
# General linear methods for time-dependent PDEs

*Alexander Jaust and Jochen Schütz*

UHasselt Computational Mathematics Preprint
Nr. UP-17-01

January 19, 2017

# General linear methods for time-dependent PDEs

Alexander Jaust*[1] and Jochen Schütz†[1]

[1] *Faculty of Sciences, Hasselt University, Agoralaan Gebouw D, BE-3590 Diepenbeek*

The hybridized discontinuous Galerkin method has been successfully applied to time-dependent problems using implicit time integrators. These integrators stem from the 'classical' class of backward differentiation formulae (BDF) and diagonally implicit Runge-Kutta (DIRK) methods. We extend this to the class of general linear methods (GLM) that unify multistep and multistage methods into one framework. We focus on diagonally implicit multistage integration methods (DIMSIM) that can have the same desirable stability properties like DIRK methods while also having high stage order. The presented numerical results confirm that the applied DIMSIMs achieve expected approximation properties for linear and nonlinear problems.

## 1 Introduction

Discontinuous Galerkin methods have been recognized as powerful discretization methods for differential equations stemming from a variety of applications [1, 2, 14, 16, 17, 19, 20, 31, 32]. A severe drawback of these methods is the large numer of unknowns compared to other numerical schemes. This becomes particularly problematic for steady-state problems or stiff time-dependent problems, because there it is common to use implicit solution techniques that couple the unknowns globally.

The number of globally coupled unknowns may be greatly reduced by hybridization [13]. This leads to the class of so called hybridized discontinuous Galerkin methods (HDG) [3, 26–29, 33] which has shown to be competitive with other discretization schemes [25, 41]. HDG has initially been developed for steady-state problems, it can also be applied to time-dependent problems though, yielding a differential algebraic equation (DAE). The latter needs

*Electronic address: alexander.jaust@uhasselt.be
†Electronic address: jochen.schuetz@uhasselt.be

time integration schemes able to handle stiff problems. Good results have been obtained with backward differentiation formulae (BDF) and diagonally implicit Runge-Kutta schemes (DIRK) [23, 26, 29]; an approach using multiderivative time integrators has been studied in [24].

In this work we study the coupling of an HDG method to a diagonally implicit multistage integration method (DIMSIM) in Nordsieck representation [4, 7]. These methods are a subclass of general linear methods (GLM) [5, 6, 10, 12, 21]. GLMs are a generalization of multistep and multistage methods and therefore contain these schemes as special cases. Their advantage over classical time integrators such as BDF and DIRK methods is that one can obtain methods with high accuracy that are $L$- and $A$-stable while having stage order $q > 1$, being particularly important for very stiff problems. Moreover, there are techniques available to adapt the time step size and the order of the scheme [8, 21, 22] and an extension to implicit explicit (IMEX) methods exists [36] as well.

This work is structured as follows: First, we briefly introduce the HDG method and show its coupling to a GLM. Then, numerical results are shown. We end the work with conclusions and outlook.

## 2 Numerical method

We consider partial differential equations of convection-diffusion type that can be written as

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

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

on a domain $\Omega \subset \mathbb{R}^2$. The system consists of $m \geq 1$ equations; the functions $f$ and $f_v$ are given, possibly nonlinear functions. Both Euler and Navier-Stokes equations fall within this framework. As it is frequently done, we reformulate the PDE (1) as a first-order PDE by introducing the additional unknown $\sigma := \nabla w$:

$$\sigma = \nabla w \qquad \forall\ (x,t) \in \Omega \times [0,T] \tag{3}$$

$$w_t + \nabla \cdot (f(w) - f_v(w, \sigma)) = 0 \qquad \forall\ (x,t) \in \Omega \times [0,T] \tag{4}$$

$$w(x,0) = w_0(x) \quad \forall\ x \in \Omega. \tag{5}$$

If the system is first order, i.e. $f_v \equiv 0$, then (3) is not needed and (4) simplifies accordingly.

## 2.1 The hybridized discontinuous Galerkin method

For a proper discretization the domain $\Omega$ has to be partitioned into a set of subdomains such that

$$\Omega = \bigcup_{k=1}^{N} \Omega_k. \tag{6}$$

The number of subdomains is denoted by $N$. For a hybridized discretization we also need the set of all edges $\Gamma$. It contains the edges $\Gamma_k$ of intersecting subdomains $\Omega_k \cap \Omega_{k'}$ and subdomains intersecting the domain boundary $\Omega_k \cap \partial\Omega$. The number of all edges is given by $\widehat{N} = |\Gamma|$. Furthermore, we need necessary spaces for the approximations of $w$, $\sigma$ and the additional hybrid unknown $\lambda$ on the edges. The following standard spaces are considered:

$$H_h := \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^P(\Omega_k)\ \forall k = 1, \dots, N\}^{2m} \tag{7}$$

$$V_h := \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^P(\Omega_k)\ \forall k = 1, \dots, N\}^{m} \tag{8}$$

$$M_h := \{f \in L^2(\Omega) \mid f_{|\Gamma_k} \in \Pi^P(\Gamma_k)\ \forall k = 1, \dots, \widehat{N},\ \Gamma_k \in \Gamma\}^{m}. \tag{9}$$

For a shorter notation we also define the following abbreviations for standard scalar products on elements and edges

$$(h_1, h_2) := \sum_{k=1}^{N} \int_{\Omega_k} h_1 \cdot h_2 \, dx,$$

$$\langle h_1, h_2 \rangle_\Gamma := \sum_{k=1}^{\widehat{N}} \int_{\Gamma_k} h_1 \cdot h_2 \, d\sigma, \quad \langle h_1, h_2 \rangle_{\partial\Omega_k} := \sum_{k=1}^{N} \int_{\partial\Omega_k} h_1 \cdot h_2 \, d\sigma.$$

Applying the HDG method in a standard way yields the task of finding $\sigma_h \in H_h$, $w_h \in V_h$ and $\lambda_h \in M_h$ such that

$$(\sigma_h - \nabla w_h, \tau_h) - \langle \lambda_h - w_h^-, \tau_h^- \cdot n \rangle_{\partial\Omega_k} = 0 \quad \forall \tau_h \in H_h \tag{10}$$

$$((w_h)_t, \varphi_h) - (f(w_h) - f_v(w_h, \sigma_h), \nabla\varphi_h) + \langle (\hat{f} - \hat{f}_v) \cdot n, \varphi_h^- \rangle_{\partial\Omega_k} = 0 \quad \forall \varphi_h \in V_h \tag{11}$$

$$\langle [\![ \hat{f} - \hat{f}_v ]\!] \cdot n, \mu_h \rangle_\Gamma = 0 \quad \forall \mu_h \in M_h \tag{12}$$

is fulfilled for all times $t \in [0, T]$. The fluxes on element boundaries $\partial\Omega_k$ have been replaced by numerical fluxes

$$\hat{f} := f(\lambda_h) - \alpha(\lambda_h - w_h^-)n, \quad \hat{f}_v := f_v(\lambda_h, \sigma_h^-) + \beta(\lambda_h, w_h^-)n \tag{13}$$

with positive real parameters $\alpha$ and $\beta$. The parameters have to be chosen carefully to ensure stability of the scheme. For a detailed description on how boundary conditions are incorporated, we refer to further publications [34, 38].

Note that a time-derivative of only $w_h$ occurs in the equation. Therefore, the equations form a set of differential algebraic equations (DAEs).

The number of unknowns in (10)–(12) is larger than for the initial problem where $\lambda_h$ would be absent. However, this formulation allows to apply static condensation such that the global number of unknowns can be greatly reduced [13].

In order to obtain a more compact notation we will abbreviate the set of ansatz and test spaces by

$$\mathbb{X}_h := H_h \times V_h \times M_h \tag{14}$$

and the vector of unknowns by

$$\mathbf{w}_h := (\sigma_h, w_h, \lambda_h). \tag{15}$$

Then, we can write (10)–(12) compactly in an ODE-like way as

$$\mathcal{T}((w_h)_t, \varphi_h) + \mathcal{N}(\mathbf{w}_h; \mathbf{x}_h) = 0, \quad \forall \mathbf{x}_h \in \mathbb{X}_h \tag{16}$$

where $\mathcal{T}$ is the vector containing time derivatives and $\mathcal{N}$ represents the spatial discretization of the problem.

## 2.2 General linear methods

The differential algebraic equation obtained from the HDG discretization requires good stability properties of the applied time integrator. Therefore, mostly schemes that are at least $A(\alpha)$-stable are used. Popular methods that have been used with HDG discretizations are DIRK or BDF methods [23, 26, 29].

In this work we want to discretize (16) using general linear methods [6, 9, 10, 21, 22]. These can be seen as generalization of standard methods like multistage (such as DIRK) or multistep (such as BDF) methods. Multistage methods rely on only $r = 1$ external approximation – the solution at the previous time step – but compute $s \geq 1$ internal approximations during stages. Multistep methods rely on $r \geq 1$ external approximations that are passed from one time step to another, but have only $s = 1$ internal approximation that equals the solution at the new time step. General linear methods allow the usage of several internal approximations $s \geq 1$ and external approximations $r \geq 1$.

In order to give a brief idea of the method we start with an ordinary differential equation (ODE)

$$y'(t) = f(t, y), \quad y(0) = y_0 \tag{17}$$

with unknown $y$, time $t$ and a given initial condition $y_0$. The ODE shall be solved on a uniform grid in time with $t_n := t_0 + n \cdot \Delta t$. An approximation

using a GLM is obtained via

$$Y_i = \sum_{j=1}^{s} a_{ij}\Delta t F_j + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}, \quad i = 1, \dots, s \tag{18}$$

$$y_i^{[n]} = \sum_{j=1}^{s} b_{ij}\Delta t F_j + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]}, \quad i = 1, \dots, r \tag{19}$$

[6]. External approximations are stored in $y_j^{[n-1]}$ for $j = 1, \dots, r$. Additionally, in each time step $s$ internal approximations $Y_i$ $(i = 1, \dots, s)$ are computed. This is similar to Runge-Kutta methods. Once all internal approximations are known the external approximations $y_i^{[n]}$ are updated. The collection of $Y_i$, $F_j$, $y_j^{[n-1]}$ and $y_j^{[n]}$ are often written as single vectors consisting of the data

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_s \end{bmatrix}, \quad y^{[n-1]} = \begin{bmatrix} y_1^{[n-1]} \\ y_2^{[n-1]} \\ \vdots \\ y_r^{[n-1]} \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ \vdots \\ y_r^{[n]} \end{bmatrix}. \tag{20}$$

$F_j := f(Y_j)$ is often referred to as stage derivative because, due to the ODE (17), it describes the derivative of $Y_j$ in stage $j$. The shape of the method depends heavily on the choice of values to be stored in $y^{[n]}$ and $y^{[n-1]}$. This also depends on the method and the way it is represented [10]. Pure multistep methods may store solutions at previous times $y_{n-r}, y_{n-r+1}, \dots, y_{n-1}$, the corresponding derivatives $f(y_{n-r}), f(y_{n-r+1}), \dots, f(y_{n-1})$ or both. Pure multistage methods only need to store the solution of the previous time $y_{n-1}$.

Order and stability of the method depends on the careful choice of real coefficients $a_{ij}$, $u_{ij}$, $b_{ij}$ and $v_{ij}$. The coefficients of the method can be compactly written as matrices

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix} \quad \text{or} \quad \begin{array}{c|c} A & U \\ \hline B & V \end{array}, \quad A \in \mathbb{R}^{s\times s}, B \in \mathbb{R}^{r\times s}, U \in \mathbb{R}^{s\times r}, V \in \mathbb{R}^{r\times r}. \tag{21}$$

In this work we focus on a special class of general linear methods that are also called diagonally implicit multistage integration methods (DIMSIM) [4, 7, 21]. These are closely related to (singly) diagonally implicit Runge-Kutta methods in the sense that

$$A = \begin{bmatrix} \lambda & & & \\ a_{21} & \lambda & & \\ \vdots & \ddots & \ddots & \\ a_{s1} & \dots & a_{s(s-1)} & \lambda \end{bmatrix}$$

is a lower triangular matrix with nonzero entries on the diagonal. This allows to solve a system in each stage instead of getting a much larger system in case

of nonzero entries on the upper triangular part. Furthermore, it is possible to choose the other coefficients in such a way that stability properties are equal to DIRK methods, i.e., $A$- and $L$-stable DIMSIMs are available. We use DIMSIMs of order $p = 1$ to $p = 3$ that were presented in [21]. For the applied DIMSIMs the stage order $q$ is equal to $p$. We will refer to the methods as DIMSIM1, DIMSIM2 and DIMSIM3 to distinguish between the schemes of different order. Each method has $s = p$ internal and $r = p + 1$ external approximations. These DIMSIMs are formulated in Nordsieck formulation [7, 30] which means that $y^{[n]}$ is the Nordsieck vector

$$y^{[n]} = \begin{bmatrix} y(t_n) \\ \Delta t y'(t_n) \\ \Delta t^2 y^{(2)}(t_n) \\ \vdots \\ \Delta t^r y^{(r)}(t_n) \end{bmatrix} \tag{22}$$

that stores the derivatives of $y$. Using the specific form (22) has the advantage that time step adaptation can be easily incorporated since it only requires rescaling of the Nordsieck vector. This has been successfully applied in [8, 11, 21, 22] to ODEs. In this work, we do not pursue this any further, and leave it for future work.

Because it is extremely unhandy to compute higher derivatives of the ODE's right-hand side, in practice, one usually uses an approximation to the Nordsieck vector [7]. In the case of the first order method with $r = 2$ it is self-starting since the Nordsieck vector is given by

$$y^{[0]} = \begin{pmatrix} y_0 \\ \Delta t f(y_0) \end{pmatrix}. \tag{23}$$

Higher order methods different require a different approach. In [40], the author constructed special Runge-Kutta schemes that compute an approximation to the Nordsieck vector at $t = 0$. In [21, 22], the author describes an approach where the higher order DIMSIMs are started from lower order DIMSIMs. In this work, we use an approach similar to the starting procedure of backwards differentiate formulae. We use an already available DIRK scheme of suitable order and compute $r$ equidistant approximations to the solution at times $t_i = i \cdot \Delta t,\ i = 1, \ldots, r - 1$. These values are used together with the given initial data to construct an approximation to the Nordsieck vector using interpolation.

## 2.3 Applying DIMSIMs to the HDG method

In (16) the semi-discrete form of the equations is already in the shape of a DAE. Therefore, we have to solve (18)–(19) with slightly modified notation.

In each stage $i$ of the method we compute an internal approximation by solving

$$\mathcal{T}(w_h^{n,i}, \varphi_h) = -\Delta t \sum_{j=1}^{i} a_{ij} \mathcal{N}(\mathbf{w}_h^{n,i}; \mathbf{x}_h) + \sum_{j=1}^{r} u_{ij} \mathcal{T}(y_j^{[n-1]}, \varphi_h), \quad \forall \mathbf{x}_h \in \mathbb{X}_h. \tag{24}$$

(Note that we have defined $\mathbf{x}_h = (\tau_h, \varphi_h, \mu_h)$.) Once all stage values $w_h^{n,i}$ are known we obtain the updated solution from

$$\mathcal{T}(y_i^{[n]}, \varphi_h) = -\sum_{j=1}^{s} b_{ij} \Delta t \mathcal{N}(\mathbf{w}_h^{n,i}; \mathbf{x}_h) + \sum_{j=1}^{r} v_{ij} \mathcal{T}(y_j^{[n-1]}, \varphi_h) \tag{25}$$

which only requires the local inversion of a mass matrix on each element. Here, $y^{[n-1]}$ stores an approximation to the Nordsieck vector,

$$y^{[n-1]} = \begin{bmatrix} w_h^{n-1} \\ \Delta t \frac{\partial}{\partial t} w_h^{n-1} \\ \vdots \\ \Delta t^r \frac{\partial^r}{\partial t^r} w_h^{n-1} \end{bmatrix} + \mathcal{O}(\Delta t^{p+1}). \tag{26}$$

Note that $p$ is the order of the applied DIMSIM.

# 3 Numerical results

In this section we present numerical results obtained from the HDG discretization with DIMSIM time integrators in order to verify the approach. The first test case is a linear convection-diffusion equation where the exact solution is known. In the second test case, the more involved Navier-Stokes equations are solved and the results are compared to other numerical experiments. The system of equations is solved using Newton's method until the residual drops below $10^{-10}$. The arising linear system is then solved with a restarted GMRES until the relative residual drops below $10^{-12}$ for the first and $10^{-10}$ for the second test problem.

## 3.1 Linear convection-diffusion equation

We first consider a 2D Gaussian that rotates on the domain $\Omega = [-0.5, 0.5]^2$ in counter-clockwise direction. The same problem has been studied for HDG for different time integrators also in [23, 27, 35]. The flux functions are

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

with given diffusion constant $\varepsilon = 10^{-3}$. The computation is run until final time $T = \frac{\pi}{4}$ and Dirichlet boundary conditions are specified everywhere. The solution to this problem is known what allows us to compute the error of our

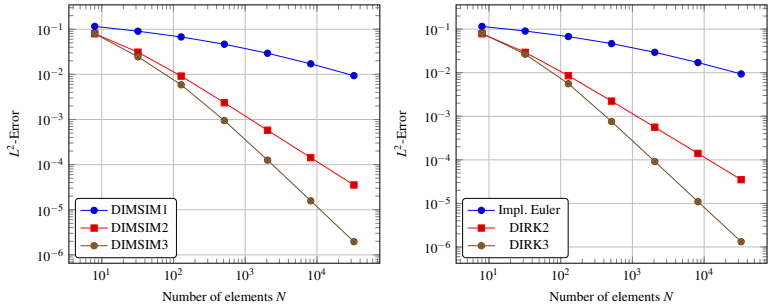


Figure 1: Errors of DIMSIM schemes of order 1 to 3 (left) and Runge-Kutta schemes of same order (right) are presented.

method and check for correct order of convergence. The coarsest grid has $N = 8$ triangular elements and the time step on this grid is $\Delta t = \frac{\pi}{16}$. We use polynomials of degree $P = 2$ for the spatial discretization such that the expect spatial order of convergence is $P + 1 = 3$. This shall not affect the order of convergence in time because we consider methods of order $p = 3$ at most.

In Fig. 1 we present the solution obtained under uniform refinement. On the left we show the solution for DIMSIM time integration and on the right the solution for DIRK time integration with same order of accuracy in time. We observe that the methods retain the correct order of convergence in time. Moreover, the errors produced are almost identical to the ones obtained from the classical DIRK discretizations.

## 3.2 Navier-Stokes equations

As second test case we consider the compressible Navier-Stokes equations in two space dimensions. A description of the fluxes can be found in [33]. We consider the flow around a cylinder at Reynolds number $\mathrm{Re} = 180$ and Mach number $\mathrm{Ma} = 0.2$. At these flow conditions vortices shed from the cylinder what is known as Kármán vortex street.

We compute the solution on a mesh that extends to 20 diameters around the cylinder. The mesh has $N = 2916$ elements and it is the same that has been used in previous publications [23, 39]. We use polynomials of degree $P = 1, 2, 3$. The flow field is initialized with free stream conditions. At simulation time around $t \approx 750$ the vortex street develops. We look at the fully evolved vortex street in the interval $t \in [1,000; 10,000]$ and compute the mean drag coefficient $c_D$ and the Strouhal number Sr. These can be compared to data from the literature given in Tab. 1.

The results from our computations are given in Tab. 2–4. The values obtained for the drag coefficient and the Strouhal number are reasonable compared to data from literature. This is especially true when the time step

Table 1: Values of the Strouhal number Sr and the drag coefficient from the literature.

| | Sr | $c_D$ |
|---|---|---|
| Gopinath [15] | 1.3406 | 0.1866 |
| Henderson [18] | 1.336 | — |
| Williamson [37] | — | 0.1919 |

Table 2: Values of the Strouhal number Sr and the drag coefficient for DIMSIM1.

| $P=1$ | Sr | $c_D$ | $P=2$ | Sr | $c_D$ | $P=3$ | Sr | $c_D$ |
|---|---|---|---|---|---|---|---|---|
| $\Delta t=1$ | 0.1733 | 1.2110 | $\Delta t=1$ | 0.1733 | 1.2164 | $\Delta t=1$ | 0.1733 | 1.2171 |
| $\Delta t=5$ | 0.0000 | 0.9723 | $\Delta t=5$ | 0.1222 | 0.9542 | $\Delta t=5$ | 0.1238 | 0.9492 |
| $\Delta t=10$ | 0.0000 | 0.9723 | $\Delta t=10$ | 0.0000 | 0.9228 | $\Delta t=10$ | 0.0000 | 0.9181 |

Table 3: Values of the Strouhal number Sr and the drag coefficient for DIMSIM2.

| $P=1$ | Sr | $c_D$ | $P=2$ | Sr | $c_D$ | $P=3$ | Sr | $c_D$ |
|---|---|---|---|---|---|---|---|---|
| $\Delta t=1$ | 0.1898 | 1.3455 | $\Delta t=1$ | 0.1898 | 1.3649 | $\Delta t=1$ | 0.1898 | 1.3651 |
| $\Delta t=5$ | 0.1849 | 1.3449 | $\Delta t=5$ | 0.1882 | 1.3714 | $\Delta t=5$ | 0.1882 | 1.3727 |
| $\Delta t=10$ | 0.1733 | 1.3317 | $\Delta t=10$ | 0.1774 | 1.3401 | $\Delta t=10$ | 0.1774 | 1.3405 |

Table 4: Values of the Strouhal number Sr and the drag coefficient for DIMSIM3.

| $P=1$ | Sr | $c_D$ | $P=2$ | Sr | $c_D$ | $P=3$ | Sr | $c_D$ |
|---|---|---|---|---|---|---|---|---|
| $\Delta t=1$ | 0.1898 | 1.3448 | $\Delta t=1$ | 0.1898 | 1.3645 | $\Delta t=1$ | 0.1898 | 1.3649 |
| $\Delta t=5$ | 0.1832 | 1.3216 | $\Delta t=5$ | 0.1882 | 1.3377 | $\Delta t=5$ | 0.1882 | 1.3385 |
| $\Delta t=10$ | 0.1667 | 1.2803 | $\Delta t=10$ | 0.1708 | 1.2840 | $\Delta t=10$ | 0.1708 | 1.2845 |

size $\Delta t$ is decreased. The DIMSIM1 fails to obtain an unsteady solution for low polynomial degrees and/or large time step because it is not accurate enough to catch the time-dependent features of the solution. We have observed similar behavior for time steps $\Delta t > 10$ even for higher order time integrators before [23]. Therefore, it is crucial to choose a discretization in time that is sufficiently accurate. This may be achieved by reducing the time step size $\Delta t$ or by using time integrators of higher accuracy.

## 4 Conclusion and outlook

We have presented the application of general linear methods to an HDG discretization in space. The resulting system of equations is similar to the system one obtains from classical BDF of DIRK methods. The numerical experiments confirm the expected order of convergence in time and the plausibility of the results.

Future work will include the evaluation of the performance of DIMSIM time integrators for HDG schemes. In this setting time step adaption is crucial in order to be competitive to other methods. Another interesting topic is the coupling of implicit explicit (IMEX) general linear methods with the hybridized discontinuous Galerkin methods.

## Acknowledgements

This study was supported by the Special Research Fund (BOF) of Hasselt University. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation – Flanders (FWO) and the Flemish Government – department EWI.

## References

[1] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39:1749–1779, 2002.

[2] F. Bassi and S. Rebay. A high-order accurate discontinuous Finite-Element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[3] T. Bui-Thanh. From Godunov to a unified hybridized discontinuous Galerkin framework for partial differential equations. *Journal of Computational Physics*, 295:114–146, 2015.

[4] J. Butcher. Diagonally-implicit multi-stage integration methods. *Applied Numerical Mathematics*, 11(5):347 – 363, 1993.

[5] J. Butcher. General linear methods. *Computers & Mathematics with Applications*, 31(4):105–112, 1996.

[6] J. Butcher. General linear methods. *Acta Numerica*, 15:157–256, 2006.

[7] J. Butcher, P. Chartier, and Z. Jackiewicz. Nordsieck representation of DIMSIMs. *Numerical Algorithms*, 16(2):209–230, 1997.

[8] J. Butcher and H. Podhaisky. On error estimation in general linear methods for stiff odes. *Applied Numerical Mathematics*, 56(3):345–357, 2006.

[9] J. C. Butcher. On the convergence of numerical solutions to ordinary differential equations. *Mathematics of Computation*, 20(93):1–10, 1966.

[10] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2nd edition edition, August 2008.

[11] J. C. Butcher and Z. Jackiewicz. A reliable error estimation for diagonally implicit multistage integration methods. *BIT Numerical Mathematics*, 41(4):656–665, 2001.

[12] A. Cardone, Z. Jackiewicz, J. H. Verner, and B. Welfert. Order conditions for general linear methods. *Journal of Computational and Applied Mathematics*, 290:44 – 64, 2015.

[13] B. Cockburn, J. Gopalakrishnan, and 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.

[14] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p–Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 207:92–113, 2005.

[15] A. Gopinath and A. Jameson. Application of the time spectral method to periodic unsteady vortex sheeding. AIAA Paper 06-0449, 2006.

[16] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: Method formulation. *International Journal for Numerical and Analytical Modeling*, 3(1):1–20, 2006.

[17] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation. *International Journal for Numerical and Analytical Modeling*, 3(2):141–162, 2006.

[18] R. D. Henderson. Details of the drag curve near the onset of vortex shedding. *Physics of Fluids*, 7:2102–2104, 1995.

[19] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Number 54 in Texts in Applied Mathematics. Springer Verlag, 2008.

[20] P. Houston and E. Süli. hp-adaptive discontinuous Galerkin Finite Element methods for first order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23:1226–1252, 2001.

[21] Z. Jackiewicz. Implementation of DIMSIMs for stiff differential systems. *Appl. Numer. Math.*, 42(1-3):251–267, Aug. 2002.

[22] Z. Jackiewicz. Construction and implementation of general linear methods for ordinary differential equations: A review. *Journal of Scientific Computing*, 25(1):29–49, 2005.

[23] A. Jaust and J. Schütz. A temporally adaptive hybridized discontinuous Galerkin method for time-dependent compressible flows. *Computers and Fluids*, 98:177–185, 2014.

[24] A. Jaust, J. Schütz, and D. C. Seal. Implicit multistage two-derivative discontinuous Galerkin schemes for viscous conservation laws. *Journal of Scientific Computing*, 69:866–891, 2016.

[25] R. M. Kirby, S. J. Sherwin, and B. Cockburn. To CG or HDG: A comparative study. *Journal of Scientific Computing*, 51(1):183–212, 2012.

[26] N. C. Nguyen and J. Peraire. Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. *Journal of Computational Physics*, 231:5955–5988, 2012.

[27] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228:3232–3254, 2009.

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

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

[30] A. Nordsieck. On numerical integration of ordinary differential equations. *Mathematics of Computation*, 16:22–49, 1962.

[31] D. D. Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin Methods*, volume 69. Springer Science & Business Media, 2011.

[32] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory, 1973.

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

[34] J. Schütz and G. May. An adjoint consistency analysis for a class of hybrid mixed methods. *IMA Journal of Numerical Analysis*, 34(3):1222–1239, 2014.

[35] J. Schütz, M. Woopen, and G. May. A combined hybridized discontinuous Galerkin / hybrid mixed method for viscous conservation laws. In F. Ancona, A. Bressan, P. Marcati, and A. Marson, editors, *Hyperbolic Problems: Theory, Numerics, Applications*, pages 915–922. American Institute of Mathematical Sciences, 2012.

[36] P. E. Vos, C. Eskilsson, A. Bolis, S. Chun, R. M. Kirby, and S. J. Sherwin. A generic framework for time-stepping partial differential equations (PDEs): general linear methods, object-oriented implementation and application to fluid problems. *International Journal of Computational Fluid Dynamics*, 25(3):107–125, 2011.

[37] C. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477–539, 1996.

[38] M. Woopen, A. Balan, G. May, and J. Schütz. A comparison of hybridized and standard DG methods for target-based *hp*-adaptive simulation of compressible flow. *Computers and Fluids*, 98:3–16, 2014.

[39] M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76:811–834, 2014.

[40] W. Wright. *General linear methods with inherent Runge-Kutta stability*. PhD thesis, University of Auckland, 2002.

[41] S. Yakovlev, D. Moxey, R. Kirby, and S. Sherwin. To CG or to HDG: A comparative study in 3D. *Journal of Scientific Computing*, 2015.

# UHasselt Computational Mathematics Preprint Series

## 2017

UP-17-01 *Alexander Jaust and Jochen Schütz*, **General linear methods for time-dependent PDEs**, 2017

## 2016

UP-16-06 *Klaus Kaiser and Jochen Schütz*, **A high-order method for weakly compressible flows**, 2016

UP-16-05 *Stefan Karpinski, Iuliu Sorin Pop, Florin A. Radu*, **A hierarchical scale separation approach for the hybridized discontinuous Galerkin method**, 2016

UP-16-04 *Florin A. Radu, Kundan Kumar, Jan Martin Nordbotten, Iuliu Sorin Pop*, **Analysis of a linearization scheme for an interior penalty discontinuous Galerkin method for two phase flow in porous media with dynamic capillarity effects** , 2016

UP-16-03 *Sergey Alyaev, Eirik Keilegavlen, Jan Martin Nordbotten, Iuliu Sorin Pop*, **Fractal structures in freezing brine**, 2016

UP-16-02 *Klaus Kaiser, Jochen Schütz, Ruth Schöbel and Sebastian Noelle*, **A new stable splitting for the isentropic Euler equations**, 2016

UP-16-01 *Jochen Schütz and Vadym Aizinger*, **A hierarchical scale separation approach for the hybridized discontinuous Galerkin method**, 2016