nabla w$, i.e., in the sequel, we consider

$$\sigma = \nabla w, \quad w_t + \nabla \cdot (f(w) - f_v(w,\sigma)) = g \qquad \forall (x,t) \in \Omega \times (0,T), \tag{6}$$

$$w(x,0) = w_0(x) \quad \forall x \in \Omega. \tag{7}$$

3. A hybridized DG method

3.1. Semi-Discrete Method

In this section, we formulate the semi-discrete version of our hybridized DG method. To this end, we need a triangulation that is defined in the sequel:

Definition 1. *We assume that Ω is triangulated as*

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

We define an edge e_k to be either an intersection of two neighboring elements, or the intersection of an element with the physical boundary $\partial\Omega$, having positive one-dimensional measure. Γ denotes the collection of all these intersections, while $\Gamma_0 \subset \Gamma$ denotes those $e_k \in \Gamma$ that do not intersect the physical boundary $\partial\Omega$ of the domain. We define $\widehat{N} := |\Gamma|$ to be the number of edges in Γ.

For the ease of presentation, we introduce the following standard abbreviations for integration:

$$(f_1, f_2) := \sum_{k=1}^{N} \int_{\Omega_k} f_1 \cdot f_2 \, dx, \quad \langle f_1, f_2 \rangle_\Gamma := \sum_{k=1}^{\widehat{N}} \int_{e_k} f_1 \cdot f_2 \, d\sigma, \quad \langle f_1, f_2 \rangle_{\partial\Omega_k} := \sum_{k=1}^{N} \int_{\partial\Omega_k} f_1 \cdot f_2 \, d\sigma$$

In the method to be presented, both σ and w are approximated explicitly. Additionally, we introduce a variable λ that has support on the skeleton of the mesh only, $\lambda := w_{|\Gamma_0}$. The resulting algorithm will thus approximate the quantity

$$\boldsymbol{w} := (\sigma, w, w_{|\Gamma_0}). \tag{9}$$

On first sight, this seems like a tremendous increase in degrees of freedom, as one does not only approximate w (which is usually done in DG methods), but also σ and λ. However, the hybridized DG algorithm is constructed in such a way that one can locally eliminate both approximations to σ and w in favor of the approximation to λ. The only coupled degrees of freedom are those associated to the approximation of λ.

In the sequel, we define the correct approximation spaces:

Definition 2. *Let the approximation to $\boldsymbol{w}(\cdot, t)$ at some fixed time t,*

$$\boldsymbol{w}_h(\cdot, t) := (\sigma_h(\cdot, t), w_h(\cdot, t), \lambda_h(\cdot, t)) \tag{10}$$

be in $\mathbb{X}_h := H_h \times V_h \times M_h$, where

$$\begin{aligned}
H_h &:= \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^p(\Omega_k) \quad \forall k = 1, \dots N\}^{2\cdot m} \\
V_h &:= \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^p(\Omega_k) \quad \forall k = 1, \dots N\}^{m} \\
M_h &:= \{f \in L^2(\Gamma) \mid f_{|e_k} \in \Pi^p(e_k) \quad \forall k = 1, \dots \widehat{N}, e_k \in \Gamma\}^{m}.
\end{aligned}$$

Remark 1. *Whenever we use bold letters for a function, we think of a triple of functions. See the definitions* (9) *and* (10) *of $\boldsymbol{w}$ and $\boldsymbol{w}_h$, respectively, for examples.*

Based on these approximation spaces, and following Nguyen et al.'s and our previous work [13, 14, 19], we can define the semi-discretization in the sequel:

Definition 3. *A semi-discrete approximation* $\boldsymbol{w}_h(\cdot,t) := (\sigma_h(\cdot,t), w_h(\cdot,t), \lambda_h(\cdot,t)) \in \mathbb{X}_h$ *to* (6) *using the hybridized DG method is defined as the function* $\boldsymbol{w}_h$*, such that for all* $t \in (0,T)$*:*

$$(\sigma_h - \nabla w_h, \tau_h) - \langle \lambda_h - w_h^-, \tau_h^- \cdot n \rangle_{\partial\Omega_k} = 0 \qquad \forall \tau_h \in H_h \tag{11}$$

$$((w_h)_t, \varphi_h) - (f(w_h) - f_v(w_h, \sigma_h), \nabla\varphi_h) + \langle (\widehat{f} - \widehat{f_v}) \cdot n, \varphi_h^- \rangle_{\partial\Omega_k} = (g, \varphi_h) \quad \forall \varphi_h \in V_h \tag{12}$$

$$\langle [(\widehat{f} - \widehat{f_v})] \cdot n, \mu_h \rangle_\Gamma = 0 \qquad \forall \mu_h \in M_h. \tag{13}$$

Numerical fluxes $\widehat{f}$ *and* $\widehat{f_v}$ *are defined as*

$$\widehat{f} := f(\lambda_h) - \delta\left(\lambda_h - w_h^-\right) n, \quad \widehat{f_v} := f_v(\lambda_h, \sigma_h^-) + \tau\left(\lambda_h - w_h^-\right) n.$$

Both δ *and* τ *are real parameters that depend on* f *and* f_v*, respectively.*

Remark 2. *In the limiting cases* $f \equiv 0$*, one has* $\delta = 0$*, while for* $f_v \equiv 0$*,* $\tau = 0$*. Boundary conditions are incorporated into the definition of the fluxes* $\widehat{f}$ *and* $\widehat{f_v}$*, i.e., the definition of* $\widehat{f}$ *and* $\widehat{f_v}$ *is altered on* $\Gamma \backslash \Gamma_0$*. This is done in such a way that the method is* adjoint consistent*. For the ease of presentation, we neglect the details which can be found in, e.g., [19, 28]. The fluxes are such that* w_h^+ *and* w_h^- *(and* σ_h^+ *and* σ_h^-*, respectively), are not directly coupled. This allows for a static condensation, so that the only globally coupled degrees of freedom are those associated to* λ_h*. This is in general a reduction of degrees of freedom in comparison to traditional DG methods.*

For the ease of presentation, we rewrite (11)-(13) as

$$\mathcal{T}((w_h)_t, \varphi_h) + N(\boldsymbol{w}_h; \boldsymbol{x}_h) = 0 \quad \forall \boldsymbol{x}_h \in \mathbb{X}_h, \tag{14}$$

where $\boldsymbol{x}_h := (\tau_h, \varphi_h, \mu_h)$ is just a convenient shortcut for the test functions. $\mathcal{T}$ denotes the vector having 0 entries for the first and the last equation, i.e.,

$$\mathcal{T}((w_h)_t, \varphi_h) := (0, ((w_h)_t, \varphi_h), 0)^T,$$

and $N(\boldsymbol{w}_h; \boldsymbol{x}_h)$ is the remaining part belonging to the discretization of the stationary convection-diffusion equation.

Remark 3. *A straightforward method of lines approach can only be applied if* $\mathcal{T}$ *does not have the zero entries in the first and the last argument. At least in principle, one could derive equations for* σ_t *and* λ_t *and try to incorporate them. However, this would require a large amount of derivatives which will most likely deteriorate the order of the scheme. To this end, we will in the sequel rely on* implicit *time discretization methods and Jameson's idea of dual time-stepping [23].*

3.2. (Embedded) DIRK discretization

In this section, we explain how time is discretized in our setting, so that in the end we get a fully discrete algorithm. In an earlier publication [24], we have used backward difference formulae for the discretization of the temporal part. These methods seemed particularly suited

to the method at hand, and showed very good stability and accuracy results. However, there are a couple of drawbacks, one of them being the expensive initial step(s) to obtain suitable start values. Another drawback is the complicated incorporation of temporal adaptivity. In this publication, inspired by the recent work of Nguyen and Peraire [16], we use diagonally implicit Runge-Kutta (DIRK) methods and, more specifically, *embedded* DIRK methods to achieve an adaptive temporal discretization of (potentially) high order.

We consider an adaptive sequence of time instances $0 = t^0 < t^1 < \ldots < t^M = T$, where both $\Delta t^n := t^{n+1} - t^n$ and M depend on the solution to be approximated. We define $\boldsymbol{w}_h^n$ to be the approximation of $\boldsymbol{w}_h(\cdot, t^n)$ at time instance t^n.

We use standard embedded DIRK schemes present in the literature [25, 26, 27]:

Definition 4. *An embedded DIRK method is given by its Butcher tableau with a lower triangular matrix $A \in \mathbb{R}^{k\times k}$, a node vector $\beta \in \mathbb{R}^k$ and two weighting vectors $\gamma_1, \gamma_2 \in \mathbb{R}^k$. Frequently, the Butcher tableau is given as*

$$\begin{array}{c|cccc} \beta_1 & A_{11} & & & \\ \beta_2 & A_{21} & A_{22} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \beta_k & A_{k1} & \ldots & \ldots & A_{kk} \\ \hline & \gamma_{11} & \ldots & \ldots & \gamma_{1k} \\ \hline & \gamma_{21} & \ldots & \ldots & \gamma_{2k} \end{array} . \tag{15}$$

Remark 4. *Applying an embedded DIRK method to an ordinary differential equation*

$$y'(t) = f(t, y(t))$$

amounts to approximating two values y_1^{n+1} and y_2^{n+1}. Those two values are given by

$$y_1^{n+1} = y^n + \Delta t^n \sum_{i=1}^{k} \gamma_{1i} f(t^n + \beta_i \Delta t^n, y^{n,i}) \tag{16}$$

$$y_2^{n+1} = y^n + \Delta t^n \sum_{i=1}^{k} \gamma_{2i} f(t^n + \beta_i \Delta t^n, y^{n,i}) \tag{17}$$

with the intermediate values $y^{n,i}$ implicitly given by

$$y^{n,i} = y^n + \Delta t^n \sum_{j=1}^{i} A_{ij} f(t^n + \beta_j \Delta t^n, y^{n,j}). \tag{18}$$

Note that each step of computing the $y^{n,i}$ is basically an implicit Euler step.

Remark 5. *Note that the two approximations to $y(t^{n+1})$ are supposed to have different orders of accuracy. More precisely, we have chosen to enumerate in such a way that y_1^{n+1} is the more accurate approximation. This means that $\|y_1^{n+1} - y_2^{n+1}\|$ can serve as a measure of consistency error. Furthermore, we choose our schemes in such a way that the DIRK method corresponding to γ_1 is both A- and L-stable. Usually, γ_{2k} is zero, so that the embedded method has actually only $k - 1$ stages.*

It is straightforward to apply the DIRK method to fully discretize (14): As in Rem. 4, $\boldsymbol{w}_h(\cdot, t^{n+1})$ is approximated by two different values $\boldsymbol{w}_{h,1}^{n+1}$ and $\boldsymbol{w}_{h,2}^{n+1}$. The approximations are such that

$$\mathcal{T}\left(w_{h,1}^{n+1} - w_h^n, \varphi_h\right) + \Delta t^n \sum_{i=1}^{k} \gamma_{1i} N(\boldsymbol{w}_h^{n,i}; \boldsymbol{x}_h) = 0 \quad \forall \boldsymbol{x}_h \in \mathbb{X}_h \tag{19}$$

$$\mathcal{T}\left(w_{h,2}^{n+1} - w_h^n, \varphi_h\right) + \Delta t^n \sum_{i=1}^{k} \gamma_{2i} N(\boldsymbol{w}_h^{n,i}; \boldsymbol{x}_h) = 0 \quad \forall \boldsymbol{x}_h \in \mathbb{X}_h\,. \tag{20}$$

The intermediate stages $\boldsymbol{w}_h^{n,i}$ are defined via the equation

$$\mathcal{T}\left(w_h^{n,i} - w_h^n, \varphi_h\right) + \Delta t^n \sum_{j=1}^{i} A_{ij} N(\boldsymbol{w}_h^{n,j}; \boldsymbol{x}_h) = 0 \quad \forall \boldsymbol{x}_h \in \mathbb{X}_h\,. \tag{21}$$

Remark 6. *Note that the computation of $\boldsymbol{w}_h^{n,i}$, see* (21)*, reduces to solving a stationary system, and thus fits very nicely into the framework of our steady-state solver [19]. Note furthermore that* (19)-(20) *is an explicit step (up to the inversion of a mass matrix) and corresponds to* (16)-(17)*.*

Remark 7. *In our numerical computations, we use three different embedded DIRK schemes, one taken from the book by Hairer and Wanner [25], one taken from Al-Rabeh [26] and one taken from Cash [27]. The corresponding tableaus are given in Appendix A. The design orders are $q = 3$ for Cash's method and $q = 4$ for the other two methods.*

As we are using an embedded DIRK scheme, it is our desire to adaptively control the timestep Δt^n. To this end, we define the following error estimation based on the quantities $w_{h,1}^{n+1}$ and $w_{h,2}^{n+1}$:

$$e_h^n := \|w_{h,1}^{n+1} - w_{h,2}^{n+1}\|_{L^2}.$$

Note that the use of a non-bold w is not a typo, we only use the second component of $\boldsymbol{w}_{h,i}^{n+1}$ which represents the solution within the elements, and is probably the most important quantity. As is customary in the use of embedded Runge-Kutta methods, a timestep is rejected (i.e., repeated with half the timestep size) if

$$e_h^n > \Delta t^n \cdot \mathrm{tol}$$

for a user-defined tolerance tol. This approach guarantees that $\sum_{i=0}^{M-1} e_h^n < T \cdot \mathrm{tol}$. If the timestep is accepted,we take $\boldsymbol{w}_{h,1}^{n+1}$ to be the new approximate value $\boldsymbol{w}_h^{n+1}$ and compute the new timestep, based on the old timestep, as

$$\Delta t^{n+1} = \alpha \Delta t^n \left(r_h^n\right)^{-\frac{1}{q}}.$$

α is a safety factor given by

$$\alpha = 0.9 \frac{2n_{it,max} + 1}{2n_{it,max} + n_{it}},$$

where we take the maximum number of Newton steps per stage n_{it}, and the maximum allowable number of Newton steps $n_{it,max}$ in our nonlinear solver into account. r_h^n is defined by

$$r_h^n := \frac{e_h^n}{\text{tol}\,\Delta t^n}. \tag{22}$$

The underlying paradigm is that r_h^n is close to one, because if $e_h^{n+1} \approx e_h^n$, this allows the largest possible timestep such that e_h^{n+1} is not too big.

For a detailed derivation, see standard textbooks such as [25]. q denotes the design accuracy of the DIRK scheme. As usual, Δt^n is 'limited' such that it does not exceed a maximum and a minimum value.

Remark 8. *We control the higher-order Runge-Kutta method with the lower-order one. Strictly speaking, it should be the other way around. Nevertheless, it has been done in literature as it is desirable to keep the higher-order approximation, and we do it for the same reason.*

4. Numerical Results

In this section, we show numerical results obtained with our method. We start from a simple, convection-diffusion problem to test the accuracy of our method and also the performance of the timestep control. Then, we show results for both Euler and Navier-Stokes equations.

4.1. Scalar Convection-Diffusion Equation (Rotating Gaussian)

As a scalar convection-diffusion equation, we present a test case that has previously been investigated by Nguyen et al. [13]. The problem is both scalar and linear, with convective and viscous flux vector, respectively, given as

$$f(w) = (-4y, 4x)^T w, \quad f_v(w, \nabla w) = 0.1\nabla w.$$

The source term g is set to zero, and we consider the domain $\Omega = [-0.5, 0.5]^2$. The final time T is defined as $T = \frac{\pi}{4}$. Note that this test case is very interesting as, in the vicinity of the origin, the problem is diffusion-dominated, while, away from the origin, it is convection-dominated. Initial data are given by a (scaled) Gaussian distribution, i.e.,

$$w(x, y, 0) := e^{-100(x^2+y^2)}. \tag{23}$$

An exact solution to this problem is known, and on $\partial\Omega$, we impose Dirichlet boundary conditions that we choose to be this exact solution. In the numerical results, we use the L^2- norm of $w - w_h$ at final time T as a measure of error.

In Fig. 1, we demonstrate that the fully discrete scheme (without time-step control) converges under both uniform spatial and temporal refinement. The design accuracy of Cash's DIRK scheme is q = 3, while the design accuracy of the other two DIRK schemes is q = 4. The convergence order to be expected is thus $\min(q, p + 1)$. For piecewise quadratic polynomials as ansatz functions, i.e., for $p = 2$, one can thus expect third order of convergence, while for piecewise cubic polynomials, i.e., for $p = 3$, one can expect third and fourth order of convergence, respectively. Numerical results confirm this expectation.

The next test is about the timestep adaptation per se. We take a fixed spatial mesh consisting of 512 elements and cubic ansatz functions, and only refine in time. For $\Delta t \to 0$, this will yield

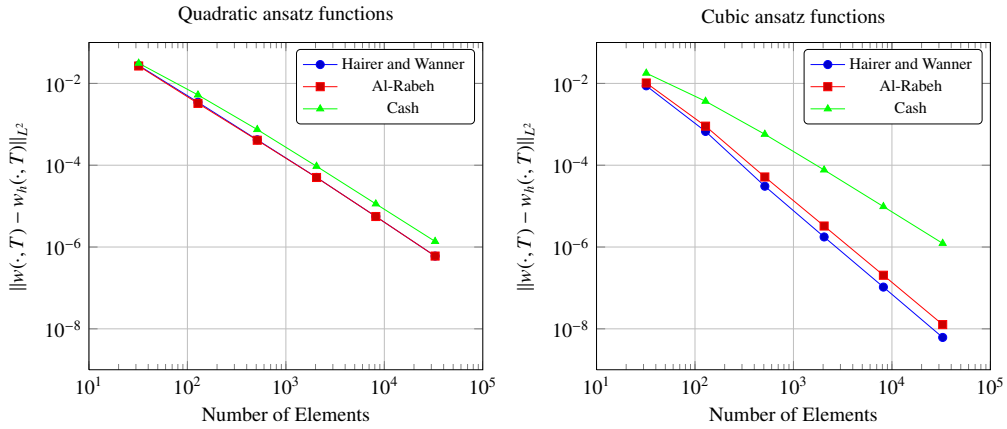
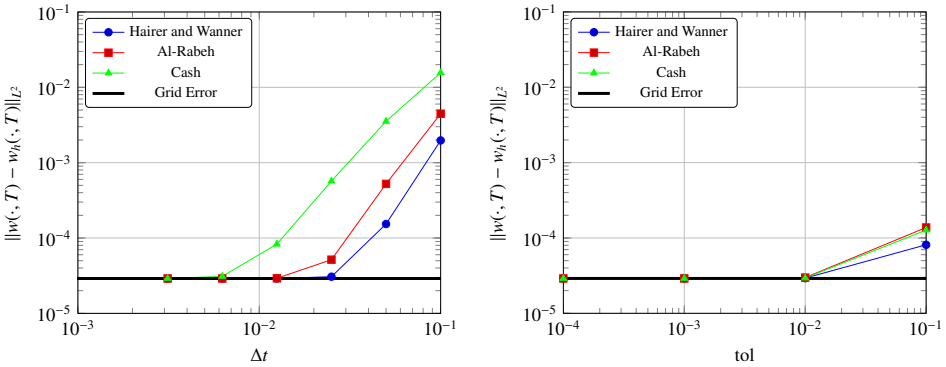


Figure 1: Rotating Gaussian: Convergence of $\|w(\cdot,T) - w_h(\cdot,T)\|_{L^2}$ under uniform spatial and temporal refinement.

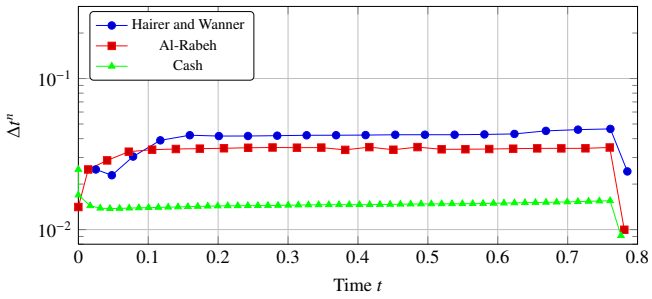
the spatial error. In Fig. 2(a), we plot the error evolution for a fixed timestep Δt, while in Fig. 2(b), we plot the error versus different tolerances for all three DIRK methods. One can clearly see that all the methods are able to obtain the spatial error with only a moderate degree of tolerance. The quasi-optimal timestep is determined automatically. Furthermore, it can be observed that the method by Hairer and Wanner has the best convergence properties, both uniform and adaptive. In Figs. 3(a)-3(b), we plot the evolution of the timestep for tol $= 10^{-1}$ and tol $= 10^{-2}$. The test case under consideration is actually quite homogeneous in its temporal behavior. As a consequence, one can see that after an initial increase, Δt^n remains nearly constant. Also here, Hairer and Wanner's method performs best, as it yields the largest timestep without sacrificing accuracy. The kink which can can be observed in the plots of the timestep size at the right is not an artifact, but in fact the result of a fixed final time T which has to be reached by the last timestep.



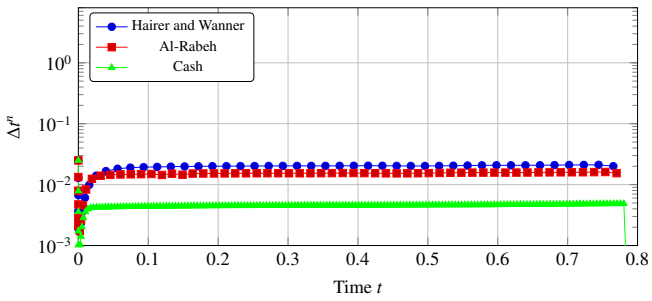
(a) Uniform temporal refinement

(b) Adaptive temporal refinement

Figure 2: Rotating Gaussian: Convergence of $\|w(\cdot,T) - w_h(\cdot,T)\|_{L^2}$ for a fixed mesh with $N = 512$ elements and cubic ansatz functions.



(a) tol = 10^{-1}



(b) tol = 10^{-2}

Figure 3: Rotating Gaussian: Timestep size for a fixed mesh with $N = 512$ elements and cubic ansatz functions for different values of tol

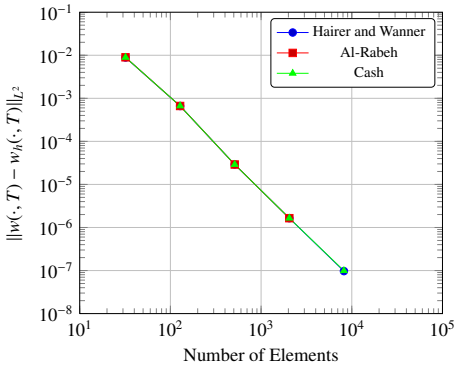


Figure 4: Rotating Gaussian: Convergence of $\|w(\cdot,T)-w_h(\cdot,T)\|_{L^2}$ for cubic polynomials under both mesh and tolerance refinement.

Using a fixed tolerance tol is obviously not enough if one considers mesh refinement. With an increasing spatial resolution, the tolerance should decrease. We choose to set tol $= O(h^{\min(q,p+1)})$. We start on an initial mesh with $N = 32$ elements and an initial tolerance tol $= 10^{-1}$. Convergence results can be seen in Fig. 4. The surprising outcome of this is that the error values nearly lie on top of each other. This means that the timestep adaptation really performs well, as it obviously minimizes the temporal error. Comparing with Fig. 1, one can see that this adaptive approach performs better than even the uniform refinement.

4.2. Euler equations (Radial Expansion Wave)

The next test case has been proposed for both first and second high order workshop [20]. It is to compute a spherically symmetric, inviscid flow using the Euler equations on domain $\Omega = [-4,4]^2$. For the standard choice of $\gamma = 1.4$, the flow ceases to be smooth but its derivative becomes discontinuous. To this end, it has been proposed to use $\gamma = 3$, which will be our choice in the sequel. The final time T is set to $T = 2$. The flow is supersonic throughout the domain, so it is fully specified by its initial conditions (for simplicity, $r \equiv r(x,y) := \sqrt{x^2+y^2}$ denotes radius):

$$q(r) := \begin{cases} 0, & 0 \le r < \frac{1}{2} \\ \frac{1}{\gamma}\left(1+\tanh\left(\frac{r-1}{0.25-(r-1)^2}\right)\right), & \frac{1}{2} \le r < \frac{3}{2} \\ \frac{2}{\gamma}, & r \ge \frac{3}{2} \end{cases}, \tag{24}$$

$$u(x,y,0) := \frac{x}{r}q(r), \qquad v(x,y,0) := \frac{y}{r}q(r), \tag{25}$$

$$\rho(x,y,0) := \gamma\left(1-\frac{\gamma-1}{2}q(r)\right)^{\frac{2}{\gamma-1}}, \qquad p(x,y,0) := \frac{\rho(x,y,0)\left(1-\frac{\gamma-1}{2}q(r)\right)^2}{\gamma}. \tag{26}$$

See Fig. 5 for a picture of initial and final density. As the flow remains smooth, at least for $\gamma = 3$, the entropy

$$s := \ln\left(\frac{p}{\rho^\gamma}\right) \tag{27}$$

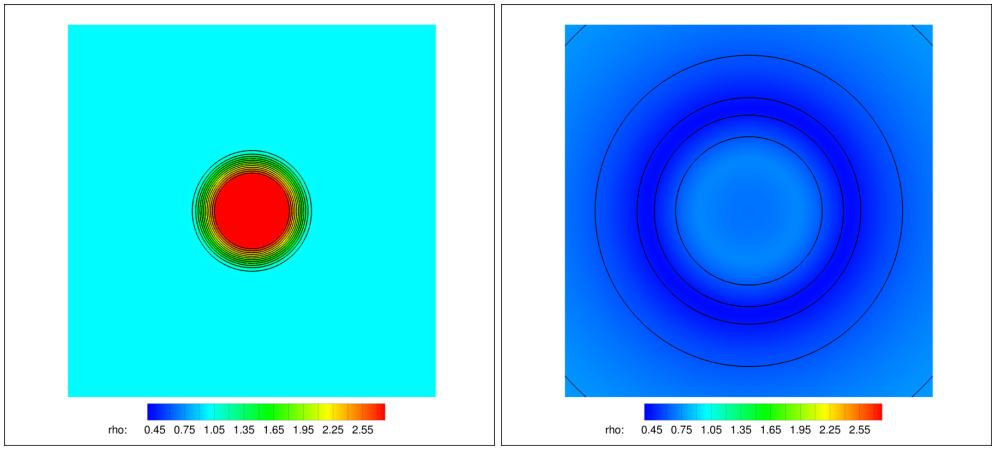


Figure 5: Density of radial expansion wave at time $t = 0$ and $t = 2$, respectively.

should remain constant. This constant is denoted by s_0.

It is expected for this test case to monitor the L^2-norm of the entropy error. We perform this exercise for two different (uniform) grids, one having 2048 elements, and the other having 8192 elements, and for quadratic and cubic ansatz functions. Entropy is monitored for all the adaptive DIRK methods available and compared against a BDF2 and BDF3 scheme, respectively, see Figs. A.9-A.12. Tolerance tol was chosen to be tol = 0.5 for quadratics and 2048 elements, and tol = 10^{-1} for the other computations. It can be clearly seen that the adaptive DIRK methods perform as well as the BDF schemes, except for the last test case, where only the Runge-Kutta method by Hairer and Wanner performs as good as BDF3. Nevertheless, the deviation of Cash's and Al-Rabeh's method from BDF3 is not too extreme. What can again be observed is the fact that the adaptive methods choose an appropriate timestep that is an order of magnitude larger than the one corresponding to the BDF schemes. (Note that the timestep size that corresponds to BDF is chosen in such a way that temporal resolution has minimal effect on the accuracy of the entropy. This is one of the requirements from the high order workshop.) Furthermore, for $t \to \infty$, the flow gets more and more trivial. This is, for all the methods, reflected in the timestep size that is increasing. The most expensive part of an implicit time integration method is the Newton steps. For this reason, we document the cumulated number of Newton iterations over time, including the rejected steps, see Figs. A.9-A.12. Even in this not so long-term run, it can be seen that there is a clear advantage of the adaptive methods, especially for the methods by Hairer and Wanner and Al-Rabeh. Furthermore, the curves associated to the adaptive schemes have a much smaller slope than the curve associated to the BDF methods. For long-time runs, this constitutes a clear advantage.

4.3. Navier-Stokes equations (Von Kármán vortex street)

The last numerical test case computes a von Kármán vortex street. It is well-known that for Re > 50, flow around a circular cylinder gets unstable and the process of vortex shedding begins. We choose free stream Mach *Ma* and Reynolds number Re, respectively, as Ma = 0.2 and Re = 180. The employed mesh consists of 2916 elements and extends to 20 diameters away from

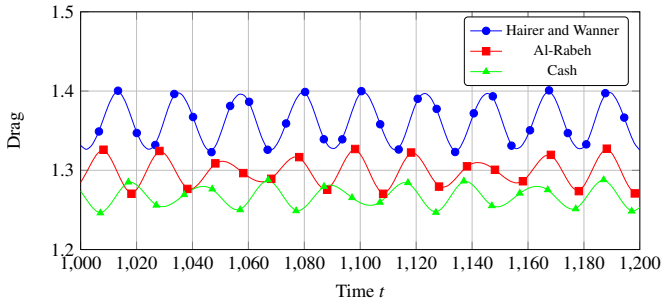


Figure 6: Von Kármán vortex street: Evolution of mean drag coefficient

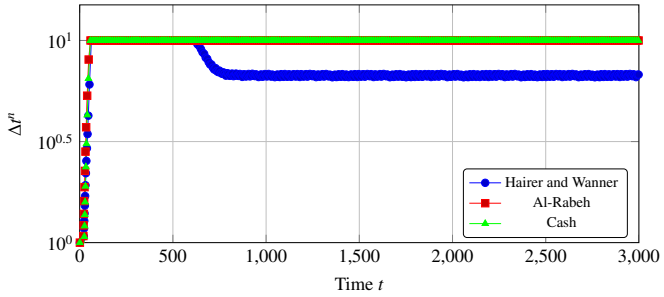


Figure 7: Von Kármán vortex street: Timestep evolution

the cylinder. We used the same mesh in our earlier work, see [24]. Computations are performed with cubic ansatz functions, and a tolerance of tol = 10^{-1}. Timestep Δt^n is limited to 10, because otherwise, the physics of the flow are not correctly captured. There is vast literature on this test case. For the free stream values as indicated one can, e.g., compute mean drag coefficients and Strouhal numbers. Reference values have been reported in literature, see Tbl. 1. Next to these reference values, we have tabulated our computational values. Furthermore, in Fig. 6, we have plotted the drag evolution. Both from the table and the figure, it can be seen that Hairer and Wanner's method performs best. Furthermore, it can be seen from Fig. 7 that this method has the smallest Δt^n. In combination, this indicates that the error indicator performs best. Fig. 8 shows a Mach number plot for this test case, computed with Hairer and Wanner's method.

Experiment	c_D	Sr
Gopinath [29]	1.3406	0.1866
Henderson [30]	1.336	-
Williamson [31]	-	0.1919

Time Discretization	c_D	Sr
Hairer and Wanner	1.3621	0.1920
Al-Rabeh	1.2983	0.1865
Cash	1.2669	0.1766

Table 1: Mean drag coefficients and Strouhal numbers from literature and computations

5. Conclusions and Outlook

In this publication, we have developed a combination of a hybridized discontinuous Galerkin method and an embedded diagonally implicit Runge-Kutta method. We have shown numerical

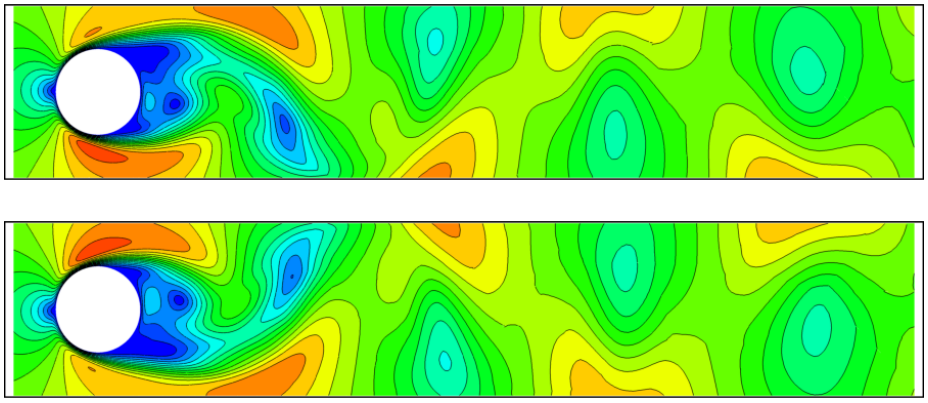


Figure 8: Von Kármán vortex street: Mach number at two instances.

results that demonstrate how different Runge-Kutta methods perform. It seems that the method taken from the book by Hairer and Wanner [25] performs best, whereas the method by Cash [27] often overestimates the error and therefore predicts an unnecessarily small timestep.

Adaptation in the temporal domain alone is obviously not enough to get the full potential of a method. Spatial adaptation is already available in the solver [32], and temporal adaptation has been investigated in this publication. Future work should therefore couple both ingredients in a clever way to achieve maximum efficiency. One idea is to use an adjoint-based error indicator in both space and time to optimally design the spatial mesh. Adaptation in time can then be performed by a mixture of the adjoint and the DIRK timestep prediction.

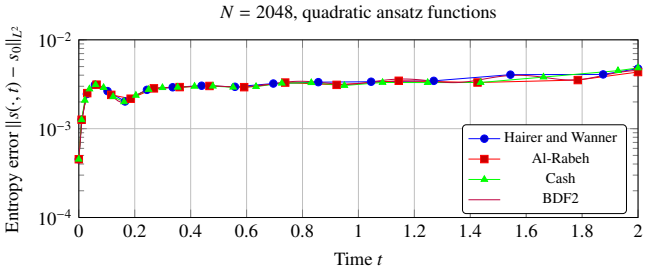
Appendix A. Embedded DIRK schemes

In this short appendix, we have listed the embedded DIRK schemes that we employ in our numerical results section: The first tableau can be found in the classical book by Hairer and Wanner [25], its design order of accuracy is 4 and 3, respectively.

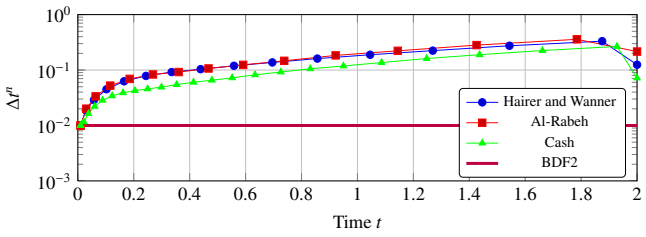
$$\begin{array}{c|ccccc} \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{11}{20} & \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} & 0 \\ 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & \frac{-85}{12} & \frac{1}{4} \\ \hline \gamma_1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\ \gamma_2 & \frac{59}{48} & -\frac{17}{96} & \frac{225}{32} & -\frac{85}{12} & 0 \end{array} \tag{A.1}$$

The second tableau is due to Al-Rabeh [26], its design order of accuracy is 4 and 3, respectively.

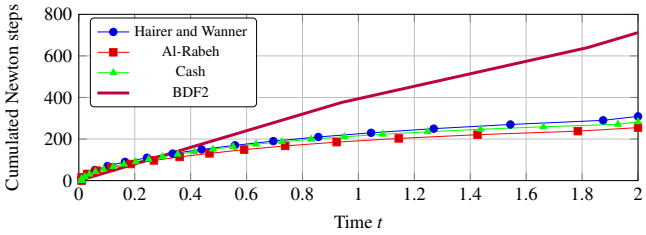
$$\begin{array}{c|cccc} 0.4358665 & 0.4358665 & 0 & 0 & 0 \\ 0.0323722 & -0.4034943 & 0.4358665 & 0 & 0 \\ 0.9676278 & -0.3298751 & 0.8616364 & 0.4358665 & 0 \\ 0.5641335 & 0.5575315 & -0.1930865 & -0.2361781 & 0.4358665 \\ \hline \gamma_1 & 0.3153914 & 0.1846086 & 0.1846086 & 0.3153914 \\ \gamma_2 & 0.6307827 & 0.1413538 & 0.2278634 & 0 \end{array} \tag{A.2}$$



(a) Evolution of entropy error



(b) Timestep evolution



(c) Cumulated Newton steps

Figure A.9: Radial Expansion Wave: 2048 elements and quadratic ansatz functions.

The last tableau we use is due to Cash [27], its design order of accuracy is 3 and 2, respectively.

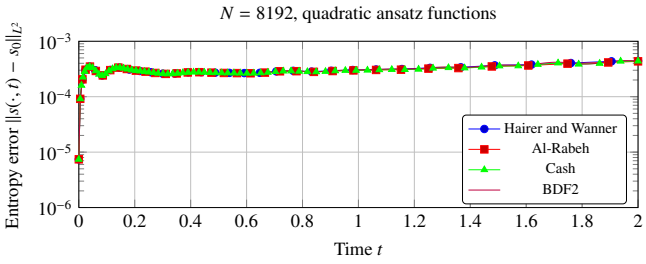
$$
\begin{array}{c|ccc}
0.435866521508 & 0.435866521508 & 0 & 0 \\
0.717933260755 & 0.2820667320 & 0.435866521508 & 0 \\
1.0 & 1.208496649 & -0.6443632015 & 0.435866521508 \\
\hline
\gamma_1 & 1.208496649 & -0.6443632015 & 0.435866521508 \\
\gamma_2 & 0.77263013745746 & 0.22736986254254 & 0.0
\end{array}
\tag{A.3}
$$

References

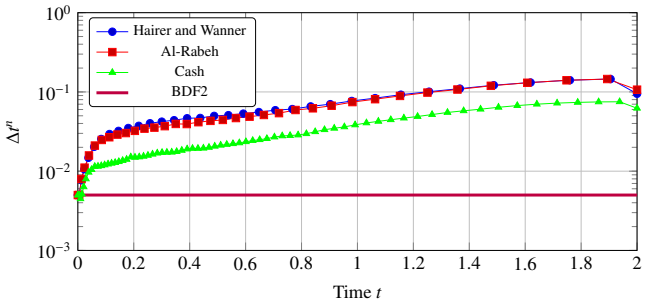
[1] Z. J. Wang, L. Zhang, Y. Liu., High-order spectral volume method for 2D Euler equations, AIAA Paper 03-3534 (2003).

[2] H. T. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, AIAA Paper 09-403 (2009).

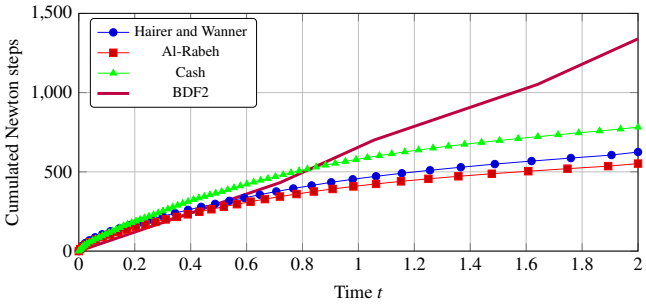
[3] R. Hartmann, P. Houston, Adaptive Discontinuous Galerkin Finite Element methods for the compressible Euler Equations, Journal of Computational Physics 183 (2002) 508–532.



(a) Evolution of entropy error

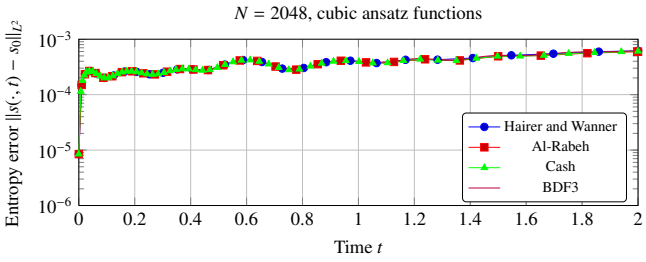


(b) Timestep evolution

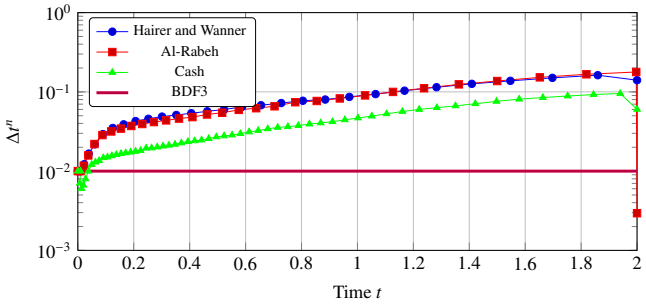


(c) Cumulated Newton steps

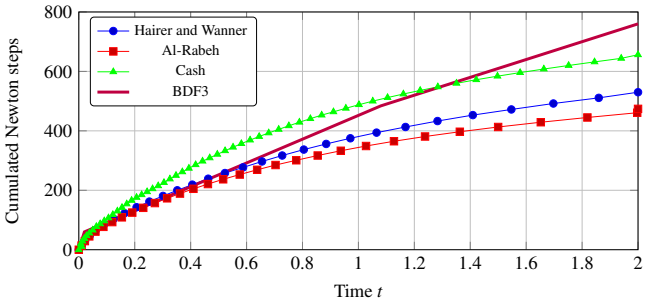
Figure A.10: Radial Expansion Wave: 8192 elements and quadratic ansatz functions.



(a) Evolution of entropy error

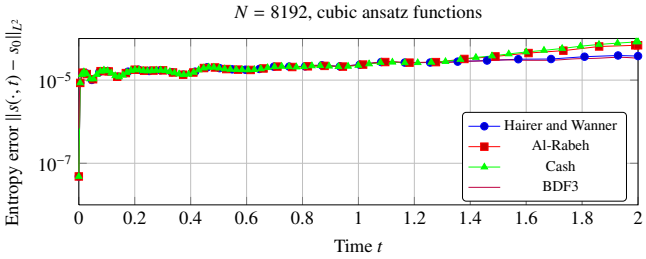


(b) Timestep evolution

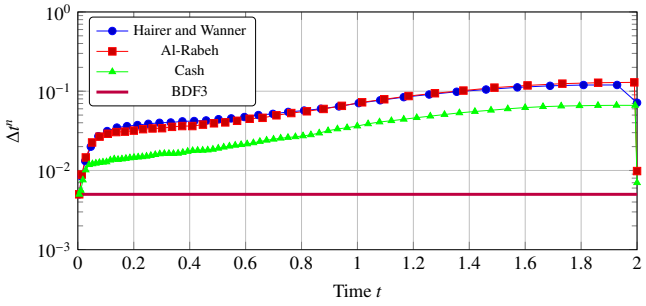


(c) Cumulated Newton steps

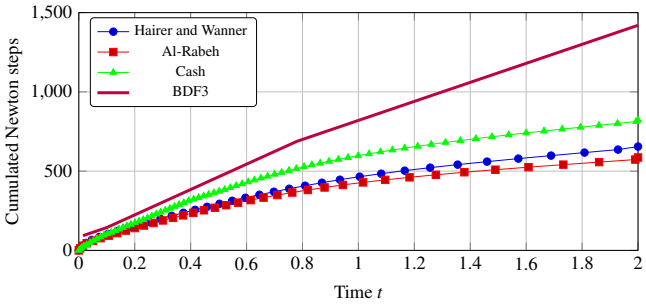
Figure A.11: Radial Expansion Wave: 2048 elements and cubic ansatz functions.



(a) Evolution of entropy error



(b) Timestep evolution



(c) Cumulated Newton steps

Figure A.12: Radial Expansion Wave: 8192 elements and cubic ansatz functions.

[4] W. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Tech. rep., Los Alamos Scientic Laboratory (1973).
[5] F. Bassi, S. Rebay, A high-order accurate discontinuous Finite-Element method for the numerical solution of the compressible Navier-Stokes equations, Journal of Computational Physics 131 (1997) 267–279.
[6] B. Cockburn, C.-W. Shu, The Runge-Kutta local projection p^1-Discontinuous Galerkin Finite Element Method for Scalar Conservation Laws, RAIRO - Modélisation mathématique et analyse numérique 25 (1991) 337–361.
[7] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for Conservation Laws II: General framework, Mathematics of Computation 52 (1988) 411–435.
[8] B. Cockburn, S. Y. Lin, TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for Conservation Laws III: One dimensional systems, Journal of Computational Physics 84 (1989) 90–113.
[9] B. Cockburn, S. Hou, C.-W. Shu, The Runge–Kutta local projection Discontinuous Galerkin finite element method for Conservation Laws IV: The multidimensional case, Mathematics of Computation 54 (1990) 545–581.
[10] B. Cockburn, C.-W. Shu, The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems, Mathematics of Computation 141 (1998) 199–224.
[11] C. Baumann, J. Oden, An adaptive-order Discontinuous Galerkin method for the solution of the Euler equations of gas dynamics, International Journal for Numerical Methods in Engineering 47 (2000) 61–73.
[12] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of Discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.
[13] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous galerkin method for linear convection-diffusion equations, Journal of Computational Physics 228 (2009) 3232–3254.
[14] 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 (2009) 8841–8855.
[15] J. Peraire, N. C. Nguyen, B. Cockburn, A hybridizable Discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations, AIAA Paper 10-363 (2010).
[16] N. C. Nguyen, J. Peraire, Hybridizable discontinuous galerkin methods for partial differential equations in continuum mechanics, Journal of Computational Physics 231 (2012) 5955–5988.
[17] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of Discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2009) 1319–1365.
[18] H. Egger, J. Schöberl, A hybrid mixed Discontinuous Galerkin Finite Element method for convection-diffusion problems, IMA Journal of Numerical Analysis 30 (2010) 1206–1234.
[19] J. Schütz, G. May, A Hybrid Mixed Method for the Compressible Navier-Stokes Equations, Journal of Computational Physics 240 (2013) 58–75.
[20] Z.J.Wang, K.Fidkowski, R.Abgrall, F.Bassi, D.Caraeni, A.Cary, H.Deconinck, R.Hartmann, K.Hillewaert, H.T.Huynh, N.Kroll, G.May, P.Persson, B. Leer, M.Visbal, High-order cfd methods: Current status and perspective, International Journal for Numerical Methods in Fluids 72 (2013) 811–845.
[21] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[22] L. Ferracina, M. N. Spijker, An extension and analysis of the Shu-Osher representation of Runge-Kutta methods, Mathematics of Computation 74 (2004) 201–219.
[23] A. Jameson, Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, AIAA Paper 91-1596 (1991).
[24] J. Schütz, M. Woopen, G. May, A combined hybridized discontinuous Galerkin / hybrid mixed method for viscous conservation laws, Tech. rep., AICES (2012).
[25] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer Series in Computational Mathematics, 1991.
[26] A. H. Al-Rabeh, Embedded dirk methods for the numerical integration of stiff systems of odes, International Journal of Computer Mathematics 21 (1987) 65–84.
[27] J. Cash, Diagonally implicit runge-kutta formulae with error estimates, J. Inst. Maths Applics 24 (1979) 293–301.
[28] J. Schütz, G. May, An adjoint consistency analysis for a class of hybrid mixed methods, Tech. rep., IGPM, accepted by IMA Journal of Numerical Analysis in 2013 (2012).
[29] A. Gopinath, A. Jameson, Application of the time spectral method to periodic unsteady vortex sheeding, AIAA Paper 06-0449 (2006).
[30] R. D. Henderson, Details of the drag curve near the onset of vortex shedding, Physics of Fluids 7 (1995) 2102–2104.
[31] C. Williamson, Vortex dynamics in the cylinder wake, Annual review of fluid mechanics 28 (1996) 477–539.
[32] A. Balan, M. Woopen, G. May, Adjoint-based hp-adaptation for a class of high-order hybridized finite element schemes for compressible flows, AIAA Paper 13-2938 (2013).