Towards a Simulation System for Virtual Flight - Dynamic High-Order Overset Grids Method

Moritz Spraul[*], Anton Stephan[†], Frank Holzäpfel[‡]
Institut für Physik der Atmosphäre, Deutsches Zentrum für Luft- und Raumfahrt, 82234 Oberpfaffenhofen, Germany

A virtual flight through a realistic atmosphere is of vital interest not only for aircraft design but also for airports and airlines regarding runway capacity and safety. In the future virtual flight shall allow a simulation of a flight through a realistic atmosphere including the reaction of the aircraft to aerodynamic forces and the resulting trailing vortices from their generation until final decay. For realization, a bidirectional coupling between the atmospheric domain and a moving aircraft-centered domain is required. In this paper, we present a dynamic high-order overset grids method including hole-cutting, implemented in a discontinuous Galerkin solver for the compressible Navier-Stokes equations. Using discontinuous Galerkin for spatial discretization removes the requirement for halo elements at the artificial boundaries and simplifies the construction of the interface. The number of solution interpolation points at the interface is lower than in conventional Finite-Volume schemes using volume interpolation, resulting in less computational effort. The method can handle any kind of movement on a trajectory altered during run-time. Special emphasis is put on the ability of the interface to transfer turbulent structures without significant disturbance. The presented test cases include a 3D convergence test and the Taylor-Green vortex, both with a grid moving along a non-trivial trajectory.

I. Nomenclature

χ	=	Cells in receiver mesh
e	=	Energy density
$\boldsymbol{F}^C$	=	Convective flux
$\boldsymbol{F}^D$	=	Diffusive flux
δ_{ij}	=	Kronecker delta
$\boldsymbol{\xi}$	=	Coordinate in reference space
E_k	=	Kinetic energy
$\mathcal{E}$	=	Enstrophy
ϵ	=	Dissipation rate
J	=	Jacobian of the mapping
k	=	Heat conductivity
κ	=	Ratio of specific heats
l	=	Lagrangian basis functions
L	=	Distance
Ma	=	Mach number
μ	=	Dynamic viscosity
N	=	Polynomial degree
$\boldsymbol{n}$	=	Face normal vector
Ω	=	Cells in donor mesh
p	=	Static pressure
ψ	=	Test functions
$\boldsymbol{q}$	=	Heat flux vector

[*] PhD Candidate (Moritz.Spraul@dlr.de)

[†] Research Scientist

[‡] Senior Scientist, Associate Fellow AIAA

Re	=	Reynolds number
ρ	=	Density
R	=	Specific gas constant
$\tilde{R}$	=	Reference element
$\boldsymbol{\tau}$	=	Stress tensor
T	=	Static temperature
$\boldsymbol{U}$	=	Vector of conserved variables
$\boldsymbol{v}$	=	Vector of velocity components
V	=	Volume
VF	=	Virtual Flight
$\boldsymbol{x}$	=	Coordinate in physical space
ζ	=	Surface

II. Introduction

The realization of a virtual flight (VF) is a vital step towards a virtual product, designating a digital representation of the physical aircraft using computer-aided design and computational fluid dynamics (CFD). Integrated simulation environments already cover important areas of aircraft development, including aerodynamic properties. However, a true virtual product is only achieved when an aircraft is also integrated into the simulation of a realistic atmosphere. This important component is located at the interface between atmospheric physics and vehicle design and can only be served holistically.

The successful implementation of a virtual flight simulation environment will answer many scientific questions referring to the interaction of the aircraft with complex atmospheric structures. First, structured and coherent disturbances in the form of gusts, gravity waves and aircraft wake vortices[1], a special case of which presents the aerodynamics of formation flight [2]. Second, unstructured disturbances like different types of turbulence, including convective turbulence, breaking gravity waves, thunderstorm-induced turbulence, boundary layer and wake turbulence. Furthermore, the virtual flight will enable the virtual design of the aircraft with regard to extreme events and fatigue strength, as well as predictive maintenance by replicating realistic load cases, without the imminent need for flight experiments. This could not only enhance the knowledge of flight characteristics including the properties of the generated wake vortices but also increase efficiency by vastly accelerating the optimization process and supporting the specific development of control mechanisms and on-board sensors. At last, VF potentially will be a valuable tool to simplify the certification process, through computational validation.

A key aspect of VF is the coupling of an atmospheric with an aerodynamic simulation on moving grids. The use of two grids allows the specific use of different numerical models for aircraft and atmosphere, which significantly increases the efficiency of the simulation [3]. The atmospheric component is governed by atmospheric turbulence, and it may also include the wake induced by some vortex generator. On the other hand, the aerodynamic part is characterized by the boundary layer of a moving geometry and its disturbance through the atmosphere. Interaction between the two regimes may lead to a change of flight trajectories driven by aerodynamic forces and moments, which are to be captured by an appropriate scheme.

The realistic simulation of VF poses a challenging task in the field of computational fluid dynamics, given the multiscale nature of the problem in time and space. The relevant time scales range from fast vortex generation and roll up, through a comparatively long phase of persistence, and finally, the decay of the vortex by disturbances such as the Crow instability [4]. For the investigation of the interaction of an aircraft with the wake generated by a leading aircraft, a vortex of a certain age is needed. An aircraft centered reference frame would need an excessively large computational domain to generate such a vortex. This is deemed all the more unfeasible regarding computational cost for a simulation covering the whole vortex evolution from vortex generation until final decay. However, a setup comprising a ground-fixed domain with one or more aircraft flying through is a viable solution using the computational resources available today [5, 6].

Moving geometries can be found in many engineering problems e.g. helicopters, gas, and wind turbines. One way to realize these movements in computational fluid dynamics is the Arbitrary Lagrangian-Eulerian technique [7]. It is applicable for rather small displacements only, since the connectivity between grid cells is not to be changed or dynamic re-meshing is necessary. For larger shifts two approaches, sliding mesh or overset grids (chimera) methods, are suitable. For the sliding mesh method the domain is subdivided into non-overlapping subdomains [8–10]. Movement is only possible along the common interfaces of these partitions. Since aircraft affected by wake vortices experience large translatory movements as well as rolling moments, the sliding mesh technique is regarded as insufficient to capture this complex behavior. Overset grids connect an arbitrary number of computational grids by a set of overlapping mesh elements. Coupling is achieved by interpolation of the conserved variables in overlapping areas [11–13]. The movement of the different grids is completely independent of each other and without any restrictions regarding direction and type of motion. Therefore, an overset grids method was selected to further investigate the interaction between external disturbances like turbulence or wake vortices and aircraft.

High-order methods bring along superior accuracy and efficiency compared to conventional methods for computational fluid dynamics and high-performance computing [14]. Many of these approaches like WENO schemes employ rather large stencils for solution reconstruction to obtain a high-order [15]. However, a small spatial stencil is desired for overset grids methods in order to simplify the construction of the chimera interface [16]. The discontinuous Galerkin (DG) method approximates the solution in space by cell-local polynomials. Coupling between the cells is realized by a numerical flux obtained by approximate Riemann solvers [17, 18]. To obtain the flux, only information from direct neighbors is needed. Hence, for the extension to overset grids the flux computation procedure can be used without modification. The only adaption is the interpolation of the adjacent solution at a chimera boundary from other grids. Naturally, we choose DG for spatial discretization, as it simplifies the design of the chimera interface. In addition, this scheme features desirable properties like low numerical dissipation and a compact spatial stencil [19]. With the outlined tool, a high fidelity simulation of an aircraft interacting with atmospheric disturbances is deemed to be reachable in order to enable the investigation of the highly nonlinear and unsteady flow physics.

In this work we use FLEXI, a high-order solver for the compressible Navier-Stokes equations applying the discontinuous Galerkin spectral element method on unstructured grids. FLEXI provides excellent parallel scaling properties as well as a robust spatial discretization even when running severely under-resolved computations [20]. Especially parallel efficiency is an important property for virtual flight given the large scale of such a simulation. FLEXI is developed at the Numerics Research Group at the University of Stuttgart and available on GitHub*. The purpose of this paper is to test the application of the DG overset grids method with regard to VF. Our key advancement is the implementation of the dynamic DG chimera approaches of Crabill et al. [13] and Noack et al. [21] in FLEXI.

The outline of this paper will be as follows: In Section III, we introduce the governing equations, while in Section IV.A, we give a short introduction to the used spatial discretization scheme. The coupling interface will be shown in Section IV.B, with its spatial location and movement of the interface explained in Section IV.C. Section V validates the error convergence of the method, investigates the conservation error and shows the ability of the coupling interface to accurately transfer turbulence between the grids. In Section VI, the paper concludes with a proposal of possible improvements and future extensions.

III. Governing Equations

If not mentioned otherwise we consider the three-dimensional, compressible Navier-Stokes equations

$$\boldsymbol{U}_t + \nabla_x \cdot \boldsymbol{F}^C(\boldsymbol{U}) - \nabla_x \cdot \boldsymbol{F}^D(\boldsymbol{U}, \nabla_x \boldsymbol{U}) = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{U} = [\rho, \rho v_1, \rho v_2, \rho v_3, \rho e]^T$ denotes the vector of conserved variables. The convective fluxes $\boldsymbol{F}^C$ and the diffusive fluxes $\boldsymbol{F}^D$ are given in terms of density ρ, velocity vector $\boldsymbol{v} = (v_1, v_2, v_3)^T$ and energy density e for three spatial directions $i = 1, 2, 3$

$$\boldsymbol{F}_i^C = \begin{pmatrix} \rho v_i \\ \rho v_1 v_i + \delta_{1i} p \\ \rho v_2 v_i + \delta_{2i} p \\ \rho v_3 v_i + \delta_{3i} p \\ \rho e v_i + p v_i \end{pmatrix}, \ \boldsymbol{F}_i^D = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} v_j - q_i \end{pmatrix}. \tag{2}$$

With τ_{ij} being the stress tensor

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \lambda \delta_{ij} \frac{\partial v_k}{\partial x_k} \tag{3}$$

*https://github.com/flexi-framework/flexi and q_i the heat flux

$$q_i = -k \frac{\partial T}{\partial x_i}. \tag{4}$$

In the equations above k denotes the heat conductivity, μ the dynamic viscosity and T the static temperature. Assuming a Newtonian fluid and Stokes hypothesis yields $\lambda = -\frac{2}{3}\mu$.

Closure of the equation system is achieved by the equation of state for a perfect gas

$$p = \rho R T = \rho(\kappa - 1)\left(e - \frac{1}{2}(v_1^2 + v_2^2 + v_3^2)\right) \tag{5}$$

with κ the ratio of specific heats and R the specific gas constant.

For a moving reference, there is an additional flux term over cell boundaries to be considered originating in said movement. Using the Arbitrary Lagrangian-Eulerian (ALE) formulation [22], a second convective flux term F_i^G is needed incorporating the velocity $\boldsymbol{w}$ of the computational grid and leading to

$$\boldsymbol{F}_i^G = w_i \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho e \end{pmatrix}. \tag{6}$$

With this new flux term Eq. (1) becomes

$$\boldsymbol{U}_t + \nabla_x \cdot \left(\boldsymbol{F}^C(\boldsymbol{U}) - \boldsymbol{F}^G(\boldsymbol{U})\right) - \nabla_x \cdot \boldsymbol{F}^D(\boldsymbol{U}, \nabla_x \boldsymbol{U}) = \mathbf{0}. \tag{7}$$

IV. Numerical Methods

A. Discontinuous Galerkin Spectral Element Method

For the discontinuous Galerkin (DG) discretization every subdomain is divided into non overlapping elements Ω^n. In a next step a polynomial mapping from physical space $\boldsymbol{x} = (x_1, x_2, x_3)^T$ to the logical space $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$ is introduced, to transform the physical elements to a reference element $\tilde{R} = [-1, 1]^3$. Using the Jacobian of the mapping $J = \det\left(\frac{\partial x_i}{\partial \xi_j}\right)$ Eq. (7) becomes

$$J(\boldsymbol{\xi})\boldsymbol{U}_t + \nabla_{\xi} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{U}, \nabla_{\xi} \boldsymbol{U}) = \mathbf{0}. \tag{8}$$

with $\boldsymbol{\mathcal{F}}$ being the contra-variant fluxes on which a detailed explanation can be found in [23]. Projection onto the space of test functions $\psi \in \mathbb{P}$ and integration by parts over the logical element $\tilde{R}$ leads to the weak formulation of the equation

$$\int_{\tilde{R}} J(\boldsymbol{\xi})\boldsymbol{U}_t \psi(\boldsymbol{\xi}) d\tilde{R} + \int_{\partial \tilde{R}} (\boldsymbol{\mathcal{F}} \cdot \boldsymbol{n})\psi(\boldsymbol{\xi}) dS - \int_{\tilde{R}} \boldsymbol{\mathcal{F}}(\boldsymbol{U}) \cdot \left(\nabla_{\xi}\psi(\boldsymbol{\xi})\right) d\tilde{R} = \mathbf{0}, \tag{9}$$

where $\boldsymbol{n}$ is the surface normal vector of an element boundary. The fluxes across the element faces are approximated by Roe's Riemann solver [24] employing an entropy fix of Harten and Hyman [25]. Approximation of the conserved and the covariant fluxes is attained by a piecewise polynomial ansatz

$$\boldsymbol{U}(\boldsymbol{\xi}, t) \approx \sum_{i,j,k=0}^{N} \hat{\boldsymbol{U}}_{ijk} l_i(\xi^1) l_j(\xi^2) l_k(\xi^3), \tag{10}$$

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\xi}, t) \approx \sum_{i,j,k=0}^{N} \hat{\boldsymbol{\mathcal{F}}}_{ijk} l_i(\xi^1) l_j(\xi^2) l_k(\xi^3), \tag{11}$$

using a tensor basis of one-dimensional Lagrange polynomials l which entail the cardinal property on a certain set of interpolation points ξ_i^n. Since we employ a Galerkin method, the test functions ψ are chosen identically to the polynomial ansatz functions. Integrals in Eq. (9) are approximated using numerical quadrature. The efficiency of the

DGSEM scheme is based on collocation of integration and interpolation on the same point set which results in a reduced number of operations per degree of freedom (DOF). For the treatment of diffusive fluxes the solution gradients are provided by the "first" method of Bassi and Rebay [26]. Temporal evolution is achieved by explicit time integration using a low-storage Runge-Kutta scheme [27]. If not mentioned otherwise we use the Legendre-Gauß-Lobatto interpolation points with the Split-DG scheme of Pirozolli [28]. A general overview of the framework as well as validation and performance analysis of the baseline code is given in [20].

B. Overset Grids Coupling

A central matter when dealing with simulations on multiple moving grids is the coupling between these. The proposed overset grids scheme follows the AB-Method of Galbraith et al. [29], where the computational grids are only coupled by the numerical fluxes between the cells at the artificial boundary. For a mesh internal element the conserved variables and their gradients left/right of a cell boundary are used to evaluate the numerical flux $\boldsymbol{\mathcal{F}}(\boldsymbol{U}^-, \nabla\boldsymbol{U}^-, \boldsymbol{U}^+, \nabla\boldsymbol{U}^+)$ over the considered edge. This procedure is kept at artificial boundaries, however, an adjacent solution from a neighbor cell is not readily available. Instead, the solution needs to be supplied by another computational grid overlapping in this area, see Fig. 1.

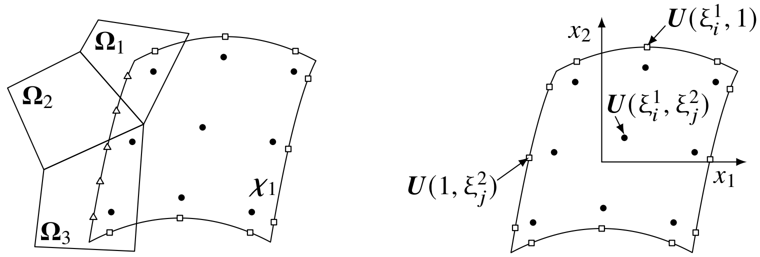


Fig. 1 Left: Two overlapping grids showing one artificial boundary with degree $N = 2$ and $N_{AB} = 4$ in physical space. Volume solution points are denoted by dots, flux evaluation points by squares and artificial boundary points by triangles. Right: An internal cell without artificial boundaries is shown for comparison.

The interface itself is loaded with $N_{AB} + 1$ flux evaluation points according to the chosen solution point distribution. Their physical coordinates $\boldsymbol{x}_{AB} = (x_1, x_2, x_3)$ are sent to the other computational grids. For parallel calculations, it is desirable to limit the communication to a reasonable sized region around the subdomain of one specific processor containing artificial boundaries. This is done by underlying a cartesian processor background mesh (BGM). Every process determines the bounding box of its subdomain using it to calculate an extent inside the BGM in terms of an index range per spatial direction. This information is then communicated globally, so every processor is aware of its neighbors in a certain area.

Even with the communication limited by the BGM, possible donor partitions might still receive a considerable amount of points to be evaluated. To speed up this procedure. we first check the bounding box of an element versus the coordinate $\boldsymbol{x}_{AB}$. If this test is successful Newtons Method is used to invert the cell local mapping and determine the logical coordinates $\boldsymbol{\xi}(\boldsymbol{x}_{AB})$. Since $\xi \in [-1, 1]^3$ holds, this enables the final determination if a point resides inside the element. Using the polynomial ansatz the conserved variables and their gradients can be interpolated for the determined donor. These values are sent back to the receiving mesh, where the solution is projected from the artificial boundary polynomial degree N_{AB} to the actual degree of the scheme N as depicted in Fig. 2. Then the numerical flux at the artificial boundaries can be obtained following the procedure for the internal cell edges.

Besides the notable computational effort needed to find the right element for a desired donor point, the presented approach offers some unique advantages. First is the absence of halo elements needed for solution reconstruction in Finite-Volume schemes. The number of flux evaluation points is by a factor of $N + 1$ smaller compared to the number of degrees of freedom needed when evaluating the complete volume solution. In addition, the spatial stencil size of the FV operator increases with the order of the scheme which in turn demands more halo cells at the overset grids boundary. A DG overset grids scheme with the presented AB method retains an arbitrary order using only the given flux evaluation points.

Handling the artificial boundaries simply as a kind of special boundary condition greatly simplifies the implementation of the method in an already parallelized code using the Message Passing Interface (MPI). The handling is similar to already present MPI boundaries between processor domains. Another notable property is the absence of a requirement

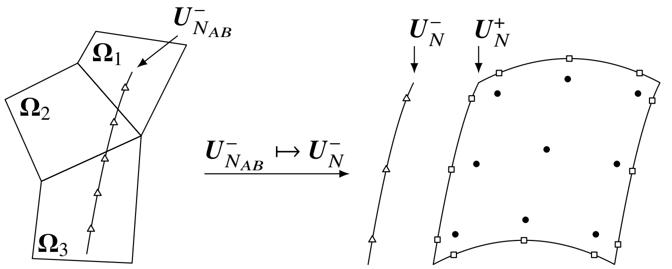


Fig. 2 Artificial boundary flux calculation for two overlapping grids with degree $N = 2$ and $N_{AB} = 4$ in physical space. Volume solution points are denoted by dots, flux evaluation points by squares and artificial boundary points by triangles.

for large overset areas between the two grids needed in conventional chimera schemes like the one found in [3], in the limiting case they can even abut. Therefore, a larger amount of cells can be cut from the underlying grid which results in fewer degrees of freedom to be computed.

When simply evaluating the solution in a pointwise manner on an artificial boundary over multiple donor cells one introduces an error. This is due to the possible discontinuous nature of the solution between the cells and the fact that numerical quadrature is only valid for continuous functions. To remedy this problem Farell et al. [30] and Crabill et al. [31] propose the use of local Galerkin projection. The basic idea is instead of a pointwise projection, a local so-called "super mesh" is introduced with one supercell only covering the intersection between a donor and a receiver cell. The solution is then integrated in each super mesh element separately. While this approach reduces the conservation error introduced by the chimera interface, the implementation is not trivial. Also, even more extra operations for the elements at the artificial interface are introduced. A general problem of the chimera method is an imbalance in computational load per element introduced by the local operations for the overset coupling. When using an even distribution of elements per CPU, this leads to a bad parallel scaling behavior and therefore the waste of resources.

Yet another source of conservation error is the absence of a unique flux between the grids. For the internal cell boundaries, the numerical flux is calculated only once and used to update the solution in both cells ensuring conservation. When using the outlined approach, two different fluxes are employed at the artificial boundaries, which results in conservation errors. Flux reconstruction was presented by Crabill et al. [31] as a possible solution. The basic idea is to interpolate a corrected flux to the artificial boundary instead of a discontinuous solution. However, this procedure is accompanied by a more strict requirement of the extent of the grid overlap and possible stability issues that need to be addressed. Galbraith et al. [29] showed that the mentioned errors decrease with the theoretical order of convergence for the scheme. In this work, we will use this comparatively simple approach and show that these errors are acceptable for the investigated test cases.

C. Hole-Cutting

1. Static Hole-Cutting

In the previous section, the chimera interface was introduced as a kind of special boundary condition. A vital part of the overset grids scheme is to determine which sides should be flagged to be chimera boundaries. When dealing with only two grids, the procedure is simple: blank every element of the underlay mesh which is completely overcast by the overlay mesh and exclude them from the computation. All sides between a hole (blanked elements) and a field (non-blanked elements) cell are artificial boundaries required to execute the AB coupling introduced in the previous section. A simple example of such a cut is shown in Fig. 3.

In the following, the term overlay denotes the subdomain containing a geometry cutting the static underlay grid without solid boundaries. However, the suggested approach is not limited in the number of grids used or if a geometry is present. Only the designation of which grid is employed for cutting in a certain area needs to be determined. One possible principle is to choose the grid with the highest resolution to be kept.

According to Noack et al. [21] there are three basic types of hole cutting. First are search-based techniques where for every grid point in the underlay mesh a possible donor element is searched for. In regions with multiple meshes present this will be successful, classifying the corresponding cells as hole. Some kind of geometry approximation is needed as some points may reside inside a geometry and will therefore find no donor although being inside an overlay

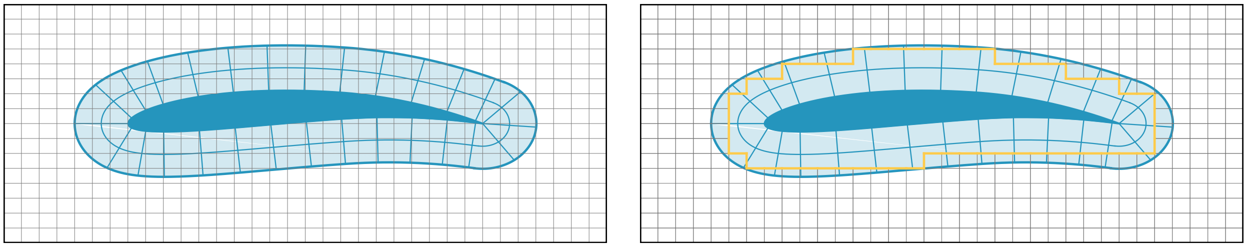


Fig. 3 Basic overset grids setup in 2D, showing a meshed NACA profile and a cartesian background grid in (a). The resulting cut and artificial boundaries in yellow are shown in b).

mesh. For this method to work properly a minimal extent of overlap is needed, otherwise the process might fail.

The query-cut approach builds upon an approximation of the cutting surface. This can be done by using an analytical sphere or cartesian approximations for example. Care has to be taken to describe the interface with sufficient accuracy to accurately cut the underlying mesh. At the same time, it is desired not to add too much complexity through the approximation to keep the evaluation and therefore the hole cutting computationally fast.

When employing a direct-cut type method, all elements intersected by the cutting interface are determined. The remaining cells inside the overlay are then cut and removed from the computation. To achieve this some kind of paint-fill algorithm is needed to identify all overcast cells. This is the most accurate method since no approximation of the chimera interface is needed.

The approach used in this work is of the direct-cut type and follows Crabill et al. [13], who builds upon ideas from Galbraith et al. [32] and Noack et al. [21] but with the restriction to linear cells. Elements with curved surfaces are needed to approximate geometries with reasonable accuracy. For the application of virtual flight, non-trivial wall boundaries will only be present in the overlay mesh. While it is desirable to reduce the amount of elements around the moving geometry some grid layers are still needed. As outlined in [20] a common approach is to curve 2-4 layers adjacent to the geometry, while keeping the remaining layers linear. This results in a linear artificial boundary for the overlay mesh, significantly reducing the effort needed for a direct cut.

The general idea for the hole cutting is to start with several simple and fast comparisons, determining the status of the majority of elements. Subsequent filters reduce the number of remaining elements while moving to computationally more expensive operations until the final blanking status for the whole domain is determined. The purpose of the multistage filtering is the reduction of overall computation time needed for the hole cutting.

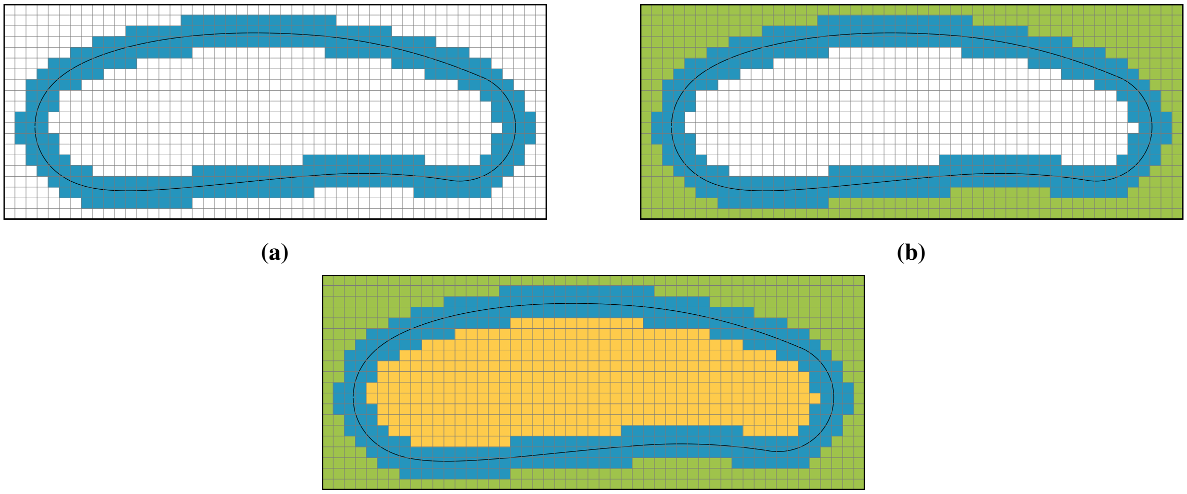
As a prepossessing step the bounding box (BB) along the axes of the global coordinate system of all elements contained in the subdomain of one process $\boldsymbol{B}_P$ and for the whole overlay mesh $\boldsymbol{B}_{OL}$ are computed. If these do not intersect, all elements in the underlay subdomain can be marked as field cells. The remaining partitions test for overlap with neighbor overlay subdomains, which are determined using the BGM mentioned in the preceding section. Further on, the center of all sides considered to be artificial boundaries and the corresponding normal vectors are sent to the remaining partitions near the interface which might need cutting data.

So far the status of elements far away from the artificial boundaries has been determined. For the cells closer, we employ a hole map which is a Cartesian approximation of the interface. Being Cartesian with a constant step size and dimensions simplifies the determination of the status of a point inside this map. The first parameter to be determined is the resolution of the map, which is a compromise between accuracy and computational overhead. It is desirable to have some kind of relation between the dimensions of the artificial surfaces ζ and the hole map, to automate its generation for a given overlay mesh. The greatest distance L^ζ from the center of a side $\boldsymbol{x}_O^\zeta$ to the corners $\boldsymbol{x}_C^\zeta$ is taken. From all surfaces in question the smallest distance of all distances $\boldsymbol{L}$ is chosen to be the step size ds for all spatial dimensions.

$$L^\zeta = max\left(\left\|\boldsymbol{x}_O^\zeta - \boldsymbol{x}_C^\zeta\right\|\right), \quad ds = min(\boldsymbol{L}) \tag{12}$$

The number of elements per spatial direction $\boldsymbol{n} = [n_1, n_2, n_3]$ is chosen by a relation using the extent of the overlay bounding box $\boldsymbol{B}_{OL}$ and adding four elements of padding

$$\boldsymbol{n} = \left\lfloor \frac{(\boldsymbol{B}_{OL}^+ - \boldsymbol{B}_{OL}^-)}{ds} \right\rfloor + 4 \tag{13}$$



(a)

(b)

(c)

Fig. 4 Example of hole map construction. Elements intersecting with the boundary and their direct neighbors are colored blue. Using a paint-fill process all elements outside the blue area are marked as field and indicated by green. All remaining elements in yellow are considered to have hole status.

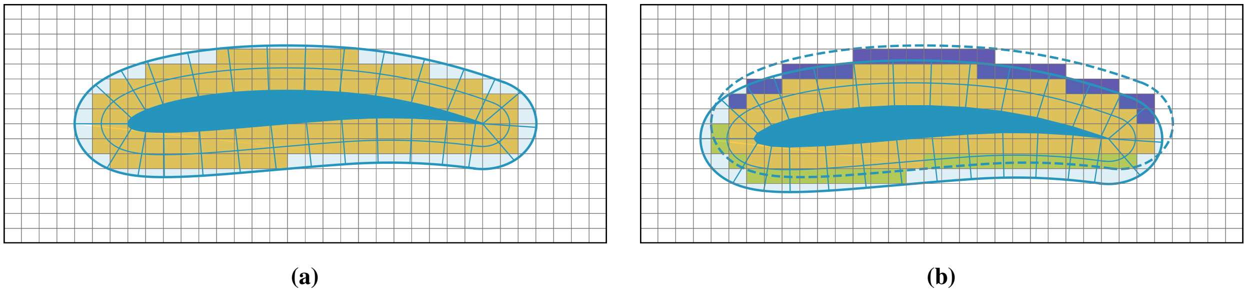
With the hole map in place, the status of its elements can be determined. Since the map is Cartesian, this is done by calculating an index for a point using the extent and spatial step of the map, which is a fast and computationally cheap operation. For the artificial interface, we use again the coordinates of the corners and the center of the surfaces, called cutting points, for index calculation. All hole map elements intersecting the interface are tagged with the color blue as shown in Fig. 4. To increase the robustness of the scheme we add one element of padding in every spatial direction. This procedure results in a closed Cartesian approximation of the chimera interface. Starting at the corners of the hole map and therefore ensuring to be outside the interface area, a paint-fill procedure is applied to label all elements with the color green until reaching interface elements with the tag blue. This process will give erroneous results if there are holes in the interface approximation. Therefore, the choice of ds in combination with the number of cutting points per surface, outlined in the paragraph before, is essential for ensuring a proper hole map. All remaining cells reside inside the overset boundary and are therefore considered possessing hole status and tagged yellow.

When the construction of the hole map is finished, the remaining partitions calculate indices for the element corners concerning the coordinate system of the map. If all corners are labeled yellow or green, the status of the cell is considered to be safely determined by the hole map. Also, if at least one corner is tagged green, the element is a field element, regardless of the other corners. Any combination of pure blue tags or blue/yellow leaves the element status to be determined by the last cutting step.

All remaining cells without a clear hole status are located near the artificial boundary. To decide their final blanking state a simple inside-out test via dot product is used. Previous to the hole map evaluation all partitions near cutting surfaces already received the corresponding face center coordinates and normals. For every corner of a filtered underlay element, the vector from the cutting surface center to the corner is computed. The evaluation of the corner status is done by a dot product with the normal vector. If one corner resides outside the cutting surface, the elements' status is saved and the next cell is evaluated. It might be desirable to further reduce the number of required dot products since the loop over all cutting sides from a nearby partition can be quite extensive. This could be achieved by utilizing the already present hole map and only considering surfaces in surrounding elements. However, the approach seems to be working quite fast for the tests performed. No dramatic increase in computational time is observed when comparing moving overset grids setup with a static one requiring hole cutting only once.

2. Dynamic Hole-Cutting

When one or multiple of the used grids are moving some modifications to the overall process are necessary. As the overlay mesh moves through the domain, elements with field status at the beginning of the simulation will be overcast at some point. The blanking of these cells is rather simply achieved by setting the corresponding solution to zero and



(a)

(b)

Fig. 5 Example of a moving overlay grid. In a) the grid at t^n is shown with the hole cells tagged yellow. The new status at t^{n+1} is given in b), the old position shown using dashed lines with elements to unblank (violet) before the execution of the time step and elements to blank (green) after this time step.

excluding the element from further computation. When a grid cell reappears from below the overlay it needs to be unblanked and filled with a valid solution. For this purpose the artificial boundary method from Section IV.B is applied to the volume Gauss points and the values for the conserved variables donated by the overlaying grid. An example of elements which need to perform the blanking/unblanking process between two time steps is given in Fig. 5.

Given that the movement of the geometry is a reaction to aerodynamic forces, the cells to cut cannot be determined through preprocessing. Therefore, the steps outlined for static hole cutting need to be performed every time step. The focus of this work lies on explicit time stepping using multi-stage Runge-Kutta methods. The calculation of the solution at the new time t^{n+1} is done by evaluation of the spatial operator at a number of intermediate sub-time steps. During these substages, the blanking status must be held constant. This assumption only holds if the overlay grid is not moving over the whole length of any cell in the underlay mesh during one time step. Since the time step is strictly limited by the information propagation speed using the CFL- condition, this conjecture is justified. As a result, the unblanking process is executed rather infrequently. Even though the number of points per element to which a solution needs to be supplied is by a factor $N + 1$ higher than for a surface solution in the AB context, the overall cost is regarded as low since the number of elements is also low.

The overall procedure is as follows: At time t^n the overlay grid is moved to its new position, hole cutting is performed as described in the preceding section. If the status of an element is switching hole to field, this cell is tagged to be unblanked before execution of the next time step. When an element is completely overcast at t^{n+1}, the unblanking is performed after the temporal evolution of the solution, since the element is needed for artificial boundary operations in the substages of the RK scheme. When the unblanking is finished, the grid is moved back and the actual time integration is started. The overlay is moved in every RK substage according to the sub time step, the reason being the artificial boundaries need the actual coordinates at this time to obtain a time accurate solution.

When performing a dynamic hole cutting some operations, which can be considered preprocessing in the static case, need to be executed every time step. Given that the grids are not deforming, the BB of the subdomains, grids and the hole map can be simply moved according to the displacement of the overlay. This approach saves some operations needed for recomputing, which is especially important for the hole map.

V. Results

This section presents validation test cases for the outlined DG overset grids method. The first experiment shows that the method retains high-order accuracy on moving meshes. Convergence of the conservation error is also presented. We then investigate the method's ability to transfer turbulent structures across the artificial boundary with the Taylor-Green Vortex. Finally, the ability to move a geometry through space and its interaction with the surrounding fluid will be shown.

If not mentioned otherwise time integration is performed using a fourth order, low storage Runge-Kutta scheme by Carpenter et al. [33]. We use the Legendre-Gauß-Lobatto interpolation points with the Split-DG scheme of Pirozolli [28] and Roe's approximate Riemann solver employing an entropy fix by Harten and Hyman [25].

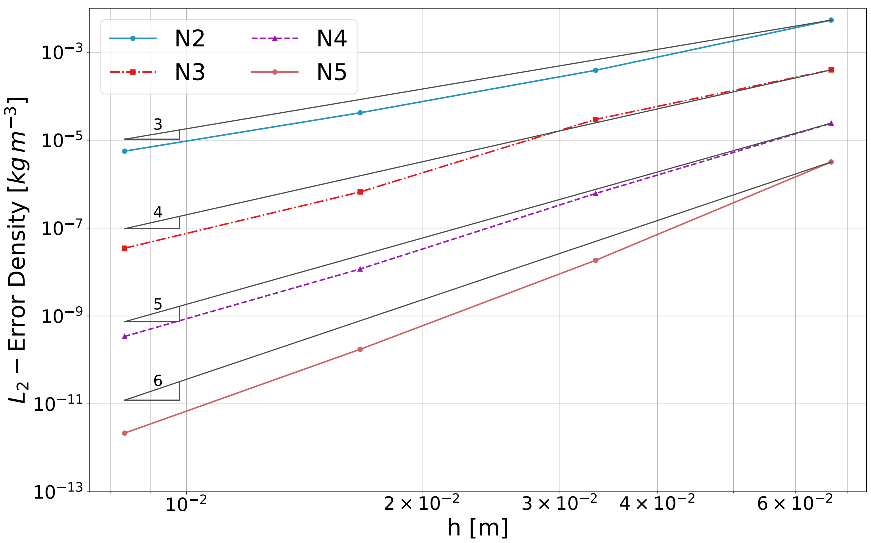


Fig. 6 **L_2 error of the density for different polynomial degrees N at $t = 1\ s$ for the manufactured solution over spatial stepsize h. The theoretical EOC is denoted using black lines.**

A. Convergence Test

To verify the high-order character of the DG chimera scheme, we will use the method of manufactured solutions (MFS) to compute the experimental order of convergence (EOC), following Gassner et al. [33]. The general idea of this method is to modify the equation system by a source term to force a prescribed, non-trivial solution. Since the exact solution of the modified equation system is known, it can readily be used to compute the EOC. We choose an oblique sine wave as the solution for all variables, advected in 3D with a constant velocity $\mathbf{v}_{adv} = [1, 1, 1]$ ms^{-1}, an amplitude $\alpha = 0.1 \cdot \boldsymbol{U}_0$ regarding the initial mean value of the solution $\boldsymbol{U}$ and a frequency $\omega = 1$. The specific gas constant, rate of specific heats and dynamic viscosity are given by $R = 287.058$ Jkg^{-1}K^{-1}, $\kappa = 1.4$ and $\mu = 0.001$ kg m^{-1}s^{-1}.

The underlay domain is laid out as a cube with dimensions $\mathbf{x}_{UL} = [-1, 1]^3$ m employing periodic boundary conditions in all spatial directions and a cartesian, uniform grid. A second cube which extents $\mathbf{x}_{UL} = [-0.4, 0.4]^3$ m serving as overlay domain is placed at $\mathbf{x}_{t0} = [-0.5, -0.5, -0.5]$ m at the beginning of the simulation. The domain moves with a constant velocity $\mathbf{v}_{trans} = [1, 1, 1]$ ms^{-1} to the opposite corner of the underlay domain until $t = 1$ s. We start with a resolution of five elements per spatial direction and double the resolution with every refinement step. The polynomial degree at the artificial interface is chosen to be $N_{AB} = 2N$ to reduce conservation errors introduced by artificial boundaries.

The time step was scaled down to limit the temporal error to a magnitude where the spatial error will dominate the overall error. In Fig. 6 the L_2 error at $t = 1$ s is shown, confirming the scheme is maintaining the EOC when using the overset grids method on moving meshes with unblanking.

As mentioned in IV.B the scheme is by design non-conservative. However, according to Galbraith et al. [29] the conservation error should decrease with the EOC of the numerical scheme. To validate this we choose again the method of MFS. In this case the sinus prescribed by the MFS is only avdvected in x_2 direction with $\mathbf{v}_{adv} = [0, 1, 0]$ ms^{-1}. The overlay mesh is placed at $\mathbf{x}_{t0} = [-0.5, 0, 0]$ m, all remaining discretization parameters and resolutions stay the same. The movement of the overlay mesh in x_2, x_3 direction is given by a sine

$$x_1 = x_1 + u_0 \cdot t, \tag{14}$$

$$x_2 = x_2 + 0.4L \cdot sin(2\pi \cdot t), \tag{15}$$

$$x_3 = x_3 + 0.4L \cdot sin(2\pi \cdot t). \tag{16}$$

with $u_0 = 1$ ms^{-1} and $L = 1$ m. The conservation error is caused by the numerical integration of a discontinuous solution on the artificial boundary as well as the non-unique flux between the grids, see Section IV.B for a more detailed explanation. The conservation error depends therefore on the position of the volume and surface solution points in

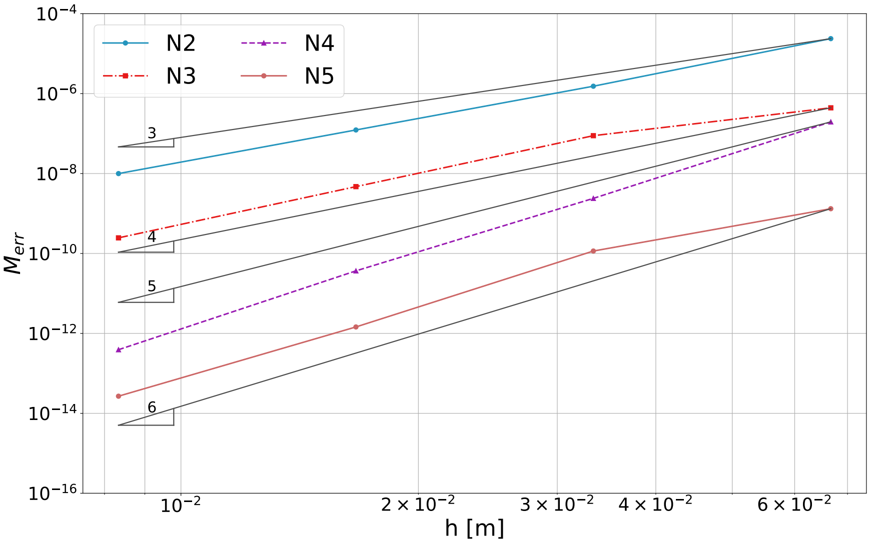


Fig. 7 Conservation error of for different polynomial degrees N with $N_{AB} = 2N$ at $t = 1\,s$ for the manufactured solution over spatial stepsize h. The theoretical EOC is denoted using black lines.

relation to the corresponding donor elements and the smoothness of the solution between them. We chose this movement to have a more general trajectory than the rather simple movement along a line from before.

In Fig. 7 the conservation error over different resolutions and polynomial degrees is shown. The error is defined in Eq. (17) with M_e being the exact mass contained in the mesh at $t = 0$ s and M_h the mass contained in both meshs at final time $t = 1$ s. In general, the mass error decreases with the expected EOC, with exception to the first two data points. This is most likely due to insufficient spatial resolution. Also notable is the greater than expected EOC for $N = 4$ which might be caused by the of $N_{AB} = 2N$. Overall we would like to emphasize that the conservation error decreases with increasing mesh resolution even for a non-trivial trajectory and the non-conservative AB method.

$$M_{err} = \frac{M_e - M_h}{M_e} \tag{17}$$

B. Taylor-Green Vortex

The Taylor-Green Vortex (TGV) is a common testcase for the 3D Navier-Stokes equations at moderate Reynolds numbers. From a simple sinus distribution of the velocity components, the flow transitions through a short laminar phase to full turbulence. The TGV is one of the problems used for the evaluation of CFD codes at the International Workshops on High-Order CFD Methods [34, 35]. It is chosen because of its simple setup, all periodic boundary conditions and readily available DNS results from [36]. An important property of the outlined overset grids scheme is the correct transfer of turbulent structures crossing artificial boundaries. The TGV is, therefore, particularly attractive to test the influence of the moving reference frame on the turbulent spectrum. For comparison, the testcase can also easily be computed employing only a single grid. Another interesting aspect is the influence of the non-conservative scheme on the turbulence parameters.

Initial conditions are prescribed using primitive variables

$$v_1 = u_0\, sin\left(\frac{x_1}{L}\right) cos\left(\frac{x_2}{L}\right) cos\left(\frac{x_3}{L}\right), \tag{18}$$

$$v_2 = -u_0\, sin\left(\frac{x_1}{L}\right) cos\left(\frac{x_2}{L}\right) cos\left(\frac{x_3}{L}\right), \tag{19}$$

$$v_3 = 0, \tag{20}$$

$$p = p_0 + \frac{\rho_0 {u_0}^2}{16}\left[cos\left(\frac{2x_1}{L}\right) + cos\left(\frac{2x_2}{L}\right)\right]\left[cos\left(\frac{2x_3}{L}\right) + 2\right] \tag{21}$$

and the perfect gas law $\rho = \frac{p}{RT_0}$ with specific gas constant $R = 17.857$ Jkg^{-1}K^{-1} and ratio of specific heats $\kappa = 1.4$. The physical conditions $T_0, p_0, \rho_0 u_0$ are chosen to achieve a Mach number of $Ma = 0.1$ corresponding to u_0. The Reynolds number governing the flow is $Re = 1600$ in regard to u_0 and L is set by $\mu = 6.25 \cdot 10^{-4}\, \frac{kg}{ms}$.

The underlay domain is an all periodic cube with dimensions $\mathbf{x}_{UL} = [-L\pi, L\pi]^3$ using a total of 32, 64 and 96 elements per spatial direction. When employing a polynomial degree of $N = 3$ and $N_{AB} = 6$ this corresponds to 128^3, 256^3 and 384^3 degrees of freedom. For the underlay, we choose a box with magnitude $\mathbf{x}_{OL} = [-0.4\, L\pi, 0.4\, L\pi]^3$ and place it at the origin for $t = 0\, s$. The number of cells is selected, ensuring a uniform resolution across the domain. During the computation, the overlay grid translates in a figure 8 pattern in the x_1, x_2 plane while traveling up and down along the x_3 axis. The setup is similar to Crabill et al. [13] with parametrization of the movement being

$$x_1 = x_1 + \frac{5\,\pi\, L}{10}\, sin\Big(\frac{20\pi^2}{169}\, t\Big), \tag{22}$$

$$x_2 = x_2 + \frac{3\,\pi\, L}{10}\, sin\Big(\frac{40\pi^2}{169}\, t\Big), \tag{23}$$

$$x_3 = x_3 + \frac{5\,\pi\, L}{10}\, sin\Big(\frac{10\pi^2}{169}\, t\Big). \tag{24}$$

All computations were performed until the time $t = 20\ s$ was reached.

Various quantities can be analyzed to quantify the performance of the scheme. First, the enstrophy, averaged by the volume V

$$\mathcal{E} = \frac{1}{\rho_0 V} \int_v \rho \frac{\omega \cdot \omega}{2} dV. \tag{25}$$

Assuming incompressible flow, which is a valid assumption at $Ma = 0.1$ the dissipation rate of the kinetic energy can be computed using the enstrophy with the relation

$$\epsilon = -\frac{dE_k}{dt} = 2\frac{\mu}{\rho_0}\mathcal{E}. \tag{26}$$

In Fig. 8 the temporal evolution of the energy dissipation rate is visualized using the overset grids setup. For comparison, the results for a single grid using the same resolution and a DNS result are also given. When the spatial resolution is increased, ϵ approaches the DNS solution which is the expected behavior, also visible in a detail view in Fig. 9. Notable is the good agreement of single and overset grid results, even though the overset simulation is not conservative. Some minor deviations can be observed which indicate that the AB method is only slightly dissipative. These results demonstrate that the artificial interface can transfer turbulent structures across the mesh boundaries without signification disturbances. The scheme should therefore also be able to couple the turbulent flow around a wing with the surrounding atmosphere.

C. Cylinder

The last test case to be investigated is a cylinder moving through a fluid. It shows the method's ability to transfer the wake generated by a geometry into the underlying domain. While this set up is pseudo-3D with only one layer of cells in the x_3 direction it is a first step in the direction of wake vortex generation of virtual flight.

The relevant part of the underlay domain, the area where the overlay is moving, is shown in Fig. 10. It extends to $x_1 = [-4L, 4L]$ with $n_{x1} = 160$ elements and $x_2 = [-2.5L, 2.5L]$ with $n_{x2} = 100$ elements, where L corresponds to the diameter of the cylinder. Outside this region, the resolution is drastically decreased to dampen acoustics originating from the start-up process. On the right-hand side boundary, a constant inflow with zero flow angle and $Ma = 0.1$ is prescribed, by setting total temperature and pressure according to their initial, static counterparts inside the domain resulting in an inflow velocity u_0. The remaining boundaries in x_1, x_2 direction specify a constant back pressure p, in x_3 direction periodic boundaries are used.

The dimension of the overlay extends between the radii of $r_{inner} = 0.5L$ and $r_{outer} = 0.9L$, while the domain is subdivided in $n_t = 135$ elements in circumferential and $n_r = 20$ elements in radial direction. A visualization of the mesh can be found in Fig. 10. At the inner radius a no-slip wall BC is employed with two layers of curved cells normal to the wall, at the outer radius artificial boundaries are found while in x_3 direction again periodic BCs are used. The ratio of specific heats is $\kappa = 1.4$, the specific gas constant $R = 287.058$ Jkg^{-1}K^{-1}, while the Reynolds number regarding

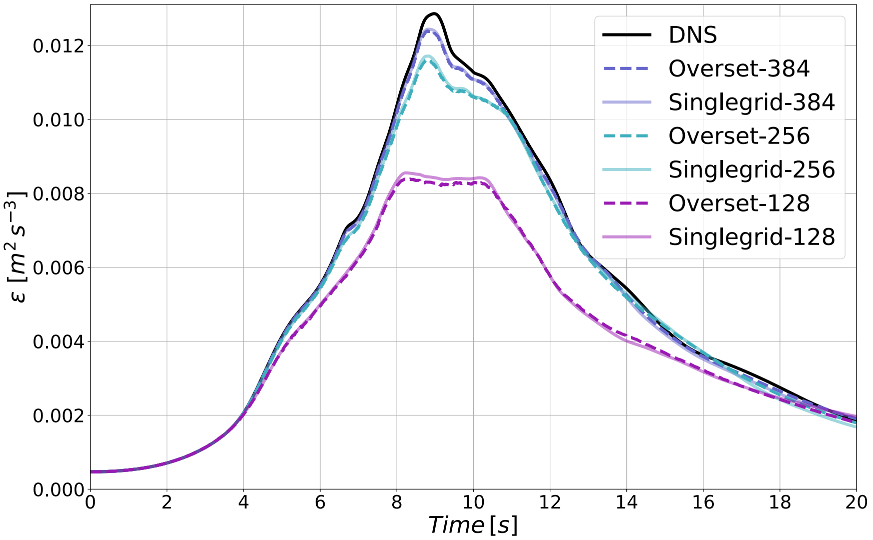


Fig. 8 Energy dissipation rate for the TGV using two grids coupled by the AB method, with 128^3, 256^3 and 384^3 DOF's while using $N = 3$. For comparison, the single grid and DNS results are also provided.

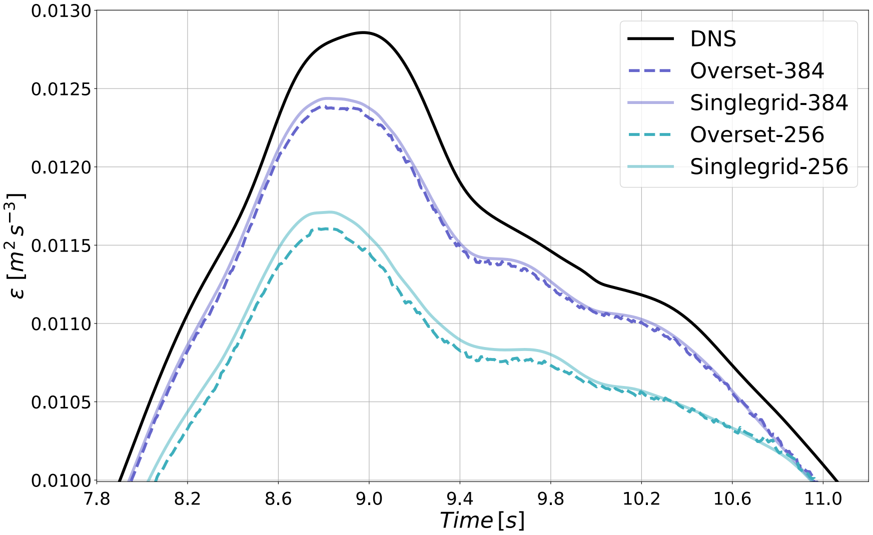
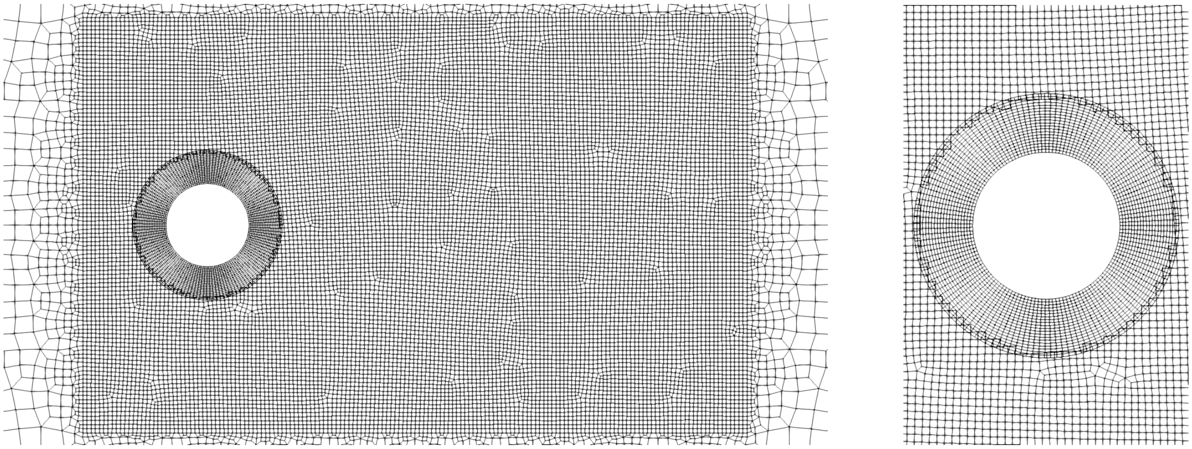


Fig. 9 Detailed view of the energy dissipation rate for the TGV using two grids coupled by the AB method, with 256^3 and 384^3 DOF's using $N = 3$. For comparison, the single grid and DNS results are also provided.



(a) (b)

Fig. 10 Section of the unstructured underlay grid a) in which the cylinder is moving. The structured overlay grid with the corresponding hole cut is shown in b).

the cylinder's diameter of L and the inflow velocity in addition to the mesh velocity in x_1 direction is $Re = 50000$. We use a polynomial degree of $N = 6$ in combination with $N_{AB} = 10$. The movement of the cylinder follows a sine curve described by

$$x_1 = x_1 + u_0 \cdot t, \tag{27}$$

$$x_2 = x_2 + 0.75L \cdot sin\left(\frac{2\pi}{5L} \cdot t\right). \tag{28}$$

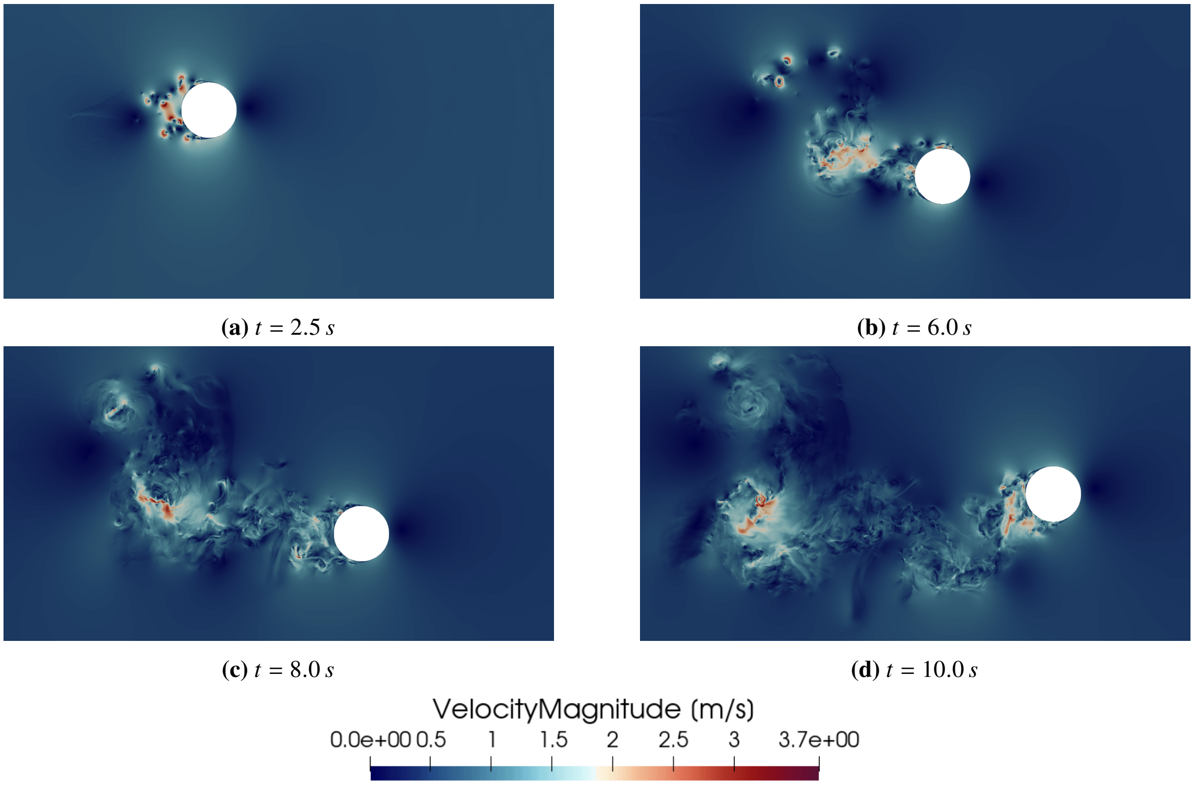
In Fig. 11 the velocity magnitude is shown at different times. While there is no reference to validate these results, they show that the presented method is indeed able to capture the interaction between a moving geometry and the surrounding fluid. Eddies detach from the cylinder and are transferred into the underlying domain without visible artifacts. While not visible in the static plots the transport of turbulent structures from the underlay back into the overlay is also possible. Additionally, detached eddies crossing the interface from the underlying domain to the moving grid inside the recirculation region remain stable and do not show a nonphysical deformation.

VI. Conclusion

A high-order discontinuous Galerkin overset grids method intended for the simulation of virtual flight has been presented. This approach enables the coupling between multiple, unstructured computational grids without any restriction regarding the type of movement and trajectory. The hole cutting is realized with a focus on simplicity and speed, by starting with a couple of simple tests to filter elements far away from the cutting surface and only perform costly checks on elements near the artificial boundary.

The method of manufactured solutions has been used to show, that the method retains its experimental order of convergence (EOC) on moving grids with blanking and unblanking of elements besides the introduction of a conservation error by the artificial interface. Convergence was also demonstrated for the conservation error. Simulating the Taylor-Green Vortex showed turbulent structures can indeed be transferred via this interface with minor interference. At last the methods ability to transport the turbulent wake generated by a cylinder moving through a fluid into the underlying grid was presented. The results of these test cases also proof the applicability of the hole cutting procedure to 3D cases while reducing the user inputs needed for tuning the algorithm.

Next will be the evaluation of the parallel performance and the comparison with single grid performance. Challenging in this context is the dynamic communication pattern originating in the movement of the grids. While the amount of communication is already reduced by employing the processor background mesh, some improvements are still to be made. Especially regarding the number of dot products needed for the direct cut of cells near the artificial boundary. Another topic to be addressed in a future publication is the load imbalance, designating not evenly distributed


(a) $t = 2.5\,s$

(b) $t = 6.0\,s$

(c) $t = 8.0\,s$

(d) $t = 10.0\,s$

Fig. 11 Cylinder moving along a sine at $Re = 50000$, using $N = 6$ and $N_{AB} = 10$. Shown is the velocity magnitude.

workload across processes. This imbalance is introduced by the chimera interface and the interpolation of the solution in overlapping cells, which is only necessary near the overlay mesh. A possible approach is dynamic load balancing as outlined in [37]. Since the trajectories of the moving geometry are predictable with sufficient accuracy for a couple of time steps, the load balancing might be achieved in a priori fashion.

When the issues regarding computational performance are eliminated we plan to use this method on a range of interesting cases with wake vortex encounters during approach and landing or in formation flight. This will enable the accurate simulation of the interaction between a wing and atmospheric disturbances, which is so far restricted in our present two-way coupling approach [3].

VII. Acknowledgments

The authors would like to thank the Numerics Research Group at the University of Stuttgart for providing the FLEXI chimera branch and their continued assistance. Special thanks to Nico Krais for implementing the static overset grids scheme. We would also like to thank the Leibniz Supercomputing Center for their support and supply of computing time on the SuperMUC-NG supercomputer through the project "pn98me".

References

[1] Gerz, T., Holzäpfel, F., and Darracq, D., "Commercial aircraft wake vortices," *Progress in Aerospace Sciences*, Vol. 38, No. 3, 2002, pp. 181–208. https://doi.org/https://doi.org/10.1016/S0376-0421(02)00004-0.

[2] Unterstrasser, S., and Stephan, A., "Far field wake vortex evolution of two aircraft formation flight and implications on young contrails," *The Aeronautical Journal*, Vol. 124, No. 1275, 2020, p. 667–702. https://doi.org/10.1017/aer.2020.3.

[3] Zholtovski, S., Stephan, A., and Holzäpfel, F., "Towards Virtual Flight in Realistic Environments: A Hybrid Coupled Simulation Method," *AIAA Journal*, Vol. 58, No. 3, 2020, pp. 1266–1277. https://doi.org/10.2514/1.J058390.

[4] Crow, S. C., "Stability Theory for a Pair of Trailing Vortices," *AIAA Journal*, Vol. 8, No. 12, 1970, pp. 2172–2179. https://doi.org/https://doi.org/10.2514/3.6083.

[5] Misaka, T., Holzäpfel, F., and Gerz, T., "Large-Eddy Simulation of Aircraft Wake Evolution from Roll-Up Until Vortex Decay," *AIAA Journal*, Vol. 53, No. 9, 2015, pp. 2646–2670. https://doi.org/10.2514/1.J053671.

[6] Stephan, A., Holzäpfel, F., and Misaka, T., "Hybrid simulation of wake-vortex evolution during landing on flat terrain and with plate line," *International Journal of Heat and Fluid Flow*, Vol. 49, 2014, pp. 18–27. https://doi.org/https://doi.org/10.1016/j.ijheatfluidflow.2014.05.004, 8th Symposium on Turbulence and Shear Flow Phenomena (TSFP8).

[7] Minoli, C. A. A., and Kopriva, D. A., "Discontinuous Galerkin spectral element approximations on moving meshes," *Journal of Computational Physics*, Vol. 230, No. 5, 2011, pp. 1876–1902. https://doi.org/https://doi.org/10.1016/j.jcp.2010.11.038.

[8] van der Weide, E., Kalitzin, G., Schluter, J., and Alonso, J., *Unsteady Turbomachinery Computations Using Massively Parallel Platforms*, 2006, p. 421. https://doi.org/10.2514/6.2006-421.

[9] Duerrwaechter, J., Kurz, M., Kopper, P., Kempf, D., Munz, C.-D., and Beck, A., "An Efficient Sliding Mesh Interface Method for High-Order Discontinuous Galerkin Schemes," 2020.

[10] McNaughton, J., Afgan, I., Apsley, D. D., Rolfo, S., Stallard, T., and Stansby, P. K., "A simple sliding-mesh interface procedure and its application to the CFD simulation of a tidal-stream turbine," *International Journal for Numerical Methods in Fluids*, Vol. 74, No. 4, 2014, pp. 250–269. https://doi.org/https://doi.org/10.1002/fld.3849.

[11] Benek, J., Steger, J., and Dougherty, F., *A flexible grid embedding technique with application to the Euler equations*, 1983. https://doi.org/10.2514/6.1983-1944.

[12] Galbraith, M. C., Benek, J. A., Orkwis, P. D., and Turner, M. G., "A discontinuous Galerkin scheme for Chimera overset viscous meshes on curved geometries," *Computers and Fluids*, Vol. 119, 2015, pp. 176–196. https://doi.org/https://doi.org/10.1016/j.compfluid.2015.07.002.

[13] Crabill, J., Witherden, F., and Jameson, A., "A parallel direct cut algorithm for high-order overset methods with application to a spinning golf ball," *Journal of Computational Physics*, Vol. 374, 2018, pp. 692–723. https://doi.org/https://doi.org/10.1016/j.jcp.2018.05.036.

[14] Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845. https://doi.org/https://doi.org/10.1002/fld.3767.

[15] Liu, X.-D., Osher, S., and Chan, T., "Weighted Essentially Non-oscillatory Schemes," *Journal of Computational Physics*, Vol. 115, No. 1, 1994, pp. 200–212. https://doi.org/https://doi.org/10.1006/jcph.1994.1187.

[16] Sherer, S. E., "Investigation of high-order and optimized interpolation methods with implementation in a high-order overset grid fluid dynamics solver," Ph.D. thesis, The Ohio State University, 2002.

[17] Reed, W. H., and Hill, T. R., "Triangular mesh methods for the neutron transport equation," 1973. URL https://www.osti.gov/biblio/4491151.

[18] Cockburn, B., and Shu, C.-W., "The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems," *Journal of Computational Physics*, Vol. 141, No. 2, 1998, pp. 199–224. https://doi.org/https://doi.org/10.1006/jcph.1998.5892.

[19] Gassner, G., and Kopriva, D. A., "A Comparison of the Dispersion and Dissipation Errors of Gauss and Gauss–Lobatto Discontinuous Galerkin Spectral Element Methods," *SIAM Journal on Scientific Computing*, Vol. 33, No. 5, 2011, pp. 2560–2579. https://doi.org/10.1137/100807211.

[20] Krais, N., Beck, A., Bolemann, T., Frank, H., Flad, D., Gassner, G., Hindenlang, F., Hoffmann, M., Kuhn, T., Sonntag, M., and Munz, C.-D., "FLEXI: A high order discontinuous Galerkin framework for hyperbolic–parabolic conservation laws," *Computers and Mathematics with Applications*, Vol. 81, 2021, pp. 186–219. https://doi.org/https://doi.org/10.1016/j.camwa.2020.05.004.

[21] Noack, R., *A Direct Cut Approach for Overset Hole Cutting*, 18th AIAA Computational Fluid Dynamics Conference, 2018. https://doi.org/10.2514/6.2007-3835.

[22] Hirt, C., Amsden, A., and Cook, J., "An arbitrary Lagrangian-Eulerian computing method for all flow speeds," *Journal of Computational Physics*, Vol. 14, No. 3, 1974, pp. 227–253. https://doi.org/https://doi.org/10.1016/0021-9991(74)90051-5.

[23] Kopriva, D. A., *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, 1st ed., Springer Publishing Company, Incorporated, 2009.

[24] Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. https://doi.org/https://doi.org/10.1016/0021-9991(81)90128-5.

[25] Harten, A., and Hyman, J., "Self-Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 50, 1983, pp. 235–269. https://doi.org/10.1016/0021-9991(83)90066-9.

[26] Bassi, F., and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 131, 1997, pp. 267–279. URL https://api.semanticscholar.org/CorpusID:123455836.

[27] Carpenter, M., and Kennedy, C., "Fourth-order 2N-storage Runge-Kutta schemes," *Nasa reports TM, 109112,*, 1994.

[28] Pirozzoli, S., "Numerical Methods for High-Speed Flows," *Annual Review of Fluid Mechanics*, Vol. 43, No. 1, 2011, pp. 163–194. https://doi.org/10.1146/annurev-fluid-122109-160718.

[29] Galbraith, M. C., Benek, J. A., Orkwis, P. D., and Turner, M. G., "A Discontinuous Galerkin Chimera scheme," *Computers and Fluids*, Vol. 98, 2014, pp. 27–53. https://doi.org/https://doi.org/10.1016/j.compfluid.2014.03.014, 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics.

[30] Farrell, P., and Maddison, J., "Conservative interpolation between volume meshes by local Galerkin projection," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 1, 2011, pp. 89–100. https://doi.org/https://doi.org/10.1016/j.cma.2010.07.015.

[31] Crabill, J. A., Sitaraman, J., and Jameson, A., *A High-Order Overset Method on Moving and Deforming Grids*, 2016. https://doi.org/10.2514/6.2016-3225.

[32] Galbraith, M. C., "A Discontinuous Galerkin Chimera Overset Solver," 2013. URL https://api.semanticscholar.org/CorpusID:63435004.

[33] Gassner, G. J., Lörcher, F., Munz, C.-D., and Hesthaven, J. S., “Polymorphic nodal elements and their application in discontinuous Galerkin methods,” *Journal of Computational Physics*, Vol. 228, No. 5, 2009, pp. 1573–1590. https://doi.org/https://doi.org/10.1016/j.jcp.2008.11.012.

[34] “4th International Workshop on High-order CFD methods,” At the Foundation for Research and Technology Hellas (FORTH) in Heraklion (Crete), 2016. URL https://how4.cenaero.be/.

[35] “5th International Workshop on High-order CFD methods,” At the 2018 AIAA SciTech in Kissimmee, Florida, 2018. URL https://how5.cenaero.be/.

[36] van Rees, W. M., Leonard, A., Pullin, D. I., and Koumoutsakos, P., “A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers,” *Journal of Computational Physics*, Vol. 230, No. 8, 2011, pp. 2794–2805. https://doi.org/10.1016/j.jcp.2010.11.031.

[37] Mossier, P., Appel, D., Beck, A., and Munz, C.-D., “An Efficient hp-Adaptive Approach for Compressible Two-Phase Flows using the Level-Set Ghost Fluid Method,” , 2023. https://doi.org/10.23967/admos.2023.039.