
Oral presentation | Mesh generation/adaptation

Mesh generation/adaptation-II

Wed. Jul 17, 2024 2:00 PM - 3:30 PM Room C

[8-C-01] Towards Robust Time-Accurate Anisotropically Adaptive Hybridized Discontinuous Galerkin Method

*Tomáš Levý¹, Georg May² (1. University of West Bohemia, 2. von Karman Institute for Fluid Dynamics)

Keywords: Anisotropic mesh adaptation, Solution transfer, Hybridized discontinuous Galerkin method, Time-dependent balance laws

Towards Robust Time-Accurate Anisotropically Adaptive Hybridized Discontinuous Galerkin Method

T. Levý* and G. May**

Corresponding author: levyto@kme.zcu.cz

* Department of Mechanics, Faculty of Applied Sciences,
University of West Bohemia, Pilsen, Czech Republic.

** Aeronautics and Aerospace Department,
von Karman Institute for Fluid Dynamics, Sint-Genesius-Rode, Belgium.

Abstract: Metric-based anisotropic mesh adaptation has proven effective for the solution of both steady and unsteady problems in terms of reduced computational time and accuracy gain. Especially for time-dependent problems, its generalization to implicit high-order space and time discretizations is, nevertheless, still a challenging task as it requires a great care to preserve consistency and stability of the numerical solution. In this regard, the objective of the present paper is twofold. First, to devise an accurate unsteady mesh adaptation algorithm, and second, to introduce a new solution transfer between anisotropic meshes, which preserves the local minima and maxima. Our findings are based on a hybridized discontinuous Galerkin (HDG) solver with diagonally implicit Runge-Kutta (DIRK) time integration, whereas the main focus is on problems for two-dimensional Euler equations including moving shocks.

Keywords: Anisotropic mesh adaptation, Time-dependent conservation laws, Solution transfer, Hybridized discontinuous Galerkin method.

1 Introduction

The use of second-order methods, such as finite volume schemes, has become a standard for the simulation of compressible flows during the last couple of decades. These methods offer a trade-off between the resolution of the numerical solution and the stability of the solution process. Although the intrinsic artificial diffusivity of these methods greatly contributes to the stability, it can, however, also exclude important features of the flow field from the numerical solution.

On the other hand, high-order methods offer superior resolution of the flow field at reduced number of degrees of freedom for a given problem, provided the solution is smooth enough [1]. Since high-speed compressible flow is often accompanied by the presence of shock waves, which in turn disrupts the local regularity of the solution, a finer local resolution is essential for the high-order method to obtain a reasonably accurate numerical solution. Of course, this need comes with an increased computational cost. Besides the shocks, there could also be other highly anisotropic features of the flow such as boundary layers or wakes in the case of viscous flow simulation, requiring locally refined computational mesh.

The ideal high-order method then should be adaptive in the sense that it takes advantage of the smooth flow regions where the resolution may not be as fine as in the regions where rapid changes in the solution are present. One such approach is the use of anisotropically adapted meshes where the mesh elements align with the features of the flow based on the numerical solution at hand [2].

The resolution problem is probably even more significant for time-dependent simulations as the anisotropic features of the flow can move over time. In order to fully benefit from the adaptation strategy, the mesh has to be adapted numerous times during the simulation. Such anisotropic mesh refinement generally leads to highly skewed unstructured grids. Hence, attention must also be paid to the accuracy of the overall numerical method in order to retain the solution precision after each adaptation step.

Here, we restrict ourselves to semi-discretization of time-dependent partial differential equations in space, thus constructing a system of ordinary differential equations, which is further discretized in time. This procedure is sometimes called the *method of lines* [3]. Another possibility is to consider a space-time high-order method [4] requiring a $(d + 1)$ -dimensional mesh where d is the space dimension of a given problem.

An essential pitfall of the separate space and time discretization is that the remeshing procedure requires interpolation of the solution from the previous mesh to the adapted mesh that will be used in the subsequent time step. Inappropriate interpolation methods can disrupt conservation of physical quantities of interest, which may ultimately lead to a loss of accuracy. Although a number of conservative solution transfer operators coupled with time-dependent anisotropic mesh refinement can be found in the literature [5, 6], these are usually designed for locally piecewise-linear representation of the solution, and the interpolation then relies on the Hessian of the solution only [7].

In this work, we consider the hybridized discontinuous Galerkin (HDG) method introduced by Nguyen et al. [8] to discretize conservation laws of general form in space. In the context of discontinuous Galerkin methods, the resulting numerical solution on a particular mesh is represented by a polynomial of degree at most p on each element. We further extend the work of Rangarajan et al. [9] where an anisotropic mesh adaptation methodology is built on an HDG solution of steady balance laws. Following the work of Dolejší [10], the adaptation procedure utilizes reconstructed directional derivatives of order $p + 1$, which allow mesh optimization using higher degree of polynomial approximation, simplifying to the more classical Hessian-based anisotropic adaptation approaches for $p = 1$, see e.g. [11]. The aim of the present work is to couple the anisotropic mesh adaptation with the numerical solution of time-dependent conservation laws using high-order diagonally implicit Runge-Kutta (DIRK) methods for time discretization, while the main focus is on problems with moving discontinuities present in the solution.

For such piecewise polynomial approximations, the Galerkin projection, which is optimal in the L^2 -norm, has proven to be well-suited for interpolation of fields between meshes [12]. We follow the work of Farrell et al. [13], who constructs a geometry, known as the supermesh, defined as a triangulation of intersection polygons of the input meshes. The basis functions of both meshes are, by construction, continuous polynomials on each supermesh element, allowing high-accuracy evaluation of the projection integrals. With regard to piecewise discontinuous approximations, the main drawback is that the Galerkin projection itself is not bounded. Although a bounded variant of the algorithm has been devised in [14], it is based on a linear solution representation on each element of the mesh leading to a global decrease of the projection accuracy. Hence, the aim of this work is to design a bounded solution transfer approach, which preserves the order of the solution projection where it is considered smooth, and which eliminates the local minima and maxima overshoots by a limiting procedure.

Regarding the coupling of time advancement with the anisotropic mesh adaptation, most of the existing work is based on either global or local remeshing, [5]. Probably the simplest and most straightforward approach is to successively generate meshes to be used in time step $n + 1$ based the solution from time step n , [15]. Such approach, however, results in a large number of mesh adaptations, which can tremendously increase the computational time. Additionally, excessive number of mesh adaptations comes with increased error in the solution due to the solution transfers between meshes. Reducing the number of mesh adaptations by remeshing every $m > 1$ time steps only intensifies the problem at hand since the optimal mesh is computed based on solution from an earlier time, which results in a time shift between the mesh-solution pair. In fact, this delay exists already for the case of remeshing in every time step. Furthermore, iterative mesh adaptation, which is usually introduced for anisotropic mesh adaptation in steady problems addressing their nonlinear nature [16], is not considered in this approach.

Naturally, one can think of some form of a solution prediction for the purpose of mesh adaptation in chosen time interval. Similar idea has been proposed by Alauzet et al. in [17] where the authors has introduced so-called *transient fixed-point scheme* for unsteady mesh adaptation. They split the simulation time into subintervals, inside which the solution is repeatedly sought by a second-order finite volume method with explicit time marching, generating an adapted mesh every time the end of the subinterval is reached. Note the mesh is then pseudo-optimal for the entire subinterval.

The present paper aims at generalization of the outlined idea of time-dependent anisotropic mesh adaptation for implicit high-order methods. We propose an efficient mesh adaptation algorithm based on a lower-order solution prediction, which together with the new solution transfer approach results in an accurate and stable adaptive numerical method taking full advantage of the high-order approximation. We also provide a detailed comparison of the above-mentioned mesh adaptation approaches for time-dependent problems with both smooth and discontinuous solutions.

The paper is organized as follows. In Section 2 and 3, we briefly discuss the space and time discretization and the baseline anisotropic mesh adaptation algorithm. In Section 4, the new bounded solution transfer is introduced. A detailed description of the two unsteady mesh adaptation approaches considered in this work is given in Section 5. In Section 6, we return our attention to the discretization with regard to its stabilization for the case of solutions including shocks. Finally, in Section 7, we present the numerical results demonstrating advantages and limitations of the presented algorithms.

2 Hybridized Discontinuous Galerkin Method

In this work, we focus on the solution of general time-dependent convection-diffusion systems of equations of the form

$$\partial_t \mathbf{w} + \nabla \cdot (\mathbf{f}_c(\mathbf{w}) - \mathbf{f}_v(\mathbf{w}, \nabla \mathbf{w})) = \mathbf{0} \quad (1)$$

with appropriate initial and boundary conditions defined on $\partial\Omega$, the boundary of an open bounded domain $\Omega \in \mathbb{R}^d$, $d = 2$. Here, $\mathbf{w} \in \mathbb{R}^m$ is the vector of conservative variables, $\mathbf{f}_c : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times d}$ is the convective flux, and $\mathbf{f}_v : \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^{m \times d}$ is the diffusive flux, m being the number of conservative variables.

Let the computational domain Ω be partitioned into a collection \mathcal{T}_h of nonoverlapping elements such that $\Omega = \bigcup_{\kappa \in \mathcal{T}_h} \kappa$ with $\partial\mathcal{T}_h := \{\partial\kappa \setminus \partial\Omega : \kappa \in \mathcal{T}_h\}$ being the set of all edges of the elements, excluding the boundary. We denote the set of all interior edges of \mathcal{T}_h by \mathcal{E}_h and the set \mathcal{E}_h^∂ contains all boundary edges.

Following the work of Nguyen et al. [18, 8], the conservation law (1) can be rewritten as a first-order system by introducing an auxiliary variable \mathbf{q} , which represents the gradient of the solution,

$$\begin{aligned} \mathbf{q} - \nabla \mathbf{w} &= \mathbf{0}, \\ \partial_t \mathbf{w} + \nabla \cdot (\mathbf{f}_c(\mathbf{w}) - \mathbf{f}_v(\mathbf{w}, \mathbf{q})) &= \mathbf{0}. \end{aligned} \quad (2)$$

Before the weak formulation will be introduced, let us define the discontinuous function spaces in which the solution of system (2) will be sought. Let $\mathcal{P}^p(\mathcal{D})$ denote a set of polynomials of total degree at most p on some domain \mathcal{D} . For a given computational mesh, we consider the following approximation spaces

$$\Sigma_h := \{\boldsymbol{\tau} \in [L^2(\Omega)]^{m \times d} : \boldsymbol{\tau}|_\kappa \in [\mathcal{P}^p(\kappa)]^{m \times d}, \forall \kappa \in \mathcal{T}_h\}, \quad (3)$$

$$V_h := \{\mathbf{v} \in [L^2(\Omega)]^m : \mathbf{v}|_\kappa \in [\mathcal{P}^p(\kappa)]^m, \forall \kappa \in \mathcal{T}_h\}. \quad (4)$$

Functions $\boldsymbol{\tau} \in \Sigma_h$ and $\mathbf{v} \in V_h$ are then piecewise polynomials of degree p , which can be discontinuous across element edges.

In the HDG method, additional unknown $\boldsymbol{\lambda}$ is introduced at the element edges. The system of equations is then closed by weakly enforcing continuity of the normal component of convective and diffusive numerical fluxes at the edges. Hence, we define yet another discontinuous function space

$$\Lambda_h := \{\boldsymbol{\mu} \in [L^2(\mathcal{E}_h)]^m : \boldsymbol{\mu}|_e \in [\mathcal{P}^p(e)]^m, \forall e \in \mathcal{E}_h\}, \quad (5)$$

where functions $\boldsymbol{\mu} \in \Lambda_h$ are now piecewise polynomials of degree p , which can be discontinuous across element vertices.

Now, the goal is to find an approximation $\mathfrak{x}_h := (\mathbf{q}_h, \mathbf{w}_h, \boldsymbol{\lambda}_h) \in \mathbb{X}_h := \Sigma_h \times V_h \times \Lambda_h$ such that

$$\begin{aligned} 0 &= (\mathbf{q}_h, \boldsymbol{\tau}_h)_{\mathcal{T}_h} + (\mathbf{w}_h, \nabla \cdot \boldsymbol{\tau}_h)_{\mathcal{T}_h} - \langle \boldsymbol{\lambda}_h, \boldsymbol{\tau}_h \cdot \mathbf{n} \rangle_{\partial\mathcal{T}_h} \\ &+ (\partial_t \mathbf{w}_h, \mathbf{v}_h)_{\mathcal{T}_h} - \left(\mathbf{f}_c(\mathbf{w}_h) - \mathbf{f}_v(\mathbf{w}_h, \mathbf{q}_h), \nabla \mathbf{v}_h \right)_{\mathcal{T}_h} + \left\langle \left(\widehat{\mathbf{f}}_c - \widehat{\mathbf{f}}_v \right) \cdot \mathbf{n}, \mathbf{v}_h \right\rangle_{\partial\mathcal{T}_h} \\ &+ \left\langle \left[\widehat{\mathbf{f}}_c - \widehat{\mathbf{f}}_v \right], \boldsymbol{\mu}_h \right\rangle_{\mathcal{E}_h} + \mathcal{N}_h^{\text{BC}}(\mathbf{q}_h, \mathbf{w}_h; \boldsymbol{\tau}_h, \mathbf{v}_h) + \mathcal{N}_h^{\text{SC}}(\mathbf{q}_h, \mathbf{w}_h; \mathbf{v}_h) \end{aligned} \quad (6)$$

holds for all $\mathfrak{y}_h := (\boldsymbol{\tau}_h, \mathbf{v}_h, \boldsymbol{\mu}_h) \in \mathbb{X}_h$. We have used the abbreviation (\cdot, \cdot) and $\langle \cdot, \cdot \rangle$ to distinguish between element- and edge-oriented inner products, and the jump operator for the vector valued function is defined as $[\![\mathbf{v}]\!] := \mathbf{v}^+ \cdot \mathbf{n} + \mathbf{v}^- \cdot \mathbf{n}^-$, where the signs \pm correspond to elements κ^+ and κ^- separated by edge e .

On the edges of the mesh, the analytical flux functions are replaced by the numerical fluxes. In our work, the choice of the convective and diffusive numerical fluxes corresponds to Lax-Friedrichs flux and LDG flux, respectively, i.e.,

$$\widehat{\mathbf{f}}_c(\boldsymbol{\lambda}_h, \mathbf{w}_h) = \mathbf{f}_c(\boldsymbol{\lambda}_h) \cdot \mathbf{n} - \alpha_c (\boldsymbol{\lambda}_h - \mathbf{w}_h), \quad (7)$$

$$\widehat{\mathbf{f}}_v(\boldsymbol{\lambda}_h, \mathbf{w}_h, \mathbf{q}_h) = \mathbf{f}_v(\boldsymbol{\lambda}_h, \mathbf{q}_h) \cdot \mathbf{n} + \alpha_v (\boldsymbol{\lambda}_h - \mathbf{w}_h), \quad (8)$$

where we assume α_c and α_v to be constant scalar values.

The boundary conditions are incorporated into the scheme by evaluating the exact flux functions with a suitably chosen vector of conservative variables at the domain boundary $\mathbf{w}_{h,\partial\Omega} \equiv \mathbf{w}_{h,\partial\Omega}(\mathbf{w}_h)$, and

its gradient $\mathbf{q}_{h,\partial\Omega} \equiv \mathbf{q}_{h,\partial\Omega}(\mathbf{q}_h)$ depending on a particular type of the boundary condition,

$$\mathcal{N}_h^{\text{BC}}(\mathbf{q}_h, \mathbf{w}_h; \boldsymbol{\tau}_h, \mathbf{v}_h) = \langle \mathbf{w}_{h,\partial\Omega}, \boldsymbol{\tau}_h \cdot \mathbf{n} \rangle_{\mathcal{E}_h^\partial} + \left\langle (\mathbf{f}_c(\mathbf{w}_{h,\partial\Omega}) - \mathbf{f}_v(\mathbf{w}_{h,\partial\Omega}, \mathbf{q}_{h,\partial\Omega})) \cdot \mathbf{n}, \mathbf{v} \right\rangle_{\mathcal{E}_h^\partial}. \quad (9)$$

See the work of Nguyen et al. [8] for details.

The operator $\mathcal{N}_h^{\text{SC}}$ in (6) represents the shock-capturing terms used for stabilization of the solution in case of a presence of discontinuities in the solution. The shock-capturing approach utilized in this work is described in detail in section 6.

The presence of the unsteady term $(\partial_t \mathbf{w}_h, \mathbf{v}_h)_{\mathcal{T}_h}$ in the semi-discretized scheme (6) gives rise to a system of differential-algebraic equations of index 1. Hence the classical explicit time integration schemes cannot be used, and one has to choose a sufficiently stable implicit method for the solution of time-dependent problems. In this work, we consider the diagonally implicit Runge-Kutta (DIRK) methods for the time discretization [19].

If we split the discretization of the term with the time derivative and the remaining part, then applying a general s -stage DIRK method to formulation (6) results in seeking an approximation $\mathbb{x}_h^{n,i} \in \mathbb{X}_h$ such that

$$\frac{1}{a_{ii}\Delta t^n}(\mathbf{w}_h^{n,i}, \mathbf{v}_h)_{\mathcal{T}_h} + \mathcal{N}(\mathbb{x}_h^{n,i}; \mathbb{y}_h) = \frac{1}{a_{ii}\Delta t^n}(\mathbf{w}_h^n, \mathbf{v}_h)_{\mathcal{T}_h} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} \mathcal{N}(\mathbb{x}_h^{n,j}; \mathbb{y}_h), \quad (10)$$

for all $\mathbb{y}_h \in \mathbb{X}_h$ and for all stages $i = 1, \dots, s$ of the DIRK method. The resulting system of nonlinear equations (10) in each stage of the DIRK method is solved by a damped Newton method. The coefficients a_{ij} , $i, j = 1, \dots, s$ are given by a specific DIRK method and the approximation $\mathbb{x}_h^{n,i}$ represents the intermediate solution at each stage. We further consider special class of DIRK methods, which are so-called stiffly accurate [19], resulting in the fact that the intermediate solution of the last stage is identical to the numerical solution at time step $n + 1$, i.e. $\mathbb{x}_h^{n,s} = \mathbb{x}_h^{n+1}$. In particular, we use

- the backward differentiation formula of first order (BDF1), i.e. the backward Euler method, [19],
- the two-stage second-order DIRK(2,2) method of Alexander [20],
- the three-stage third-order DIRK(3,3) of Cash [21],
- and the five-stage fourth-order DIRK(5,4) method of Hairer and Wanner [19].

Note the BDF1 method can be seen as a limiting case of DIRK method having only a single stage. Hence, it naturally fits into the same implementation approach.

3 Anisotropic Mesh Adaptation

Let us now focus on the metric-based mesh adaptation, of which particular elements are used in our anisotropic mesh adaptation procedure. The fundamental concept is the mesh-metric duality and the continuous mesh model introduced by Loseille and Alauzet [22].

Given a triangulation \mathcal{T}_h of the computational domain $\Omega \in \mathbb{R}^2$, each nondegenerate triangular element can be characterized by symmetric positive definite matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ such that each of its edges \mathbf{e}_i , $i = 1, 2, 3$ has a constant length C under the norm induced by the metric

$$\|\mathbf{e}_i\|_{\mathbf{M}} = \sqrt{\mathbf{e}_i^T \mathbf{M} \mathbf{e}_i} = C, \quad i = 1, 2, 3. \quad (11)$$

The mesh element can be inscribed into an ellipse with its origin located at the element centroid, h_1 and h_2 being its principal axes and θ its orientation in the Cartesian coordinate system. These quantities can be extracted from the spectral decomposition of the matrix, $\mathbf{M} = \mathbf{Q}^T \boldsymbol{\Lambda} \mathbf{Q}$, where

$$\mathbf{Q} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} 1/h_1^2 & 0 \\ 0 & 1/h_2^2 \end{pmatrix}. \quad (12)$$

One can also define the aspect ratio $\beta := h_2/h_1$ and the local density $d := 1/(h_1 h_2)$, which is proportional to the inverse of the element area since $|\kappa| = \frac{3\sqrt{3}}{4} h_1 h_2$. The tuple (d, β, θ) now completely defines the

metric M and hence the mesh element $\kappa \in \mathcal{T}_h$. Quantities β and θ can be denoted as the anisotropy of the triangle, while the local density d alone corresponds to its size.

Our adaptation methodology is based on evaluation of the indicated metric for each element of the current mesh and generation of an optimized metric, which is obtained using the solution error estimate, corresponding to the desired mesh element. After a suitable interpolation, continuous metric field can then be used in a metric-based mesh generator to produce the desired mesh. In the present work, we use the BAMG mesh generator of Hecht [23].

The aim of the anisotropic mesh adaptation is therefore to optimize both the size (d) and the anisotropy (β, θ) of the mesh elements in such a way that the resulting mesh will have improved approximation properties. In our case, the goal is to obtain triangles that give the smallest interpolation error into the piecewise polynomial approximation space (4) in the L^q -norm.

The optimal metric is obtained by a two-step optimization. First, a locally optimal anisotropy (β^*, θ^*) is obtained from a bound on the interpolation error based on the work of Dolejší [10]. Using a remainder of the Taylor series of order p centered at $\mathbf{x} \in \Omega$, the local error model is expressed by

$$|e_{\mathbf{x},p}(\mathbf{y})| = \frac{1}{(p+1)!} |u^{(p+1,\varphi)}(\mathbf{x})| |\mathbf{y} - \mathbf{x}|^{p+1}, \quad \forall \mathbf{y} \in \Omega, \quad (13)$$

which is valid for some sufficiently smooth function $u(\mathbf{x})$. The directional derivative $u^{(k,\varphi)}(\mathbf{x})$ of k -th order is given by

$$u^{(k,\varphi)}(\mathbf{x}) = \sum_{\ell=0}^k \binom{k}{\ell} \frac{\partial^\ell u(\mathbf{x})}{\partial x_1^\ell \partial x_2^{k-\ell}} (\cos \varphi)^\ell (\sin \varphi)^{k-\ell}, \quad \varphi \in [0, 2\pi). \quad (14)$$

By the estimates introduced in [10], one can obtain an approximate bound for the interpolation error in terms of three parameters (A_p, ρ_p, φ_p), which are related to the anisotropy of the element (β, θ). Let φ_p correspond to the angle where the directional derivative is reaching its maximum,

$$\varphi_p(\mathbf{x}) := \operatorname{argmax}_{\varphi \in [0, 2\pi)} |u^{(p+1,\varphi)}(\mathbf{x})|, \quad (15)$$

and let φ_p^\perp represent the direction orthogonal to φ_p . Then, A_p is the maximum of the scaled $(p+1)$ -st directional derivative along direction φ_p , $A_p^\perp(\mathbf{x})$ is the corresponding maximum along direction φ_p^\perp , and ρ_p is their ratio, i.e.

$$A_p(\mathbf{x}) := \frac{1}{(p+1)!} |u^{(p+1,\varphi_p)}(\mathbf{x})|, \quad A_p^\perp(\mathbf{x}) := \frac{1}{(p+1)!} |u^{(p+1,\varphi_p^\perp)}(\mathbf{x})|, \quad \rho_p(\mathbf{x}) := \frac{A_p(\mathbf{x})}{A_p^\perp(\mathbf{x})}. \quad (16)$$

The optimal value of the error bound in L^q -norm is then achieved for

$$\beta^*(\mathbf{x}) = (\rho_p(\mathbf{x}))^{1/(p+1)} \quad \text{and} \quad \theta^*(\mathbf{x}) = \varphi_p(\mathbf{x}) - \frac{\pi}{2}. \quad (17)$$

The anisotropy parameters (β, θ) are thus eliminated and replaced by higher derivatives of u with the help of parameters ($A_p(\mathbf{x}), \rho_p(\mathbf{x})$). We refer to [10] for details. Note the definition of function u is optional and can be chosen as either primary or derived variable of the numerical solution.

With fixed optimal anisotropy, the corresponding error estimate is then valid for a family of ellipses, where their size or the local density $d(\mathbf{x})$ and, hence, the size of the inscribed triangle, is now a free parameter that can be determined based on some appropriate optimization strategy. Having a continuous error model, one may apply analytic optimization techniques to obtain the optimal size distribution given the number of mesh elements as a constraint. In earlier work of Rangarajan et al. [9], the following optimal local density is proposed

$$d^*(\mathbf{x}) = K \left(\frac{A_p(\mathbf{x})}{\sqrt{\rho_p(\mathbf{x})}} \right)^{\frac{2q}{q(p+1)+2}} \quad \text{with} \quad K = N \left(\int_{\Omega} \left(\frac{A_p(\mathbf{x})}{\sqrt{\rho_p(\mathbf{x})}} \right)^{\frac{2q}{q(p+1)+2}} d\mathbf{x} \right)^{-1}. \quad (18)$$

The constant N represents the mesh complexity, which is related to the desired number of mesh elements. Note the minimization of the interpolation error is done in L^q norm, $q \in [1, \infty]$. In the present work, we choose $q = 2$.

4 Solution Transfer

In Section 2, we have seen that in order to solve equation (10) for the intermediate solutions in each stage of the DIRK method, the solution of the previous time step has to be known. Hence, unlike computations with static meshes, the use of anisotropic mesh refinement in unsteady simulations relies on the solution transfer between the previous and the current mesh.

Furthermore, this interpolation step needs a special care since one would like to avoid the disruption of the design order of convergence of the HDG method and/or the DIRK method. Another important property whose absence can have fatal consequences for the evolution of numerical solution in time is the conservativity of the solution transfer. Unlike for isotropic mesh adaptation, the pair of anisotropically adapted meshes does not share the same topology, which can be major source of errors during the transfer of the solution, negatively affecting the numerical solution in the subsequent time steps.

4.1 Galerkin Projection

When prescribing the initial conditions for a given problem, it is usual in the context of discontinuous Galerkin methods to perform a local L^2 -projection, also called Galerkin projection, onto the space of polynomials $\mathcal{P}^p(\kappa)$, $\kappa \in \mathcal{T}_h$, of degree at most p forming the global function spaces given by equations (3) - (5). Consider a given continuous function $u(\mathbf{x})$, $\mathbf{x} \in \kappa$. The aim is to obtain interpolant $\Pi_h u \in \mathcal{P}^p(\kappa)$, which is optimal in the L^2 -norm, i.e.

$$\|u - \Pi_h u\|_{L^2(\kappa)} = \min_{v \in \mathcal{P}^p(\kappa)} \|u - v\|_{L^2(\kappa)}. \quad (19)$$

The problem at hand can be equally formulated in the weak sense by

$$\int_{\kappa} u \varphi^{(i)} d\mathbf{x} = \int_{\kappa} \Pi_h u \varphi^{(i)} d\mathbf{x}, \quad i = 1, \dots, N, \quad (20)$$

where $N = \dim(\mathcal{P}^p(\kappa))$, $\varphi^{(i)}$ being the basis functions of $\mathcal{P}^p(\kappa) = \text{span}\{\varphi^{(i)}\}$. Relation (20) represents a linear system for the unknown coefficients $U^{(i)}$ of the linear expansion

$$\Pi_h u = \sum_{i=1}^N U^{(i)} \varphi^{(i)}. \quad (21)$$

This approach directly leads to locally conservative projection of function u , provided the unit constant function is contained in the polynomial space $\mathcal{P}^p(\kappa)$. In this work, we employ the Dubiner basis functions for which this requirement is satisfied.

The solution transfer between two distinct meshes used in this study relies on the local Galerkin projection and it is based on the work of Farrell [13, 14]. Compared to formulation (20), we will now focus on the case where the function to be projected onto the new mesh is given by the HDG solution of the last time step. Thus, when performing the Galerkin projection locally on an element of the new mesh, the underlying solution is generally discontinuous along the intersection lines of the two meshes under consideration. The situation will be analogous when a discontinuous initial condition is prescribed in case the edges of an initial mesh does not perfectly align with the discontinuity.

4.2 Broken Galerkin Projection

Consider field $w_1 = w_1(\mathbf{x}, t^n)$, which represents a component of the vector of conservative variables appearing in the conservation law (1) defined on mesh \mathcal{T}_h^1 , resulting from the n -th time step of a simulation of an unsteady problem. After a mesh adaptation step, a new adapted triangulation \mathcal{T}_h^2 based on the solution w_1 is obtained. The goal is to get interpolated solution $w_2 = w_2(\mathbf{x}, t^n)$ defined on \mathcal{T}_h^2 .

Let each element $\bar{\kappa}$ of the previous mesh \mathcal{T}_h^1 have $N_1^{\bar{\kappa}}$ degrees of freedom corresponding to the basis functions $\varphi_1^{(j)}$, $j = 1, \dots, N_1^{\bar{\kappa}}$. Similarly, for the case of the currently adapted mesh, let N_2^{κ} be the degrees of freedom of each element $\kappa \in \mathcal{T}_h^2$ with basis functions $\varphi_2^{(i)}$, $i = 1, \dots, N_2^{\kappa}$. Both fields, w_1 and w_2 , can then be represented on each element by the linear expansions

$$w_1(\mathbf{x}, t^n) = \sum_{j=1}^{N_1^{\bar{\kappa}}} W_1^{(j)}(t^n) \varphi_1^{(j)}(\mathbf{x}), \quad \mathbf{x} \in \bar{\kappa}, \quad \bar{\kappa} \in \mathcal{T}_h^1, \quad (22)$$

$$w_2(\mathbf{x}, t^n) = \sum_{i=1}^{N_2^\kappa} W_2^{(i)}(t^n) \varphi_2^{(i)}(\mathbf{x}), \quad \mathbf{x} \in \kappa, \quad \kappa \in \mathcal{T}_h^2. \quad (23)$$

Returning to equation (20), we can write

$$\int_{\kappa} w_1 \varphi_2^{(i)} d\mathbf{x} = \int_{\kappa} w_2 \varphi_2^{(i)} d\mathbf{x}, \quad i = 1, \dots, N_2^\kappa, \quad (24)$$

The integral on the left-hand side now should be performed over discontinuous function $w_1(\mathbf{x}, t^n)$, $\mathbf{x} \in \kappa$ since the basis functions $\varphi_1^{(j)}$, $j = 1, \dots, N_1^\kappa$ in (22) are associated with elements $\bar{\kappa} \in \mathcal{T}_h^1$ and therefore are piecewise polynomials on new-mesh element $\kappa \in \mathcal{T}_h^2$. Thus, even when the Gaussian quadrature rule used for numerical integration will be exact for polynomials of degree p , integrating the product of different basis functions numerically by evaluating $\varphi_1^{(j)}$ at each quadrature point of element κ would result in an inexact integration, and hence the loss of accuracy and conservativity of the resulting projection.

In order to evaluate the corresponding integral, we utilize the concept of a supermesh [14]. The basis functions $\varphi_2^{(i)}$ are naturally continuous over κ , and hence the idea is to split the element κ into triangular intersections with the elements of \mathcal{T}_h^1 and evaluate the integral as a sum of integrals over these intersections while avoiding the discontinuities in the basis functions. Hence, the numerical integration on each of these intersections will be exact when quadrature rule of appropriate order is used.

The supermesh is defined as the mesh of intersection polygons of the elements of \mathcal{T}_h^1 and \mathcal{T}_h^2 . We denote the local supermesh of element $\kappa \in \mathcal{T}_h^2$ by $\mathcal{H}_\kappa = \{K_\ell, \ell = 1, \dots, N_\kappa\}$ where N_κ is the number of local supermesh elements. The supermesh construction has been described in the previous work [24], see Fig. 1.

Finally, broken Galerkin projection can be formulated based on (24). Replacing w_1 and w_2 by their finite element representations (22) and (23), respectively, and breaking the integrals \int_{κ} into $\sum_{N_\kappa} \int_{K_\ell}$ with respect to the supermesh elements yields

$$\sum_{\ell=1}^{N_\kappa} \int_{K_\ell \in \mathcal{H}_\kappa} \sum_{j=1}^{N_1^\kappa} W_1^{(j)} \varphi_1^{(j)} \varphi_2^{(i)} d\mathbf{x} = \sum_{\ell=1}^{N_\kappa} \int_{K_\ell \in \mathcal{H}_\kappa} \sum_{k=1}^{N_2^\kappa} W_2^{(k)} \varphi_2^{(k)} \varphi_2^{(i)} d\mathbf{x}, \quad i = 1, \dots, N_2^\kappa. \quad (25)$$

This representation gives rise to a linear system for the vector of expansion coefficients $\mathbf{W}_2 \in \mathbb{R}^{N_2^\kappa}$ on each element $\kappa \in \mathcal{T}_h^2$ of the form

$$\mathbf{M}_2 \mathbf{W}_2 = \mathbf{M}_{1,2} \mathbf{W}_1, \quad (26)$$

where $\mathbf{M}_2 \in \mathbb{R}^{N_2^\kappa \times N_2^\kappa}$ is the mass matrix defined with the set of basis functions φ_2 and $\mathbf{M}_{1,2} \in \mathbb{R}^{N_2^\kappa \times N_1^\kappa}$ is the mixed mass matrix of both sets of basis functions φ_2 and φ_1 .

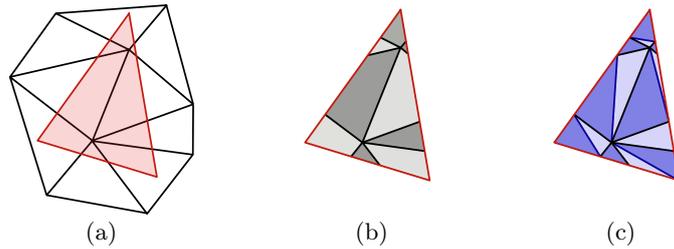


Figure 1: Illustration of a local supermesh \mathcal{H}_κ . Part of the initial mesh \mathcal{T}_h^1 is depicted in black and element $\kappa \in \mathcal{T}_h^2$ is indicated in red (a). At first, intersection polygons of both meshes are found (b). Local supermesh is then formed by triangulation of the intersection polygons. (c).

4.3 Limiting

The Galerkin projection in general does not preserve local minima and maxima of the field defined on the old mesh. While this property does not cause major problems when the solution is smooth enough and the underlying mesh has a reasonably fine resolution, since the difference between the old and new extrema will be negligible, one has to be careful when solving problems with discontinuous solutions.

The lack of boundedness leads to the formation of Gibbs phenomenon in terms of oscillations of the projection around a jump discontinuity.

In this work, we suppress the creation of the spurious oscillations by the application of limiting procedure, which is known from the second-order finite volume method with piecewise linear reconstruction of the solution, to the locally projected solution. However, we would like to preserve the accuracy in regions where the solution is smooth and thus well-approximated.

For this purpose, we utilize the discontinuity sensor introduced by Persson and Peraire [25], which has been originally used for detection of the amount of artificial viscosity applied in an elementwise manner to stabilize discontinuous Galerkin methods. The sensor is based on the rate of the decay of the expansion coefficients in (23). Once the broken Galerkin projection on element κ is performed, we compute the quantity

$$s_\kappa = \log_{10} \left(\frac{(w_2 - \widehat{w}_2, w_2 - \widehat{w}_2)_\kappa}{(w_2, w_2)_\kappa} \right), \quad (27)$$

where \widehat{w}_2 corresponds to the truncated expansion of the same projection, only containing terms up to order $p - 1$,

$$\widehat{w}_2 = \sum_{i=1}^{N_{2,p-1}^\kappa} W_2^{(i)} \varphi_2^{(i)} \quad \text{with} \quad N_{2,p-1}^\kappa = \dim(\mathcal{P}^{p-1}(\kappa)). \quad (28)$$

The element will be marked for limiting when $s_\kappa > s_0$, where s_0 is empirically determined constant. By analogy with Fourier coefficients in 1D, which decay at the rate $\sim 1/p^2$, it is expected that the quantity s_κ defined by (27) should roughly scale as $\sim \log_{10}(1/p^4)$. In this work, we set $s_0 = -(2.5 + 4 \log_{10}(p))$.

When the troubled element is detected, we truncate the projection expansion to linear polynomial corresponding to $p = 1$ and apply the slope-limiting procedure of Barth and Jespersen [26]. The truncated projection on element κ will be of the form

$$w_2 = W_2^{(1)} \varphi_2^{(1)} + \alpha \left(W_2^{(2)} \varphi_2^{(2)} + W_2^{(3)} \varphi_2^{(3)} \right), \quad \mathbf{x} \in \kappa, \quad \kappa \in \mathcal{T}_h^2. \quad (29)$$

where

$$\alpha = \min_{j \in \mathcal{I}(\partial\kappa)} \max(\alpha_j, 0). \quad (30)$$

The set $\mathcal{I}(\partial\kappa)$ contains the integration points \mathbf{x}_j lying on the edges of element κ . Furthermore,

$$\alpha_j = \begin{cases} \frac{M - W_2^{(1)}}{w_2(\mathbf{x}_j) - W_2^{(1)}}, & \text{if } w_2(\mathbf{x}_j) - M > 0, \\ \frac{m - W_2^{(1)}}{w_2(\mathbf{x}_j) - W_2^{(1)}}, & \text{if } w_2(\mathbf{x}_j) - m < 0, \\ 1, & \text{otherwise,} \end{cases} \quad (31)$$

and m and M is the minimum and maximum value of solution w_1 over all the integration points defined on the intersection edges forming the supermesh of the new element.

The resulting projected local solution w_2 is then linear and bounded by the minimum and maximum of the underlying solution defined on the old mesh. Note the solution transfer is still conservative even after the limiting step since the Dubiner basis functions are orthogonal and truncation of the linear combination will affect only the order of the solution. We rely on the assumption that the mesh is clustered in the region of the discontinuity after the anisotropic adaptation and hence the limiting procedure will be applied on a small fraction of the mesh elements. In theory, reducing the polynomial degree of the solution locally to $p = 1$ will not affect its accuracy since the order of the numerical method is in general reduced to one in the vicinity of the shocks.

In case a discontinuous initial condition is prescribed, we follow the same strategy. Once the L^2 -projection of the initial condition is performed by (20), we detect the elements with low resolution by the sensor (27) of Persson and Peraire and we then truncate the projection on these elements to a linear representation. When applying the Barth-Jespersen limiter, we take the quantities m and M in (31) as minimum and maximum of the initial condition evaluated on both element and edge integration points.

5 Anisotropic Mesh Adaptation for Time-Dependent Problems

In this section, we describe how the time-dependent HDG solution process can be coupled with the anisotropic mesh adaptation with the aid of the solution transfer discussed in the previous section. The idea is to devise a solver, which is robust enough to be able to generate a sufficiently accurate numerical solution of general time-dependent conservation laws represented by equation (1) even when a coarse, easily-generated initial mesh is supplied.

The common approach for steady problems is to provide such a mesh to a solver, solve the discretized problem at hand and finally generate a new mesh with the help of a metric-based mesh adaptation algorithm. The adapted mesh with ideally improved approximating properties according to a given problem is then used in the subsequent computation to generate a new numerical solution and consequently another adapted mesh, which has the potential of even better resolution. This process is iteratively repeated usually until some prescribed tolerance on the properly defined solution error norm is reached. Hence, we can talk about a *fixed-point iteration* method, which is ideally converging to an optimal mesh-solution pair with respect to a given error estimate.

Extension of this approach to unsteady problems is not straightforward since the discretized system (10) involve solutions in two consecutive time steps. In theory, both of these solutions would comply to a different adapted mesh. Hence, there are multiple options on how to appropriately couple the anisotropic mesh adaptation with solution of time-dependent problems. Next, we introduce and compare two different approaches in detail.

5.1 Immediate Remeshing

Probably the simplest approach to couple our anisotropic mesh adaptation methodology described in Section 3 with the numerical solution of time-dependent problems is the following. Once the HDG solution of (10) at a given time t^n is found, the mesh is adapted based on this solution, and the solution in time t^{n+1} is sought on the resulting adapted mesh. This algorithm is repeated every N_A -th time step of the simulation.

The algorithm is depicted in Fig. 2. The mesh adaptation step is straightforward. First, an optimal metric \mathcal{M}^{n+1} is computed based on the last known solution \mathbf{w}^n at time level t^n . The metric field \mathcal{M}^{n+1} is then passed to BAMG to generate mesh \mathcal{T}_h^{n+1} to be used in the subsequent time step. Once the new mesh \mathcal{T}_h^{n+1} is available, solution \mathbf{w}^n is transferred by the Galerkin projection (Section 4) from mesh \mathcal{T}_h^n to the new mesh \mathcal{T}_h^{n+1} , which is indicated by the operator $\mathcal{S}_n^{n+1}(\cdot)$.

The simplicity of this approach is, however, redeemed by the fact that such a procedure results in a mesh adaptation delay compared to the solution itself since the generated mesh is optimal only for the solution of the previous time step and not the solution of the subsequent time steps to be sought. The obvious way to minimize this effect is to adapt the mesh in every time step, corresponding to $N_A = 1$, which can cause computational overhead. On the other hand, when there are discontinuities or high gradients present in the flow field, the mesh adaptation has to be frequent in order to follow the moving local rapid changes by the mesh resolution and hence to accurately capture these phenomena.

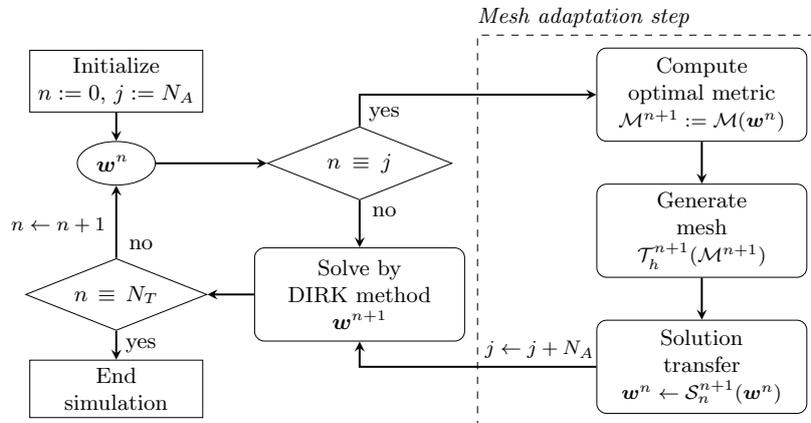


Figure 2: Anisotropic mesh adaptation with immediate remeshing for the solution of time-dependent problems

To achieve a reasonable initial element distribution within the mesh, we perform the anisotropic mesh adaptation based on the prescribed initial condition at the beginning of the simulation. The initial mesh adaptation is repeated several times for a given problem based on the user input.

5.2 A Numerical Example

To show the limits of the presented algorithm, we consider the rotating Gaussian problem from [18] for 2D scalar linear convection-diffusion equation of the form

$$\frac{\partial w(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u}(\mathbf{x})w(\mathbf{x}, t)) = \varepsilon \nabla^2 w(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0. \quad (32)$$

The test case involves rotational transport of a Gaussian pulse inside the domain $\Omega = [-1, 1]^2$. The exact solution with the velocity field $\mathbf{u}(x, y) = (-4y, 4x)$ is given by

$$w(x, y, t) = \frac{2\sigma^2}{2\sigma^2 + 4\varepsilon t} \exp\left(-\frac{(\hat{x} - x_c)^2 + (\hat{y} - y_c)^2}{2\sigma^2 + 4\varepsilon t}\right), \quad (33)$$

where

$$\hat{x} = x \cos(4t) + y \sin(4t) \quad \text{and} \quad \hat{y} = -x \sin(4t) + y \cos(4t). \quad (34)$$

The initial center of the pulse is chosen as $(x_c, y_c) = (-0.5, 0)$, the standard deviation of the Gaussian distribution is set to $\sigma = 0.05$ resulting in steep gradients close to its center, and the diffusivity constant is set to $\varepsilon = 1 \cdot 10^{-5}$ causing very small diffusion effects. Dirichlet boundary conditions taken from the exact solution are assumed at $\partial\Omega$.

We use $p = 2$ polynomials to represent the solution on each element of the mesh and we limit the number of the mesh elements to roughly 500. The structured initial mesh with 2048 triangular elements supplied to our solver is depicted in Fig. 3(a). The mesh is initially adapted three times based on the initial condition given by the exact solution, see Fig. 3(b) - 3(d). The close-up view of the area of interest is in the lower right corner of each figure. The representation of the initial condition on the third initially adapted mesh is illustrated in Fig. 3(e).

The problem is solved by the third-order DIRK(3,3) method and the final time is set to $T = \pi/4$, which corresponds to one-half rotation of the pulse in counterclockwise direction. We are using an adaptive time step size based on a classical CFL condition with $\text{CFL} = 1$ and we would like to analyze the effect of different number of time steps N_A , after which the mesh is anisotropically adapted. For some cases, especially when the size of the time step is adaptive, it is more convenient to prescribe an adaptation interval Δt_A , after which the mesh is adapted. In the following we will thus interchange the mesh adaptation after N_A time steps and after Δt_A time interval.

Note the implicit methods used for time integration in this work are generally stable for even higher CFL number but usually with negative effect on the accuracy of the numerical solution. Since here we are aiming at comparison of mesh-solution pairs after different adaptation intervals, such discrepancies would naturally translate to the mesh adaptation yielding significant and undesirable source of error.

To demonstrate how sensitive is the remeshing approach to the size of adaptation interval, we adapt the mesh after time intervals $T/12$, $T/16$, $T/32$, $T/64$, and $T/128$ where the denominator indicates the number of mesh adaptations and hence the number of required solution transfers. Additionally, we compute the solution with mesh adaptation in every single time step ($N_A = 1$).

The comparison of all computations is shown in Tab. 1. To compare adaptation after specified

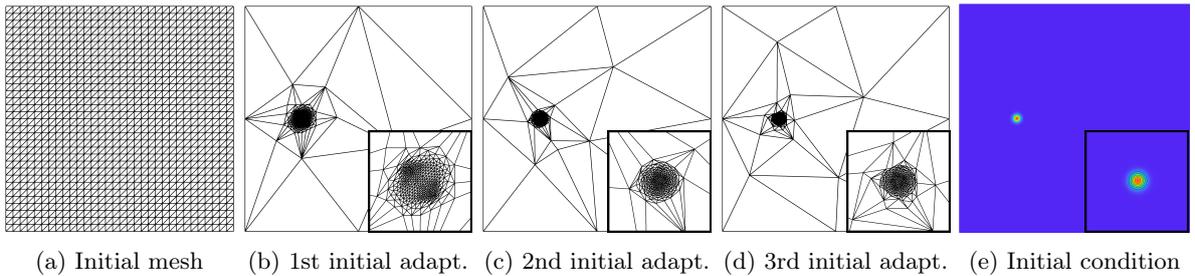


Figure 3: Initial mesh adaptation for the rotating Gaussian test case

number of time steps and time interval, we present also the average number of time steps between the adaptations for each case and the overall number of adaptations for the entire simulation (excluding the initial adaptations). We can see that the L_2 -norm error at the final time $\|w - w_h\|_2$ for the case with 11 adaptations ($\Delta t_A = T/12$) is by two orders higher than for the case with 127 adaptations ($\Delta t_A = T/128$). For this particular case, reducing the size of the adaptation interval below $T/128$ does not improve the accuracy since the error at the final time is comparable with adaptation in every time step. Moreover, the number of adaptations drastically increases without any benefit. Note the prescribed number of elements affects not only the resulting mesh element size but also the size of the time step via the CFL condition. This is why the average number of time steps in Δt_A is the same for $\Delta t_A = T/12$ and $\Delta t_A = T/16$.

The meshes at the final time with detail of the area of interest in the lower left corner are shown in Fig. 4. One can see how the increase of the adaptation interval results in more diffusive mesh resolution. While this problem can be circumvented by introducing the fixed point iteration in every adaptation interval similarly as in the steady case, the mesh-solution shift would still be present. Thus, this step would possibly generate even larger error since the mesh would have better approximating properties for the solution in the n -th time step only.

Table 1: Comparison of various adaptation time intervals for the immediate remeshing approach

	Adaptation interval Δt_A	Time steps within Δt_A	Number of adaptations	Average Δt	Error at final time
$\Delta t_A = T/12$	$6.545 \cdot 10^{-2}$	43	11	$1.528 \cdot 10^{-3}$	$1.658 \cdot 10^{-2}$
$\Delta t_A = T/16$	$4.909 \cdot 10^{-2}$	43	15	$1.148 \cdot 10^{-3}$	$7.643 \cdot 10^{-3}$
$\Delta t_A = T/32$	$2.454 \cdot 10^{-2}$	30	31	$8.472 \cdot 10^{-4}$	$9.845 \cdot 10^{-4}$
$\Delta t_A = T/64$	$1.227 \cdot 10^{-2}$	15	63	$8.338 \cdot 10^{-4}$	$2.950 \cdot 10^{-4}$
$\Delta t_A = T/128$	$6.136 \cdot 10^{-3}$	8	127	$8.294 \cdot 10^{-4}$	$1.723 \cdot 10^{-4}$
$N_A = 1$	$8.259 \cdot 10^{-4}$	1	950	$8.259 \cdot 10^{-4}$	$1.795 \cdot 10^{-4}$

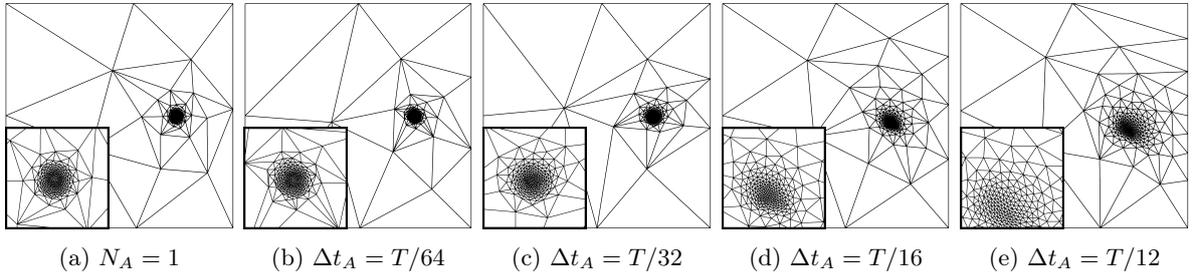


Figure 4: Anisotropically adapted meshes at the final time T with different adaptation intervals

5.3 Mesh Predictor

The immediate remeshing approach has the disadvantage that an adapted mesh is optimal only for the solution in the n -th time step whereas we are using the same mesh to seek the solution in $n + m$, $m \geq 1$ time steps. If we focus on the case $N_A = 1$, i.e. adaptation in every time step, to increase the robustness of a solver, the mesh has to be adapted based on a solution not only at time level t^n but also at t^{n+1} level to accurately capture the evolution of a numerical solution. Hence, we are looking for a mesh predictor, which would provide estimate of the solution w^{n+1} and which could be used to generate a mesh being optimal for the time interval $[t^n, t^{n+1}]$. Ultimately, we would like to be able to predict the solution for the time interval $[t^n, t^n + \Delta t_A]$, which minimizes the number of solution transfers and hence increases the accuracy compared to the immediate remeshing approach.

The need for a mesh predictor is motivated especially by the problems including moving shocks as the shock easily propagates out of the position, based on which the mesh has been previously adapted. In such a situation, the immediate remeshing is likely to fail since the shock has to be captured on a relatively coarse mesh.

Note that we can use the same method as before, DIRK(3,3), as the predictor of the solution. However, the predicted evolution of the solution does not need to be of high order since it is only used

to generate a mesh with better approximating properties. The predictor method should primarily have the following properties:

- robustness, i.e. to be stable on moderately coarse meshes,
- simplicity, i.e. the computational time of the predictor should be kept as low as possible,
- appropriate accuracy, i.e. to still be able to properly capture the solution.

Designing a suitable predictor method requires to make a compromise with respect to the properties listed above. Naturally, a good candidate is a time-integration method of order lower than the order the method used to obtain the time-dependent numerical solution. Second-order DIRK(2,2) method comprises of two stages and, based on our experience, tends to amplify possible oscillations around discontinuities in the solution.

Thus, we propose a BDF1-based mesh predictor as the backward Euler method is the simplest implicit method in considered class of time-integration methods. The quality of the adapted mesh will then greatly depend on the size of the adaptation interval. Therefore, to overcome the fact that BDF is only first-order accurate, we add the possibility of a fixed point iteration over a given adaptation time interval Δt_A . In the present work, we choose a fixed number of iterations N_{iter} for which the predictor step is repeated before a final mesh is obtained and used to seek the solution in the forthcoming time steps.

In order to generate a mesh, which is optimal for an entire time interval, we rely on the metric intersection algorithm devised by Alauzet et al. [17]. Formally speaking, we seek an ellipse corresponding to metric intersection $\mathcal{M}_1 \cap \mathcal{M}_2$, which lies in a geometrical intersection of ellipses corresponding to metrics \mathcal{M}_1 and \mathcal{M}_2 and has the maximal possible area. This algorithm is often used also for the case when a mesh adaptation is based on more than one flow variable. In our situation, we perform the intersection of metrics stemming from different time levels.

The anisotropic mesh adaptation with mesh predictor is depicted in Fig. 5. At the beginning of the adaptation step, an optimal metric $\mathcal{M}(\mathbf{w}^n)$ is computed based on the solution from the previous time step \mathbf{w}^n . Next, the mesh-predictor step is performed. In the first inner time step, we solve the discretized problem by the predictor method yielding numerical solution $\tilde{\mathbf{w}}^{n+1}$. Then we compute the optimal metric $\mathcal{M}(\tilde{\mathbf{w}}^{n+1})$ based on the solution $\tilde{\mathbf{w}}^{n+1}$. Finally, a metric intersection $\mathcal{M}(\mathbf{w}^n) \cap \mathcal{M}(\tilde{\mathbf{w}}^{n+1})$ is obtained followed by the next inner time step, in which the procedure is repeated. Once the time of the next mesh adaptation is reached for the predictor solution, the predictor time stepping is stopped. Note the nomenclature in Fig. 5 is simplified as some of the fields used in the algorithm can be reused for purpose of the implementation.

The result of the predictor step is the metric intersection

$$\mathcal{M}^{n+1} := \mathcal{M}(\mathbf{w}^n) \cap \mathcal{M}(\tilde{\mathbf{w}}^{n+1}) \cap \mathcal{M}(\tilde{\mathbf{w}}^{n+2}) \cap \dots \cap \mathcal{M}(\tilde{\mathbf{w}}^{n+N_A}), \quad (35)$$

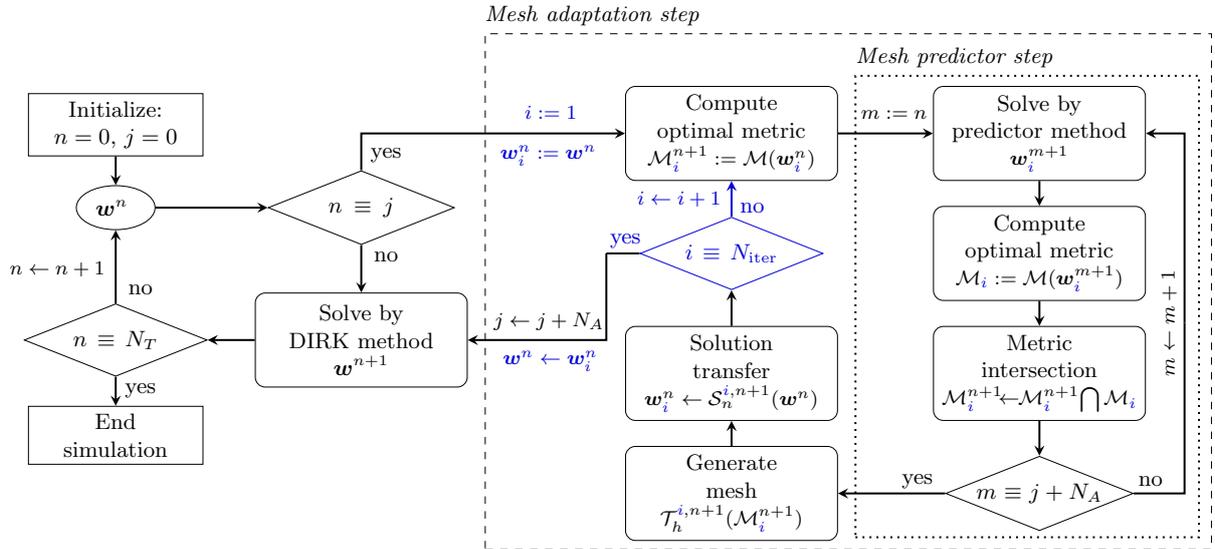


Figure 5: Anisotropic mesh adaptation with predictor for the solution of time-dependent problems

which is passed to BAMG to generate the adapted mesh. This mesh is now *pseudo-optimal* for the entire time interval $[t^n, t^{n+N_A}]$. Finally, the solution from the previous time step \mathbf{w}^n is transferred onto the new adapted mesh and the problem is solved on the adaptation interval by the DIRK method until the time of the next mesh adaptation is reached. In case of a fixed-point iteration, the predictor step is repeated N_{iter} times. The data in Fig. 5 marked in blue correspond to the case $N_{\text{iter}} > 1$.

Remark. Our anisotropic mesh adaptation methodology described in Section 3 relies on the prescribed number of elements of the adapted mesh, see equation (18). Performing multiple computations of optimal metric followed by their intersection has the effect of gradual increase of the overall number of mesh elements since every optimal metric is computed with prescribed number of desired mesh elements. To control the total number of elements, we further scale the metric intersection before passing it to BAMG. Thus, the prescribed number of elements is still met for the anisotropic mesh adaptation based on mesh predictor. However, this operation causes that the mesh resolution, i.e. the smallest mesh size, depends on the size of the prescribed adaptation interval.

Remark. Note the computation of an optimal metric in every single predictor time step is not necessary as instead of increasing the accuracy, it only results in increase of computational overhead. Based on our experience, we compute the optimal metric after a time interval which corresponds to $\text{CFL} = 5$, i.e. for adaptive time step size Δt with $\text{CFL} = 1$, the optimal metric is computed every five time steps or when the time of the next mesh adaptation is reached. We recall the CFL condition is of the classical form

$$\Delta t = \min_{\kappa \in \mathcal{T}_h} \left(\text{CFL} \frac{h_\kappa}{\lambda_{\max, \kappa}} \right), \quad (36)$$

where h_κ represents the size of element κ and $\lambda_{\max, \kappa}$ the maximal wave speed in element κ .

5.4 A Numerical Example Revisited

Returning back to the rotating Gaussian test case introduced in Section 5.2, we recompute the problem with the mesh-predictor adaptation approach where now we adapt the mesh after time intervals $T/16$, $T/8$, $T/4$, $T/2$, and T . The latter case corresponds to the situation, where the predictor is used to predict the evolution of the solution in the entire simulation time interval $[0, T]$.

Although the initial mesh adaptation can also be used for this approach, it is rather inappropriate for this particular test case. As the adaptation intervals are large, the initial mesh adaptation results in coarse mesh resolution in most of the domain, which negatively affects the accuracy of the BDF1 predictor. Instead, we use the structured mesh shown in Fig. 3(a) to obtain the first predictor solution.

Here, we set $N_{\text{iter}} = 5$. In Fig. 6, an example of the mesh predictor approach is shown. We show the adapted mesh after the first predictor step for all adaptation intervals Δt_A considered. Note how the smallest element size is affected while preserving the total number of elements, which is 500 as for the immediate remeshing approach. In Fig. 7, the evolution of the mesh after each fixed-point iteration for the case $\Delta t_A = T/2$ is shown. The need for robustness of the predictor method can be seen in prediction of the mesh for the second adaptation interval. Initially, the predictor has to be able to deliver a stable numerical solution in time interval $[T/2, T]$ on the mesh in Fig. 7(e), which has a very coarse resolution in the area of interest. Subsequently, the initial mesh iterate in Fig. 7(f) is generated and further improved in the following iterations.

An analogy of the Tab. 1 for the immediate remeshing approach is shown in Tab. 2. One can see that for $\Delta t_A = T/16$ the L_2 -norm error at the final time is by more than order lower for the BDF1 predictor than for the case of immediate remeshing with the same size of the adaptation interval.

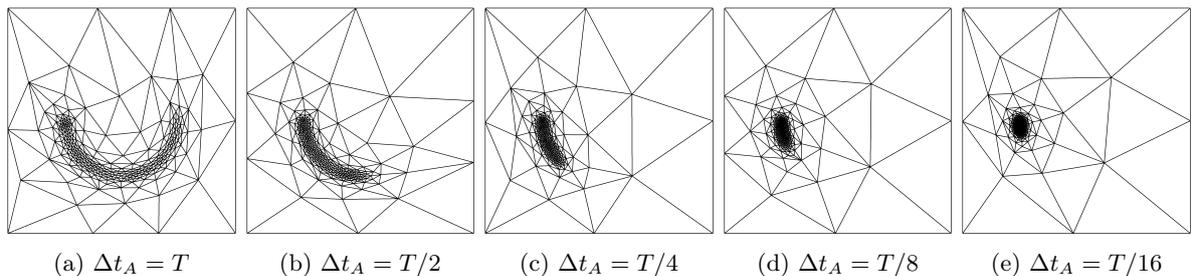


Figure 6: Mesh predictor approach. Mesh after 5th iteration for different adaptation intervals Δt_A .

Finally, we compare the effect of the number of fixed-point iterations N_{iter} and also the type of the mesh predictor. For all the previously defined adaptation intervals, we compute the test case with $N_{\text{iter}} \in \{1, 2, 3, 4, 5\}$, each with BDF1, DIRK(2,2), and DIRK(3,3) time-integration methods chosen as the predictor method. The results in terms of the L_2 -norm error at the final time are depicted in Fig. 8.

While the meshes in Fig. 7 indicate that after the fourth iteration the mesh is visually unchanged with the additional iteration, here we can see that the error is almost constant already for $N_{\text{iter}} \geq 3$ in case of BDF1 predictor with largest decrease of error between $N_{\text{iter}} = 1$ and $N_{\text{iter}} = 2$. The situation is very similar for the DIRK(2,2) and DIRK(3,3) methods used as the predictor, although the error levels are slightly lower. However, the enormous increase of computational time compared to BDF1 predictor cannot be redeemed by the accuracy gain.

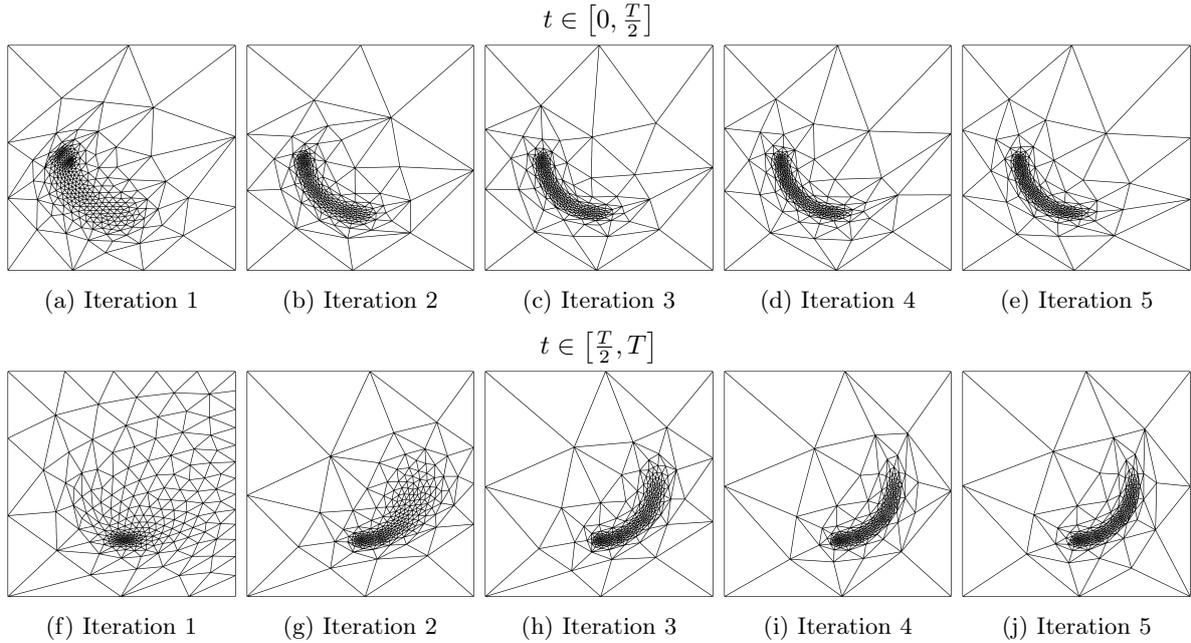


Figure 7: Mesh predictor approach. Adaptation interval is set to $T/2$. Initially, uniform triangular mesh with 2048 elements is used. The BDF1 predictor performs five iterations of the initial mesh (a) - (e). The solution is then sought by DIRK(3,3) method until $t = T/2$ is reached. Before every BDF1 predictor iteration (f) - (j), the DIRK(3,3)-solution is transferred onto the current mesh iterate. Once the 5th iteration is finished, the primary computation is resumed and the problem is solved by DIRK(3,3) until the final time T is reached.

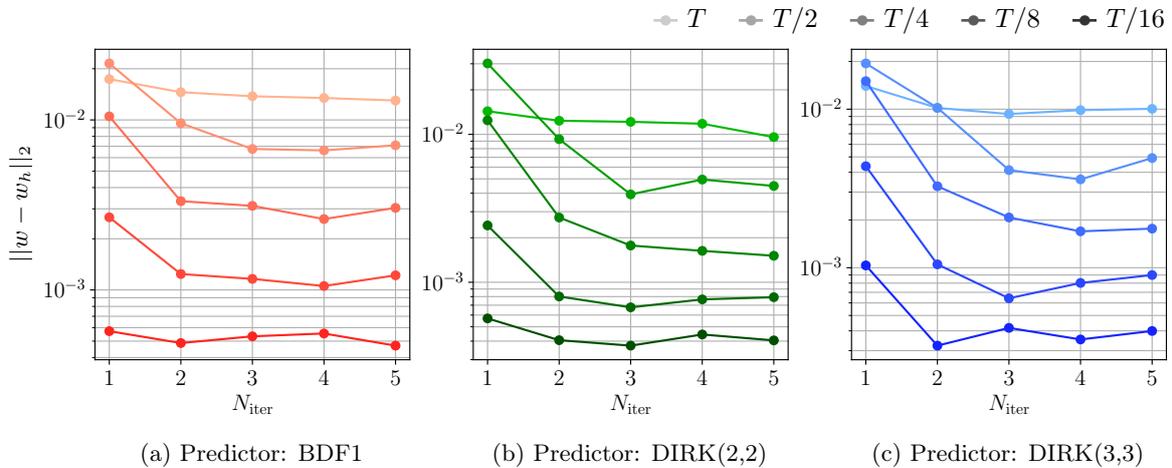


Figure 8: Mesh predictor approach. L_2 -norm error of the solution obtained by DIRK(3,3) method with mesh-predictors of different orders.

Table 2: Comparison of various adaptation time intervals for the mesh predictor approach

	Adaptation interval Δt_A	Time steps within Δt_A	Number of adaptations	Average Δt	Error at final time
$\Delta t_A = T$	$7.854 \cdot 10^{-1}$	206	1	$3.813 \cdot 10^{-3}$	$1.301 \cdot 10^{-2}$
$\Delta t_A = T/2$	$3.927 \cdot 10^{-1}$	130	2	$3.021 \cdot 10^{-3}$	$7.095 \cdot 10^{-3}$
$\Delta t_A = T/4$	$1.963 \cdot 10^{-1}$	77	4	$2.558 \cdot 10^{-3}$	$3.039 \cdot 10^{-3}$
$\Delta t_A = T/8$	$9.817 \cdot 10^{-2}$	52	8	$1.888 \cdot 10^{-3}$	$1.219 \cdot 10^{-3}$
$\Delta t_A = T/16$	$4.909 \cdot 10^{-2}$	36	16	$1.388 \cdot 10^{-3}$	$4.703 \cdot 10^{-4}$

6 Stabilization

In this work, we focus mainly on solutions of problems including moving shock waves. So far, we have seen that the mesh-solution pair obtained by the anisotropic mesh adaptation with mesh predictor tends to be more consistent with the exact solution than using the immediate remeshing approach. However, the choice of an unsteady mesh adaptation approach alone is not sufficient to achieve stability of the time-dependent numerical solutions containing discontinuities.

A necessary ingredient of high-order methods needed to stabilize the solution of such problems is some form of a shock-capturing mechanism. We return our attention to the HDG weak formulation given by equation (6). More specifically, we focus on definition of the shock-capturing term $\mathcal{N}_h^{\text{SC}}(\mathbf{q}_h, \mathbf{w}_h; \mathbf{v}_h)$.

Among other shock-capturing approaches, we follow the artificial viscosity method, which has been introduced in the context of discontinuous Galerkin methods by Persson and Peraire [25]. The idea is to introduce an artificial diffusion flux $\mathbf{f}_v = \varepsilon_{\text{AV}} \nabla \mathbf{w}$, which is proportional to a Laplacian of the solution \mathbf{w} with an artificial viscosity coefficient ε_{AV} when inserted to the conservation law (1). Now, the task at hand is to find an appropriate artificial viscosity field $\varepsilon_{\text{AV}}(\mathbf{x})$, $\mathbf{x} \in \Omega$ for given numerical method such that the numerical solution will be stable even when discontinuities are present in the flow field.

A lot of artificial viscosity definitions can be found in the literature [27, 28, 29] differing in either the variables used to quantify ε_{AV} or preprocessing of ε_{AV} before its use in assembly of the linearized system. To solve the nonlinear problem (10) by the Newton method, we use the automatic differentiation approach to assemble the Jacobian matrix. Hence, we seek an artificial viscosity, which can be computed locally based on the flow variables and, therefore, its derivatives with respect to the conservative variables are not compromised by nonlocal operations such as averaging or reconstruction based on e.g. the values at mesh vertices. Such operations would negatively affect the convergence rate or the stability of the Newton method.

Thus, we follow the approach of Moro et al. [30], which has been devised specifically for the HDG method. Let us consider the Euler equations with

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ E \end{pmatrix}, \quad \mathbf{f}_c = \begin{pmatrix} \rho \mathbf{u}^T \\ \rho \mathbf{u} \otimes \mathbf{u} + P \mathbf{I}_d \\ (E + P) \mathbf{u}^T \end{pmatrix}, \quad \mathbf{f}_v = \mathbf{0}, \quad (37)$$

where ρ is the density, $\mathbf{u} = (u, v)^T$ is the fluid velocity vector, E is the total energy, and \mathbf{I}_d is $d \times d$ identity matrix. Pressure P is related to the conservative variables by the equation of state of an ideal gas in the form

$$P = (\gamma - 1) \left(E - \frac{1}{2} \rho \mathbf{u} \cdot \mathbf{u} \right), \quad (38)$$

where γ is the ratio of specific heats and $\gamma = 1.4$ for air. Redefining the diffusion flux as $\mathbf{f}_v = \varepsilon_{\text{AV}} \nabla \mathbf{w}$, the artificial viscosity can be locally computed as

$$\varepsilon_{\text{AV}} = \frac{k_h h}{p} \sqrt{\mathbf{u} \cdot \mathbf{u} + c^2} f(\tilde{s}^*), \quad \tilde{s}^*(\mathbf{w}) = -\frac{k_h h}{p} \frac{\nabla \cdot \mathbf{u}}{c^*}, \quad f(\tilde{s}^*) = \frac{1}{\alpha} \log_{10} (1 + \exp(\alpha (\tilde{s}^* - \beta))), \quad (39)$$

where $k_h = 1.5$, $\alpha = 10^4$, $\beta = 0.01$, p is the order of the polynomials used to represent the solution, and $c^{(*)}$ is the (critical) speed of sound. Note the the shock sensor \tilde{s}^* is based on the idea that the presence of shocks is associated with strongly negative dilation, i.e. the divergence of velocity. The details can be found in [30]. To include the anisotropic nature of the mesh into the artificial viscosity definition, we

define the local length scale as $h = \frac{1}{\sqrt{\lambda}}$ where $\lambda = \max_i(\lambda_i)$, λ_i being the eigenvalues of the metric \mathbf{M} defined in Section 3, i.e. the diagonal of the matrix $\mathbf{\Lambda}$ in equation (12).

Note the Euler equations originally represent a purely inviscid mathematical model. Thus, introducing the artificial diffusion flux would significantly increase the computational cost since in such case an auxiliary variable \mathbf{q} representing the solution gradient is now an additional unknown, see equation (2). To avoid this necessity, we discretize the artificial diffusion flux in an incomplete manner by taking $\mathbf{q}_h = \nabla \mathbf{w}_h$ with the aid of the derivatives of the basis functions. More precisely, we define the shock-capturing term in (6) by

$$\mathcal{N}_h^{\text{SC}}(\mathbf{q}_h, \mathbf{w}_h; \mathbf{v}_h) = (\varepsilon_{\text{AV}}(\mathbf{w}_h, \mathbf{q}_h) \mathbf{q}_h, \nabla \mathbf{v}_h)_{\mathcal{T}_h}, \quad (40)$$

such that artificial viscosity of neighbouring elements does not communicate with each other as the integrals over element edges are neglected. A similar simplification has been used in the work of Hartmann, [27].

As we have seen in Section 5.4, the mesh predictor approach relies on a stable predictor time-integration method. The least favorable situation that can occur is that the predictor has to solve given problem on a mesh, which is not yet aligned with the solution anisotropy of the intended adaptation interval. In other words, when one has an accurately adapted mesh for time interval $[t^n, t^n + \Delta t_A]$, the task of the very first predictor iteration is to ensure a stable, most probably inaccurate solution can be obtained even in time interval $[t^n + \Delta t_A, t^n + 2\Delta t_A]$ followed by generation of the first mesh iterate. This problem is even more challenging in case of a presence of shock waves as the above defined shock-capturing approach may not be sufficient to stabilize the solution on a coarse section of the mesh. To ensure a reasonable mesh iterate will be generated, we apply a constant artificial viscosity value for the entire domain $\varepsilon_{\text{AV}}(\mathbf{x}) = \tilde{\varepsilon}_{\text{AV}} = \text{const}$ in the first predictor iteration. As our solver is nondimensional, we usually take $\tilde{\varepsilon}_{\text{AV}} \in (0.001, 0.1)$.

7 Numerical Results

In this section, we verify the accuracy and efficiency of the time-dependent anisotropic adaptation strategies, namely the immediate remeshing approach and the mesh predictor methodology. Verification of the order of convergence is illustrated on a problem for advection equation with smooth solution where the mesh adaptation is based on the solution itself. The rest of the numerical examples is focused on solution of problems for compressible Euler equations given by equations (1) and (37). All of these standard test problems include propagation of shock waves and the mesh is adapted based on density variable unless stated otherwise. Note the novel limited solution transfer introduced in Section 4 is applied also for the advection test case, despite the solution being smooth, as the sensor is not activated due to the mesh resolution and the solution smoothness. Thus, the limiting process does not corrupt the solution accuracy during its transfer.

7.1 Mixing of Hot and Cold Fronts

The first example is a 2D problem for a scalar linear advection equation of the form

$$\frac{\partial w(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\mathbf{u}(\mathbf{x}) w(\mathbf{x}, t)) = 0, \quad \mathbf{x} \in \Omega, \quad t > 0 \quad (41)$$

taken from [31]. On a domain $\Omega = [-4, 4]^2$, a high gradient region represented by hyperbolic tangent is initially prescribed. Moreover, a rotational velocity field of the form

$$\mathbf{u}(x, y) = \left(-\frac{y f_t}{0.385r}, \frac{x f_t}{0.385r} \right), \quad r = \sqrt{x^2 + y^2}, \quad f_t = \frac{\tanh(r)}{\cosh^2(r)}, \quad (42)$$

which has a swirling effect, is specified. The exact solution is given by

$$w(x, y, t) = -\tanh \left[y \cos \left(\frac{f_t}{0.385r} t \right) - x \sin \left(\frac{f_t}{0.385r} t \right) \right]. \quad (43)$$

Snapshots of the exact solution at time $t = 0$, $t = 2$, and the final time $T = 4$ are depicted in Fig. 9. In this test case, we would like to further analyze the properties of the immediate remeshing approach from

Section 5.1 and the mesh predictor approach described in Section 5.3 for the case of a smooth solution.

First, we compute the numerical solution using a static mesh for the entire simulation time having 128, 512, 2048, 8192, and 32768 elements. In the mesh adaptation case, we adapt the mesh in every timestep ($N_A = 1$) and limit the number of elements to 128, 254, 512, 1024, 1532, and 2048. The BDF1 mesh predictor step is performed with 3 fixed-point iterations. All these computations has been carried out for polynomial orders $p \in \{1, 2, 3, 4\}$ representing the solution on each element of the mesh. The fourth-order DIRK(5,4) method with adaptive time step size and CFL = 1 has been used for time-integration. The convergence plots for each polynomial order are shown in Fig. 10.

As expected, the order of convergence in case of the static mesh is optimal, $\mathcal{O}(p+1)$. However, in case of a time-dependent anisotropic mesh adaptation, the error reduction is much faster as the number of mesh elements is increased and the order of convergence is between $\mathcal{O}(p+1)$ and $\mathcal{O}(p+2)$. In addition, both time-dependent adaptation approaches perform similarly.

Next, we analyze the influence of the size of the adaptation interval. The L_2 -norm errors for $p = 1$ and

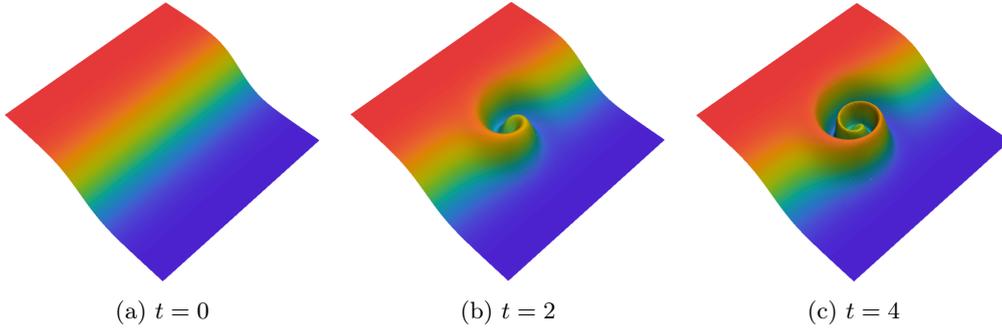


Figure 9: Exact solution of the mixing of hot and cold fronts test case. The contours range from $w = -1$ (blue) to $w = 1$ (red).

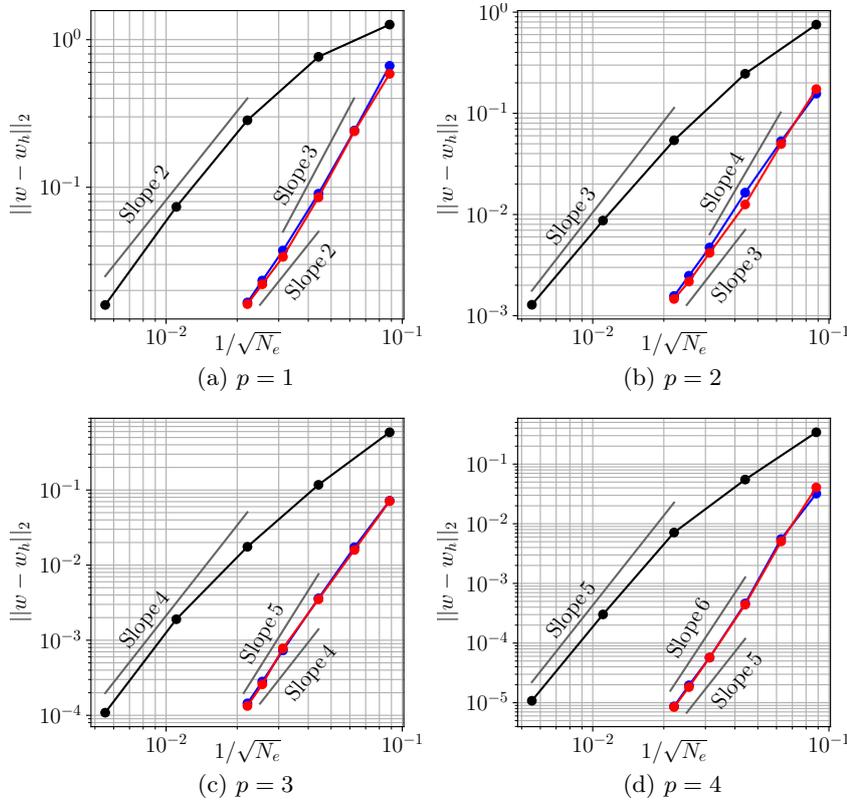


Figure 10: Convergence results in terms of L_2 -norm error of the numerical solution of the mixing of hot and cold fronts test problem. Static mesh (\bullet), adaptive mesh with immediate remeshing (\bullet), adaptive mesh with mesh predictor (\bullet).

$p = 4$ polynomials and the number of time steps $N_A \in \{1, 4, 8, 16, 32\}$, after which the mesh is adapted, are depicted in Fig. 11. One can see that increasing the size of the adaptation interval negatively affects the accuracy of both time-dependent adaptation approaches. However, the increase of error with the adaptation interval seems to be reasonable for the case of mesh predictor compared to the immediate remeshing approach.

An example of the mesh at the final time obtained with the BDF1 predictor and $N_A = 1$ is shown in Fig. 12(a). The mesh consistently follow the evolution of the underlying numerical solution. As the size of adaptation interval is increased ($N_A = 32$, Fig. 12(b)), the quality of the mesh resolution is slightly reduced, however, the anisotropy of the flow is still reflected by the element distribution. In case of the immediate remeshing with $N_A = 32$ in Fig. 12(c), the mesh at the final time shows several numerical artifacts causing reduction of the solution accuracy.

In Fig. 13, we present a work-precision diagrams in terms of the computational time for the case $N_A = 1$, $N_A = 8$, and $N_A = 32$ using $p = 4$ polynomials to represent the solution. Beside the static mesh, we compare the CPU times of immediate remeshing approach with the BDF1 mesh predictor with 1, 2, and 3 fixed-point iterations in each adaptation interval. The least demanding method for $N_A = 1$ is the immediate remeshing approach as the solution is smooth and the mesh-solution mismatch is almost suppressed.

The time-to-solution gradually increases with the number of fixed-point iterations in case of mesh predictor. However, all the time-dependent anisotropic mesh adaptation mechanisms perform more or less the same for the case with $N_A = 8$ and still outperform the use of a static mesh. As the size of the adaptation interval is further increased, the immediate remeshing approach with $N_A = 32$ is even more expensive than computation with the static mesh. This is not the case for BDF1 mesh predictor with increasing number of elements. One can also see that increase of the number of fixed-point iterations has very little effect on accuracy for this particular case.

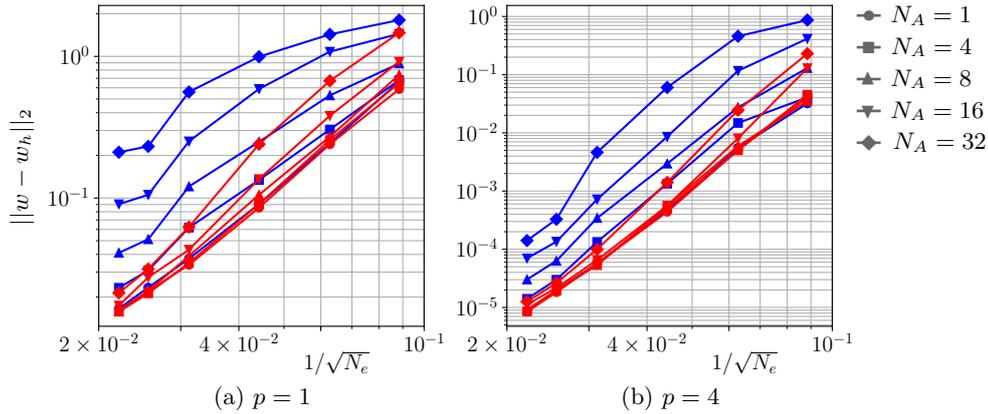


Figure 11: L_2 -norm error of the numerical solution of the mixing of hot and cold fronts test problem and for various number of time steps N_A , after which the mesh is adapted. Adaptive mesh with immediate remeshing (\bullet), adaptive mesh with mesh predictor (\blacklozenge).

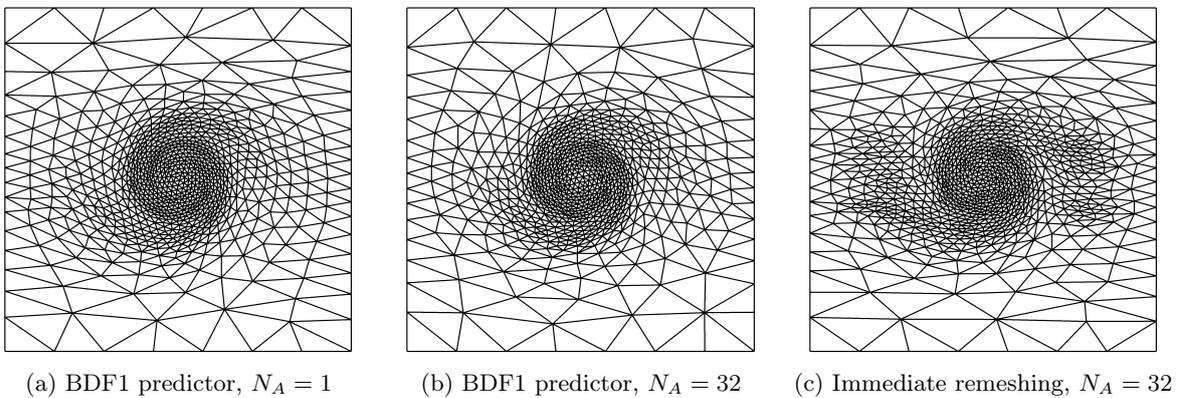


Figure 12: Example of the meshes at the final time of the mixing of hot and cold fronts test problem

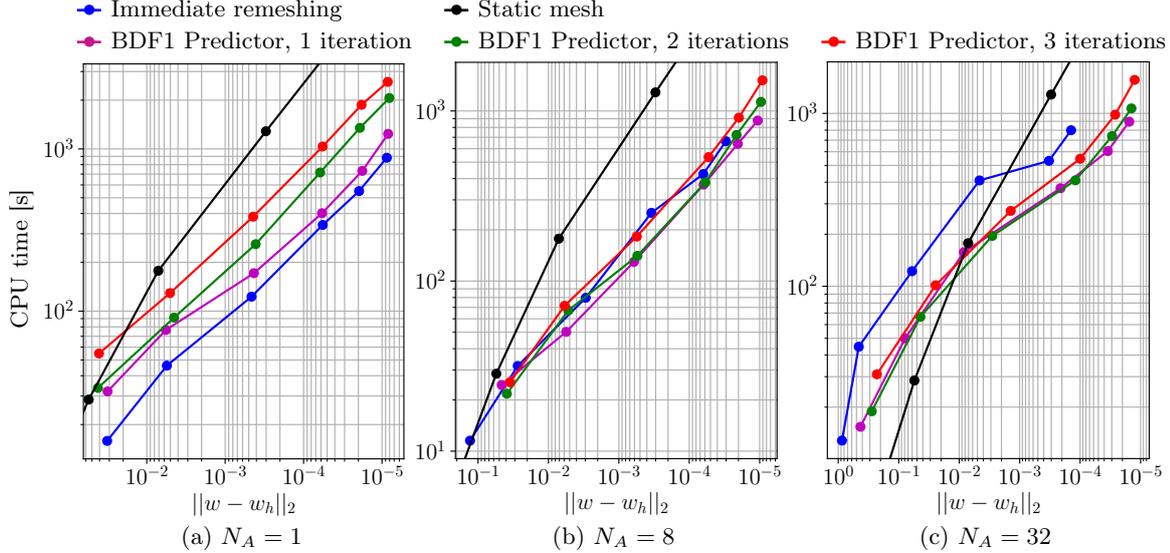


Figure 13: Mixing of hot and cold fronts test problem with $p = 4$ polynomials. Work-precision diagram in terms of CPU time.

7.2 Sod's Shock Tube

To investigate the capabilities of the two time-dependent anisotropic mesh adaptation approaches to handle discontinuities present in the solution, we consider the classical Sod's shock tube problem [32] extended to 2D. In domain $\Omega = [0, 1] \times [-0.05, 0.05]$, we solve the compressible Euler equations introduced by (37). The gas is initially at rest while there is a discontinuity in density and pressure along line $x = 0.5$. The two states are given by

$$(\rho, P) = \begin{cases} (1, 1), & x \leq 0.5, \\ (0.125, 0.1), & x > 0.5. \end{cases} \quad (44)$$

At later times, the solution develops a three-wave structure, namely the rarefaction wave, the contact discontinuity, and the shock wave.

We use $p = 2$ polynomials to represent the solution in space and first-order BDF1 method with $CFL = 1$ to evolve the solution in time until the final time $T = 0.25$ is reached. The use of high-order DIRK method will be discussed later. We compare the immediate remeshing approach and the BDF1 mesh predictor approach with three fixed-point iterations for number of time steps $N_A \in \{1, 2, 4, 8, 16\}$, after which the mesh is adapted. The solver is given an uniform mesh consisting of 1200 elements while we limit the number of elements of the resulting adapted meshes to 500. In the case of immediate remeshing, we perform three initial adaptations to obtain a reasonable initial mesh resolution to accurately represent the initial condition. When using the mesh-predictor approach, the initial adaptation is not necessary as the predictor itself will provide a reasonable mesh. However, we still perform a single initial mesh adaptation as it increases the resolution near the initial discontinuity and further reduce the number of elements, hence speeding up the predictor phase.

The plot of density along line $y = 0$ at the final time for all cases considered is depicted in Fig. 14. The exact solution to the problem is shown in black. One can see that there is no significant difference in using the immediate remeshing and the mesh predictor. Varying the number of time steps N_A , after which the mesh is adapted, also does not show notable deviation.

However, looking at the final meshes for different N_A in Fig. 15, we immediately see that the immediate remeshing approach loses the mesh-solution consistency as the number of time steps N_A is increased. In fact, even for $N_A = 1$, there are numerous small oscillations, which are detected by the optimal mesh metric computation and are further reflected in the adapted mesh topology. Due to this phenomenon, the number of mesh elements is slightly increased, which can be observed especially for $N_A = 8$, rendering the immediate remeshing an ineffective approach in case of problems with moving shock waves. In contrast, the resolution of the mesh properly follow the evolution of the solution in case of mesh predictor such that the mesh-solution pair remains consistent for the entire simulation.

To see how robust the mesh predictor approach is, we set the size of adaptation interval to $\Delta t_A = 0.025$ and increase the limit of mesh elements to 1000. In such situation, the number of solution transfers directly affecting the solution accuracy is only 10. Here, we again apply the limiting procedure consisting of finding the problematic elements and further application of the Barth-Jespersen limiter as described in Section 4.3. Note that with the previous setting and $N_A = 1$, the number of the limited solution transfers is slightly above 3000 for both the immediate remeshing and mesh-predictor approaches.

The time evolution of the mesh is depicted in Fig. 16(a). In Fig. 16(b), the mesh evolution within the last predictor step in time interval $[0.225, 0.250]$ is shown. Note the first mesh iterate is obtained by solving the problem with globally constant artificial viscosity as discussed in Section 6. The mesh also reflects the diffusion effects and consequently stabilizes the propagation of discontinuities in the next adaptation interval.

In Fig. 16(c), we also present the evolution of the mesh in the last predictor step in case of solution by the third-order DIRK(3,3) method. As can be seen, the adapted mesh is more clustered around both the contact discontinuity and the rarefaction wave than for the full BDF1 method. However, it can be observed from the Fig. 17 the contact discontinuity is captured more precisely but with spurious oscillations. The reason is that the artificial viscosity approach of Moro et al. [30] does not detect the contact discontinuities at all as the sensor is sensitive to shock waves only.

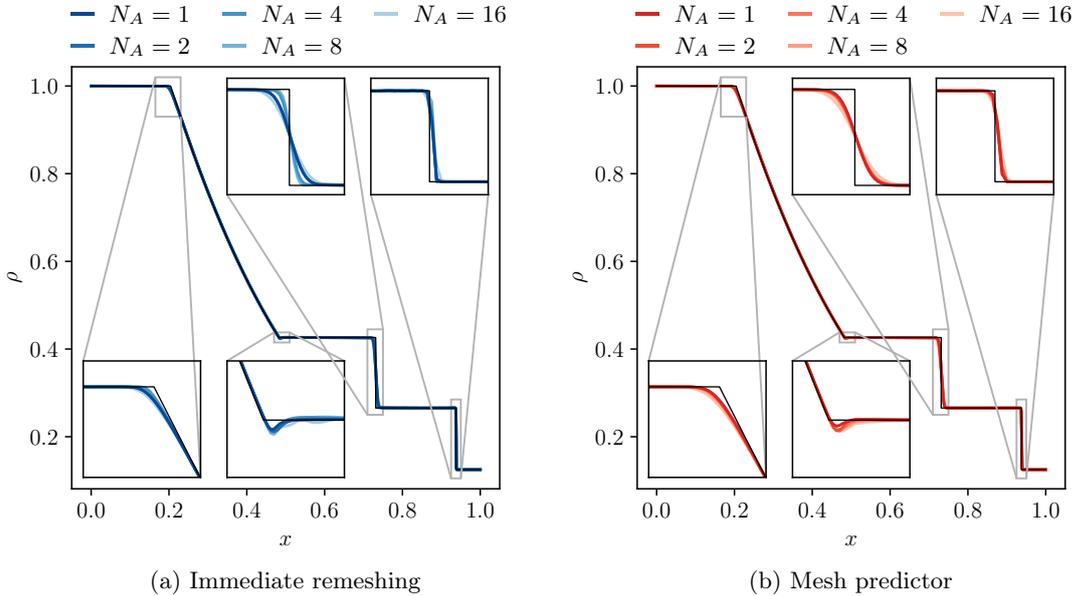


Figure 14: Sod's shock tube problem. Density at final time for the immediate remeshing approach (a) and the mesh-predictor approach (b) with a limit of 500 elements for the adapted mesh.

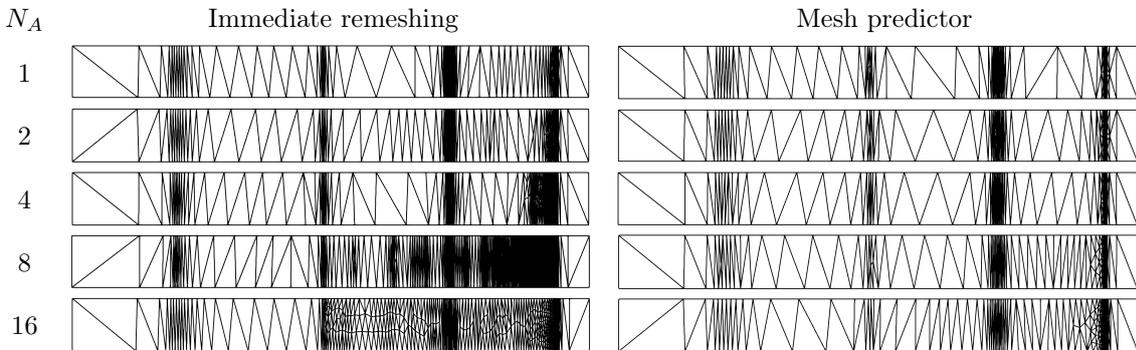


Figure 15: Sod's shock tube problem. Meshes at the final time with mesh adaptation after different number of time steps N_A .

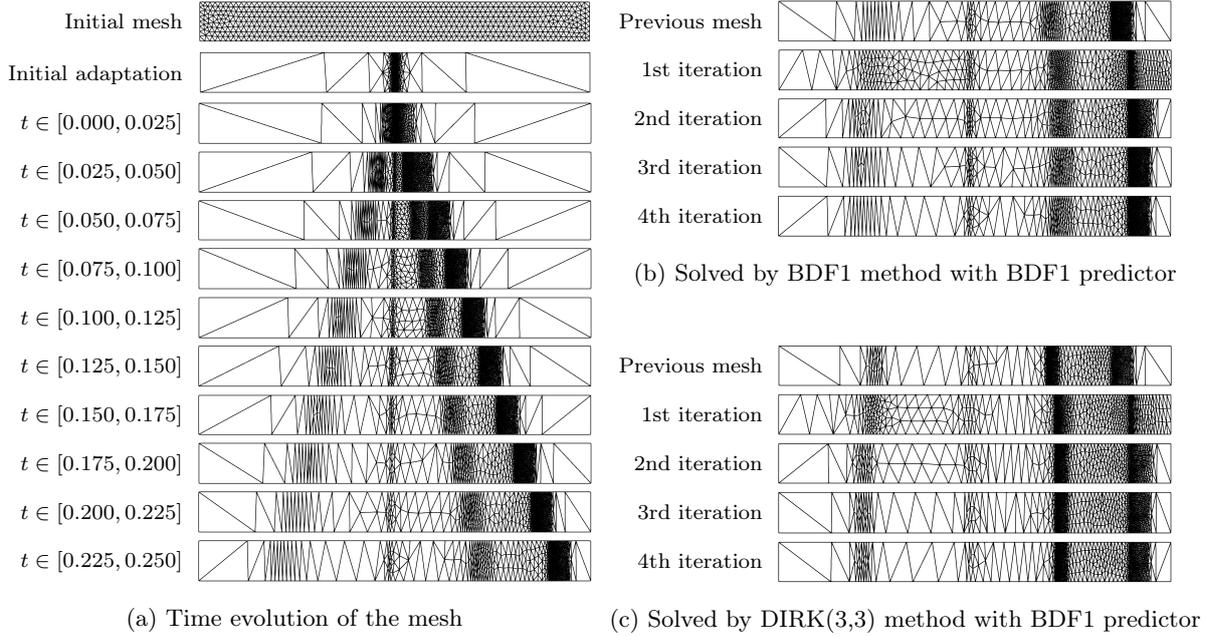


Figure 16: Mesh-predictor adaptation approach for the Sod's shock tube problem with adaptation interval $\Delta t_A = 0.025$ and 1000 elements. Meshes used in distinct adaptation intervals (a), mesh evolution in the last BDF1 predictor step, i.e. in time interval $[0.225, 0.250]$, when the problem itself is solved by the BDF1 (b) and DIRK(3,3) (c) methods, respectively.

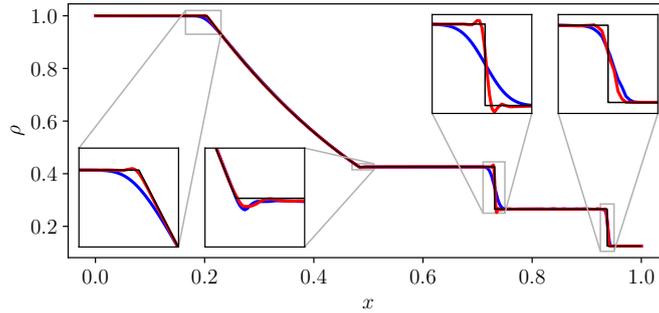


Figure 17: Density at the final time for the Sod's shock tube problem solved by first-order BDF1 method (—) and third-order DIRK(3,3) method (—), both using BDF1 mesh-predictor approach for time-dependent anisotropic mesh adaptation.

7.3 Explosion Problem

In the last two examples, we consider only the BDF1 mesh predictor approach with three fixed-point iterations. To show how our methodology handles solutions with nonplanar shocks, we consider the radially symmetric version of the Sod's problem, which is often called the explosion problem [33]. At time $t = 0$, inside a circular region of radius $r = 0.5$, high density and high pressure compared to its exterior are prescribed similarly as for the Sod's shock tube test case (44). As the solution is symmetric along lines $x = 0$ and $y = 0$, we consider only one quarter of the original domain $[-1, 1]^2$, i.e. $\Omega = [0, 1]^2$. With $p = 2$ polynomials, adaptive time step size with CFL = 0.8 and a limit of 4000 elements for the mesh adaptation, we solve the problem using both BDF1 and DIRK(3,3) methods until the final time $T = 0.25$ is reached. We set the size of the adaptation interval to $\Delta t_A = 0.005$ resulting in 50 limited solution transfers directly affecting the solution accuracy.

Considering first the results of the DIRK(3,3) method, the adapted mesh obtained in the last predictor step is shown in Fig. 18(a). The logarithm of artificial viscosity field at the final time can be seen in Fig. 18(b). The maximum of the artificial viscosity is reached close to the shock wave and has magnitude of approximately 10^{-3} . Note the low nonzero artificial viscosity values are also detected on the outermost rarefaction wave end. In Fig. 18(c), we show the elements where the projection has been limited during

the last solution transfer, at $t = 0.245$. The limiting is active primarily at the position of the shock wave. Density contours at the final time are depicted in Fig. 18(d).

In Fig. 18(e), we compare the reference solution with the numerical solution obtained by BDF1 and DIRK(3,3) methods. The resulting density exhibits similar characteristics as in the case of Sod's shock tube problem in Fig. 17. The reference solution shown in black is obtained by solving an equivalent 1D radially symmetric problem using a second-order finite volume method with very fine uniform grid.

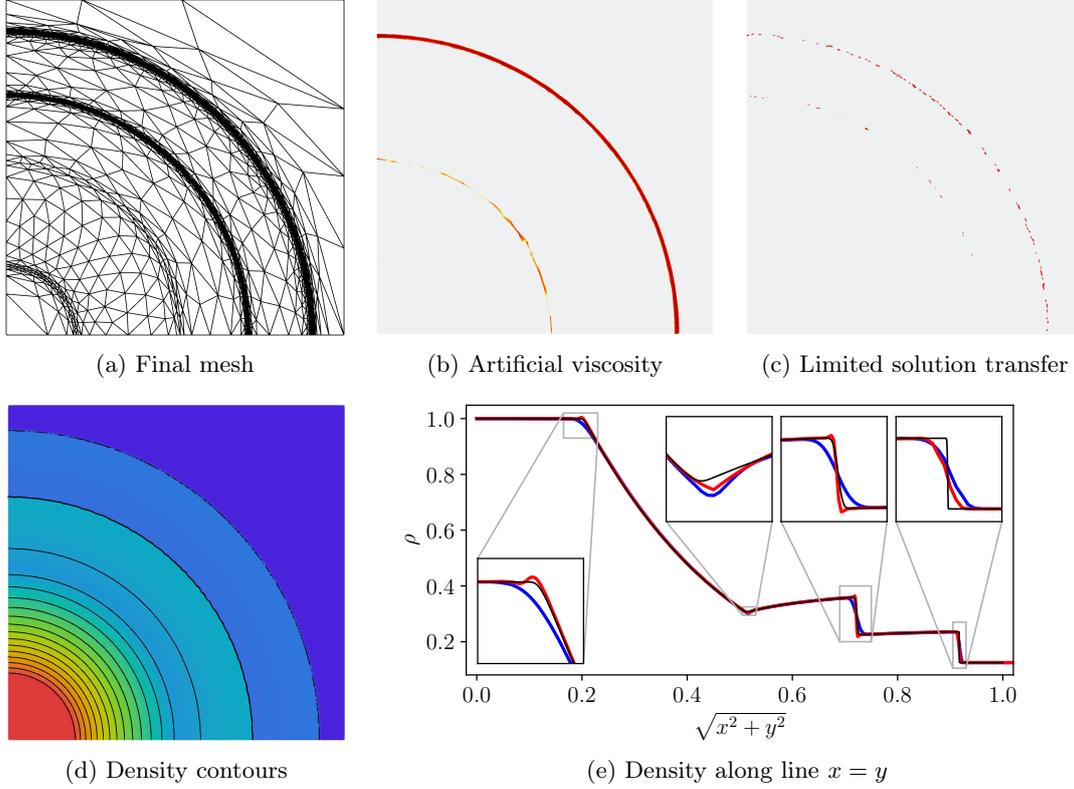


Figure 18: The results of the Explosion problem at the final time. The density plot show in (e) compares the reference solution (—) with the HDG solution obtained by BDF1 (—) and DIRK (3,3) (—) methods, both with BDF1 mesh predictor.

7.4 Shock Diffraction Problem

In the last test case, we consider diffraction of a shock wave over a backward facing step, which has been analyzed in [34] or [35]. The computational domain for this problem is $\Omega = [0, 1] \times [6, 11] \cup [1, 9] \times [0, 11]$. Note this test case is usually initialized with a right-moving shock wave of Mach number $M_s = 5.09$ located at $x = 0.5$ and $6 \leq y \leq 11$. Here, we would like to observe the ability of the present time-dependent adaptation approach to adapt the mesh with regard to various anisotropic features present in the flow field.

Hence, we assume two cases, $M_s = 1.3$ and $M_s = 2.0$. Similar calculations were conducted in the work by Ripley et al. [36]. The flow pattern shows more complex structure as the shock Mach number is increased, ranging from a vortex formed right behind the corner to the presence of a secondary shock wave. The pre-shock state is given by $(\rho_R, P_R, u_R, v_R) = (1.4, 1, 0, 0)$, whereas the post-shock state is obtained by Rankine-Hugoniot relations. We set an inflow boundary condition at $x = 0, 6 \leq y \leq 11$, and an outflow boundary condition at $x = 13, 0 \leq y \leq 11$ and $1 \leq x \leq 9, y = 0$. For the rest of the boundary, we prescribe a symmetry boundary condition. For both shock Mach numbers considered, we use $p = 2$ polynomials and adaptive time step size with $CFL = 0.5$. The mesh is adapted based on gas velocity magnitude after a time interval of size $\Delta t_A = 0.05$.

The resulting density contours and adapted meshes consisting of only 1000 elements for $M_s = 1.3$ and 2000 elements for $M_s = 2.0$ are shown in Fig. 19(a) and Fig. 20(a), respectively. Similarly as for the explosion test case in previous section, the initial discontinuity is resolved on a mesh having a given number of elements. As time marches on, more complex wave structure is developed although the limit

of the number of mesh elements remains unchanged. Therefore, the resolution of the flow phenomena is necessarily less accurate than at the beginning of the simulation.

A robust time-accurate anisotropically adaptive solver should be able to respond to the expansion or compression of the waves. Hence, to address this problem to some extent, we consider the following simple procedure to dynamically adapt the number of mesh elements to properly capture the characteristics of the flow at given time. At the beginning of the computation, we calculate the total area of elements A_{init}^* where the shock sensor given by equation (39) is activated. Just before the next mesh adaptation, we modify the desired number of elements, denoted by N in equation (18), by enforcing the ratio A^*/N remains fixed throughout the simulation. In other words, we set

$$N = \frac{A^*}{A_{\text{init}}^*} N_{\text{init}}. \quad (45)$$

before every predictor step. Obviously, features of the flow field different from shock waves are not taken into account in this approach. The results with the adaptive element number are depicted in Fig. 19(b) and Fig. 20(b).

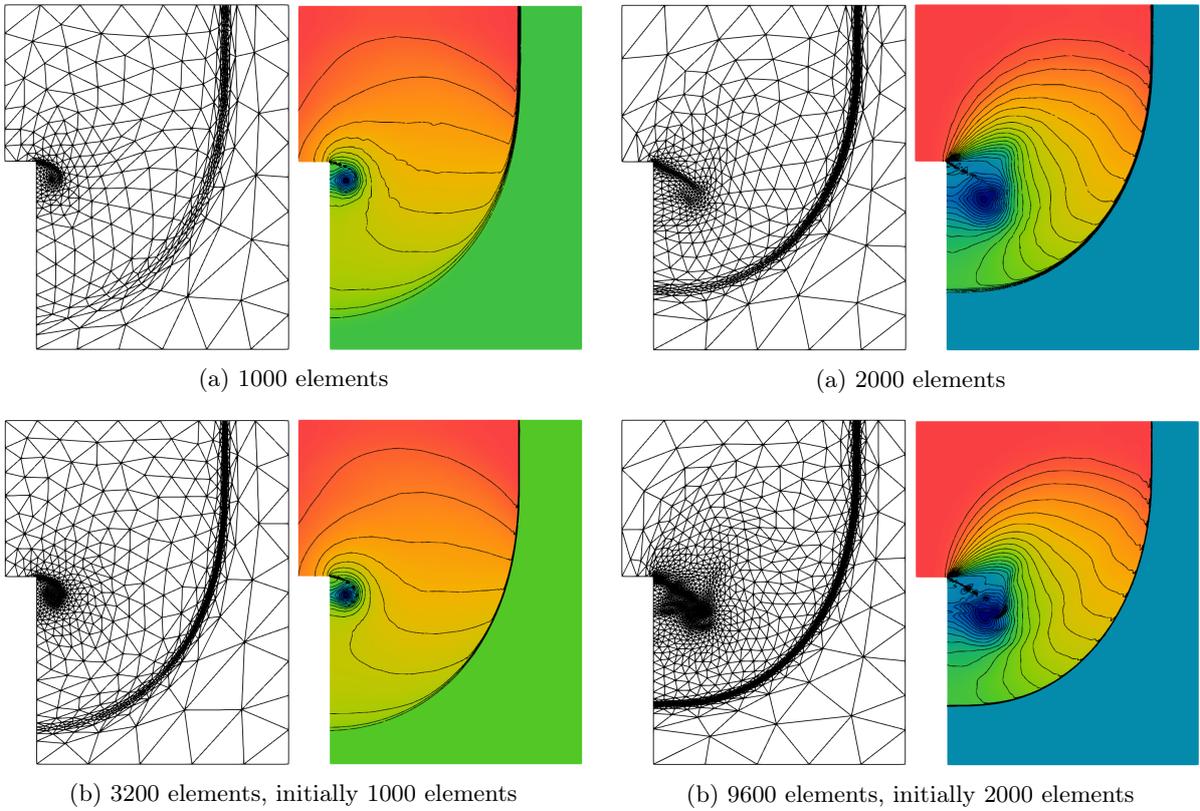


Figure 19: Shock diffraction with $M_s = 1.3$. Mesh and density contours at time $t = 5$, $\rho \in [0.7, 2.1]$.

Figure 20: Shock diffraction with $M_s = 2$. Mesh and density contours at time $t = 3.5$, $\rho \in [0.3, 3.7]$.

8 Conclusion and Future Work

In this paper, we have presented an improved time-dependent anisotropic mesh adaptation algorithm for high-order methods based on a lower-order solution prediction with focus on propagating shock waves emerging in initial-boundary value problems for compressible Euler equations. Additionally, we have introduced a novel bounded Galerkin projection algorithm for the transfer of the solution between anisotropically adapted meshes.

Numerical results show the mesh predictor approach produces a consistent mesh-solution pair, whereas the immediate remeshing, in which the mesh is adapted based on a solution from the previous time step, fails in the case of moving shock waves. However, both approaches outperform the use of static meshes and achieve similar accuracy for smooth solutions with moderate CFL numbers used to estimate the size

of an adaptive time step and fairly frequent mesh adaptations. The immediate remeshing approach loses accuracy as the adaptation interval is increased, which can ultimately destabilize the solution process.

Furthermore, the limited variant of the solution transfer does not disrupt the order of the solution in case of smooth flow. For discontinuous solutions, it rather helps to stabilize the high-order numerical solution. As the artificial viscosity approach is somewhat ad-hoc numerical technique, it is by itself not able to guarantee a nonphysical solutions will be avoided during the solution process whenever strong shocks are present in the flow field. Hence, the future work will focus on the improvement of the stabilization of the high-order solution by means of more sophisticated algorithms utilizing some form of positivity-preserving property [37].

Moreover, the mesh predictor approach using fixed-point iteration method to sequentially improve the adapted mesh resolution is only controlled by given number of iterations in this work. The prediction of the solution could also be used to estimate the spatial error and to terminate the iteration process when a given error tolerance is reached. The adaptive time step size could be found in a similar way.

Acknowledgments

The first author appreciate the financial support provided by the student project SGS-2022-008 of the University of West Bohemia.

References

- [1] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, F. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P. O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.
- [2] F. Alauzet and A. Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, 72:13–39, 2016.
- [3] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, Berlin, 1993.
- [4] P. Caplan, R. Haimes, and D. Darmofal. Four-Dimensional Anisotropic Mesh Adaptation. *Computer-Aided Design*, 129:102915, 07 2020.
- [5] J. F. Remacle, X. Li, M. Shephard, and J. Flaherty. Anisotropic Adaptive Simulation of Transient Flows using Discontinuous Galerkin Methods. *Int. J. Numer. Meth. Engng.*, 00:1–6, 01 2000.
- [6] F. Alauzet, A. Loseille, and G. Olivier. Time-accurate multi-scale anisotropic mesh adaptation for unsteady flows in CFD. *Journal of Computational Physics*, 373(C):28–63, 2018.
- [7] F. Alauzet. A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 299:116–142, 2016.
- [8] 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(23):8841–8855, 2009.
- [9] A. M. Rangarajan, A. Balan, and G. May. Mesh adaptation and optimization for discontinuous Galerkin methods using a continuous mesh model. AIAA Modeling and Simulation Technologies Conference, 2016.
- [10] V. Dolejší. Anisotropic hp -adaptive method based on interpolation error estimates in the L^q -norm. *Applied Numerical Mathematics*, 82:80–114, 2014.
- [11] A. Loseille and F. Alauzet. Continuous mesh framework part II: Validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011.
- [12] P. L. George and H. Borouchaki. *Delaunay triangulation and meshing: Application to finite elements*. Hermès, 1998.
- [13] P. E. Farrell and J. R. Maddison. Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, 200:89–100, 2011.
- [14] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman, and C. R. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2632–2642, 2009.
- [15] J. Carpio and J. L. Prieto. An anisotropic, fully adaptive algorithm for the solution of convection-dominated equations with semi-Lagrangian schemes. *Computer Methods in Applied Mechanics and Engineering*, 273:77–99, 2014.

**Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024**

- [16] A. Rangarajan and G. May. *Metric Construction for Error Control of Finite Element Solutions*. AIAA Aviation 2019 Forum.
- [17] F. Alauzet, P. J. Frey, P. L. George, and B. Mohammadi. 3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations. *Journal of Computational Physics*, 222(2):592–623, 2007.
- [18] 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(9):3232–3254, 2009.
- [19] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems*. Springer-Verlag, Berlin, 1996.
- [20] R. Alexander. Diagonally implicit Runge–Kutta methods for stiff O.D.E.’s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.
- [21] J. R. Cash. Diagonally implicit Runge-Kutta formulae with error estimates. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)*, 24(3):293–301, 1979.
- [22] A. Loseille and F. Alauzet. Continuous mesh framework part I: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011.
- [23] F. Hecht. BAMG: Bidimensional anisotropic mesh generator. Technical report, INRIA, Rocquencourt, France, 1998.
- [24] T. Levý and G. May. Conservative Solution Transfer Between Anisotropic Meshes for Adaptive Time-Accurate Hybridized Discontinuous Galerkin Methods. AIAA SciTech 2023 Forum, 2023.
- [25] P. O. Persson and J. Peraire. Sub-Cell Shock Capturing for Discontinuous Galerkin Methods. 44th AIAA Aerospace Sciences Meeting and Exhibit, 2012.
- [26] T. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. 27th Aerospace Sciences Meeting, 1989.
- [27] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.
- [28] G. E. Barter and D. L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, 2010.
- [29] P. O. Persson. *Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems*. 21st AIAA Computational Fluid Dynamics Conference.
- [30] D. Moro, N. C. Nguyen, and J. Peraire. Dilation-based shock capturing for high-order methods. *International Journal for Numerical Methods in Fluids*, 82(7):398–416, 2016.
- [31] V. Juntasaro and A. J. Marquis. Comparative Study of Flux-limiters Based on MUST Differencing Scheme. *International Journal of Computational Fluid Dynamics*, 18:569–576, 10 2004.
- [32] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, 1978.
- [33] R. Liska and B. Wendroff. Comparison of Several Difference Schemes on 1D and 2D Test Problems for the Euler Equations. *SIAM Journal on Scientific Computing*, 25(3):995–1017, 2003.
- [34] R. Hillier. Computation of shock wave diffraction at a ninety degrees convex edge. *Shock Waves*, 1(2):89–98, 1991.
- [35] B. Cockburn and C. W. Shu. The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *Journal of Computational Physics*, 141(2):199–224, 1998.
- [36] R.C. Ripley, F. S. Lien, and M. M. Yovanovich. Numerical simulation of shock diffraction on unstructured meshes. *Computers & Fluids*, 35(10):1420–1431, 2006.
- [37] T. Qin and C. W. Shu. Implicit Positivity-Preserving High-Order Discontinuous Galerkin Methods for Conservation Laws. *SIAM Journal on Scientific Computing*, 40(1):A81–A107, 2018.