F. Setzwein, P. Ess, P. Gerlinger, Adaptive numerical dissipation control for high-order k-exact reconstruction schemes on vertex-centered unstructured grids using artificial neural networks, Journal of Computational Physics, Vol. 471 (2022), 111633

The original publication is available at www.elsevier.com

https://doi.org/10.1016/j.jcp.2022.111633

Preprint accepted in Journal of Computational Physics (originally submitted in March, 2022)
Link to journal version: https://doi.org/10.1016/j.jcp.2022.111633

Adaptive numerical dissipation control for high-order k-exact reconstruction schemes on vertex-centered unstructured grids using artificial neural networks

Florian Setzwein[a,*], Peter Ess[a,1], Peter Gerlinger[a,2]

[a]*German Aerospace Center (DLR), Institute of Combustion Technology, Stuttgart, 70569, Germany*

January 22, 2024

Abstract

Due to their enhanced numerical dissipation properties, high-order discretization methods are an important prerequisite to obtain accurate results with Large-Eddy Simulations. However, the exact amount of dissipation often requires a careful tuning by the user via problem-dependent parameters. In this work we present a fully adaptive dissipation control, which ensures stability and additionally reduces the numerical dissipation to a minimum. This novel approach employs a simple feedforward neural network model, which indirectly tabulates an underlying stability equation and thus reduces the computational overhead to estimate the dissipation during runtime. The methodology is adapted for a high-order k-exact reconstruction method on fully unstructured vertex-centered grids, and it is implemented in a full production flow solver. Based on several test cases, the enhanced accuracy compared to a conventional low-order scheme is demonstrated. Especially when dealing with Large-Eddy Simulation benchmarks, significant savings in computation time and grid resolution requirements can be obtained for reaching a desired level of accuracy. Moreover, compared to a high-order reconstruction method with constant numerical dissipation, the presented adaptive approach consistently yields accurate results, regardless of the flow problem.

Keywords: High-order accuracy, Unstructured grids, Finite-volume method, Von Neumann stability analysis, Artificial neural networks, Adaptive numerical dissipation

1. Introduction

Large-Eddy Simulations (LES) are increasingly used in the industrial design process to address complex vortex-dominated flow phenomena for which the Reynolds-Averaged Navier-Stokes (RANS) approach is not sufficient [1, 2]. The increased accuracy of LES compared to RANS results from the separation of the flow field into large and small scales due to a filtering operation. While the large scales are resolved directly, the influence of the small scales on the mean flow is modeled with a subgrid-scale (SGS) model. The idea is that these small scales behave more universal, which leads to a lower modeling error than in the RANS approach [3, 4]. However, the direct resolution of the large scales puts high demands on the underlying

*PhD Student, Corresponding author: Email: florian.setzwein@dlr.de;
[1]Postdoctoral fellow
[2]Professor

numerical discretization procedure, in order to accurately predict, for example, highly vortical flows [1]. As a consequence, conventional discretization methods, which were originally established for RANS simulations, can only be used in the scope of LES at a very high computational cost, since their inherently high dissipative properties must be compensated by a large number of elements. For this reason, higher-order methods are typically used, which exhibit significantly lower numerical dissipation and which are therefore capable of reducing the number of elements and the associated computational effort [1, 5]. In the past, structured approaches were often adopted for this purpose, while the current trend is shifting towards unstructured methods which offer greater geometric flexibility in the meshing process [5, 6]. Besides preserving the order of accuracy, high-order methods must exhibit a certain robustness with regard to discontinuities. Especially in compressible flows, discontinuities occur due to the hyperbolic nature of the equations and can lead to the Gibbs phenomenon. The latter refers to parasitic oscillations in the vicinity of the discontinuity, which are attributed to the approximation of flow values with higher order reconstruction polynomials [1]. Parasitic modes can also occur in incompressible flows in the absence of shocks. For example, they arise when numerical artifacts caused by boundary conditions or when insufficient discrete conservation properties are amplified by the loss of accuracy of the numerical scheme at the smallest scales of the grid [7]. In particular, central discretization schemes are prone to such oscillations, due to their low inherent numerical dissipation. Several methods exist to suppress unwanted oscillations when dealing with high-order methods. In the scope of compressible flow problems, slope limiting approaches, such as the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) [8, 9, 10] are often employed to reduce the order of accuracy in the vicinity of shocks. A similar approach is used in weighted essentially non-oscillatory (WENO) schemes, where the solution is reconstructed from a combination of several candidate stencils to form a smooth reconstruction [11, 12]. All these approaches act in the sense of Godunov's theorem, which states that a linear scheme must be first-order accurate in order to prevent the generation of new extrema [13]. However, these approaches must be implemented carefully to prevent a loss of accuracy in smooth flow regions and typically lead to an increase in computational cost. Spurious oscillations can also be damped by the introduction of artificial numerical dissipation, which goes back to the work of Von Neumann and Richtmeyer [14] and Jameson et al. [15]. This simple, yet efficient approach is also often used in central schemes to prevent odd-even decoupling of the solution [4]. The approach has also been suggested as a better alternative for the stabilization of Discontinuous Galerkin methods [1, 16]. A disadvantage of the method is that the correct amount of numerical dissipation is often difficult to predict. Hence, it is based on empirical parameters that depend on the particular flow problem.

We present a novel approach for an adaptive numerical dissipation control and apply it to an implicit high-order k-exact finite-volume scheme on vertex-centered grids [17]. The latter has recently been implemented in DLR's finite-volume flow solver ThetaCOM, a turbulent heat release extension of DLR's TAU code in its combustion version [18, 19, 20, 21, 22, 23, 24, 25], which features a memory-efficient matrix free Krylov solver for the system of linear equations and multigrid preconditioning. The scheme employs a multiple-correction approach [26], which ensures a proper reconstruction on unstructured, median-dual grids and features efficient parallelization capabilities. This methodology has been incorporated into a fractional step strategy for the solution of the incompressible Navier-Stokes equations, which requires a Poisson equation for the pressure-velocity coupling to be solved implicitly. Furthermore, a novel discretization of the convective and diffusive fluxes ensures an accurate flux approximation, even on highly irregular grids [17, 27, 28]. In this work, a Von Neumann stability analysis for a one-dimensional linear advection-diffusion equation is used to derive a general stability criterion for the k-exact vertex-centered scheme. This criterion should predict the optimum amount of numerical dissipation required to stabilize the solution. It is based on nonlinear relations between the CFL number and the Reynolds number, both of which are based on local flow and grid properties. Since the direct evaluation of this criterion requires a considerable amount of computation time, a subset of the solution space from the underlying stability equation is tabulated indirectly in the form of a simple feedforward neural network model. This enables the minimum required numerical dissipation to be determined based on local flow conditions with little additional computational effort. It also eliminates the tedious search by the user for optimum empirical parameters to stabilize the solution. In recent years, several works utilized neural network models to detect shocks, where the solution was then stabilized in the vicinity of the discontinuity, either by introducing local numerical dissipation in the context of the

Discontinuous Galerkin method [29, 30, 31] or Fourier spectral schemes [32, 33]. Beck et al. [34] developed a shock sensor for a Discontinuos Galerkin scheme, which is based on edge detection via convolutional neural networks. Stevens and Colonius [35] used a machine-learning method to adjust the finite-volume coefficients of a WENO5-JS scheme via small perturbations that stem from a trained neural network model. This approach has been shown to improve the accuracy in resolving fine-scale flow features and discontinuities and has also been adapted in the work of Kossaczka et al. [36]. In contrast to these works, our approach is not intended to improve the capture of discontinuities with higher accuracy. Rather, we aim to determine an optimum numerical dissipation for the simulation of incompressible flows. The dissipation is intended to dampen spurious oscillations that arise from an amplification of the numerical error due to the dispersive properties of the discretization scheme at wavenumbers in the range of the grid scales. In this way, a stable, yet accurate solution can be obtained without requiring the user to specify a large number of empirical parameters.

In the following, it is explained how the filtered Navier-Stokes equations can be discretized using the novel k-exact multiple-correction method for vertex-centered grids in combination with an appropriate fractional step strategy for the pressure-velocity coupling. Subsequently, the numerical properties of the spatial discretization procedure are discussed in detail. Based on the Von Neumann stability analysis for a linear advection-diffusion equation, a stability criterion for the k-exact discretization method is derived. The relationship between the numerical stability and the introduction of the numerical dissipation as a function of the exactness of the discretization procedure is also explained in more detail. It is then shown how this stability criterion can be tabulated in the form of a neural network model, and, thus, effectively evaluated during runtime to determine the optimum numerical dissipation as a function of the local flow state. Finally, the advantages of the described approach are presented on the basis of several benchmarks. Furthermore, the advantages of the higher order of accuracy in space are compared to a conventional discretization method for unstructured grids.

2. Methods

Consider a spatial filter operator for any field variable, for example the velocity $u(\mathbf{x}, t)$

$$\tilde{u}(\mathbf{x}, t) = \iiint G(\mathbf{r}, \mathbf{x})\, u(\mathbf{x} - \mathbf{r}, t)\, \mathrm{d}\mathbf{r} \tag{1}$$

with a filter kernel $G(\mathbf{r}, \mathbf{x})$. This operator is applied to the Navier-Stokes equations for an incompressible fluid with velocity $\mathbf{u}$, pressure p, viscosity ν and constant density ρ_0, which leads to the following set of equations

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0, \tag{2}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\tilde{u}_i \tilde{u}_j\right) + \frac{\partial \tau_{ij}^r}{\partial x_j} - \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j \partial x_j} = -\frac{1}{\rho_0}\frac{\partial \tilde{p}}{\partial x_i}. \tag{3}$$

The values of τ_{ij}^r refer to the anisotropic residual-stress tensor, which results from the filter operation on the convective part of the momentum equations $\widetilde{u_i u_j} = \tilde{u}_i \tilde{u}_j + \tau_{ij}^r + \frac{2}{3}\delta_{ij} k_r$. The isotropic residual stress terms $\frac{2}{3}\delta_{ij} k_r$ are included in the filtered pressure $\tilde{p}$ [37] and the anisotropic residual-stress tensor is modeled with a linear eddy-viscosity model $\tau_{ij}^r = -2\nu_r \tilde{S}_{ij}$, where the deformation tensor of the resolved flow field is defined by $\tilde{S}_{ij} = \frac{1}{2}\left(\partial \tilde{u}_i / \partial x_j + \partial \tilde{u}_j / \partial x_i\right)$. The local eddy viscosity $\nu_r(\mathbf{x}, t)$ accounts for the influence of residual motions on the large flow scales. In this work, it is modeled using the Wall-Adapting Local Eddy-Viscosity

(WALE) model [38, 39] and is calculated according to

$$\nu_r = \Delta^2 C_w^2 \frac{\left(\tilde{S}^*_{ij}\tilde{S}^*_{ij}\right)^{3/2}}{\left(\tilde{S}_{ij}\tilde{S}_{ij}\right)^{5/2} + \left(\tilde{S}^*_{ij}\tilde{S}^*_{ij}\right)^{5/4}} \qquad \text{with} \qquad \tilde{S}^*_{ij} = \frac{1}{2}\left(\tilde{g}^2_{ij} + \tilde{g}^2_{ji}\right) - \frac{1}{3}\delta_{ij}\tilde{g}^2_{kk} \qquad \text{and} \qquad \tilde{g}^2_{ji} = \frac{\partial \tilde{u}_i}{\partial x_k}\frac{\partial \tilde{u}_k}{\partial x_j}. \tag{4}$$

The terms $\tilde{S}^*_{ij}$ denote the traceless symmetric part of the squared velocity gradient tensor and Δ refers to the employed filter width. The latter is basically a threshold at which scales are no longer resolved but modeled and it is based on the local element size. The value of C_w denotes the subgrid-scale constant, which is set to 0.5 in this work [39].

Since the continuity Equation (2) is only present in terms of a constraint, there is no explicit equation available for estimating the pressure at a future time step. Thus, a standard incremental pressure correction scheme is used for the coupling of pressure and velocity [40]. First, an interim velocity field is estimated from the pressure at an initial time step t_n. If a Crank-Nicolson scheme is used, the semi-discretized momentum predictor equation can be written in terms of

$$\frac{\tilde{u}^*_i - \tilde{u}^n_i}{\Delta t} + \frac{1}{2}\frac{\partial}{\partial x_j}\mathcal{F}_{ij}(\tilde{\mathbf{u}}^*) + \frac{1}{2}\frac{\partial}{\partial x_j}\mathcal{F}_{ij}(\tilde{\mathbf{u}}^n) = -\frac{1}{\rho_0}\frac{\partial \tilde{p}^n}{\partial x_i}, \tag{5}$$

where the fluxes reduce to $\mathcal{F}_{ij}(\tilde{\mathbf{u}}) = \tilde{u}_i\tilde{u}_j - (\nu + \nu_r)\,\partial\tilde{u}_i/\partial x_j$. The unknown pressure field at a future time step t_{n+1} is estimated with a Poisson equation that is obtained by taking the divergence of the momentum equations. After the introduction of the pressure difference $\delta\tilde{p} = \tilde{p}^{n+1} - \tilde{p}^n$, this leads to

$$\frac{\partial}{\partial x_i}\left(\frac{\partial\,\delta\tilde{p}}{\partial x_i}\right) = \frac{\rho_0}{\Delta t}\left[\frac{\partial \tilde{u}^*_i}{\partial x_i} - \frac{\partial \tilde{u}^{n+1}_i}{\partial x_i}\right]. \tag{6}$$

Since the velocity field at t_{n+1} must satisfy continuity, the second term on the right hand side reduces to zero. Once the pressure difference $\delta\tilde{p}$ has been obtained, it is finally used to correct the interim velocity field, so that it satisfies continuity:

$$\tilde{u}^{n+1}_i = \tilde{u}^*_i - \frac{\Delta t}{\rho_0}\frac{\partial\,\delta\tilde{p}}{\partial x_i}. \tag{7}$$

In the following section, we first discuss the spatial discretization procedure, by means of which the solution can be reconstructed with a higher order of accuracy. This is followed by a discussion of how this methodology can be applied to the projection method.

2.1. The multiple-correction approach

For the spatial discretization, the domain $\Omega \subset \mathbb{R}^d$ is discretized by a set of linear elements, which is referred to as primary grid $\mathcal{P}(\Omega)$. The latter consists of tetrahedrals, hexahedrals, prisms or pyramids for $d = 3$ or triangles and quadrilaterals for $d = 2$. An edge-based representation of $\mathcal{P}(\Omega)$ is obtained by constructing polyhedrons from the centroids of adjacent elements, faces and edges around each primary grid node. The resulting grid consists of N non-overlapping complex polyhedral elements and is referred to as a median-dual representation $\mathcal{D}(\Omega)$. Two elements Ω_α and Ω_β are called adjacent, if they share a common face $A_{\alpha\beta}$. All adjacent elements of a cell Ω_α are referred to as its 1^{st} neighborhood, signed as $\mathbb{V}^{(1)}_\alpha$. The n^{th} neighborhood of Ω_α is defined recursively with the neighborhoods of its adjacent elements $\mathbb{V}^{(n)}_\alpha := \bigcup_{\gamma\in\mathbb{V}^{(n-1)}_\alpha} \mathbb{V}^{(1)}_\gamma$. Figure 1 shows a primary grid $\mathcal{P}(\Omega)$, its respective median-dual tesselation $\mathcal{D}(\Omega)$ and the first and second neighborhood of a median-dual cell Ω_α for $d = 2$. Central to the k-exact discretization

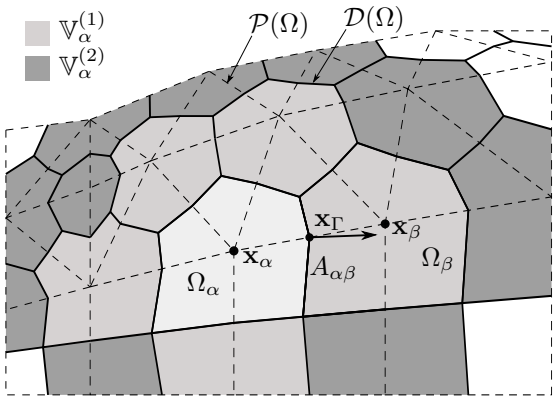


Figure 1: Median-dual grid in 2D, indicated in solid lines. The corresponding primary grid is drawn in dashed lines. The simulation variables are stored at the location of primary grid nodes, e.g. $\mathbf{x}_\alpha$ or $\mathbf{x}_\beta$.

approach is the volume-average $\overline{u}_\alpha$ in an element Ω_α

$$\overline{u}_\alpha = \frac{1}{|\Omega_\alpha|} \iiint_{\Omega_\alpha} \tilde{u}(\mathbf{x})\, \mathrm{d}V, \tag{8}$$

where $|\Omega_\alpha|$ denotes the volume of the element. For reasons of clarity, the filter operator $(\tilde{\cdot})$ will be omitted in the following. The averaging is applied to the filtered Navier-Stokes equations, such that the resulting averages in every element act as degrees of freedom in a system of N equations. The latter can then be solved essentially in three steps [41]. First, the solution of any field variable $u(\mathbf{x})$ at an initial time step t_n is reconstructed locally for every element Ω_α using a polynomial function of degree k:

$$u(\mathbf{x}) = u^{(k+1)}(\mathbf{x}; \mathbf{x}_\alpha) + \mathcal{O}\left(h^{k+1}\right), \tag{9}$$

with h being a characteristic width of the element. The superscript in brackets indicates the order of accuracy to approximate the solution. Next, the reconstruction function is used to approximate point values at element interfaces, in order to estimate the discretized fluxes $\mathcal{F}_{ij}$. Finally, the system of equations is solved to obtain the volume-averages $\overline{\mathbf{u}}_\alpha$, $\overline{p}_\alpha$ at new time step t_{n+1}. The polynomial $u^{(k+1)}(\mathbf{x}; \mathbf{x}_\alpha)$ is constrained by the conservation of the mean. It thus must satisfy the average $\overline{u}_\alpha$ with an error of $\mathcal{O}\left(h^{k+1}\right)$ when volume-integrated over Ω_α. Besides this, it must also conserve the volume-averages of elements in $\mathbb{V}_\alpha^{(k)}$. For the current approach, $u^{(k+1)}(\mathbf{x}; \mathbf{x}_\alpha)$ is defined by means of a Taylor-polynomial

$$u^{(k+1)}(\mathbf{x}; \mathbf{x}_\alpha) = u\Big|_{\mathbf{x}_\alpha}^{(k+1)} + \frac{\partial u}{\partial x_{i_1}}\Big|_{\mathbf{x}_\alpha}^{(k)} (x_{i_1} - x_{i_1,\alpha}) + \ldots + \frac{1}{k!} \frac{\partial^k u}{\partial x_{i_1} \ldots \partial x_{i_k}}\Big|_{\mathbf{x}_\alpha}^{(1)} (x_{i_1} - x_{i_1,\alpha}) \ldots (x_{i_k} - x_{i_k,\alpha}). \tag{10}$$

The point of evaluation $\mathbf{x}_\alpha$ is the primary grid position, around which Ω_α is constructed and $u|_{\mathbf{x}_\alpha}^{(k+1)}$ refers to the approximation of the point value $u(\mathbf{x}_\alpha)$ with an accuracy of $\mathcal{O}(h^{k+1})$. Respectively, the derivative terms denote k-exact differentiation operators, which approximate the n^{th} derivative of u at $\mathbf{x}_\alpha$ with an accuracy of $\mathcal{O}(h^{k-n+1})$. The point value $u|_{\mathbf{x}_\alpha}^{(k+1)}$ can be related to the volume-average $\overline{u}_\alpha$ by applying operator (8) on the reconstruction polynomial (10)

$$u\Big|_{\mathbf{x}_\alpha}^{(k+1)} = \overline{u}_\alpha - \frac{\partial u}{\partial x_{i_1}}\Big|_{\mathbf{x}_\alpha}^{(k)} \mathcal{M}_{i_1}^{(\alpha,\alpha)} - \ldots - \frac{1}{k!} \frac{\partial^k u}{\partial x_{i_1} \ldots \partial x_{i_k}}\Big|_{\mathbf{x}_\alpha}^{(1)} \mathcal{M}_{i_1 \ldots i_k}^{(\alpha,\alpha)} + \mathcal{O}\left(h^{k+1}\right). \tag{11}$$

The terms $\mathcal{M}_{i_1 \ldots i_p}^{(\alpha,\alpha)}$ refer to rank p geometric volume moment tensors. These quantities are of major importance to maintain a higher order of accuracy, especially on highly distorted grids. They are defined in a

more general way as

$$\mathcal{M}^{(\beta,\alpha)}_{i_1 i_2 \dots i_p} = \frac{1}{|\Omega_\beta|} \iiint_{\Omega_\beta} \left(x_{i_1} - x_{i_1,\alpha}\right) \left(x_{i_2} - x_{i_2,\alpha}\right) \dots \left(x_{i_p} - x_{i_p,\alpha}\right) \, \mathrm{d}V. \tag{12}$$

The first superscript β refers to the volume Ω_β for performing the averaging and the second superscript α denotes the point $\mathbf{x}_\alpha$ used for centering the moment. Further information concerning the calculation of geometric moment tensors can be found in the literature [42, 43, 44]. The rank one volume moment tensor $\mathcal{M}^{(\alpha,\alpha)}_i$ simply expresses the distance of a primary grid node $\mathbf{x}_\alpha$ towards the geometric centroid of the median-dual element Ω_α. It reduces to zero if both coincide. It remains to approximate the unknown derivative operators of the reconstruction polynomial with appropriate accuracies. In traditional approaches, this is typically addressed with a least-squares approach over a large stencil of elements in the vicinity of Ω_α. However, the non-locality of these elements can lead to a bottleneck in the parallelization and involves complicated data structures [45, 46, 47]. The multiple-correction approach avoids these problems, since unknown derivatives are calculated by successively applying a Green-Gauss gradient operator onto field variables. This only requires to exchange information between face-neighboring elements

$$\left.\frac{\partial u}{\partial x_i}\right|^{(0)}_{\mathbf{x}_\alpha} := \frac{1}{|\Omega_\alpha|} \sum_{\beta \in \mathbb{V}^{(1)}_\alpha} \frac{1}{2} \left(\overline{u}_\alpha + \overline{u}_\beta\right) \mathcal{S}^{(\alpha\beta)}_i \qquad \text{with} \qquad \mathcal{S}^{(\alpha\beta)}_i = \iint_{A_{\alpha\beta}} n_i \, \mathrm{d}A. \tag{13}$$

Here, $\boldsymbol{\mathcal{S}}^{(\alpha\beta)}$ refers to the joint normal of all sub-faces of the median-dual face $A_{\alpha\beta}$. Its norm equals the surface area $|A_{\alpha\beta}|$. As already mentioned, the superscripts of point values enclosed in brackets serve to emphasize the order of accuracy. Thus, the gradient operator in Equation (13) features a discretization error of $\mathcal{O}(1)$ as shown in a previous work [27]. The resulting gradient acts as *a priori* estimate, which must be transformed to ensure the required k-exactness constraints. This is accomplished using a correction matrix $\mathbf{G}_\alpha$, that solely depends on the mesh geometry. For a detailed derivation of $\mathbf{G}_\alpha$ for vertex-centered grids, we refer to previous works [17, 27, 28]. The correction with matrix $\mathbf{G}_\alpha$ ensures a gradient accuracy of $\mathcal{O}\left(h\right)$ on arbitrary grids. It also enforces the conservation of the mean for Ω_α and its first neighborhood with $\mathcal{O}\left(h\right)$:

$$\left.\frac{\partial u}{\partial x_i}\right|^{(1)}_{\mathbf{x}_\alpha} := G_{ij,\alpha} \left.\frac{\partial u}{\partial x_j}\right|^{(0)}_{\mathbf{x}_\alpha}. \tag{14}$$

It can be shown, that the inverse of this 1-exact gradient correction matrix $\mathbf{G}_\alpha$ must be calculated for vertex-centered grids according to

$$G^{-1}_{ij,\alpha} := \left.\frac{\partial \mathcal{M}_j}{\partial x_i}\right|^{(0)}_{\mathbf{x}_\alpha} \qquad \text{with} \qquad \left.\frac{\partial \mathcal{M}_j}{\partial x_i}\right|^{(0)}_{\mathbf{x}_\alpha} := \frac{1}{|\Omega_\alpha|} \sum_{\beta \in \mathbb{V}^{(1)}_\alpha} \frac{1}{2} \left(\mathcal{M}^{(\beta,\alpha)}_j + \mathcal{M}^{(\alpha,\alpha)}_j\right) \mathcal{S}^{(\alpha\beta)}_i. \tag{15}$$

This expression is simply a Green-Gauss gradient operation (13) applied to the respective geometric volume moment tensor entries. It should be mentioned, that the centering of the geometric volume moments in Equation (15) varies according to the element Ω_α. For a 2-exact reconstruction, the gradient of u must be provided with an accuracy of $\mathcal{O}\left(h^2\right)$ in addition to a Hessian matrix of $\mathcal{O}\left(h\right)$. The latter is again approximated through a Green-Gauss approach

$$\left.\frac{\partial^2 u}{\partial x_i \partial x_j}\right|^{(0)}_{\mathbf{x}_\alpha} := \frac{1}{|\Omega_\alpha|} \sum_{\beta \in \mathbb{V}^{(1)}_\alpha} \frac{1}{2} \left(\left.\frac{\partial u}{\partial x_i}\right|^{(1)}_{\mathbf{x}_\alpha} + \left.\frac{\partial u}{\partial x_i}\right|^{(1)}_{\mathbf{x}_\beta} \right) \mathcal{S}^{(\alpha\beta)}_j. \tag{16}$$

Similar to the gradient, there exists a linear mapping, which can be used to correct this Hessian matrix to a form that ensures an accuracy of $\mathcal{O}(h)$ on arbitrary grids. It relies on a Hessian-correction matrix $\mathbf{H}_\alpha$,

for whose detailed definition we refer to the author's previous work [17]. The definition of $\mathbf{H}_\alpha$ relies on the vectorization of the Hessian matrix into the following form

$$\widehat{\mathbf{D}}_\alpha^{(0)}(u) = \left[\left.\frac{\partial^2 u}{\partial x_1 \partial x_1}\right|_{\mathbf{x}_\alpha}^{(0)} \quad \left.\frac{\partial^2 u}{\partial x_1 \partial x_2}\right|_{\mathbf{x}_\alpha}^{(0)} \quad \left.\frac{\partial^2 u}{\partial x_1 \partial x_3}\right|_{\mathbf{x}_\alpha}^{(0)} \quad \left.\frac{\partial^2 u}{\partial x_2 \partial x_2}\right|_{\mathbf{x}_\alpha}^{(0)} \quad \left.\frac{\partial^2 u}{\partial x_2 \partial x_3}\right|_{\mathbf{x}_\alpha}^{(0)} \quad \left.\frac{\partial^2 u}{\partial x_3 \partial x_3}\right|_{\mathbf{x}_\alpha}^{(0)} \right]. \tag{17}$$

This form incorporates the symmetry properties of the Hessian matrix. Finally, a 2-exact Hessian matrix operator is introduced as

$$\widehat{D}_{i,\alpha}^{(1)}(u) := H_{ij,\alpha}\, \widehat{D}_{j,\alpha}^{(0)}(u). \tag{18}$$

During runtime, it is only necessary to compute this matrix-vector product in order to correct the Hessian operator (16) to its 2-exact form. Since this Hessian operator features an accuracy of $\mathcal{O}(h)$, it can be employed to correct the 1-exact gradient. This leads to the definition of the 2-exact gradient operator

$$\left.\frac{\partial u}{\partial x_i}\right|_{\mathbf{x}_\alpha}^{(2)} := \left.\frac{\partial u}{\partial x_i}\right|_{\mathbf{x}_\alpha}^{(1)} - \frac{1}{2} \left.\frac{\partial^2 u}{\partial x_j \partial x_k}\right|_{\mathbf{x}_\alpha}^{(1)} \left.\frac{\partial \mathcal{M}_{jk}}{\partial x_i}\right|_{\mathbf{x}_\alpha}^{(1)}. \tag{19}$$

The 1-exact gradient of the rank two volume moments must be stored in addition to the defined correction matrices for every primary grid node. It is also calculated only once prior to the simulation, if no mesh deformation is considered. Figure 2 gives an overview on the final correction procedure that is needed for a 2-exact reconstruction. For further details on the multiple-correction methodology for vertex-centered grids, we refer to [17].

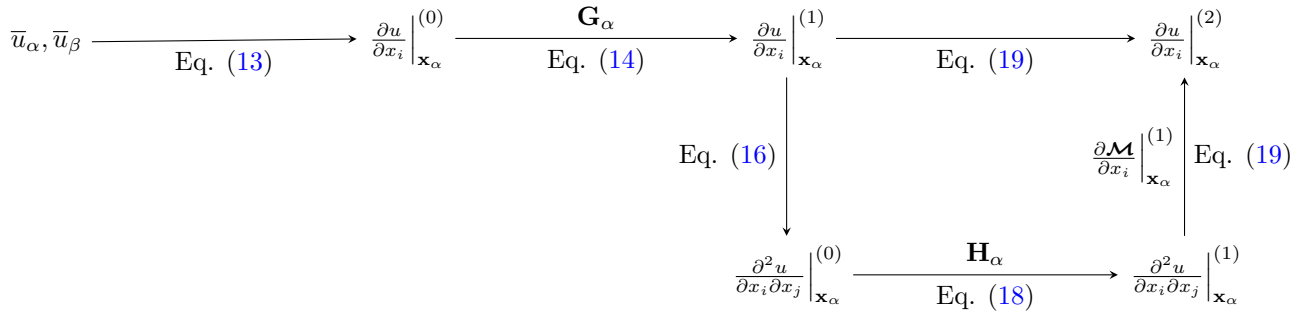


Figure 2: Procedure for the calculation of high-order accurate derivatives using the node-centered multiple-correction approach. The diagram shows the necessary steps for a 2-exact reconstruction of the gradient and the Hessian matrix.

2.2. Approximation of the momentum predictor equation

Once the derivatives are calculated with their required orders of accuracy, the discretized Navier-Stokes equations can be solved. The volume-averaged momentum predictor equations (5) then read

$$\frac{\overline{u}_{i,\alpha}^{*} - \overline{u}_{i,\alpha}^{n}}{\Delta t} + \frac{1}{|\Omega_\alpha|} \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} \left[\frac{1}{2} \iint_{A_{\alpha\beta}} \mathcal{F}_{ij}\left(\overline{\mathbf{u}}^{*}\right) n_j \,\mathrm{d}A + \frac{1}{2} \iint_{A_{\alpha\beta}} \mathcal{F}_{ij}\left(\overline{\mathbf{u}}^{n}\right) n_j \,\mathrm{d}A \right] = -\frac{1}{\rho_0} \overline{\left(\frac{\partial p^n}{\partial x_i}\right)}_\alpha. \tag{20}$$

As stated, the filter-operator has been omitted for reasons of clarity. It remains to approximate the fluxes, which are composed of convective and diffusive parts

$$\iint_{A_{\alpha\beta}} \mathcal{F}_{ij}\left(\overline{\mathbf{u}}\right) n_j \,\mathrm{d}A = \iint_{A_{\alpha\beta}} \left(u_i u_j\right) n_j \,\mathrm{d}A + \iint_{A_{\alpha\beta}} \left(\nu + \nu_r\right) \left(\frac{\partial u_i}{\partial x_j}\right) n_j \,\mathrm{d}A \tag{21}$$

In order to approximate the surface-integrals with an appropriate order of accuracy, a single-point surface integration method is applied. The approach is based on a Taylor series expansion around a single point $\mathbf{x}_\Gamma$ on a median-dual face $A_{\alpha\beta}$. The surface-integral of any flux function f_i can then be approximated by means of the reconstructed value $f_i|_{\mathbf{x}_\Gamma}$ and its derivatives at point $\mathbf{x}_\Gamma$:

$$\iint_{A_{\alpha\beta}} f_i \, n_i \, \mathrm{d}A = f_i \Big|_{\mathbf{x}_\Gamma} \mathcal{S}_i^{(\alpha\beta)} + \frac{\partial f_i}{\partial x_{j_1}}\bigg|_{\mathbf{x}_\Gamma} \mathcal{S}_{i,j_1}^{(\alpha\beta)} + \ldots + \frac{1}{k!} \frac{\partial^k f_i}{\partial x_{j_1} \ldots \partial x_{j_k}}\bigg|_{\mathbf{x}_\Gamma} \mathcal{S}_{i,j_1 \ldots j_k}^{(\alpha\beta)} + \mathcal{O}\left(h^{k+1}\right). \tag{22}$$

This integration method relies on the definition of the rank p geometric surface moments, which are generally defined for a face $A_{\alpha\beta}$ by

$$\mathcal{S}_{i,j_1 j_2 \ldots j_p}^{(\alpha\beta)} = \iint_{A_{\alpha\beta}} n_i \left(x_{j_1} - x_{j_1,\Gamma}\right)\left(x_{j_2} - x_{j_2,\Gamma}\right) \ldots \left(x_{j_p} - x_{j_p,\Gamma}\right) \mathrm{d}A. \tag{23}$$

The subscripts i and j_p are separated by a comma to highlight that i indicates the face normal direction and j_p the spatial direction of the terms $\left(x_{j_p} - x_{j_p,\Gamma}\right)$. The superscripts $(\alpha\beta)$ indicate the elements Ω_α and Ω_β adjacent to the face, on which point $\mathbf{x}_\Gamma$ for the Taylor series expansion is located. The rank zero surface moment has already been introduced as the joint normal of the face $A_{\alpha\beta}$. Note that in contrast to the geometric volume moment formulation, the surface moments are not normalized with the interface area $|A_{\alpha\beta}|$. This is due to reasons of clarity for the further derivation of the scheme. All geometric surface moments depend solely on the grid geometry and can be calculated analytically in a preprocessing step prior to the simulation. It remains to reconstruct the flux function $f_i|_{\mathbf{x}_\Gamma}$ and its derivatives at the cell interfaces, which is accomplished via the reconstruction polynomials of respective field variables from adjacent elements. After some algebra and exploiting the chain rule, the convective fluxes can be approximated for a 2-exact reconstruction approach according to

$$\iint_{A_{\alpha\beta}} (u_i u_j)\, n_j \, \mathrm{d}A = \dot{m}\Big|_{\mathbf{x}_\Gamma}^{(3)} u_i\Big|_{\mathbf{x}_\Gamma}^{(3)} + \dot{m}_j\Big|_{\mathbf{x}_\Gamma}^{(3)} \frac{\partial u_i}{\partial x_j}\bigg|_{\mathbf{x}_\Gamma}^{(2)} + \frac{1}{2}\dot{m}_{jk}\Big|_{\mathbf{x}_\Gamma}^{(3)} \frac{\partial^2 u_i}{\partial x_j \partial x_k}\bigg|_{\mathbf{x}_\Gamma}^{(1)} + |A_{\alpha\beta}|\,\mathcal{O}\left(h^3\right). \tag{24}$$

The values at the interface point $\mathbf{x}_\Gamma$ are calculated with the available reconstruction polynomials of the adjacent elements as

$$\begin{aligned}
u\Big|_{\mathbf{x}_\Gamma}^{(3)} &= \left(\frac{1+\theta}{2}\right) u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_U) + \left(\frac{1-\theta}{2}\right) u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_D), \\
\frac{\partial u}{\partial x_i}\bigg|_{\mathbf{x}_\Gamma}^{(2)} &= \left(\frac{1+\theta}{2}\right) \frac{\partial}{\partial x_i} u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_U) + \left(\frac{1-\theta}{2}\right) \frac{\partial}{\partial x_i} u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_D), \\
\frac{\partial^2 u}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}_\Gamma}^{(1)} &= \left(\frac{1+\theta}{2}\right) \frac{\partial^2}{\partial x_i \partial x_j} u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_U) + \left(\frac{1-\theta}{2}\right) \frac{\partial^2}{\partial x_i \partial x_j} u^{(3)}(\mathbf{x}_\Gamma;\mathbf{x}_D).
\end{aligned} \tag{25}$$

Here, U is the upwind and D the downwind direction at the interface. Depending on the exactness of the scheme, the respective correction operations on the derivatives are applied. For a 1-exact scheme, only first derivates and rank one geometric surface moments are used. The parameter $\theta \in [0,1]$ is used to shift the bias towards the element in the upwind direction. A value of $\theta = 0$ results in a purely central discretization, whereas with $\theta = 1$ the bias is set fully towards the upwind direction. As will be shown below, a variation of θ does not reduce the order of accuracy of the scheme, but introduces numerical dissipation. This makes it a suitable parameter to stabilize the solution. In addition, a limiter variable ψ_α is introduced for every reconstruction polynomial $u^{(k)}(\mathbf{x};\mathbf{x}_\alpha)$. This allows to reduce the order of accuracy locally, for example for

highly deformed elements. For a 2-exact reconstruction, the reconstruction polynomial reads

$$u^{(3)}(\mathbf{x};\mathbf{x}_\alpha) = \overline{u}_\alpha + \psi_\alpha \left[\frac{\partial u}{\partial x_{i_1}}\bigg|_{\mathbf{x}_\alpha}^{(2)} \left(x_i - x_{i,\alpha} - \mathcal{M}_i^{(\alpha,\alpha)} \right) + \frac{1}{2} \frac{\partial^2 u}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}_\alpha}^{(1)} \left((x_i - x_{i,\alpha})(x_j - x_{j,\alpha}) - \mathcal{M}_{ij}^{(\alpha,\alpha)} \right) \right]. \tag{26}$$

In this work, the order of accuracy is reduced locally for elements adjacent to Dirichlet boundary conditions. These elements feature small stencils and often exhibit high geometric moments due to the median-dual grid treatment at boundaries. The latter can lead to ill-conditioned correction matrices [28] which may induce instabilities in case of anisotropic flow phenomena. The limiter variable ψ_α is used only as a switch to disable the higher order accuracy in such boundary elements. It is thus only defined for the discrete values $\psi_\alpha \in \{0, 1\}$. The mass flux terms $\dot{m}$ in Equation (24) are defined by

$$\dot{m}\Big|_{\mathbf{x}_\Gamma}^{(3)} = u_i\Big|_{\mathbf{x}_\Gamma}^{(3)} \mathcal{S}_i^{(\alpha\beta)} + \frac{\partial u_i}{\partial x_j}\bigg|_{\mathbf{x}_\Gamma}^{(2)} \mathcal{S}_{i,j}^{(\alpha\beta)} + \frac{1}{2} \frac{\partial^2 u_i}{\partial x_j \partial x_k}\bigg|_{\mathbf{x}_\Gamma}^{(1)} \mathcal{S}_{i,jk}^{(\alpha\beta)}, \tag{27a}$$

$$\dot{m}_i\Big|_{\mathbf{x}_\Gamma}^{(3)} = u_j\Big|_{\mathbf{x}_\Gamma}^{(3)} \mathcal{S}_{j,i}^{(\alpha\beta)} + \frac{\partial u_j}{\partial x_k}\bigg|_{\mathbf{x}_\Gamma}^{(2)} \mathcal{S}_{j,ki}^{(\alpha\beta)}, \tag{27b}$$

$$\dot{m}_{ij}\Big|_{\mathbf{x}_\Gamma}^{(3)} = u_k\Big|_{\mathbf{x}_\Gamma}^{(3)} \mathcal{S}_{k,ij}^{(\alpha\beta)}. \tag{27c}$$

All these terms scale with $|A_{\alpha\beta}|$ due to the surface integration and the respective definition of the geometric surface moment tensors. Furthermore, the mass flux terms (27b) and (27c) scale also with h and h^2, respectively, which leads to the overall accuracy of $\mathcal{O}(h^3)$. The mass flux tensors at the interface node $\mathbf{x}_\Gamma$ are calculated with a purely central averaging of the reconstructed values from both adjacent elements ($\theta = 0$) and only once at the beginning of every time step t_n.

The Laplacian fluxes $f_i = (\nu + \nu_r)(\partial u / \partial x_i)$ are calculated using the Mathur-Murthy scheme [48, 49], that has recently been extended for the k-exact reconstruction on unstructured grids [17]. To prevent spurious oscillations, it is desired to express the surface-integral in terms of adjacent volume-averages $\overline{u}_\alpha$ and $\overline{u}_\beta$ [48]. This can be achieved through the definition of the face-dependent factor $\varepsilon^{(\alpha\beta)}$

$$\varepsilon^{(\alpha\beta)} = \frac{\mathcal{S}_i^{(\alpha\beta)} \mathcal{S}_i^{(\alpha\beta)}}{\Delta x_{j,\,\beta\alpha}\, \mathcal{S}_j^{(\alpha\beta)}}, \tag{28}$$

with the distance vector $\Delta x_{i,\,\beta\alpha} = x_{i,\,\beta} - x_{i,\alpha}$. By definition, Ω_α is the element with the outward facing normal in this context. It is now possible to express the Laplacian fluxes into three parts

$$\iint_{A_{\alpha\beta}} (\nu + \nu_r) \left(\frac{\partial u}{\partial x_i} \right) n_i \, \mathrm{d}A = (\nu + \nu_r|_\Gamma) \left[F_{L_{MM}}^{(\alpha\beta)} + F_{L_{EX1}}^{(\alpha\beta)} + F_{L_{EX2}}^{(\alpha\beta)} \right] + |A_{\alpha\beta}| \, \mathcal{O}\left(h^2\right). \tag{29}$$

Note that the error that results from approximating the local eddy viscosity $\nu_r|_\Gamma$ at the face is not considered in the error analysis here. The first term of Equation (29) recovers the original formulation of Mathur and Murthy [48]

$$F_{L_{MM}}^{(\alpha\beta)}(\overline{u}) = \varepsilon^{(\alpha\beta)} (\overline{u}_\beta - \overline{u}_\alpha) - \frac{1}{2} \left(\frac{\partial u}{\partial x_i}\bigg|_{\mathbf{x}_\alpha}^{(2)} + \frac{\partial u}{\partial x_i}\bigg|_{\mathbf{x}_\beta}^{(2)} \right) \left(\varepsilon^{(\alpha\beta)} \Delta x_{i,\,\beta\alpha} - \mathcal{S}_i^{(\alpha\beta)} \right). \tag{30}$$

The terms $F_{L_{EX1}}^{(\alpha\beta)}$ and $F_{L_{EX2}}^{(\alpha\beta)}$ are designated as contributions to enhance the accuracy in the scope of the k-exact reconstruction approach. The term $F_{L_{EX1}}^{(\alpha\beta)}$ accounts for the 1-exact reconstruction of the diffusion

flux integral

$$F_{L_{EX1}}^{(\alpha\beta)}\left(\overline{u}\right) = -\frac{1}{2}\varepsilon^{(\alpha\beta)}\left(\left.\frac{\partial u}{\partial x_i}\right|_{\mathbf{x}_\alpha}^{(2)} + \left.\frac{\partial u}{\partial x_i}\right|_{\mathbf{x}_\beta}^{(2)}\right)\left(\mathcal{M}_i^{(\beta,\,\beta)} - \mathcal{M}_i^{(\alpha,\alpha)}\right). \tag{31}$$

Further correction terms needed to preserve a 2-exact solution are incorporated in the term $F_{L_{EX2}}^{(\alpha\beta)}$

$$\begin{aligned} F_{L_{EX2}}^{(\alpha\beta)}\left(\overline{u}\right) = &\frac{1}{2}\left(\left.\frac{\partial^2 u}{\partial x_i \partial x_j}\right|_{\mathbf{x}_\alpha}^{(1)} + \left.\frac{\partial^2 u}{\partial x_i \partial x_j}\right|_{\mathbf{x}_\beta}^{(1)}\right)\left[\mathcal{S}_{i,j}^{(\alpha\beta)} - \frac{1}{2}\varepsilon^{(\alpha\beta)}\left(\mathcal{M}_{ij}^{(\beta,\,\Gamma)} - \mathcal{M}_{ij}^{(\alpha,\,\Gamma)}\right)\right] \\ &- \frac{1}{4}\left(\left.\frac{\partial^2 u}{\partial x_i \partial x_j}\right|_{\mathbf{x}_\alpha}^{(1)} - \left.\frac{\partial^2 u}{\partial x_i \partial x_j}\right|_{\mathbf{x}_\beta}^{(1)}\right)\left[\varepsilon^{(\alpha\beta)}\left(\Delta x_{i,\,\beta\alpha} + \mathcal{M}_i^{(\beta,\,\beta)} - \mathcal{M}_i^{(\alpha,\alpha)}\right) - \mathcal{S}_i^{(\alpha\beta)}\right]\Delta x_{j,\,\beta\alpha}. \end{aligned} \tag{32}$$

The derivation for the k-exact contribution terms is given in [17], where the order of accuracy for this approach is also discussed in greater detail. It remains to discretize the volume-averaged pressure gradient in Equation (20), which is treated as a source term. To maintain the order of accuracy, it was found that each gradient component must be approximated with a flux-formulation similar to the convective terms [17]

$$\overline{\left(\frac{\partial p}{\partial x_i}\right)}_\alpha = \frac{1}{|\Omega_\alpha|}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}} p\Big|_{\mathbf{x}_\Gamma}^{(3)}\mathcal{S}_i^{(\alpha\beta)} + \left.\frac{\partial p}{\partial x_j}\right|_{\mathbf{x}_\Gamma}^{(2)}\mathcal{S}_{i,j}^{(\alpha\beta)} + \frac{1}{2}\left.\frac{\partial^2 p}{\partial x_j \partial x_k}\right|_{\mathbf{x}_\Gamma}^{(1)}\mathcal{S}_{i,jk}^{(\alpha\beta)} + \mathcal{O}\left(h^3\right). \tag{33}$$

The reconstructed pressure face values and their derivatives are estimated through a central approximation ($\theta = 0$).

2.3. Approximating the pressure equation and correction of the velocity field

With the discretization methods described above, it is possible to determine the intermediate velocity field $\mathbf{u}^*$. Next, the latter is used to estimate the pressure at t_{n+1} through the discretized Poisson equation (6)

$$\sum_{\beta\in\mathbb{V}_\alpha^{(1)}}\iint_{A_{\alpha\beta}}\left(\frac{\partial\,\delta p}{\partial x_i}\right)n_i\,\mathrm{d}A = \frac{\rho_0}{\Delta t}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}}\iint_{A_{\alpha\beta}} u_i^* n_i\,\mathrm{d}A. \tag{34}$$

The integrals comprising the pressure gradients are approximated via the k-exact diffusive flux formulation (29). The surface-integral on the right hand side, which involves the interim velocity field, is approximated according to

$$\iint_{A_{\alpha\beta}} u_i^*\,n_i\,\mathrm{d}A = \dot{m}^*\Big|_{\mathbf{x}_\Gamma}^{(3)} + |A_{\alpha\beta}|\,\mathcal{O}\left(h^3\right), \tag{35}$$

where the mass flux term $\dot{m}^*\big|_{\mathbf{x}_\Gamma}^{(k+1)}$ is calculated with Equation (27a), under consideration of the intermediate velocity field u_i^*. Once the pressure difference δp has been estimated, it is used to update the velocity field

$$\overline{u}_{i,\alpha}^{n+1} = \overline{u}_{i,\alpha}^{n} - \frac{\Delta t}{\rho_0}\overline{\left(\frac{\partial\delta p}{\partial x_i}\right)}_\alpha, \tag{36}$$

where the averaged gradient of δp is calculated similarly to the pressure gradient in Equation (33). Finally, the mass flux terms $\dot{m}_i^{n+1}\big|_{\mathbf{x}_\Gamma}^{(3)}$ must be updated at every element interface. The discretization and

rearrangement of Equation (6) leads to

$$\iint_{A_{\alpha\beta}} u_i^{n+1}\, n_i \,\mathrm{d}A = \iint_{A_{\alpha\beta}} u_i^{*}\, n_i \,\mathrm{d}A - \frac{\Delta t}{\rho_0} \iint_{A_{\alpha\beta}} \left(\frac{\partial \delta p}{\partial x_i}\right) n_i \,\mathrm{d}A. \tag{37}$$

This constraint leads to the enforcement of continuity on every interface $A_{\alpha\beta}$ and has been described in the work of Jofre et al. [50].

$$\dot{m}_i^{n+1}\big|_{\mathbf{x}_\Gamma}^{(3)} = \dot{m}_i^{*}\big|_{\mathbf{x}_\Gamma}^{(3)} - \frac{\Delta t}{\rho_0}\left[F_{L_{MM}}^{(\alpha\beta)}\left(\overline{p}\right) + F_{L_{EX1}}^{(\alpha\beta)}\left(\overline{p}\right) + F_{L_{EX2}}^{(\alpha\beta)}\left(\overline{p}\right)\right]. \tag{38}$$

Formally, the truncation error of this approach is $\mathcal{O}\left(h^2\right)$ and thus larger than just employing Equation (27a). However, numerical experiments showed, that the actual numerical error is indistinguishable, whereas the numerical stability increases significantly. The higher order terms $\dot{m}_i^{n+1}\big|_{\mathbf{x}_\Gamma}^{(3)}$ and $\dot{m}_{ij}^{n+1}\big|_{\mathbf{x}_\Gamma}^{(3)}$ are simply calculated through Equations (27b) and (27c) using the updated velocity field.

3. Adaptive control of the numerical dissipation

In Equation (25), the upwind bias θ has been introduced for the calculation of convective fluxes. In this section, it will be shown how this parameter is related to numerical dissipation and how it must be controlled to ensure stability. The latter is based on a Von Neumann stability analysis for a linear advection-diffusion equation. This equation represents a simplification compared to the Navier-Stokes equations (2) and (3). Hence, while it cannot provide necessary criteria for ensuring stability, it can provide sufficient ones. Finally, a strategy is presented to adjust θ automatically. This method utilizes a simple neural network model to predict the upwind bias based on local flow conditions, such that the stability of the scheme is ensured and such that the numerical dissipation is reduced to a minimum.

3.1. Analysis of the numerical error for the convective operator

For the following analysis, consider a linear advection-diffusion equation for a scalar u with a uniform velocity U and diffusivity ν

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0. \tag{39}$$

A one-dimensional, periodic domain $x \in [0, L]$ is taken into consideration, that is discretized using a grid of elements Ω_α of size h and whose centroids x_α are placed equidistantly. The linear advection-diffusion equation is volume-averaged according to Equation (8). First, we will examine how changes in the upwind bias θ affect the numerical error of the volume-averaged convective operator $\partial u/\partial x$. Following the methodology described above, the latter can be expressed in terms of

$$\frac{1}{|\Omega_\alpha|}\int_{\Omega_\alpha}\frac{\partial u}{\partial x}\,\mathrm{d}V = \frac{1}{|\Omega_\alpha|}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}}\left(u\Big|_{x_\Gamma}\mathcal{S}_1^{(\alpha\beta)} + \frac{\partial u}{\partial x}\Big|_{x_\Gamma}\mathcal{S}_{1,1}^{(\alpha\beta)} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}\Big|_{x_\Gamma}\mathcal{S}_{1,11}^{(\alpha\beta)} + \dots\right) = \frac{1}{h}\left(u\Big|_{x_{\alpha+1/2}} - u\Big|_{x_{\alpha-1/2}}\right). \tag{40}$$

It can be shown, that the surface moments of rank higher than zero cancel out in the one-dimensional case. Thus, only the values at the element faces $x_{\alpha-1/2}$ and $x_{\alpha+1/2}$ must be reconstructed following Equation (25). Applying the limitation approach from the previous section, the 1- or 2-exact reconstruction polynomials of element Ω_α can be used to approximate the values at the element interface $x_{\alpha+1/2}$:

$$u^{(2)}(x_{\alpha+1/2}; x_\alpha) = \overline{u}_\alpha + \psi_\alpha \frac{h}{2}\frac{\partial u}{\partial x}\Big|_{x_\alpha}^{(1)} \qquad \text{and} \qquad u^{(3)}(x_{\alpha+1/2}; x_\alpha) = \overline{u}_\alpha + \psi_\alpha\left(\frac{h}{2}\frac{\partial u}{\partial x}\Big|_{x_\alpha}^{(2)} + \frac{h^2}{12}\frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha}^{(1)}\right). \tag{41}$$

Table 1: Parameters for the flux calculation with the k-exact discretization schemes.

Scheme	a_U	a_D	b_U	b_D	c_U	c_D
$k=2$	$(1+\theta)/2$	$(1-\theta)/2$	$\psi_U(1+\theta)/4$	$-\psi_D(1-\theta)/4$	$\psi_U(1+\theta)/24$	$\psi_D(1-\theta)/24$
$k=1$	$(1+\theta)/2$	$(1-\theta)/2$	$\psi_U(1+\theta)/4$	$-\psi_D(1-\theta)/4$	0	0
$k=0$	$(1+\theta)/2$	$(1-\theta)/2$	0	0	0	0

Note, that the conversion from point-values to volume-averages has been considered for $k=2$, since the geometric volume-moments reduce to $\mathcal{M}_{i_1\dots i_p}^{(\alpha,\alpha)} = \frac{h^p}{(p+1)2^{p+1}}\left[1-(-1)^{p+1}\right]$ in the one-dimensional case, to give

$$u\Big|_{\alpha} = \overline{u}_\alpha - \psi_\alpha \frac{h^2}{24}\frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha} + \mathcal{O}\left(h^4\right). \tag{42}$$

The final face value is calculated according to Equation (25), where element Ω_α is assumed to be located in the upwind direction. This leads to the following definition for the reconstructed value of u at the right face of Ω_α:

$$u\Big|_{\alpha+1/2} \approx a_U\overline{u}_\alpha + a_D\overline{u}_{\alpha+1} + \left(b_U\frac{\partial u}{\partial x}\Big|_{x_\alpha} + b_D\frac{\partial u}{\partial x}\Big|_{x_{\alpha+1}}\right)h + \left(c_U\frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha} + c_D\frac{\partial^2 u}{\partial x^2}\Big|_{x_{\alpha+1}}\right)h^2, \tag{43}$$

where the coefficients a_U, a_D, b_U, b_D, c_U and c_D are given in Table 1. Besides the upwind bias θ, these coefficients also incorporate the limiter variables ψ_U and ψ_D of the adjacent elements. The derivatives can be estimated from the application of Equations (14), (18) and (19), which simplifies to

$$\frac{\partial u}{\partial x}\Big|_{x_\alpha} = \frac{\overline{u}_{\alpha+1}-\overline{u}_{\alpha-1}}{2h} + \mathcal{O}\left(h^2\right), \qquad \frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha} = \frac{\overline{u}_{\alpha+2}-2\overline{u}_\alpha+\overline{u}_{\alpha-2}}{4h^2} + \mathcal{O}\left(h^2\right), \tag{44}$$

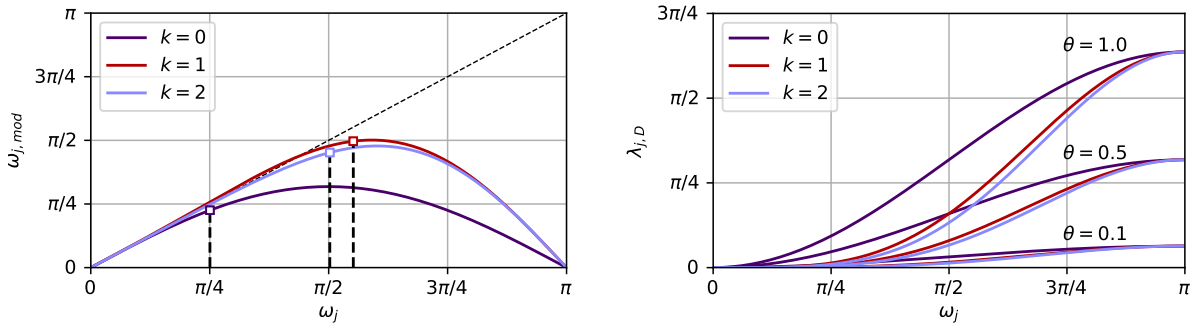
for this one-dimensional case. This is due to the fact, that the correction matrices $\mathbf{G}_\alpha$ and $\mathbf{H}_\alpha$ reduce to unit matrices. The final formulations for the left and right face values can now be used to estimate the element stencils for the volume-averaged convective operator (40). Once this is done, the influence of the upwind bias θ on the truncation error can be revealed after some algebra. For reasons of clarity, only the case $\psi_\alpha = 1$ is presented here:

$$\begin{aligned}\frac{1}{h}\left(u\Big|^{(0)}_{\alpha+1/2} - u\Big|^{(0)}_{\alpha-1/2}\right) &= \frac{1}{2h}\left(\overline{u}_{\alpha+1}-\overline{u}_{\alpha-1}\right) - \frac{\theta}{2h}\left(\overline{u}_{\alpha+1}-2\overline{u}_\alpha+\overline{u}_{\alpha-1}\right)\\ &= \overline{\left(\frac{\partial u}{\partial x}\right)}_\alpha - \frac{1}{2}\theta\, h\frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha} + \frac{1}{6}h^2\frac{\partial^3 u}{\partial x^3}\Big|_{x_\alpha} + \mathcal{O}\left(h^3\right),\end{aligned} \tag{45}$$

$$\begin{aligned}\frac{1}{h}\left(u\Big|^{(1)}_{\alpha+1/2} - u\Big|^{(1)}_{\alpha-1/2}\right) &= \frac{1}{8h}\left(-\overline{u}_{\alpha+2}+6\overline{u}_{\alpha+1}-6\overline{u}_{\alpha-1}+\overline{u}_{\alpha-2}\right) - \frac{\theta}{8h}\left(-\overline{u}_{\alpha+2}+4\overline{u}_{\alpha+1}-6\overline{u}_\alpha+4\overline{u}_{\alpha-1}-\overline{u}_{\alpha-2}\right)\\ &= \overline{\left(\frac{\partial u}{\partial x}\right)}_\alpha - \frac{1}{12}h^2\frac{\partial^3 u}{\partial x^3}\Big|_{x_\alpha} + \frac{1}{8}\theta\, h^3\frac{\partial^4 u}{\partial x^4}\Big|_{x_\alpha} + \mathcal{O}\left(h^4\right),\end{aligned} \tag{46}$$

$$\frac{1}{h}\left(u\Big|_{\alpha+1/2}^{(2)} - u\Big|_{\alpha-1/2}^{(2)}\right) = \frac{1}{96h}\left(\overline{u}_{\alpha+3} - 12\overline{u}_{\alpha+2} + 69\overline{u}_{\alpha+1} - 69\overline{u}_{\alpha-1} + 12\overline{u}_{\alpha-2} - \overline{u}_{\alpha-3}\right)$$
$$- \frac{\theta}{96h}\left(\overline{u}_{\alpha+3} - 14\overline{u}_{\alpha+2} + 47\overline{u}_{\alpha+1} - 68\overline{u}_{\alpha} + 47\overline{u}_{\alpha-1} - 14\overline{u}_{\alpha-2} + \overline{u}_{\alpha-3}\right) \quad (47)$$
$$= \overline{\left(\frac{\partial u}{\partial x}\right)}_{\alpha} + \frac{1}{12}\theta\, h^3 \frac{\partial^4 u}{\partial x^4}\bigg|_{x_\alpha} - \frac{1}{80}h^4 \frac{\partial^5 u}{\partial x^5}\bigg|_{x_\alpha} + \mathcal{O}\left(h^5\right).$$

In order to obtain these results, the conversion from point-values to volume-averages must be considered for the derivatives in the truncation error analysis, similarly to Equation (42). A related analysis has been performed in the work of Pont et al. [51], but with regard to a point-valued convective operator. For all three schemes considered, θ activates a stencil of elements which mimics a discrete operator of even order derivatives. Furthermore, for a fully central discretization ($\theta = 0$) the 0-exact scheme yields a second order of accuracy and the 2-exact scheme is even fourth order accurate. The 1-exact scheme maintains a second order accurate solution irrespective of θ. Of course, these results remain valid if the grid spacing h is kept constant. For a non-equidistant, one-dimensional grid, the varying element spacing would lead the correction matrices $\mathbf{G}_\alpha$ and $\mathbf{H}_\alpha$ to deviate from unit matrices, which in turn affects the accuracy of the discrete derivatives (Equation (44)). Besides this, the surface integration introduces further numerical error sources if the grid is defined in higher dimensions than one.



(a) Dispersive error

(b) Dissipative error

Figure 3: Results from the spectral analysis for all three schemes. The modified wavenumber $\omega_{j,mod}$ on the left reveals the dispersive properties of the discrete convective operators, which is not affected by the upwind bias θ. The squared markers refer to the cutoff wavenumber $\omega_{j,c}$ of the respective discretization schemes, at which $\omega_{j,mod}$ differs from ω_j by 10%. The dampening term $\lambda_{j,D}$ on the right can be associated to the numerical dissipation.

The spectral properties of the scheme are evaluated through a Fourier analysis [52, 53], where the scalar $\overline{u}_\alpha$ is decomposed into its Fourier modes $\hat{u}_j(t)\,\exp\left(I2\pi j x_\alpha/L\right)$. For this, the scaled wavenumber $\omega_j = 2\pi j h/L$ is introduced and the relations $x_\alpha = \alpha h$ and $I = \sqrt{-1}$ are employed. Inserting the Fourier modes into Equations (45)- (47) reveals the numerical dispersion error by comparing the Fourier coefficients of the approximate first derivative $\partial\hat{u}_j^{(k)}/\partial x = \left(\lambda_{j,D} + I\omega_{j,mod}\right)\hat{u}_j$ to their analytical counterpart $\partial\hat{u}_j/\partial x = I\omega_j\,\hat{u}_j$. The resulting effective wavenumber $\omega_{j,mod}$ should resemble the exact wavenumber ω_j as closely as possible.

Table 2: Implicit stencil coefficients for the Von Neumann stability analysis of the 1- and 2-exact schemes.

Scheme	$S_{-1}^{(IMP)}$	$S_{0}^{(IMP)}$	$S_{1}^{(IMP)}$
$k=1,2$	$-\frac{1+\theta}{4}-\frac{1}{2Re_h}$	$\frac{\theta}{2}+\frac{1}{Re_h}$	$\frac{1-\theta}{4}-\frac{1}{2Re_h}$

Table 3: Explicit stencil coefficients for the Von Neumann stability analysis of the 1- and 2-exact schemes.

Element	$S_m^{(EXP)}$ $(k=1)$	$S_m^{(EXP)}$ $(k=2)$
$\alpha-3$	0	$-\frac{\psi_U}{96}(\theta+1)$
$\alpha-2$	$\frac{\psi_U}{8}(\theta+1)$	$\frac{13\psi_U}{96}(\theta+1)+\frac{\psi_D}{96}(\theta-1)$
$\alpha-1$	$-\frac{1}{2}\frac{1}{Re_h}-\frac{1}{4}(\theta+1)-\frac{\psi_U}{8}(\theta+1)+\frac{\psi_D}{8}(\theta-1)$	$-\frac{1}{2}\frac{1}{Re_h}-\frac{1}{4}(\theta+1)-\frac{5\psi_U}{48}(\theta+1)+\frac{11\psi_D}{96}(\theta-1)$
α	$\frac{1}{Re_h}+\frac{\theta}{2}-\frac{\psi_U}{8}(\theta+1)-\frac{\psi_D}{8}(\theta-1)$	$\frac{1}{Re_h}+\frac{\theta}{2}-\frac{7\psi_U}{48}(\theta+1)-\frac{7\psi_D}{48}(\theta-1)$
$\alpha+1$	$-\frac{1}{2}\frac{1}{Re_h}-\frac{1}{4}(\theta-1)+\frac{\psi_U}{8}(\theta+1)-\frac{\psi_D}{8}(\theta-1)$	$-\frac{1}{2}\frac{1}{Re_h}-\frac{1}{4}(\theta-1)+\frac{11\psi_U}{96}(\theta+1)-\frac{5\psi_D}{48}(\theta-1)$
$\alpha+2$	$\frac{\psi_D}{8}(\theta-1)$	$\frac{\psi_U}{96}(\theta+1)+\frac{13\psi_D}{96}(\theta-1)$
$\alpha+3$	0	$-\frac{\psi_D}{96}(\theta-1)$

$$
\begin{aligned}
(\lambda_{j,D}+I\omega_{j,mod})_{k=0} &= \theta\left[1-\cos(\omega_j)\right]+I\sin(\omega_j),\\
(\lambda_{j,D}+I\omega_{j,mod})_{k=1} &= \frac{1}{4}\theta\left[3-4\cos(\omega_j)+\cos(2\omega_j)\right]+I\frac{1}{4}\left[6\sin(\omega_j)-\sin(2\omega_j)\right],\\
(\lambda_{j,D}+I\omega_{j,mod})_{k=2} &= \frac{1}{48}\theta\left[34-47\cos(\omega_j)+14\cos(2\omega_j)-\cos(3\omega_j)\right]\\
&\quad+I\frac{1}{48}\left[69\sin(\omega_j)-12\sin(2\omega_j)+\sin(3\omega_j)\right].
\end{aligned}
\tag{48}
$$

For all three schemes the upwind bias θ controls the occurence of a real-valued damping term $\lambda_{j,D}$, which can be associated with the numerical diffusion. On the other hand, the effective wavenumber $\omega_{j,mod}$ is independent of θ. Its deviation from the exact wavenumber ω_j can be related to a dispersive error. The latter is shown in Figure 3a for the three considered schemes. The dispersive error is given by means of the deviation of the modified wavenumber curve to the dashed straight line. For all three schemes, the largest deviation occurs at the maximum wavenumber $\omega_j=\pi$. The dispersion error at these wavenumber affects scales that are close to the mesh size h [7]. These errors can manifest in the generation of spurious oscillations, which in turn can lead to severe distortions of the solution. It is essential that sufficient numerical dissipation is introduced in this range to avoid the generation of such artifacts. The squared markers highlight the cutoff wavenumbers $w_{j,c}$ of the three schemes, where the modified wavenumber $\omega_{j,mod}$ and the actual wavenumber ω_j differ by 10 % [54]. The 0-exact scheme features the lowest cutoff wavenumber $\omega_{j,c}\approx\pi/4$, followed by the 2-exact scheme with $\omega_{j,c}\approx\pi/2$ and the 1-exact scheme with $\omega_{j,c}\approx 11/20\,\pi$. At first glance, the 1-exact scheme thus appears to have superior dispersive properties. However, a more detailed analysis shows that the 2-exact scheme features the lowest errors up to wavenumbers of $\omega_j\approx 3\pi/8$. As can be seen in Figure 3b, numerical dissipation can be controlled precisely at high wavenumbers by adjusting the upwind bias θ. However, it may be that the dissipation of the physical viscosity ν is sufficient for damping oscillations and thus no additional numerical dissipation is required. This also applies to the case when local eddy viscosity ν_r is introduced by the SGS model. Thus, we aim to find a relation between the upwind bias and the local flow state such that the numerical dissipation can be minimized in regions of sufficient physical dissipation. In the following section, this relation is derived using a Von Neumann analysis for the linear advection-diffusion equation (39).

3.2. Von Neumann stability analysis

In the following, we will introduce a methodology in order to control the upwind bias factor in a fully adaptive way, such that a stable solution is maintained and the required numerical dissipation is minimized.

A Crank-Nicolson approach is used for the temporal discretization of Equation (39):

$$\frac{\overline{u}_\alpha^{n+1} - \overline{u}_\alpha^{n}}{\Delta t} + \frac{U}{h}\left(u\Big|_{x_{\alpha+1/2}}^{n+1/2} - u\Big|_{x_{\alpha-1/2}}^{n+1/2} \right) - \frac{\nu}{h}\left(\frac{\partial u}{\partial x}\Big|_{x_{\alpha+1/2}}^{n+1/2} - \frac{\partial u}{\partial x}\Big|_{x_{\alpha-1/2}}^{n+1/2} \right) = 0. \tag{49}$$

The Crank-Nicolson fluxes are evaluated following a deferred correction approach [3, 55], such that only the volume-averages of face-neighboring elements are treated implicitly. Derivatives, which are used to increase the accuracy, are treated explicitly for the face-value reconstruction. This approach results in significant savings of computation time compared to a fully implicit treatment, since the derivatives are not updated in every sub-iteration of the implicit solver [17]. Using the conversion from point values to volume-averages (42), the convective fluxes can be calculated according to

$$\begin{aligned} u\Big|_{x_{\alpha+1/2}}^{n+1/2} \approx{}& \frac{1}{2}\left(a_U \overline{u}_\alpha^{n+1} + a_D \overline{u}_{\alpha+1}^{n+1}\right) \\ &+ \frac{1}{2}\left(a_U \overline{u}_\alpha^{n} + a_D \overline{u}_{\alpha+1}^{n}\right) + \left(b_U \frac{\partial u}{\partial x}\Big|_{x_\alpha}^{n} + b_D \frac{\partial u}{\partial x}\Big|_{x_{\alpha+1}}^{n} \right) h + \left(c_U \frac{\partial^2 u}{\partial x^2}\Big|_{x_\alpha}^{n} + c_D \frac{\partial^2 u}{\partial x^2}\Big|_{x_{\alpha+1}}^{n} \right) h^2, \end{aligned} \tag{50}$$

and the diffusive fluxes through

$$\frac{\partial u}{\partial x}\Big|_{x_{\alpha+1/2}}^{n+1/2} = \frac{1}{2h}\left(\overline{u}_{\alpha+1}^{n+1} - \overline{u}_\alpha^{n+1}\right) + \frac{1}{2h}\left(\overline{u}_{\alpha+1}^{n} - \overline{u}_\alpha^{n}\right). \tag{51}$$

Note, that the notation for superscripts indicating the numerical error is omitted, in order to highlight the time step. With the definition of the CFL number $\sigma = U\Delta t/h$ and the local Reynolds number $Re_h = Uh/\nu$, it is possible to write the discretized Equation (49) in terms of

$$\overline{u}_\alpha^{n+1} + \sigma \sum_{m=-1}^{1} S_m^{(IMP)}\, \overline{u}_{\alpha+m}^{n+1} = \overline{u}_\alpha^{n} - \sigma \sum_{m=-3}^{3} S_m^{(EXP)}\, \overline{u}_{\alpha+m}^{n}. \tag{52}$$

The stencil coefficients $S_m^{(IMP)}$ and $S_m^{(EXP)}$ are given in Table 2 and 3.

To analyze the stability properties of the discretization scheme, the error $\overline{\varepsilon}_\alpha^n$ is introduced as the difference between the exact and the numerical solution at time t_n. This error also evolves according to Equation (49), similarly to the actual solution $\overline{u}_\alpha$. Thus, a stable scheme should prevent $\overline{\varepsilon}_\alpha^n$ from growing indefinitely when progressing in time. This criterion can be written in terms of the amplification function $G = \hat{\varepsilon}_j^{n+1} / \hat{\varepsilon}_j^{n}$, which is estimated by inserting a Fourier mode $\hat{\varepsilon}_j(t_n)\, \exp\left(I 2\pi j x_\alpha / L\right)$ for the numerical error into Equation (52). The amplification function then reads

$$G(\omega_j) = \frac{1 - \sigma \sum_{m=-3}^{3} S_m^{(EXP)} \exp\left(I\omega_j m\right)}{1 + \sigma \sum_{m=-1}^{1} S_m^{(IMP)} \exp\left(I\omega_j m\right)}. \tag{53}$$

For a stable discretization it is required that $|G(\omega_j)| \leq 1$ [56]. This expression is squared for the following derivation, in order to avoid the emergence of imaginary terms. The stability criterion can thus be stated as

$$\left| 1 - \sigma \sum_{m=-3}^{3} S_m^{(EXP)} \exp\left(I\omega_j m\right) \right|^2 - \left| 1 + \sigma \sum_{m=-1}^{1} S_m^{(IMP)} \exp\left(I\omega_j m\right) \right|^2 \leq 0. \tag{54}$$

The substitution $X_j := cos(\omega_j)$ is employed, in order to simplify the resulting trigonometric relations. After some algebraic transformations, the stability criterion $|G(\omega_j)| \leq 1$ can be expressed in terms of a quadratic

Table 4: Definition of the coefficients a_i, b_i and c_i in Equation (56) for the 1- and 2-exact discretization scheme.

Coefficient	$k=1$	$k=2$
$a_1(X_j)$	$(X_j-1)^2(X_j+1)/8$	$-(X_j-1)^2(X_j+1)(X_j^2-37)/288$
$a_2(X_j)$	$X_j(X_j-1)^2(X_j+1)/4$	$(X_j-1)^2(X_j+1)(X_j^3-12X_j^2+35X_j+12)/144$
$a_3(X_j)$	$-(X_j-1)^2(X_j+1)/4$	$(X_j-7)(X_j-1)^2(X_j+1)/24$
$a_4(X_j)$	$(X_j-1)^2(X_j+1)/8$	$-(X_j-1)^2(X_j+1)(X_j^2-37)/288$
$a_5(X_j)$	$-(X_j-1)^2(X_j+1)/4$	$(X_j-7)(X_j-1)^2(X_j+1)/24$
$b_1(X_j)$	$-(X_j-1)^2(X_j+1)/2$	$(X_j-7)(X_j-1)^2(X_j+1)/12$
$b_2(X_j)$	$-(X_j-1)^2(X_j+1)/2$	$(X_j-7)(X_j-1)^2(X_j+1)/12$
$b_3(X_j)$	$(X_j-1)^2(X_j+1)/4$	$-(X_j-1)^2(X_j+1)(X_j^2-37)/144$
$b_4(X_j)$	0	$-(X_j-1)^2(X_j+1)/12$
$b_5(X_j)$	$-(X_j-1)^2(X_j+1)/4$	$(X_j-1)^2(X_j+1)(X_j^2-37)/144$
$b_6(X_j)$	0	$(X_j-1)^2(X_j+1)/12$
$b_7(X_j)$	$-(X_j-1)(X_j+1)/2$	$(X_j-7)(X_j-1)(X_j+1)/12$
$b_8(X_j)$	$-(X_j-1)(X_j+1)/2$	$(X_j-7)(X_j-1)(X_j+1)/12$
$b_9(X_j)$	$2(X_j-1)$	$2(X_j-1)$
$c_1(X_j)$	$-(X_j-1)^2(X_j+1)/2$	$(X_j-7)(X_j-1)^2(X_j+1)/12$
$c_2(X_j)$	$(X_j-1)^2(X_j+1)/2$	$-(X_j-7)(X_j-1)^2(X_j+1)/12$
$c_3(X_j)$	$(X_j-1)^2(X_j+1)/8$	$-(X_j-1)^2(X_j+1)(X_j^2-37)/288$
$c_4(X_j)$	$-X_j(X_j-1)^2(X_j+1)/4$	$-(X_j-1)^2(X_j+1)(X_j^3-12X_j^2+35X_j+12)/144$
$c_5(X_j)$	$(X_j-1)^2(X_j+1)/4$	$-(X_j-5)(X_j-1)^2(X_j+1)/24$
$c_6(X_j)$	$(X_j-1)^2(X_j+1)/8$	$-(X_j-1)^2(X_j+1)(X_j^2-37)/288$
$c_7(X_j)$	$(X_j-1)^2(X_j+1)/4$	$-(X_j-5)(X_j-1)^2(X_j+1)/24$
$c_8(X_j)$	$4(X_j-1)$	$4(X_j-1)$
$c_9(X_j)$	$-(X_j-1)(X_j+1)/2$	$(X_j-7)(X_j-1)(X_j+1)/12$
$c_{10}(X_j)$	$(X_j-1)(X_j+1)/2$	$-(X_j-7)(X_j-1)(X_j+1)/12$

inequality:

$$A(X_j)\,\theta^2 + B(X_j)\,\theta + C(X_j) \leq 0, \tag{55}$$

where the coefficients $A, B, C = f(X_j; \sigma, Re_h, \psi_U, \psi_D)$ are defined as

$$\begin{aligned}
A(X_j) &= \sigma^2 \left[a_1(X_j)\,\psi_U^2 + a_2(X_j)\,\psi_U\psi_D + a_3(X_j)\,\psi_U + a_4(X_j)\,\psi_D^2 + a_5(X_j)\,\psi_D\right] \\
B(X_j) &= \sigma^2 \left[b_1(X_j)\frac{\psi_U}{Re_h} + b_2(X_j)\frac{\psi_D}{Re_h} + b_3(X_j)\psi_U^2 + b_4(X_j)\,\psi_U + b_5(X_j)\,\psi_D^2 + b_6(X_j)\,\psi_D\right] \\
&\quad + \sigma\left[b_7(X_j)\,\psi_U + b_8(X_j)\,\psi_D + b_9(X_j)\right] \\
C(X_j) &= \sigma^2 \left[c_1(X_j)\frac{\psi_U}{Re_h} + c_2(X_j)\frac{\psi_D}{Re_h} + c_3(X_j)\,\psi_U^2 + c_4(X_j)\,\psi_U\psi_D + c_5(X_j)\,\psi_U + c_6(X_j)\,\psi_D^2 + c_7(X_j)\,\psi_D\right] \\
&\quad + \sigma\left[c_8(X_j)\frac{1}{Re_h} + c_9(X_j)\,\psi_U + c_{10}(X_j)\,\psi_D\right].
\end{aligned} \tag{56}$$

The coefficients $a_i(X_j)$, $b_i(X_j)$ and $c_i(X_j)$ are given Table 4. For a given set of parameters σ, Re_h, ψ_U and ψ_D, a corresponding bias value θ must be found, which satisfies the inequality (55). This inherently leads to the amplification function satisfying the condition $|G(\omega_j)| \leq 1$ and thus guarantees the stability of the underlying linear transport equation (39). To express θ as a function of X_j, we consider the case where

Equation (55) equals zero and solve for its lower root:

$$\theta(X_j) = \frac{-B(X_j) - \sqrt{B^2(X_j) - 4A(X_j)C(X_j)}}{2A(X_j)}. \tag{57}$$

In order to ensure stability for all wavenumbers, the maximum value of θ in the interval $X_j \in [0, 1]$ must be chosen, which will be referred to as θ_{max}

$$\theta_{max}(\sigma, Re_h, \psi_U, \psi_D) = \max_{X_j} \theta(X_j, \sigma, Re_h, \psi_U, \psi_D). \tag{58}$$

In this work, the values for θ_{max} are calculated numerically over a preset parameter range of $\sigma \in [10^{-9}, 5]$ and $1/Re_h \in [10^{-9}, 0.5]$. The limiter values ψ_α are restricted to either values of zero or one. Thus, only the combinations $\psi_U \times \psi_D = \{(1, 0), (0, 1), (1, 1), (0, 0)\}$ are considered.

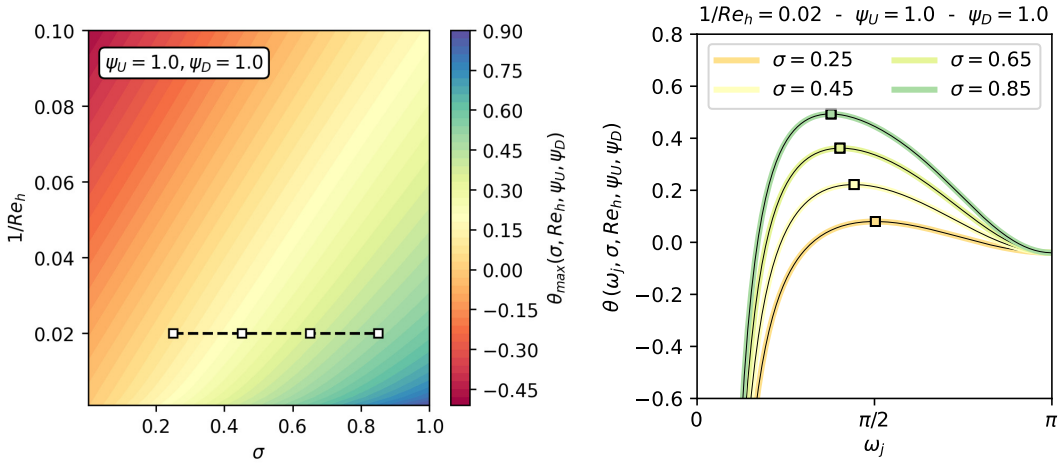
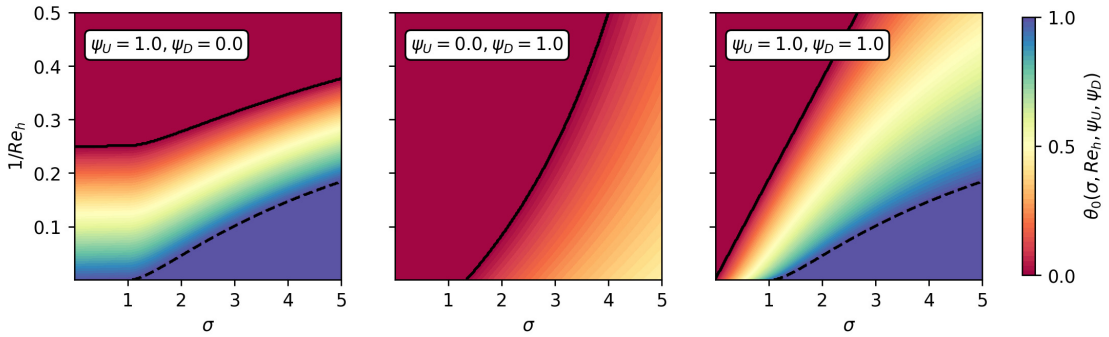


Figure 4: Maximum values for the upwind bias values θ_{max} of the 2-exact scheme that satisfy the stability equation (55) (left). The squared dots correspond to maximas, which are extracted from the respective θ-curves on the right.

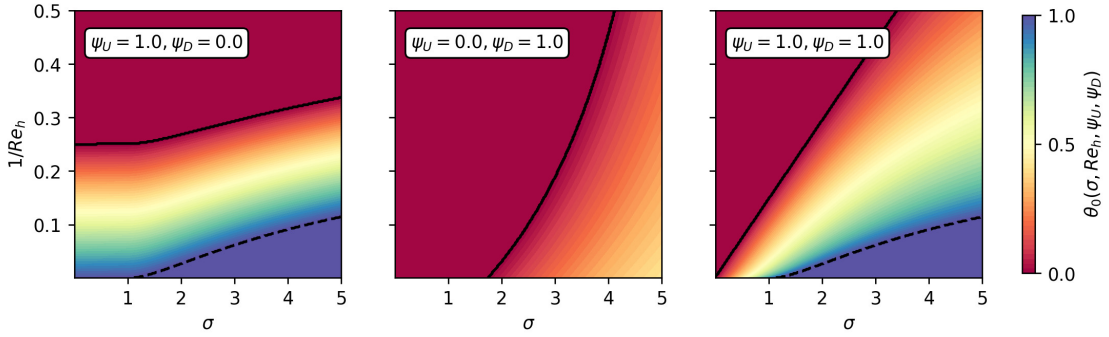
Figure 4 shows an example of how the values of θ_{max} are determined. Values for θ_{max} are shown on the left for $k = 2$, $\psi_U = 1$ and $\psi_D = 1$ and for varying values of σ and Re_h. The white squares depict the positions of the corresponding curves for θ along ω_j, which are shown on the right and which are used to estimate the values of θ_{max}. Obviously, θ_{max} exceeds the allowed bias value range of $\theta \in [0, 1]$ in certain regions. Thus, θ_{max} must be clipped to the appropriated range, where the resulting quantity will be referred to as θ_0:

$$\theta_0(\sigma, Re_h, \psi_U, \psi_D) = \max\left[0, \min\left(1, \theta_{max}\right)\right]. \tag{59}$$

Figure 5 shows θ_0 for the considered range of σ and Re_h for both the 1- and 2-exact schemes and for the different combinations of the limiter variables ψ_U and ψ_D. The red areas with $\theta_0 = 0$ refer to unconditionally stable regions, where a purely central discretization can be used. In contrast, the stability criterion can no longer be met in blue areas where θ_{max} is limited to a value of one. For $\psi_U = \psi_D = 0$ both schemes feature an unconditional stability and the upwind bias can thus be set to zero. From the figures it can be seen that the limiter variables ψ_U and ψ_D must be taken into account for the stability analysis. Interestingly, it can also be observed that the 2-exact scheme offers an enhanced stability range compared to the 1-exact scheme. This behaviour seems somehow counterintuitive, since higher order schemes are generally expected to be more prone to numerical instabilities.



(a) $k = 1$



(b) $k = 2$

Figure 5: Clipped upwind bias θ_0 for various arguments ψ_U and ψ_D and for both 1- and 2-exact schemes. For $\psi_U = \psi_D = 0$, both schemes are unconditionally stable and thus θ_0 can be set to zero. The solid lines indicate unconditionally stable regions (red), whereas dashed lines indicate unconditionally unstable regions (dark blue).

3.3. Approximation of θ_0 via a neural network approach

Based on the preceding investigations, the question arises on how the computed stability criteria can be applied in a flow solver. Since the calculation of θ_0 involves a considerable amount of computation time, a direct calculation on the fly would lead to significant performance losses. This issue can be overcome by tabulating the data from Equation (57) and (58) indirectly in a reduced order model. Due to the fact, that the underlying problem is defined analytically, a nonlinear regression approach might be a reasonable choice for the model reduction. Unfortunately this requires an explicit specification of a nonlinear fitting function, which turns out to be non-trivial to find for the given data. For example, the use of multivariate polynomials of higher degrees did not lead to a good trade-off between accuracy and compactness of the reduced order model. Other approaches, such as symbolic regression, also failed to provide satisfactory models for this problem. However, we found that the data can be approximated exceptionally well by the utilization of a simple multi-layer perceptron model, also referred to as neural network. The latter is essentially a multi-dimensional regression model for mapping the input values $\sigma, 1/Re_h, \psi_U$ and ψ_D to a respective upwind bias value θ_0. It consists of several layers of interconnected computing units, which are referred to as neurons. Every neuron calculates a weighted sum from its input values, which is either the network input data or the output values from neurons of previous layers. The weighted sum result is subsequently evaluated through a non-linear function, which is generally referred to as activation function. The weighted sum coefficients are the model unknowns which must be determined in an iterative process, referred to as training phase. For a detailed description on neural networks and the multi-layer perceptron model, we refer to the literature [57, 58, 59, 60]. As stated, the limiter values ψ_α are either zero or one, depending on the element Ω_α being adjacent to a Dirichlet boundary condition. It is thus more straightforward to create a

different network for every valid combination of ψ_U and ψ_D, which maps only the two input values σ and $1/Re_h$ to θ_0. Furthermore, for $\psi_U = \psi_D = 0$ no network is required, since this results in an unconditionally stable scheme.

The model employed in this work includes an input and an output layer, which are connected by a single intermediate layer, whose number of neurons N_n varies according to the exactness of the scheme and the chosen limiter values ψ_D and ψ_U. The input layer consists of two and the output layer of one single neuron. The entire network model forms a highly non-linear structure of nested functions, through which the upwind bias $\theta_{max}^{(NN)}$ can be calculated as follows:

$$\theta_{max}^{(NN)}(\sigma, Re_h) = b^{(2)} + \sum_{i=1}^{N_n} w_i^{(2)} g_{\text{soft}} \left(b_i^{(1)} + w_{i1}^{(1)} \sigma + w_{i2}^{(1)} (1/Re_h) \right). \tag{60}$$

Here, $w_{ij}^{(l)}$ is the weight coefficient of the connection between neuron i in layer $l-1$ and neuron j in layer l. Each layer also includes a bias weight $b_i^{(l)}$. The model emphasizes the non-linear softplus activation function

$$g_{\text{soft}}(x) = \frac{1}{1 + \widetilde{\exp}(x)} \quad \text{where} \quad \widetilde{\exp}(x) = \frac{x^2 + 6x + 12}{x^2 - 6x + 12}, \tag{61}$$

which is based on the $(2,2)$ Padé approximant of the exponential function. The latter is a reasonable approximation of the exponential function in the range $|x| < 1/2$. Preliminary numerical investigations revealed that this function offers an exceptional trade-off between accuracy and model complexity for the underlying data. During simulation, the values of $\theta_{max}^{(NN)}$ are calculated once every time step and are used for every transport equation. They are stored at every element interface $A_{\alpha\beta}$. Based on the limiter values of the adjacent elements of $A_{\alpha\beta}$, the appropriate network model is chosen and $\theta_{max}^{(NN)}$ is calculated from $\sigma = U\Delta t/h$ and $1/Re_h = \nu_{tot}/(U\,h)$. These quantities are estimated at every face from the local grid scale $h = \sqrt{\Delta x_{i,\,\beta\alpha} \Delta x_{i,\,\beta\alpha}}$, the local flow velocity $U = |\dot{m}_\Gamma|/(\rho_0 |A_{\alpha\beta}|)$ and the viscosity $\nu_{tot} = \nu + (\nu_{r,\alpha} + \nu_{r,\,\beta})/2$. The latter includes the local eddy viscosity $\nu_{r,\alpha}$ and $\nu_{r,\,\beta}$ of the residual motions of the adjacent cells Ω_α and Ω_β. The output from a trained multi-layer perceptron model is finally clipped according to Equation (59). It was found that this drastically improves the quality of results in comparison to the case where the network is trained directly with θ_0 values.

For the network training, defined input values for σ and $1/Re_h$ are presented to the model and the corresponding predicted network output $\theta_{max}^{(NN)}$ is compared with known target values θ_{max}. The latter are estimated numerically via Equation (58). The network weights are adjusted iteratively to minimize an error functional E that resembles the sum of squared errors for all K samples in a given training data set:

$$E = \frac{1}{2} \sum_{n=1}^{K} \left(\theta_{max,n} - \theta_{max,n}^{(NN)} \right)^2. \tag{62}$$

In the scope of neural network models, the minimization of E is typically approached with stochastic gradient descent (SGD) algorithms, which are especially useful for the optimization of high-dimensional problems [61]. But since the regression problem in this work features only a small input dimension, the model coefficients can be determined very efficiently via second order optimization methods [60]. For this reason, the Levenberg-Marquardt (LM) algorithm [62, 63] is used, which is suitable for the training of small- to medium-sized neural networks [64, 65]. In order to realize a cheap approximation of the upwind bias it is desirable that the models feature as few computing units as possible. This is achieved by performing several training runs with different numbers of neurons N_n and corresponding training data sets. The trained models are then tested against a validation data set, where the overall model quality is assessed with the root mean square error normalized by the standard deviation (NRMSE) and the maximum absolute error

$E_{abs}^{(max)}$

$$NRMSE = \frac{\sqrt{\frac{1}{K}\sum_n \left(\theta_{0,n} - \theta_{0,n}^{(NN)}\right)^2}}{s(\theta_0)} \qquad \text{and} \qquad E_{abs}^{(max)} = \max_n \left|\theta_{0,n} - \theta_{0,n}^{(NN)}\right|, \tag{63}$$

where $s(\cdot)$ is the standard deviation. Finally, the most compact model is selected which can approximate the validation data set with a prescribed error tolerance. However, there is one major drawback in the utilization of the LM training algorithm, which stems from the fact that the method is based on the sum of squared errors. The latter optimizes the network weights in such a way, that the upwind bias may be slightly over- or underpredicted due to approximation errors. While too high bias values are only associated with increased numerical dissipation, too low values may violate the derived stability equation. Since the underlying linear advection-diffusion equation is already a simplified model for the stability analysis of the Navier-Stokes equations, this violation could be considered negligible. In fact, preliminary numerical investigations showed that it does not significantly affect the stability of the schemes at first sight. On the other hand, a detailed analysis revealed certain numerical artifacts in the turbulent spectra of the test cases considered in this work. This behaviour is attributed to an insufficient damping of parasitic modes due to an underpredicted upwind bias. The issue can be addressed by influencing the network training with a slight adjustment of the error functional E. It requires a second term in the calculation of E, which is defined as

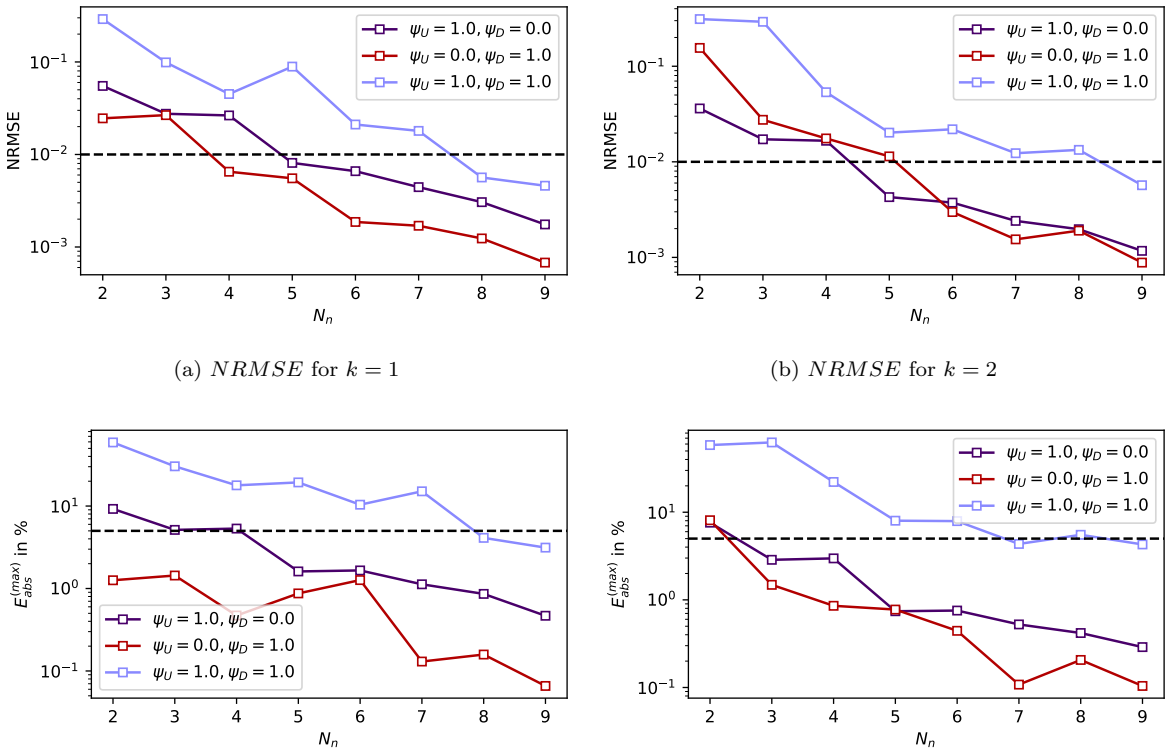
$$E = \frac{1}{2}\sum_{n=1}^{K}\left[\left(\theta_{max,n} - \theta_{max,n}^{(NN)}\right)^2 + k_p\left|\theta_{max,n} - \theta_{\mathrm{max},n}^{(NN)}\right| H\left(\theta_{max,n} - \theta_{\mathrm{max},n}^{(NN)}\right)\right]. \tag{64}$$

This additional term involves the Heaviside function $H(x)$ and a preset constant $k_p = 1{,}000$. It acts as a threshold function which is activated when predicted bias values $\theta_{\mathrm{max},n}^{(NN)}$ are lower than their corresponding target values $\theta_{max,n}$. In this way, the network weights are inherently trained to avoid an underprediction of $\theta_{max}^{(NN)}$. Unfortunately, this additional term is not easily compatible with the LM optimization method, since the latter is based on summed squared error functions [66]. In contrast, it could be used without much effort using a SGD optimization algorithm. But compared to LM, the latter method features significantly worse convergence properties for the described regression problem. We thus decided to use both methods successively in order to benefit from the respective advantages. The network training can thus be separated into the following three steps:

1. Training of several models with different numbers of neurons using the LM method and error function (62).
2. Validation of the models and selection of the most compact one, which meets the specified error tolerances.
3. Optimization of the winner model weights with a SGD method and error function (64), in order to reduce the underprediction of the upwind bias value.

3.4. Neural network training results

The results of training steps one and two are shown in Figure 6, where the model errors are given as a function of the number of network neurons. The training is realized with the LM optimization method provided by the open-source Python library Scipy [67]. Training samples are generated equidistantly over the range of σ and $1/Re_h$ and the number of samples is chosen as a function of the number of network weight coefficients. The latter significantly improves the training results in comparison to the case were the sample size is fixed for all numbers of neurons. The ratio of training samples to network coefficients is chosen to a value of 70, which has been estimated in preliminary numerical experiments. Thus, the total number of training samples varies between 630 and 2,590. The model errors are calculated with a separate validation data set, which includes 62,500 samples. The dashed lines show the error thresholds $NRMSE \leq 0.01$ and $E_{abs}^{(max)} \leq 5\%$ that are used to select the winner model. Table 5 shows the size of the resulting winner models



(a) $NRMSE$ for $k = 1$

(b) $NRMSE$ for $k = 2$

(c) $E_{abs}^{(max)}$ for $k = 1$

(d) $E_{abs}^{(max)}$ for $k = 2$

Figure 6: Calculated errors of the network model as a function of the number of neurons for the 1- and 2-exact schemes with different limiter values ψ_U and ψ_D after training the models with the LM algorithm. The dashed lines represent the error thresholds that are used to select the appropriate network models. The employed validation data set for calculating the errors comprises 62,500 samples, respectively.

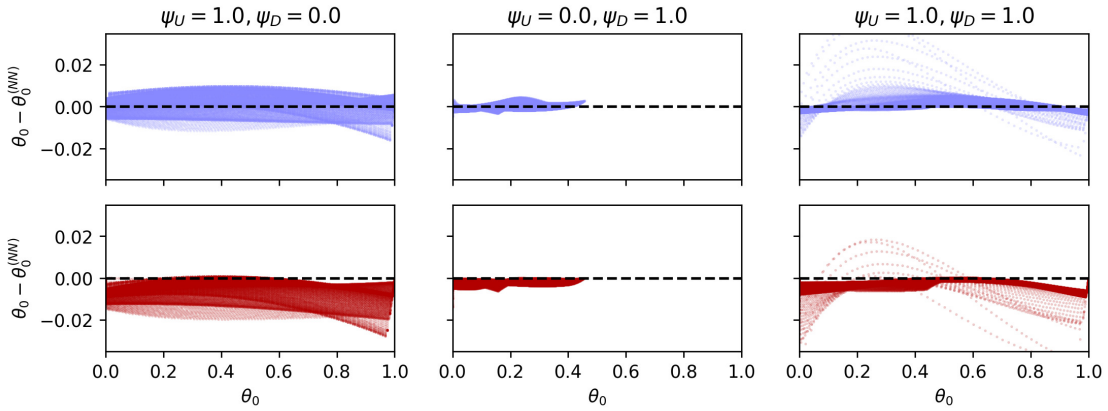
Table 5: Number of neurons of the winner models and corresponding validation errors before (LM) and after (LM+SGD) the utilization of the SGD training step.

Scheme	ψ_U	ψ_D	N_n	LM		LM+SGD	
				$E_{abs}^{(max)}$	$NRMSE$	$E_{abs}^{(max)}$	$NRMSE$
$k = 1$	1.0	0.0	5	1.61%	0.0081	2.77%	0.0134
$k = 1$	0.0	1.0	4	0.47%	0.0065	1.39%	0.0133
$k = 1$	1.0	1.0	8	4.11%	0.0056	4.08%	0.0077
$k = 2$	1.0	0.0	5	0.74%	0.0043	1.28%	0.0096
$k = 2$	0.0	1.0	6	0.44%	0.0030	0.94%	0.0077
$k = 2$	1.0	1.0	9	4.29%	0.0057	5.18%	0.0102

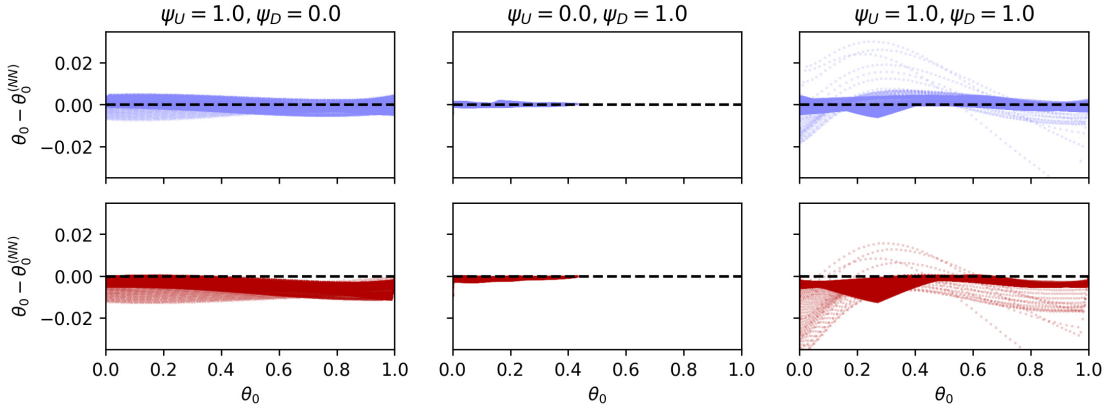
and their calculated errors. It would be possible to obtain even cheaper models by allowing larger error thresholds. Nevertheless, our primary focus is on whether the linear advection-diffusion equation is generally suitable for the determination of the upwind bias. Accordingly, we want to approximate the underlying data with errors as small as possible. In future work, the influence of errors in the reduced-order model might be further investigated with respect to the associated computation time savings.

In the third step of the model training, the weights of the winner model from the LM optimization

are adjusted with an extended SGD optimization method. For this purpose, the Adam algorithm [68] of the PyTorch framework [69] is utilized, for which the underlying error functional E from Equation (64) is applied. The same training data sets are used as for the preceding LM optimization. The effect of this additional training step is shown in Figure 7 by means of the difference between the predicted clipped upwind bias values $\theta_0^{(NN)}$ and their corresponding target values θ_0. The top figures in blue show the output of models, which are trained only with the LM algorithm, whereas the bottom figures in red show the model outputs after the applied SGD step. Apart from a few outliers, the underprediction of θ_0 is clearly reduced, in particular for the unlimited models with $\psi_U = 1$ and $\psi_D = 1$. As can be seen in Table 5, the additional SGD training step causes a slight increase in the validation errors of the models. It turns out that these errors can be further reduced by using more neurons while maintaining the underprediction properties. Nevertheless, the presented models were considered to offer a good compromise between compactness and accuracy. For this reason, it is also accepted that some LM+SGD models slightly exceed the specified error thresholds. Figure 8 further highlights the errors that are introduced by the final neural network models with respect to the given input parameters σ and $1/Re_h$. Negative (blue) regions correspond to an overprediction of θ_0 by the model and thus no violation of the stability criterion. In contrast, positive values (red) should denote regions where the predicted upwind bias is too low. Again, it can be observed that all trained models are able to approximate the upwind bias with sufficient accuracy over the considered value range, though without substantially underestimating it. The final model coefficients are given in Appendix A.



(a) $k = 1$



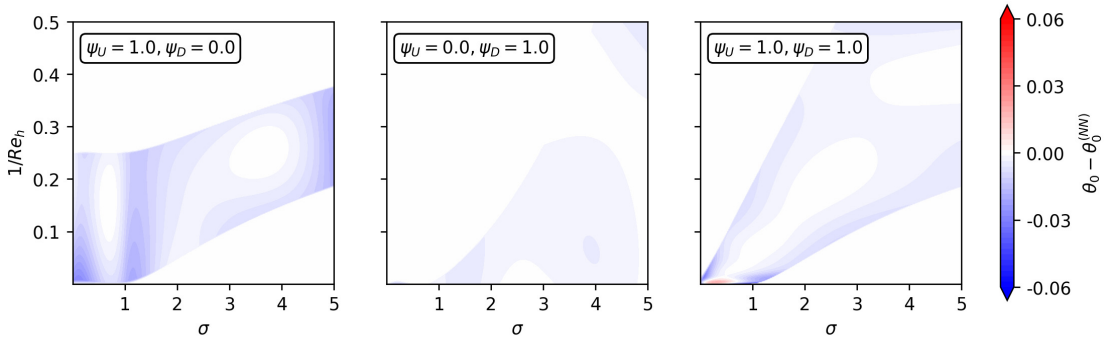
(b) $k = 2$

Figure 7: Comparison of predicted clipped upwind bias $\theta_0^{(NN)}$ by the multi-layer perceptron models and their corresponding target values θ_0. The top figures (blue) refer to the models trained with the LM algorithm, whereas the bottom figures (red) refer to the combined LM+SGD training approach. The dashed black lines refers to an ideal fit between the neural network model and the target values.

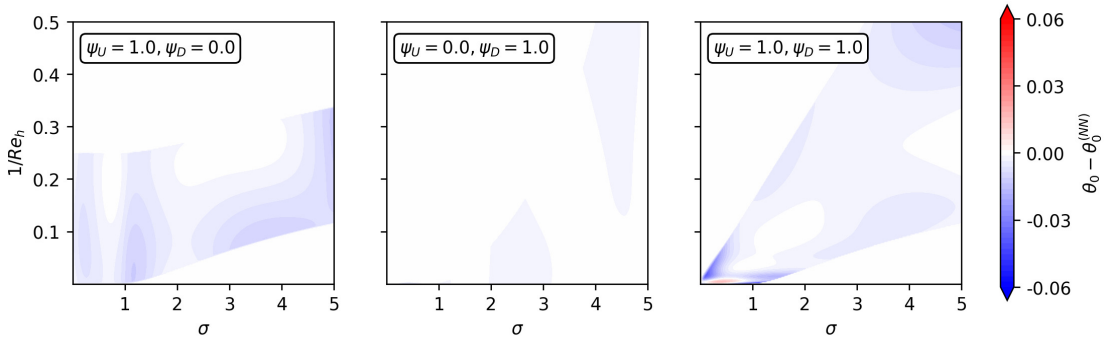
4. Numerical Benchmarks

This section is devoted to the validation of the accuracy and the performance properties of the proposed 1- and 2-exact multiple-correction schemes. The schemes have been implemented into ThetaCOM (**T**urbulent **HE**at release extension for **TA**U in its **COM**bustion version), which is developed at the DLR Institute of Combustion Technology. ThetaCOM has been used extensively for the simulation of incompressible and variable density flow problems, for example, detailed large-eddy simulations of complex combustion applications [18, 19, 20, 21, 22, 23, 24, 25]. The solver uses a collocated, edge-based representation with median-dual cells that are constructed from hybrid primary grid elements, such as tetrahedrals, hexahedrals, pyramids or prisms. The Poisson equation for the pressure is solved using the preconditioned flexible generalized minimal residual method (GMRES). As preconditioning a single multigrid V cycle is used on three grid levels. The k-exact multiple-correction is only employed on the finest grid level. All other transport equations are solved using a biconjugate gradient stabilized method (BiCGSTAB) with Jacobi preconditioning. All linear equations are formulated in a matrix-free approach, which reduces the memory requirements significantly by avoiding additional storage for sparse matrix data structures.

The 1- and 2-exact multiple-correction is tested against a conventional discretization method already



(a) $k = 1$



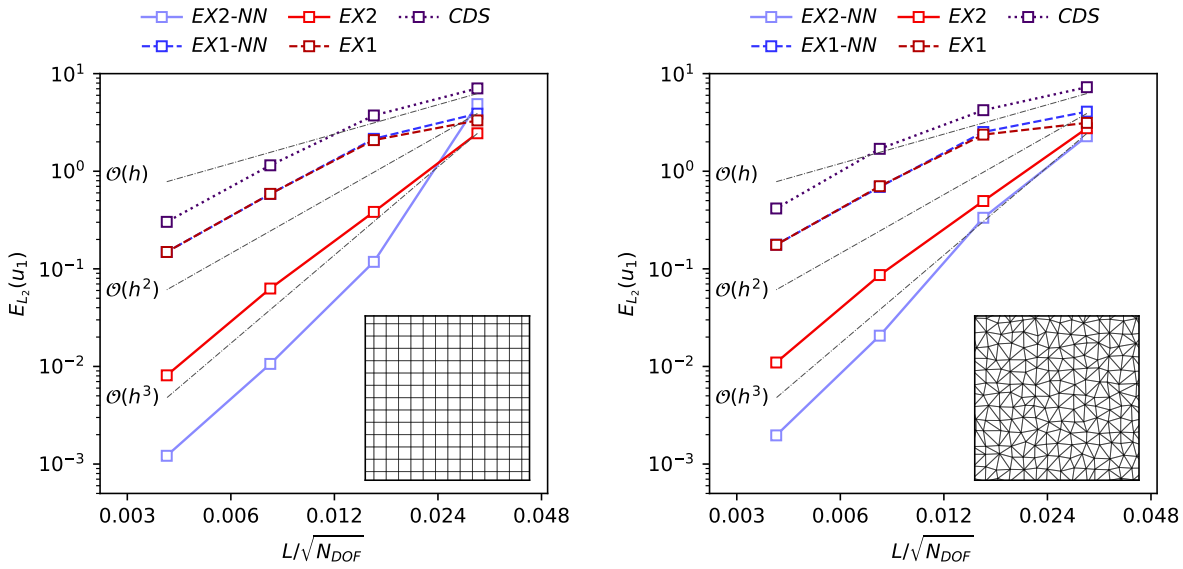
(b) $k = 2$

Figure 8: Error between predicted clipped upwind bias $\theta_0^{(NN)}$ and corresponding target values θ_0 for varying input parameters σ and $1/Re_h$.

implemented in ThetaCOM, where convective fluxes are discretized with a central differencing scheme (referred to as CDS). Diffusive fluxes are discretized in this scheme through equation (29) without using the terms contributing to the k-exact reconstruction approach. For all test cases, a Crank-Nicolson scheme is used for the temporal discretization, as proposed for the predictor step of the momentum equations (20). All implicit fluxes are calculated with a deferred-correction procedure [3, 55], where derivatives used for the reconstruction are only updated once at the end of every time step. The conventional discretization methods employ an incremental variant of the projection method as described by Knopp et al. [70], where the interpolation scheme by Rhie and Chow [71] is used to avoid spurious pressure oscillations.

4.1. Convection of a vortex on a periodic grid

The first benchmark is used to show that the presented adaptive upwind bias approach preserves the spatial accuracy of the discretized convective operator. The test case involves the convection of a two-dimensional vortex in a uniform, incompressible and inviscid flow field. This involves the solution to Equations (2) and (3) in the absence of viscosity ν and without consideration of the turbulence model. The



(a) Cartesian grid

(b) Distorted grid

Figure 9: Grid convergence study for the two-dimensional vortex convection benchmark.

velocity field and the pressure are given by

$$u_1(\mathbf{x}, t=0) = u_\infty - \frac{\Gamma}{\sigma^2}(x_2 - x_{2,c}) \exp\left[-\frac{(x_1 - x_{1,c})^2 + (x_2 - x_{2,c})^2}{2\sigma^2}\right], \tag{65a}$$

$$u_2(\mathbf{x}, t=0) = \frac{\Gamma}{\sigma^2}(x_1 - x_{1,c}) \exp\left[-\frac{(x_1 - x_{1,c})^2 + (x_2 - x_{2,c})^2}{2\sigma^2}\right], \tag{65b}$$

$$p(\mathbf{x}, t=0) = p_\infty - \frac{1}{2}\rho\left(\frac{\Gamma}{\sigma}\right)^2 \exp\left[-\frac{(x_1 - x_{1,c})^2 + (x_2 - x_{2,c})^2}{\sigma^2}\right], \tag{65c}$$

with the free-stream velocity in x_1 direction $u_\infty = 20\,\mathrm{m/s}$, the ambient pressure $p_\infty = 100,000\,\mathrm{Pa}$ and the vortex circulation $\Gamma = 0.4\,\mathrm{m}^2/\mathrm{s}$. The vortex radius is set to $\sigma = 0.005\,\mathrm{m}$ and it is initially placed at the center location $\mathbf{x}_c = (0.05\,\mathrm{m}, 0.05\,\mathrm{m})$. A grid convergence study is conducted in a two-dimensional, periodic domain $\mathbf{x} \in [0, L]^2$ with $L = 0.1\,\mathrm{m}$, where the vortex is transported for a distance of $\delta x = 0.1\,\mathrm{m}$. The benchmark is performed on two grid types: A cartesian grid that consists of quadrilateral primary grid elements and a distorted triangular grid. The convergence of the numerical schemes is examined by successively increasing the number of elements in the domain from $N = 32^2$ to $N = 256^2$. A CFL-number of 0.1 is set for all simulations, so that the influence of temporal discretization errors can be neglected. Five different discretization approaches are examined in terms of their accuracy. First, the central differencing scheme (CDS), which refers to a 0-exact reconstruction approach where the upwind bias θ is set to zero. Next, a 1- and 2-exact reconstruction scheme with a constant upwind bias $\theta = 0.2$ (referred to as $EX1$ and $EX2$), and finally, a 1- and 2-exact scheme which utilizes the respective neural-network models to predict the upwind bias (referred to as $EX1$-NN and $EX2$-NN).

Figure 9 shows the grid convergence for the various schemes on both grid types. The numerical accuracy

is assessed by means of the L_2-norm

$$E_{L_2}(\phi) = \left[\frac{\sum_{\alpha=1}^{N} \left(\overline{\phi}_\alpha - \overline{\phi}_\alpha^{\text{exact}} \right)^2 |\Omega_\alpha|}{\sum_{\alpha=1}^{N} |\Omega_\alpha|} \right]^{1/2} \tag{66}$$

with the volume-averaged exact solution $\overline{\phi}_\alpha^{\text{exact}}$ and the total number of elements N. For all schemes, the error norm drops with the corresponding orders of accuracy. The $EX2$-NN method exhibits by far the smallest errors on both grid types. For the 2-exact reconstruction approach, the utilization of the adaptive upwind bias significantly reduces the numerical error in contrast to the case where θ is held fixed. On the two finest levels of the cartesian grid, the error can be greatly reduced by an order of magnitude. Interestingly, for the 1-exact reconstruction, there is almost no change with a varying upwind bias. The improved results for the 2-exact reconstruction are achieved with only minor additional computational cost. On the finest grid levels, the $EX2$-NN scheme requires roughly 3% more time for a single iteration, compared to the $EX2$ scheme. For coarser grid levels the overhead is even lower. Concerning the computation time of the different schemes, the 2-exact schemes need about twice the time to calculate a single iteration on the same computational grid compared to the CDS scheme. However, the $EX2$-NN method can already undercut the lowest error of the CDS method with the second coarsest grid. In this way, the computation time for obtaining a solution with at least the same accuracy can be reduced by roughly 75% on the cartesian grid. The 1-exact schemes require about 45% more computation time for a single iteration compared to the CDS scheme. Compared to the conventional CDS scheme, the error is not reduced as significantly as for the 2-exact schemes.

4.2. Turbulent pipe flow

For this benchmark the turbulent flow through a pipe of radius R is investigated by means of Large-Eddy simulations. The setup features a Reynolds number $Re_\tau = u_\tau R/\nu = 180$, which is based on the pipe radius and the friction velocity u_τ. Due to the low Reynolds number, there may be a re-laminarization of the flow if, for example, a too coarse grid or a low accuracy method is used. [72, 73]. Hence, the test case is ideally suited to investigate the influence of the adaptive upwind bias approach. The length of the simulated pipe is set to $10\,R$. Periodic boundary conditions are imposed on the pipe openings and no-slip wall boundary conditions are used for the pipe walls. A constant axial pressure gradient is applied, in order to compensate for friction losses and to ensure a constant bulk velocity. A more detailed description of the setup can be found in the work of Eggels et al. [74] or Fukagata and Kasagi [75]. Direct Numerical Simulation (DNS) results of the latter are used to evaluate the outcomes of the present simulation. Two regular meshes of different resolutions are employed, where the discretization of the pipe cross-section with the coarse grid is shown in Figure 10 . Details concerning the grid spacings and respective numbers of primary grid nodes are given in in Table 6. Various simulations are carried out for the different discretization schemes, where sufficiently small time steps have been employed, in order to reduce the influence of the temporal discretization. All simulations are initialized with a velocity field of a fully developed turbulent pipe flow, which was calculated in a preliminary simulation on the finest grid. An initialization period of 30 flow through times is utilized, in order to remove effects of the initial solution. Subsequently, temporal flow field statistics are gathered for a period of 250 flow through times. The calculated statistics include velocity mean values $\langle u_i \rangle$, as well as mean velocity fluctuations $\langle u_i' u_j' \rangle$.

Figure 11 shows the instantaneous upwind bias $\theta_0^{(NN)}$, that is calculated for the 2-exact scheme on the finest grid. Due to higher flow velocities, a higher numerical dissipation is introduced in the core region of the pipe, in order to maintain a stable solution. In contrast, the numerical dissipation is reduced to a minimum in the vicinity of walls, where the scheme features inherently a higher stability due to increased viscous forces. The effect of the adaptive approach towards a fixed upwind bias in the entire domain can be observed by means of the radial flow profiles in Figure 12. The top left figure shows the normalized axial mean velocity $u^+ = \langle u_{ax} \rangle / u_\tau$ against the radial coordinate in wall units $y^+ = y u_\tau / \nu$ calculated for the coarse grid. For both 1- and 2-exact schemes, the adaptive upwind approach leads to an improved agreement

Table 6: Mesh properties for the turbulent pipe flow benchmark. N_i refers to the number of primary grid nodes and Δx_i^+ to the spacing of the first primary grid element layers adjacent to the wall, given in wall-units.

	N_{rad}	N_{tan}	N_{ax}	Δx^+_{rad}	Δx^+_{tan}	Δx^+_{ax}
Coarse grid	21	48	64	2.45	23.54	28.13
Fine grid	33	76	96	1.56	14.88	18.75
Reference DNS [75]	96	128	256	0.46	8.54	6.79

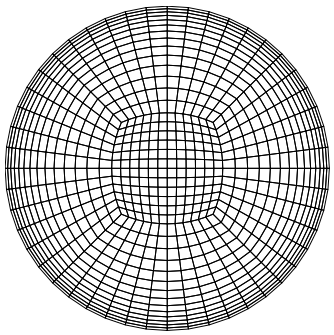


Figure 10: Cross-section of the coarse grid for the pipe flow benchmark.

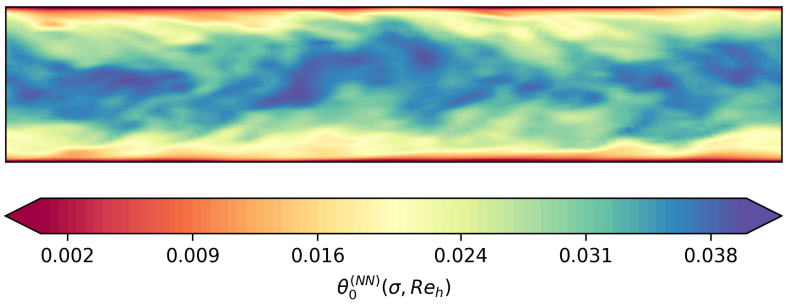


Figure 11: Instantaneous upwind bias $\theta_0^{(NN)}$ calculated with the 2-exact scheme on the fine grid.

with the DNS data by Fukagata and Kasagi [75], especially in the log-law region ($y^+ > 30$) of the flow. In contrast, the CDS scheme shows the largest deviations in both the log-law region and in the buffer layer ($5 < y^+ < 30$). Both the $EX2$-NN and $EX1$-NN schemes show similar good agreement with the DNS data. Similar results are obtained for the normalized axial and tangential velocity fluctuations in the middle and lower part of Figure 12a. Despite the low resolution properties of the coarse grid, both the $EX1$-NN and $EX2$-NN schemes lead to a remarkably good prediction of the axial velocity fluctuation maximum in the buffer layer. In contrast, the tangential fluctuations have somewhat larger deviations from the DNS data. The $EX2$ and $EX1$ schemes with a constant upwind bias also show significantly better agreement with the DNS data when compared to the conventional CDS scheme. However, the constant value of θ must be determined prior to the simulation. The choice of values too high leads to larger deviations, whereas values too low can lead to an unstable solution. Hence, the choice of the parameter must be considered carefully.This procedure is not necessary for the presented adaptive approach. Figure 12b shows the results, which have been obtained on the fine grid. Clearly, all schemes exhibit a lower error with respect to the DNS data in comparison to the coarse grid data. Both adaptive k-exact schemes capture the mean velocity and the axial fluctuations exceptionally well, especially since these results were generated with only about 8% of the primary grid nodes that were used in the reference work of Fukagata and Kasagi [75]. However, somewhat larger deviations are obtained for the tangential fluctuations for $y^+ > 20$. It should be emphasized that the profiles on the coarse grid calculated with any of the four k-exact schemes agree significantly better with the DNS data than the profiles from the conventional CDS scheme on the fine grid. In this way, it can also be clearly demonstrated how computation time can be saved by using the higher-order schemes. For example, a single iteration with the $EX2$-NN scheme on the coarse grid requires about 45% less computation time than a single iteration with the CDS scheme on the fine grid. Additionally, less iterations are required on the coarse grid to obtain a statistically stationary solution, due to the larger time step. If this is taken into account, the $EX2$-NN scheme can save roughly 55% of the wall clock time compared to the CDS scheme while also predicting the solution more accurately. Under this consideration, the wall clock time can be even reduced by 70% when using the $EX1$-NN scheme. Compared to the previous benchmark, the overhead for the adaptive calculation of the upwind bias is even less significant. The $EX2$-NN scheme

requires roughly 2% more time for a single iteration compared to the $EX2$ scheme and for the $EX1$-NN scheme the iteration time increases by 1%. Figure 13 shows one-dimensional spectra for the axial velocity $E_{11}(f)$, which are defined by [37]

$$E_{11}(f) = \frac{1}{\pi} \int_{-\infty}^{\infty} \langle u_1(t) u_1(t+\tau) \rangle \exp\left(-2\pi I f \tau\right) \mathrm{d}\tau. \tag{67}$$

The velocity spectra are calculated from temporal signals that are sampled at the pipe center at the axial location $x_1 = 5\,R$. The values are normalized with the estimated mean axial velocity on the pipe center line U_c. Figure 13a shows the spectra obtained on the coarse grid. All discretization schemes feature a distinct $-5/3$ power law behaviour, representing the inertial subrange where a constant transfer rate of energy to smaller scales is present [37]. It should be noted, that this power law behavior is affected when too small upwind bias values are calculated by the network models. This effect is counteracted by applying the SGD training step, which is described in the previous section. For the CDS scheme, the inertial subrange extends up to a cutoff frequency of $f\,R/U_c \approx 0.8$, whereas the k-exact schemes resolve this region up to $f\,R/U_c \approx 1.5$. Interestingly, the ratio of these cutoff frequencies is approximately two, which is consistent with the ratios of the respective cutoff wavenumbers $\omega_{j,c}$ in Figure 3. This suggests that this difference can be attributed to the lower dispersive error of the 1- and 2-exact schemes. The spectra do not show a clear difference between the 1- and 2-exact discretziation. However, the use of the adaptive upwind bias slightly improves the resolution of the inertial subrange in the low frequency range for both schemes. Figure 13b shows similar spectra, which are obtained on the fine grid. The cutoff frequencies of all discretization methods are shifted towards lower values due to the smaller filter width of the fine mesh. Nevertheless, similar cutoff frequency ratios between the CDS scheme and the k-exact schemes can be observed on the fine grid.

4.3. Incompressible round jet at $Re = 10,000$

The purpose of this benchmark is to demonstrate the applicability of the presented methodology for LES with a high-order accurate spatial discretization on truly unstructured grids. We consider a three-dimensional turbulent jet at Reynolds number $Re = \Delta U\, D/\nu = 10,000$, which is based on the jet diameter D, as well as the difference $\Delta U = (U_{\text{jet}} - U_{\text{coflow}})$ between the jet center velocity U_{jet} and the coflow velocity U_{coflow}. Detailed experimental measurements for this test case are available due to the work of Wygnanski and Fiedler [76] or Panchapakesan and Lumley [77]. Furthermore, an extensive review on the general theory of turbulent jets can be found in the work of Lipari and Standsby [78]. A round jet at this Reynolds number has also been studied in numerous other papers by means of LES [79, 80, 81, 82, 83]. The fluid is ejected from a nozzle into an external coflowing ambient. In the near field, the jet undergoes a transition from flat square to a round profile. This transition takes place in the range $0 \leq x_1/D \leq 25$, which is generally referred to as the initial development region [37]. Further downstream at $x/D > 30$, the self-similar region emerges due to an equilibrium between the turbulent kinetic energy generated from the main flow and the viscous dissipation at the smallest scales [82, 84]. In this region the flow profiles collapse onto a single curve, when plotted against the cross-stream similarity variable η

$$\eta = \frac{r}{x_1 - x_t} \qquad \text{with} \qquad r = \sqrt{x_2^2 + x_3^2}. \tag{68}$$

The point x_t refers to the virtual origin of the self-similar region [37]. In addition to the mean flow variables, the self-similarity also applies for the Reynolds stresses [37, 82]. Figure 14 shows the domain extension that is considered for this test case, as well as an exemplary unstructured grid that is utilized. In total, four grids are considered, which differ in the number of their primary grid nodes. The self-similarity of the flow is taken into account for the estimation of the grid element size h, such that elements become larger as the distance to the inlet and the jet centerline increases. For this purpose, the longitudinal and lateral integral length scales L_{11} and L_{22} in the self-similar regions are based on the relations $L_{11} \approx 0.038 x_1 + 0.035 r$ and $L_{22} \approx 0.016 x_1 + 0.015 r$, which have been approximated from experimental data by Wygnanski and Fiedler [76]. In the initial development region of the flow the element size is chosen in relation to the jet diameter D. Table 7 shows the corresponding ratios of D/h in the initial development region and the ratios

Table 7: Properties of the utilized meshes for the round jet benchmark.

Mesh	Primary grid nodes	Tetrahedra	Initial development region	Self-similar region	
			$D\ /\ h$	L_{11}/h	L_{22}/h
$150k$	$147,811$	$874,125$	3.5	1.8	0.8
$260k$	$257,744$	$1,528,089$	4.4	2.3	1.0
$450k$	$453,878$	$2,696,141$	5.5	2.8	1.2
$800k$	$816,944$	$4,858,854$	6.9	3.5	1.5

of L_{11}/h and L_{22}/h in the self-similar region that are prescribed for the four grids employed. The inlet velocity profile is imposed by the following equation [80]

$$u_{\text{inlet}}(r) = U_{\text{coflow}} + \frac{U_{\text{jet}}}{2}\left[1 + \tanh\left(\frac{D-2r}{4\delta_m}\right)\right], \tag{69}$$

with a momentum thickness $\delta_m = D/20$. A velocity ratio of $\Delta U/(U_{\text{jet}} + U_{\text{coflow}}) = 0.99$ between jet and coflow is applied. For every simulation, the flow field is initialized for 9 flow through times and then subsequently 174 flow through times are calculated, in order to obtain temporal statistics of the field variables. The time steps are chosen, such that a maximum CFL number of 0.5 is obtained for each grid. Numerical experiments have shown that this value is small enough to keep temporal discretization errors negligible.

Figure 15 shows the iso-contours of the vorticity magnitude $|\boldsymbol{\omega}|$ scaled by $(x_1/\Delta U)$, which refer to a value of two. The simulation results calculated with the $EX2\text{-}NN$ scheme on the finest grid are presented in Figure 15a. The initial development region can be observed close to the inlet, where the jet exhibits a coherent structure. Further downstream, the self-similar region clearly emerges, which can be seen by means of the strongly folded turbulent structures. Both the $EX1\text{-}NN$ or the conventional CDS scheme feature similar iso-contours of the scaled vorticity magnitude. This indicates that these schemes also exhibit sufficiently low numerical dissipation so that turbulent structures can emerge. The influence of an excessive amount of numerical dissipation, on the other hand, can clearly be observed in Figure 15b. This flow field is obtained on the same mesh, but with the $EX2$ scheme and a fixed upwind bias value $\theta = 0.2$. Substantially less turbulent structures are generated in the entire flow field and due to the high numerical dissipation, turbulent fluctuations in the initial development layer are overly damped. This leads to the fact that turbulence is generated much further downstream and only in very coarse and coherent structures. A similar behaviour can be observed for the $EX1$ scheme with a constant upwind bias value of 0.2. To achieve a result of the same quality as with the $EX2\text{-}NN$ scheme, the upwind bias θ must be reduced significantly. However, this could also lead the scheme to becoming unstable. This clearly demonstrates the advantage of the adaptive upwinding approach, where the numerical stability is reduced sufficiently and in an autonomous way, such that turbulent structures can develop physically.

Figure 16 shows mean flow profiles that are obtained with the $EX2\text{-}NN$, the $EX1\text{-}NN$ and the conventional CDS scheme. Due to the great influence of the numerical dissipation on this test case mentioned above, the k-exact reconstruction schemes with a fixed upwind bias are not further considered. The top figures show the normalized inverse mean centerline velocity $\Delta U/u_c$ along the axial jet direction, with the solution of the coarsest grid $(150k)$ on the left and the solution of the finest grid $(800k)$ on the right. For comparison, the experimental data has been shifted by $10D$ in the axial direction. The initial development region is located in the range $0 < x_1/D < 10$ and the transition to the self-similar regime is visible as $\Delta U/u_c$ shifts from a constant to a linear profile. For the coarse grid, the centerline profiles obtained with the $EX2\text{-}NN$ and the $EX1\text{-}NN$ scheme agree well with the experimental data, whereas the decay of the mean centerline velocity is predicted slightly too high with the CDS scheme. On the fine grid, all discretization schemes agree fairly well with the experimental data. Furthermore, it can be observed that that the length of the initial development region slightly increases for all three methods. This could be due to the fact that numerical errors caused by the non-regular grid have a smaller influence compared to the coarse grid. Thus, the initial shape of the jet can persist for a longer distance. The middle and bottom plots in Figure 16 show the axial and radial mean velocity $\langle u_{ax} \rangle$ and $\langle u_{rad} \rangle$ of the jet along the cross-stream

similarity variable η. Similarly to the work of Bogey et al. [80], the profiles are averaged over the range $50 \leq x_1/D \leq 70$. Both k-exact schemes show an exceptional agreement towards the experimental data on the coarse grid, whereas greater deviations are present for the CDS scheme. The latter agrees with the reference data only at a significantly higher grid resolution, as shown by the mean profiles in Figure 16b. A similar behaviour applies to the profiles of mean velocity fluctuations in axial and radial directions $\langle u'_{ax}u'_{ax}\rangle$ and $\langle u'_{rad}u'_{rad}\rangle$ as well as for the Reynolds Stresses $\langle u'_{ax}u'_{rad}\rangle$, which are shown in Figure 17. Again, there is an excellent agreement between simulation and experimental data for both the $EX1$-NN and the $EX2$-NN schemes, even on the coarsest grid. The CDS method requires considerably more elements for an accurate calculation of the fluctuations. These results clearly show the advantage of the higher order approach in contrast to the conventional discretization method. A certain quality of the solution can be achieved with considerably fewer elements, which results in significant savings in memory and computation time. In addition, the adaptive bias approach eliminates the need for extensive trial-and-error runs to determine any empirical parameters for the discretization scheme.

Figure 18 highlights the relationship between the three discretization schemes, the employed computational grids and the computation time in greater detail. The figures show estimated L_2-errors between simulation and experimental data, which have been calculated for various flow variables along the cross-stream similarity variable η. The data is presented against the average computation time of a single iteration τ_{it}. The latter is normalized by the maximum time for a single iteration $\tau_{it,max}$, which was encountered for the $EX2$-NN scheme on the finest grid. Remarkably, the error for the CDS scheme decreases strongly from the coarsest to the finest grid for all considered quantities. For both k-exact methods, on the other hand, a very small error is calculated already on the coarsest grid and with a further refinement the errors decrease only slightly. This is probably related to the small number of experimental reference points, which are used to calculate the L_2-errors. Nevertheless, it can be emphasized that both k-exact methods produce lower errors on the coarsest grid ($150k$) than the conventional CDS scheme on a grid with about three times as many nodal points ($450k$). This is true for the mean flow quantities as well as for the fluctuations. As a result, the $EX2$-NN method can reduce the average iteration time by roughly 60%, in order to generate a solution with at least the quality of the $450k$ grid and the CDS scheme. With the $EX1$-NN method, the average iteration time is even reduced by 70% in this case. When considering wall clock time, the savings from k-exact discretization schemes become even more significant. On coarser grids, the required CFL criterion can be met with larger time step sizes, such that the simulation time for gathering mean statistics can be achieved with fewer time steps overall. Assume that a statistically stationary solution of similar quality to the CDS scheme and the $450k$ grid is to be calculated. This can be achieved with the $EX2$-NN method already on the coarsest grid with larger time steps, thus saving roughly 75% of CPU hours. In this way, the $EX1$-NN scheme even saves 80% of CPU hours. Regarding the grid size, the elements for the k-exact schemes can be about 1.5 times larger than those for the CDS scheme, while the quality of the solution does not decrease.

5. Conclusion

In this paper, a new methodology to automatically control the numerical dissipation for vertex-centered k-exact reconstruction schemes has been introduced. The approach maintains the stability of the discretization scheme, while numerical dissipation errors are reduced to a minimum. It was shown by means of a Fourier analysis of the convective operator how the upwind bias factor θ of the discretization scheme is connected to the introduction of numerical dissipation. Based on a Von Neumann analysis for the linear advection-diffusion equation, a stability equation was derived, which maps the required upwind bias values to a local CFL and Reynolds number. In this way, the optimum upwind bias can be related to the local flow state, such that the scheme remains stable and the numerical dissipation is minimized. This enhances the overall accuracy of the scheme and avoids the introduction of empirical parameters for stability control. Even though the linear advection-diffusion equation is only of model character, the approach could be applied successfully to the incompressible Navier-Stokes equations. To calculate the required upwind bias with minimum computation time, the outcome of the stability equation has been tabulated indirectly by means of an artificial neural network model. It was shown that the trained networks reproduce the underlying stability

equation over a wide range of the CFL number and the local grid Reynolds number with a maximum error of less than 5%. In particular, a strategy was presented in which the network models are trained with different sequential optimization algorithms, where a penalty term is introduced in the objective function. This considerably reduced the undershoot of the minimum required numerical dissipation due to approximation errors of the model.

The novel approach was examined in three benchmark test cases. The convection of a vortex on a periodic grid revealed that the order of accuracy is maintained when the upwind bias is adjusted adaptively, both on a cartesian and a fully unstructured grid. For this test case, the accuracy of the 2-exact discretization scheme could also be significantly increased by using the adaptive dissipation control. In a further test case, the turbulent flow through a periodic pipe was assessed by means of LES, where the WALE model was used for subgrid-scale modeling. It was shown that for both the 1- and 2-exact approach the use of the adaptive upwind bias significantly enhanced the obtained flow profiles with very little overhead compared to the case with a fixed upwind bias. Even on a relatively coarse grid, the reference DNS data could be reproduced with very good accuracy. Furthermore, the adaptive approach enabled to obtain the solution without requiring time-consuming fine-tuning of empirical discretization parameters. All the employed k-exact methods enabled massive accuracy improvements when compared to a conventional CDS discretization scheme, even at lower grid resolution levels. This demonstrated, that the chosen high-order approach enables significant computation time savings for this benchmark, since less elements were required to meet a desired level of accuracy for the solution. The analysis of velocity spectra showed, that the k-exact schemes feature a higher cutoff frequency for the resolution of the inertial subrange than the CDS scheme. Finally, a benchmark for a turbulent round jet was conducted on four fully unstructured grids with different grid scales. This test case further highlighted the importance of using a parameter-free approach for the control of the numerical dissipation of the k-exact schemes. Using a fixed value for θ, which yielded good results in the previous benchmarks, led to a strong damping of turbulent structures and thus to an invalid solution. In contrast, highly accurate results could be obtained with both the 1- and 2-exact adaptive discretization approach. The agreement of flow profiles with experimental data demonstrated significant improvement in accuracy of both adaptive k-exact methods compared to the conventional CDS scheme. Both the adaptive 1- and 2-exact procedures accurately predicted the averaged flow profiles as well as higher statistical moments, even on the coarsest computational grid. On the other hand, the CDS method required a computational grid with three times the number of nodes to achieve similar levels of accuracy. Again, this showed a high potential of the adaptive k-exact schemes to save computation time, since these enabled to obtain a statistically stationary solution with sufficient accuracy on the coarse grid with fewer time steps. Moreover, the average iteration time with this coarse grid and the k-exact methods was even lower than the iteration time with the conventional scheme on a sufficiently fine grid. For future works, the employed k-exact reconstruction approach will be extended for the simulation of variable-density flows. This will enable to further investigate the influence of the presented high order approach on the simulation of turbulent reactive flows in complex geometries.

Appendix A. Neural network weights

In the following, the trained neural network weights for the 1- and 2-exact discretization schemes are given. The indices indicate the corresponding combinations of the limiter variables $\{\psi_U, \psi_D\}$. The weights for the $EX1\text{-}NN$ scheme are given by:

$$\mathbf{b}^{(1)}_{\{1,0\}} = \begin{bmatrix} 5.03737211 \\ 6.32343674 \\ 0.50794315 \\ -19.97484780 \\ -12.92175290 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{1,0\}} = \begin{bmatrix} 2.25994802 & 0.97043753 \\ -0.34634137 & 2.45287037 \\ -0.74894267 & -1.58069396 \\ -9.59228420 & 33.69055560 \\ 0.72971189 & 15.89204600 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{1,0\}} = \begin{bmatrix} 26.49484630 \\ 73.63108060 \\ -5.27458429 \\ 0.69498062 \\ -1.34468186 \end{bmatrix} \quad b^{(2)}_{\{1,0\}} = -83.19099430$$

$$\mathbf{b}^{(1)}_{\{0,1\}} = \begin{bmatrix} 1.15107429 \\ 0.42764404 \\ 5.70031977 \\ 0.77053821 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{0,1\}} = \begin{bmatrix} -4.05483866 & 0.20609209 \\ -0.29656017 & -1.89375186 \\ 3.96923113 & -2.37948823 \\ -1.31519699 & -1.94232666 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{0,1\}} = \begin{bmatrix} 0.41760081 \\ 2.14565444 \\ -7.33920145 \\ 1.06283844 \end{bmatrix} \quad b^{(2)}_{\{0,1\}} = 4.14726830$$

$$\mathbf{b}^{(1)}_{\{1,1\}} = \begin{bmatrix} 3.60439992 \\ 3.04572892 \\ 13.47363760 \\ -1.70139468 \\ 1.17576802 \\ -3.19013000 \\ 1.19367719 \\ 4.03480864 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{1,1\}} = \begin{bmatrix} -0.43598291 & 214.59857200 \\ -0.03501525 & -7.88090563 \\ 8.97540951 & -18.85119820 \\ 0.42859894 & 25.66828160 \\ 4.03518677 & -3.53732133 \\ 0.41089782 & -214.85285900 \\ 4.03685284 & -3.59446573 \\ -0.46742636 & 215.02186600 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{1,1\}} = \begin{bmatrix} -836.66748000 \\ 0.36775029 \\ -5.56125069 \\ 0.19995812 \\ 76.90611270 \\ -348.61282300 \\ -78.34106450 \\ 496.30502300 \end{bmatrix} \quad b^{(2)}_{\{1,1\}} = 348.56750500$$

The weights for the $EX2\text{-}NN$ scheme are given by:

$$\mathbf{b}^{(1)}_{\{1,0\}} = \begin{bmatrix} -0.13289030 \\ 2.19299674 \\ 2.85025001 \\ 10.00010490 \\ 0.53818160 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{1,0\}} = \begin{bmatrix} 0.18972446 & -2.26156664 \\ -0.38291270 & -2.68230867 \\ 2.60307598 & -6.84366083 \\ -0.46268842 & -11.01203350 \\ -1.80168939 & -1.99879658 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{1,0\}} = \begin{bmatrix} 6.66433859 \\ 2.75568151 \\ 0.25530335 \\ 3.01141453 \\ 0.55966282 \end{bmatrix} \quad b^{(2)}_{\{1,0\}} = -7.46983719$$

$$\mathbf{b}^{(1)}_{\{0,1\}} = \begin{bmatrix} -0.79968619 \\ 1.66791654 \\ 0.91030502 \\ 1.01154685 \\ -0.23419440 \\ 2.04055214 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{0,1\}} = \begin{bmatrix} -3.24619746 & 1.40960002 \\ -0.36590472 & -1.61762512 \\ 3.06775546 & -1.61764526 \\ 2.91098762 & -1.81616390 \\ -0.13441297 & -1.55384851 \\ 0.34248400 & -1.27468443 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{0,1\}} = \begin{bmatrix} -74.03594970 \\ -2.55656266 \\ -156.37689200 \\ 81.80142970 \\ 5.91900158 \\ 5.08785439 \end{bmatrix} \quad b^{(2)}_{\{0,1\}} = 69.54219820$$

$$\mathbf{b}^{(1)}_{\{1,1\}} = \begin{bmatrix} -0.70265764 \\ 3.04374647 \\ 3.72396946 \\ -1.18680501 \\ -0.74276817 \\ -0.46596319 \\ -3.68388486 \\ 2.15279818 \\ 5.65592003 \end{bmatrix} \quad \mathbf{w}^{(1)}_{\{1,1\}} = \begin{bmatrix} 0.17748395 & -56.74145890 \\ -0.37777513 & 144.93638600 \\ -0.47338566 & 146.23722800 \\ 0.24627742 & -109.39630100 \\ -0.43466511 & -9.22253513 \\ -5.33628941 & 6.81360531 \\ -3.16690898 & -4.61405945 \\ -0.27552703 & 120.41154500 \\ 3.77051163 & -3.82512641 \end{bmatrix} \quad \mathbf{w}^{(2)}_{\{1,1\}} = \begin{bmatrix} 5.34142637 \\ -237.00431800 \\ 97.57673650 \\ 58.33208850 \\ -1.11101890 \\ 0.46680856 \\ -5.22708130 \\ 198.47425800 \\ -9.44149303 \end{bmatrix} \quad b^{(2)}_{\{1,1\}} = -53.47458270$$

References

[1] J. A. Ekaterinaris, High-order accurate, low numerical diffusion methods for aerodynamics, Progress in Aerospace Sciences 41 (3-4) (2005) 192–300.

[2] W. McMullan, G. Page, Towards large eddy simulation of gas turbine compressors, Progress in Aerospace Sciences 52 (2012) 30–47.

[3] J. H. Ferziger, M. Perić, R. L. Street, Computational Methods for Fluid Dynamics, Vol. 3, Springer, 2002.

[4] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Butterworth-Heinemann, 2015.

[5] Z. J. Wang, High-order methods for the euler and navier–stokes equations on unstructured grids, Progress in Aerospace Sciences 43 (1-3) (2007) 1–41.

[6] H. Huynh, Z. J. Wang, P. E. Vincent, High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids, Computers & fluids 98 (2014) 209–220.

[7] E. Lamballais, V. Fortuné, S. Laizet, Straightforward high-order numerical dissipation via the viscous term for direct and large eddy simulation, Journal of Computational Physics 230 (9) (2011) 3270–3275.

[8] B. Van Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method, Journal of Computational Physics 32 (1) (1979) 101–136.

[9] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace Sciences Meeting, 1989, p. 366.

[10] K. Michalak, C. Ollivier-Gooch, Limiters for unstructured higher-order accurate solutions of the euler equations, in: 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008, p. 776.

[11] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1) (1994) 200–212.

[12] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, Journal of Computational Physics 126 (1) (1996) 202–228.

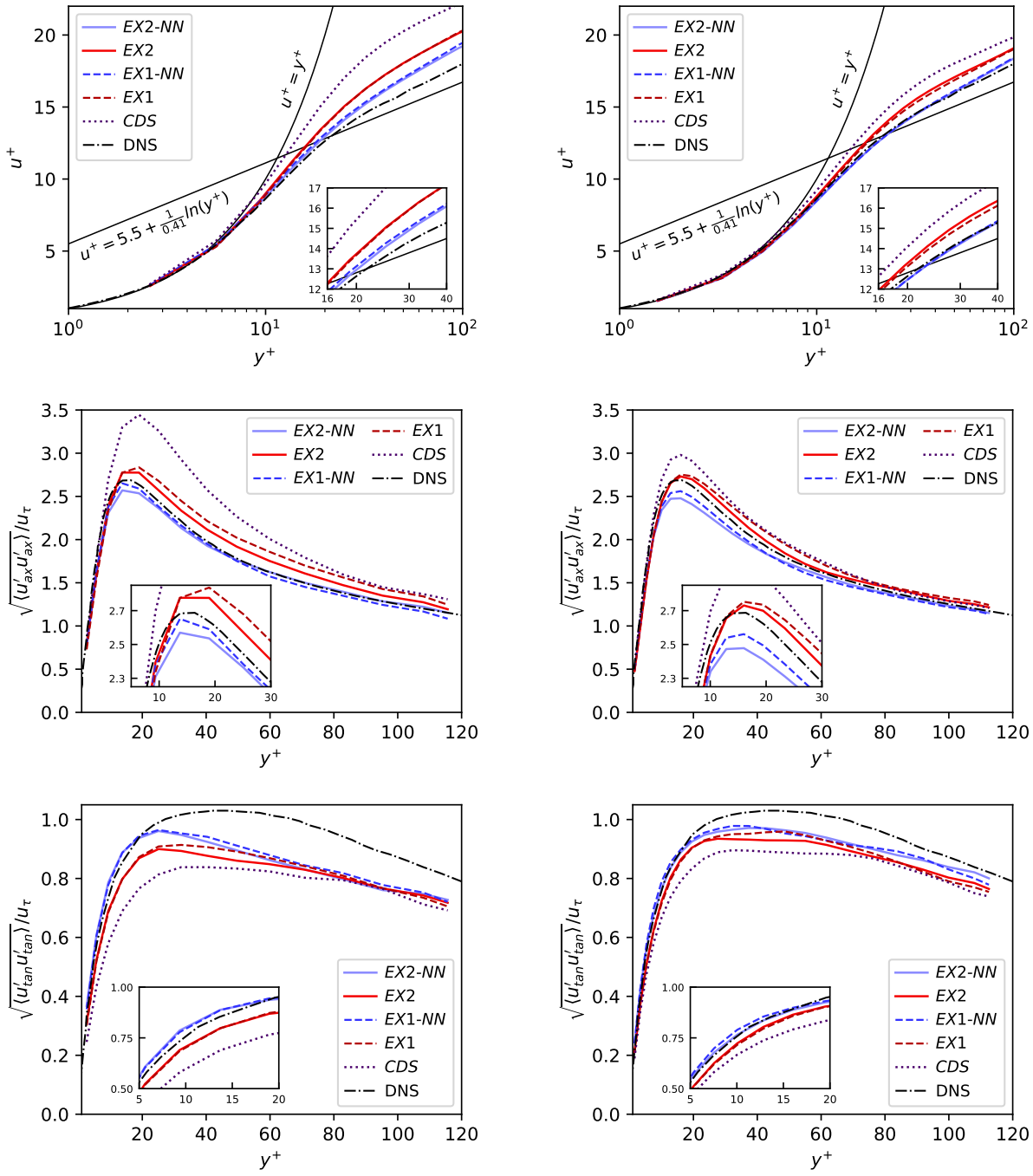
[13] S. K. Godunov, A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations, Math. Sbornik 47 (1959) 271–306.

[14] J. VonNeumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of Applied Physics 21 (3) (1950) 232–237.

[15] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes, in: 14th fluid and plasma dynamics conference, 1981, p. 1259.

[16] J. J. Van der Vegt, H. Van der Ven, Space–time discontinuous galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation, Journal of Computational Physics 182 (2) (2002) 546–585.
[17] F. Setzwein, P. Ess, P. Gerlinger, An implicit high-order k-exact finite-volume approach on vertex-centered unstructured grids for incompressible flows, Journal of Computational Physics 446 (2021) 110629.
[18] J.-M. Lourier, A. Huber, B. Noll, M. Aigner, Numerical analysis of indirect combustion noise generation within a subsonic nozzle, AIAA Journal 52 (10) (2014) 2114–2126.
[19] A. Fiolitakis, P. R. Ess, P. Gerlinger, M. Aigner, Modeling of heat transfer and differential diffusion in transported pdf methods, Combustion and Flame 161 (8) (2014) 2107–2119.
[20] J.-M. Lourier, M. Stöhr, B. Noll, S. Werner, A. Fiolitakis, Scale adaptive simulation of a thermoacoustic instability in a partially premixed lean swirl combustor, Combustion and Flame 183 (2017) 343–357.
[21] C. Eberle, P. Gerlinger, K. P. Geigle, M. Aigner, Toward finite-rate chemistry large-eddy simulations of sooting swirl flames, Combustion Science and Technology 190 (7) (2018) 1194–1217.
[22] G. Eckel, P. Le Clercq, T. Kathrotia, A. Saenger, S. Fleck, M. Mancini, T. Kolb, M. Aigner, Entrained flow gasification. part 3: Insight into the injector near-field by large eddy simulation with detailed chemistry, Fuel 223 (2018) 164–178.
[23] G. Eckel, J. Grohmann, L. Cantu, N. Slavinskaya, T. Kathrotia, M. Rachner, P. Le Clercq, W. Meier, M. Aigner, Les of a swirl-stabilized kerosene spray flame with a multi-component vaporization model and detailed chemistry, Combustion and Flame 207 (2019) 134–152.
[24] M. Grader, C. Eberle, P. Gerlinger, Large-eddy simulation and analysis of a sooting lifted turbulent jet flame, Combustion and Flame 215 (2020) 458–470.
[25] B. Enderle, B. Rauch, F. Grimm, G. Eckel, M. Aigner, Non-intrusive uncertainty quantification in the simulation of turbulent spray combustion using polynomial chaos expansion: A case study, Combustion and Flame 213 (2020) 26–38.
[26] G. Pont, D. Puech, P. Brenner, Hybrid rans/les simulation of a space launcher using a high order finite volume scheme and grid intersections technique, in: Symposium on Hybrid RANS-LES Methods, Springer, 2016, pp. 347–356.
[27] F. Setzwein, P. Ess, P. Gerlinger, High-order k-exact finite volume scheme for vertex-centered unstructured grids, in: AIAA Scitech 2020 Forum, 2020, p. 1785.
[28] F. Setzwein, M. Spraul, P. Ess, P. M. Gerlinger, On the structure of correction matrices for a k-exact high-order finite-volume scheme on vertex-centered unstructured grids, in: AIAA Scitech 2021 Forum, 2021, p. 1548.
[29] N. Discacciati, J. S. Hesthaven, D. Ray, Controlling oscillations in high-order discontinuous galerkin schemes using artificial viscosity tuned by neural networks, Journal of Computational Physics 409 (2020) 109304.
[30] D. Ray, J. S. Hesthaven, An artificial neural network as a troubled-cell indicator, Journal of Computational Physics 367 (2018) 166–191.
[31] J. Yu, J. S. Hesthaven, C. Yan, A data-driven shock capturing approach for discontinuous galekin methods, Tech. rep. (2018).
[32] O. P. Bruno, J. S. Hesthaven, D. V. Leibovici, Fc-based shock-dynamics solver with neural-network localized artificial-viscosity assignment, arXiv preprint arXiv:2111.01315.
[33] L. Schwander, D. Ray, J. S. Hesthaven, Controlling oscillations in spectral methods by local artificial viscosity governed by neural networks, Journal of Computational Physics 431 (2021) 110144.
[34] A. D. Beck, J. Zeifang, A. Schwarz, D. G. Flad, A neural network based shock detection and localization approach for discontinuous galerkin methods, Journal of Computational Physics 423 (2020) 109824.
[35] B. Stevens, T. Colonius, Enhancement of shock-capturing methods via machine learning, Theoretical and Computational Fluid Dynamics 34 (2020) 483–496.
[36] T. Kossaczká, M. Ehrhardt, M. Günther, Enhanced fifth order weno shock-capturing schemes with deep learning, arXiv preprint arXiv:2103.04988.
[37] S. B. Pope, Turbulent Flows, Cambridge University Press, 2000.
[38] F. Ducros, F. Nicoud, T. Poinsot, Wall-adapting local eddy-viscosity models for simulations in complex geometries, Numerical Methods for Fluid Dynamics VI (1998) 293–299.
[39] F. Nicoud, F. Ducros, Subgrid-scale stress modelling based on the square of the velocity gradient tensor, Flow, turbulence and Combustion 62 (3) (1999) 183–200.
[40] J.-L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering 195 (44-47) (2006) 6011–6045.
[41] T. J. Barth, Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations.
[42] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, Journal of Computational Physics 181 (2) (2002) 729–752.
[43] J. M. Pozo, M.-C. Villa-Uriol, A. F. Frangi, Efficient 3d geometric and zernike moments computation from unstructured surface meshes, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 471–484.
[44] M. R. Charest, T. R. Canfield, N. R. Morgan, J. Waltz, J. G. Wohlbier, A high-order vertex-based central eno finite-volume scheme for three-dimensional compressible flows, Computers & Fluids 114 (2015) 172–192.
[45] F. Haider, P. Brenner, B. Courbet, J.-P. Croisille, Parallel implementation of k-exact finite volume reconstruction on unstructured grids, in: High Order Nonlinear Numerical Schemes for Evolutionary PDEs, Springer, 2014, pp. 59–75.
[46] Q. Wang, Y.-X. Ren, W. Li, Compact high order finite volume method on unstructured grids i: Basic formulations and one-dimensional schemes, Journal of Computational Physics 314 (2016) 863–882.
[47] Compact high order finite volume method on unstructured grids iii: Variational reconstruction, Journal of Computational physics 337 (2017) 1–26.
[48] S. Mathur, J. Murthy, A pressure-based method for unstructured meshes, Numerical heat transfer 31 (2) (1997) 195–215.
[49] H. Nishikawa, A hyperbolic poisson solver for tetrahedral grids, Journal of Computational Physics (2020) 109358.

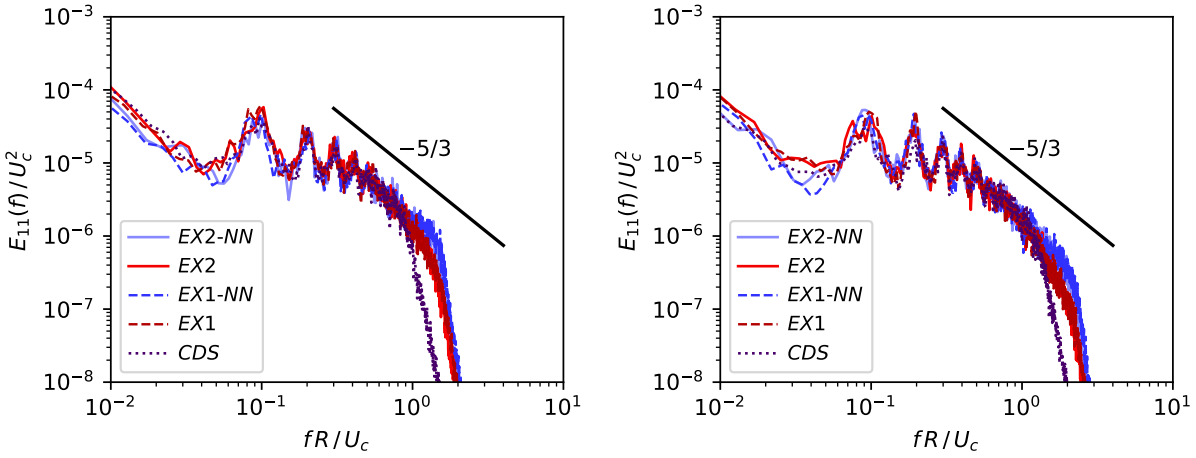
[50] L. Jofre, O. Lehmkuhl, J. Ventosa, F. X. Trias, A. Oliva, Conservation properties of unstructured finite-volume mesh schemes for the navier-stokes equations, Numerical Heat Transfer, Part B: Fundamentals 65 (1) (2014) 53–79.
[51] G. Pont, P. Brenner, P. Cinnella, B. Maugars, J.-C. Robinet, Multiple-correction hybrid k-exact schemes for high-order compressible rans-les simulations on fully unstructured grids, Journal of Computational Physics 350 (2017) 45–83.
[52] S. K. Lele, Compact finite difference schemes with spectral-like resolution, Journal of computational physics 103 (1) (1992) 16–42.
[53] R. Vichnevetsky, J. B. Bowles, Fourier analysis of numerical approximations of hyperbolic equations, SIAM, 1982.
[54] J. Froehlich, Large eddy simulation turbulenter Stroemungen, Vol. 1, Springer, 2006.
[55] P. Khosla, S. Rubin, A diagonally dominant second-order accurate implicit scheme, Computers & Fluids 2 (2) (1974) 207–209.
[56] C. Hirsch, Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows, Vol. 1 and 2, John Wiley & Sons, Chichester, New York, 1990.
[57] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, 1995.
[58] H. Simon, Neural Networks: A Comprehensive Foundation, Macmillan Coll Div, 1995.
[59] R. Beale, T. Jackson, Neural Computing - An Introduction, Institute of Physics Publishing, 1990.
[60] J. Watt, R. Borhani, A. K. Katsaggelos, Machine learning refined: Foundations, algorithms, and applications, Cambridge University Press, 2020.
[61] L. Bottou, O. Bousquet, The tradeoffs of large scale learning, Advances in neural information processing systems 20.
[62] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Quarterly of applied mathematics 2 (2) (1944) 164–168.
[63] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, Journal of the society for Industrial and Applied Mathematics 11 (2) (1963) 431–441.
[64] M. T. Hagan, M. B. Menhaj, Training feedforward networks with the marquardt algorithm, IEEE transactions on Neural Networks 5 (6) (1994) 989–993.
[65] H. Yu, B. M. Wilamowski, Levenberg–marquardt training, in: Intelligent systems, CRC Press, 2018, pp. 12–1.
[66] K. L. Priddy, P. E. Keller, Artificial neural networks: an introduction, Vol. 68, SPIE press, 2005.
[67] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.
[68] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[69] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32.
[70] T. Knopp, X. Zhang, R. Kessler, G. Lube, Enhancement of an industrial finite-volume code for large-eddy-type simulation of incompressible high reynolds number flow using near-wall modelling, Computer methods in applied mechanics and engineering 199 (13-16) (2010) 890–902.
[71] C. Rhie, W. L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA journal 21 (11) (1983) 1525–1532.
[72] Y. Morinishi, T. S. Lund, O. V. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, Journal of computational physics 143 (1) (1998) 90–124.
[73] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low mach number turbulent flows, Journal of Computational Physics 227 (15) (2008) 7125–7159.
[74] J. Eggels, F. Unger, M. Weiss, J. Westerweel, R. J. Adrian, R. Friedrich, F. Nieuwstadt, Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment, Journal of Fluid Mechanics 268 (1994) 175–210.
[75] K. Fukagata, N. Kasagi, Highly energy-conservative finite difference method for the cylindrical coordinate system, Journal of Computational Physics 181 (2) (2002) 478–498.
[76] I. Wygnanski, H. Fiedler, Some measurements in the self-preserving jet, Journal of Fluid Mechanics 38 (3) (1969) 577–612.
[77] N. R. Panchapakesan, J. L. Lumley, Turbulence measurements in axisymmetric jets of air and helium. part 1. air jet, Journal of Fluid Mechanics 246 (1993) 197–223.
[78] G. Lipari, P. K. Stansby, Review of experimental data on incompressible turbulent round jets, Flow, turbulence and combustion 87 (1) (2011) 79–114.
[79] M. Olsson, L. Fuchs, Large eddy simulation of the proximal region of a spatially developing circular jet, Physics of Fluids 8 (8) (1996) 2125–2137.
[80] C. Bogey, C. Bailly, Turbulence and energy budget in a self-preserving round jet: direct evaluation using large eddy simulation, Journal of Fluid Mechanics 627 (2009) 129–160.
[81] J. Kim, H. Choi, Large eddy simulation of a circular jet: effect of inflow conditions on the near field, Journal of Fluid Mechanics 620 (2009) 383–411.
[82] T. B. Gohil, A. K. Saha, K. Muralidhar, Large eddy simulation of a free circular jet, Journal of Fluids Engineering 136 (5).
[83] N. A. Loppi, F. D. Witherden, A. Jameson, P. E. Vincent, A high-order cross-platform incompressible navier–stokes solver via artificial compressibility with application to a turbulent jet, Computer Physics Communications 233 (2018) 193–205.
[84] H. J. Hussein, S. P. Capp, W. K. George, Velocity measurements in a high-reynolds-number, momentum-conserving, axisymmetric, turbulent jet, Journal of Fluid Mechanics 258 (1994) 31–75.



(a) Coarse grid

(b) Fine grid

Figure 12: Radial flow profiles of the normalized axial mean velocity $u^+ = \langle u_{ax} \rangle / u_\tau$, the normalized axial and tangential velocity fluctuations $\sqrt{\langle u'_{ax} u'_{ax} \rangle}$ and $\sqrt{\langle u'_{tan} u'_{tan} \rangle}$ against the radial coordinate in wall units $y^+ = y u_\tau / \nu$ for the turbulent pipe flow. Results for the coarse grid are shown on the left and for the fine grid on the right.



(a) Coarse grid

(b) Fine grid

Figure 13: One-dimensional energy spectra $E_{11}(f)$ obtained for the turbulent pipe flow test case with various discretization schemes.

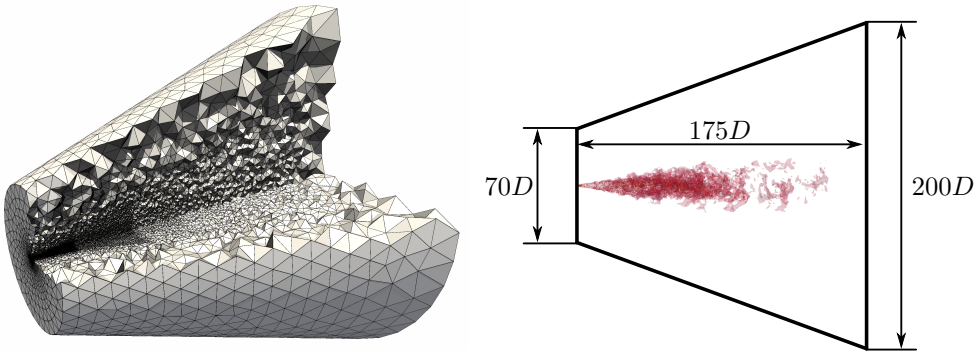
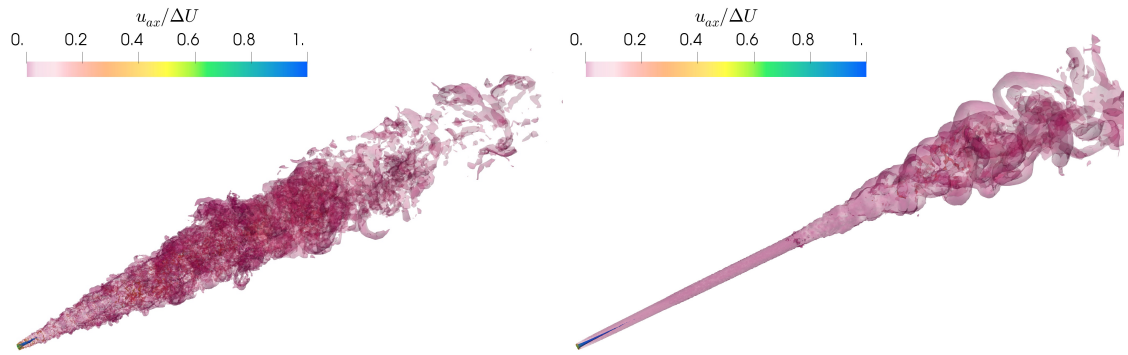


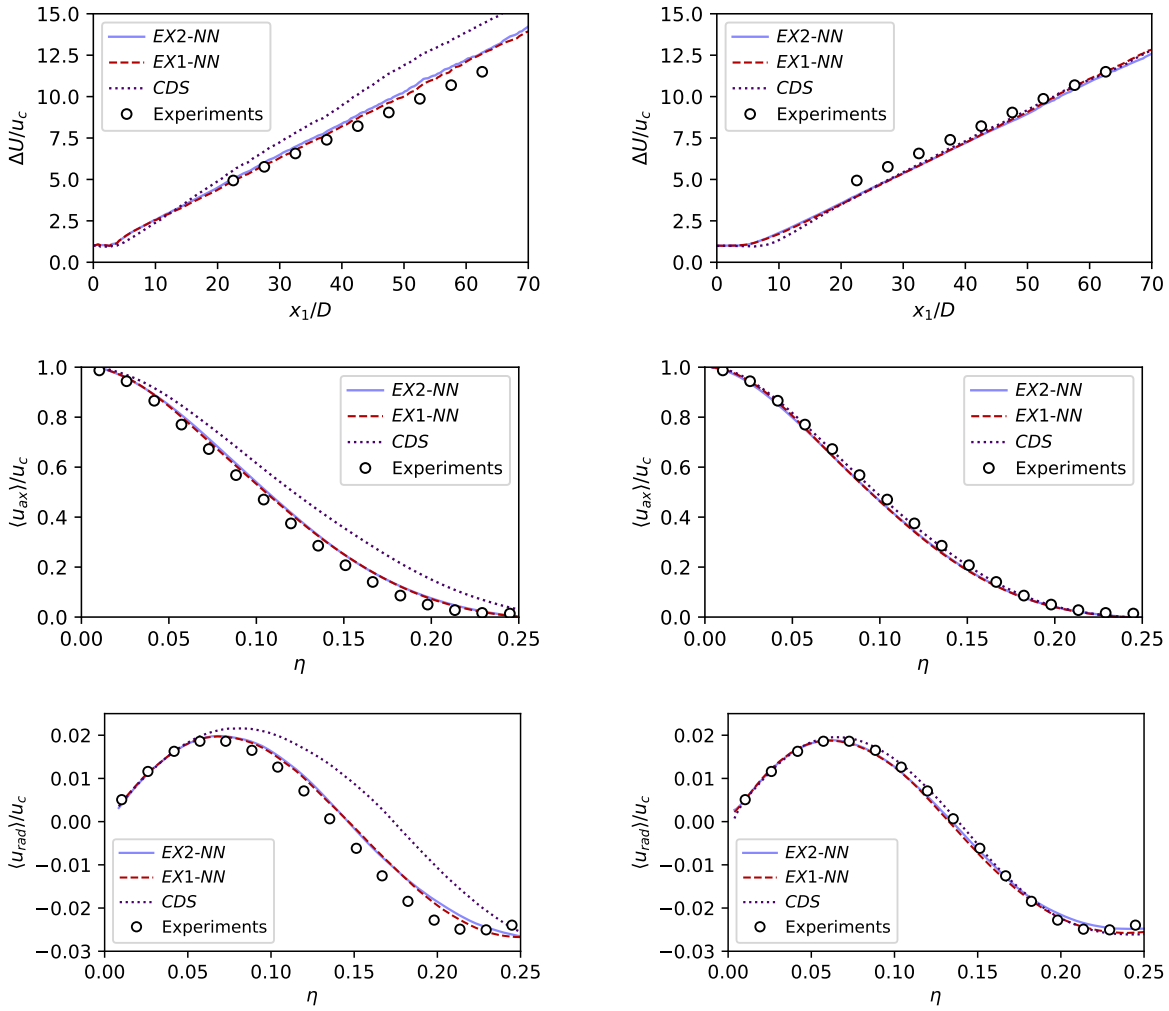
Figure 14: Employed coarse mesh for the round jet benchmark (left) and corresponding domain size (right).



(a) $EX2\text{-}NN$ scheme with fully adaptive upwind bias $\theta_0^{(NN)}$

(b) $EX2$ scheme with fixed upwind bias $\theta = 0.2$

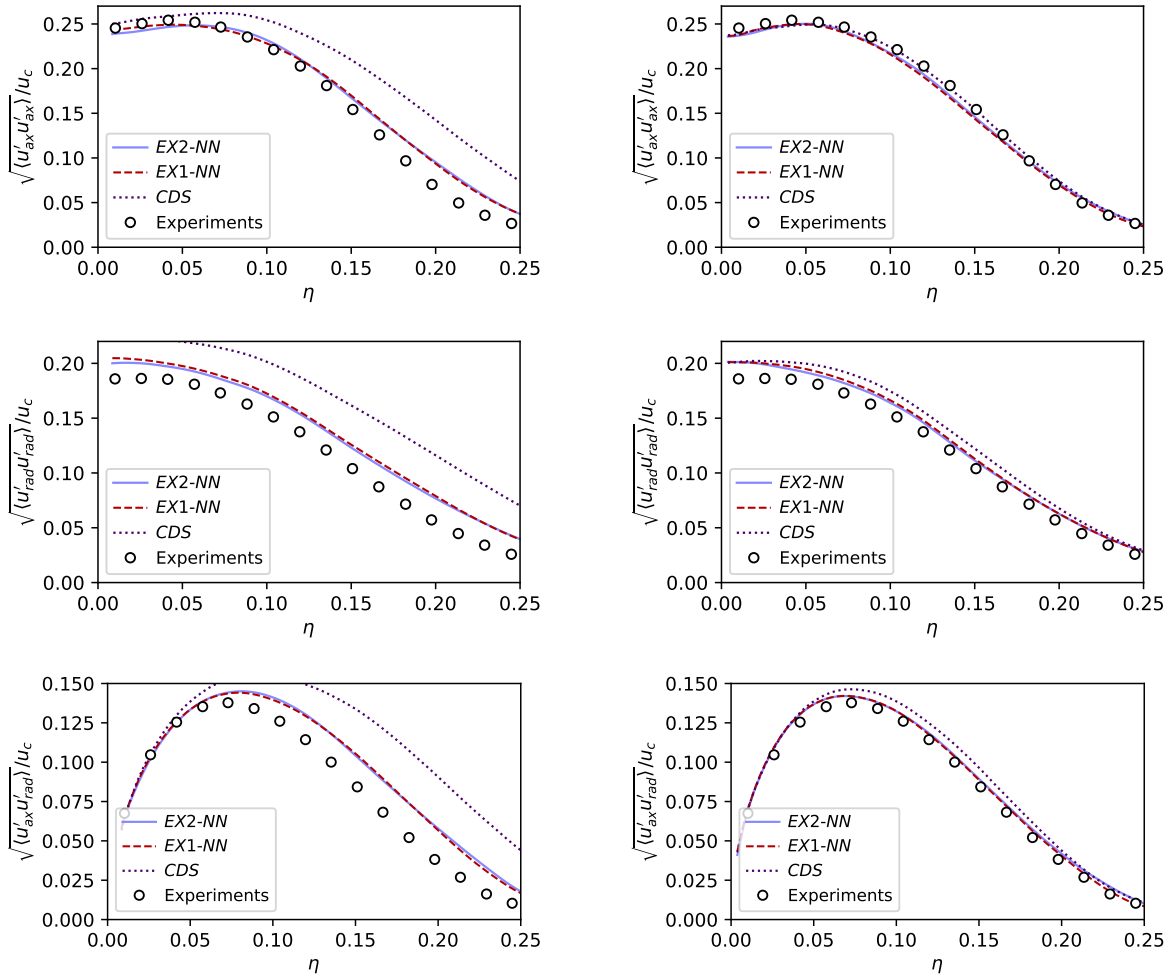
Figure 15: Scaled vorticity field $|\boldsymbol{\omega}|\,(x_1/\Delta U)$, represented by a contour that is associated with a value of two. Surfaces are coloured by means of the normalized axial velocity $u_1/\Delta U$. The results are obtained on the finest mesh with $800,000$ primary grid nodes.



(a) 147,811 grid nodes

(b) 816,944 grid nodes

Figure 16: Mean flow profiles for the turbulent round jet test case with experimental reference data of Panchapakesan and Lumley [77]. The top figures shows the normalized inverse centerline mean velocity $\Delta U/u_c$. The figures in the middle and bottom figures show profiles of the normalized axial and radial mean velocity $\langle u_{ax}\rangle/u_c$ and $\langle u_{rad}\rangle/u_c$ along the cross-stream similarity variable η. Results for the coarse $150k$ grid are shown on the left and for the fine $800k$ grid on the right.



(a) 147,811 grid nodes

(b) 816,944 grid nodes

Figure 17: Profiles of mean velocity fluctuations for the turbulent round jet test case with experimental reference data of Panchapakesan and Lumley [77]. The top and middle figures show axial and radial fluctuations $\langle u'_{ax}u'_{ax}\rangle$ and $\langle u'_{rad}u'_{rad}\rangle$ along the cross-stream similarity variable η, whereas the bottom figure shows the Reynolds Stresses $\langle u'_{ax}u'_{rad}\rangle$. Results for the coarse $150k$ grid are shown on the left and for the fine $800k$ grid on the right.

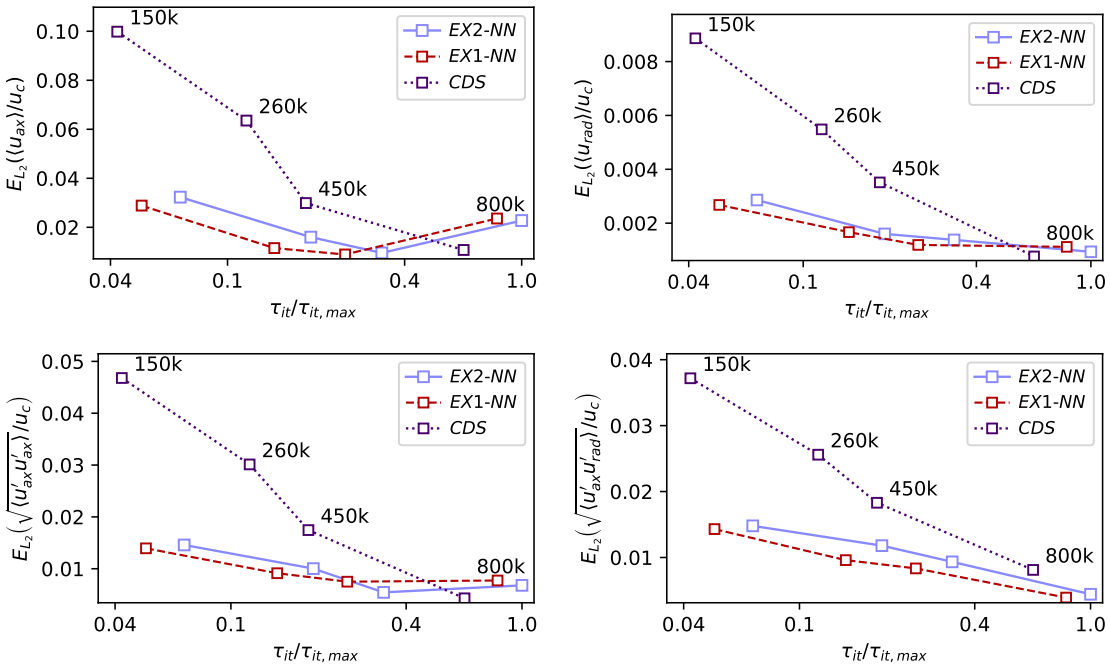


Figure 18: L_2-norm error between simulation and experimental data of the normalized mean radial velocity (left) and the normalized Reynolds stresses (right) in the self-similar region. The data is shown along the normalized computation time $\tau_{it}/\tau_{it,max}$ that is obtained for the respective grids and discretization schemes.