INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **00**:1–24
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

Implicit Finite Element Schemes for the Stationary Compressible Euler Equations

Marcel Gurris[1]* Dmitri Kuzmin[2], Stefan Turek[1]

[1] *Dortmund University of Technology, Institute of Applied Mathematics, Vogelpothsweg 87, D-44227 Dortmund, Germany*
[2] *University of Erlangen-Nürnberg, Chair of Applied Mathematics III, Haberstr. 2, D-91058 Erlangen, Germany*

SUMMARY

Semi-implicit and Newton-like finite element methods are developed for the stationary compressible Euler equations. The Galerkin discretization of the inviscid fluxes is potentially oscillatory and unstable. To suppress numerical oscillations, the spatial discretization is performed by a high-resolution finite element scheme based on algebraic flux correction. A multidimensional limiter of TVD type is employed. An important goal is the efficient computation of stationary solutions in a wide range of Mach numbers, which is a challenging task due to oscillatory correction factors associated with TVD-type flux limiters. A semi-implicit scheme is derived by a time-lagged linearization of the nonlinear residual, and a Newton-like method is obtained in the limit of infinite CFL numbers.
Special emphasis is laid on the numerical treatment of weakly imposed characteristic boundary conditions. A boundary Riemann solver is used to avoid unphysical boundary states. It is shown that the proposed approach offers unconditional stability as well as higher accuracy and better convergence behavior than in the case of algorithms in which the boundary conditions are implemented in a strong sense. The spatial accuracy of the whole scheme and the boundary conditions is analyzed by grid convergence studies. Copyright © 2010 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Euler equations, high-resolution schemes, boundary conditions, implicit schemes, stationary flow, compressible flow

1. INTRODUCTION

In recent years, significant advances have been made in the development of numerical methods for convection-dominated transport problems. In particular, Kuzmin et al. [16, 17, 18, 19, 21, 15, 20] have developed a new approach to the design of high-resolution finite element schemes. Within the framework of algebraic flux correction, the coefficients of a standard Galerkin discretization are constrained using flux limiters based on a generalization of flux-corrected transport (FCT) algorithms and total variation diminishing (TVD) methods. As demonstrated by John and Schmeyer [13], algebraic flux correction is more reliable than mainstream stabilization/shock capturing techniques. The linearized FCT algorithm [15, 21] is to be recommended for transient flows, whereas flux limiters of TVD type are better suited for the computation of steady-state solutions which is of primary interest in the present numerical study. Hence, the latter limiting strategy is adopted.

In contrast to their superior approximation properties, TVD-like flux limiters are known to

*Correspondence to: E-mail: marcel.gurris@math.uni-dortmund.de

inhibit steady-state convergence [36]. It is not unusual that the stationary residual decreases by a few orders of magnitude, after which convergence stalls due to the oscillatory correction factors. The lack of convergence compromises the advantages of flux limiting and increases the computational cost. Therefore, it is essential to find the reasons for and remedies to the above convergence problems [8]. In real-life applications, the computational mesh is frequently unstructured, and adaptive mesh refinement [25] is a must. This calls for the use of implicit schemes since the time step size in explicit computations is restricted by stability constraints based on the size of the smallest mesh cells.

Implicit schemes have the potential of being unconditionally stable without CFL restrictions if they are designed in a proper way. Since the convergence to steady-state depends on the propagation speed of the error waves, large CFL numbers accelerate the convergence to steady-state. This is primarily important at the sonic point and in low Mach number regions, where wave speeds are close to zero. Moreover, the implicit Euler approach corresponds to upwinding in time and therefore enjoys very useful numerical properties. This makes the backward Euler scheme a favorable choice in steady-state computations and it is therefore applied in this study. On the other hand, nonlinear systems must be solved and the computation of the nonlinear preconditioner is a challenging task. Therefore, the implementation of implicit schemes is far from being trivial.

In this work a semi-implicit approach is presented to circumvent the computationally expensive nonlinear iterations. Implicit and semi-implicit schemes require a very powerful linear solver since the Jacobians usually lack diagonal dominance at least at high CFL numbers. This has an adverse effect on the convergence of many iterative solvers. Implicit solvers are still rarely used for the computation of stationary solutions to the Euler equations. Their development has been pursued by several groups [3, 5, 26, 35]. However, many existing schemes employ linearizable/differentiable limiters, are conditionally stable, and the rate of steady-state convergence deteriorates if the CFL number exceeds a certain upper bound. The scheme presented here converges for arbitrary CFL numbers despite oscillatory correction factors and the rate of steady-state convergence does not deteriorate for large CFL numbers.

The development of robust and accurate boundary conditions is a goal of primary importance and sufficient care must be taken in the numerical implementation. The accuracy, robustness, stability, and convergence of an implicit solver are strongly influenced by the boundary treatment. As shown in [9], strongly imposed boundary conditions may inhibit convergence to a steady state. Thus, it is worthwhile to use flux boundary conditions of Neumann type. The weak type of boundary conditions turns out to be much more stable and flexible than its strong counterpart. When boundary conditions are prescribed in a weak sense, only the boundary integral of the weak formulation is affected by the boundary conditions, while the volume integrals remain unchanged. This is similar to the boundary treatment, which is usually implemented in finite volume schemes. In the finite volume framework the boundary fluxes are directly overwritten by the imposed boundary conditions [30, 26, 31, 32].

The Neumann-type of boundary conditions, based on the weak formulation, can be treated implicitly and incorporated into the matrix in a physical way. It improves the convergence rates and does not affect the matrix properties or give rise to stability restrictions in contrast to the strong type of boundary conditions. Dolejší and Feistauer [3] observe a stability restriction of $CFL \lesssim 6$ for an explicit implementation of weak wall boundary conditions, while the stability is significantly improved for a semi-implicit version up to $CFL \approx 100$. This emphasizes the importance of an implicit treatment of boundary conditions for the numerical performance, which is presented in this study. We recommend a boundary Riemann solver to compute the boundary fluxes in the boundary integrals to avoid unphysical effects particularly at large CFL numbers. To define a boundary Riemann problem the concept of ghost nodes is introduced. We show that a suitable treatment of boundary conditions makes it possible to achieve unconditional stability.

DOI: 10.1002/fld

In spite of their major importance boundary conditions for hyperbolic systems are rarely analyzed and discussed in the literature particularly in the framework of finite elements. Selmin et al. [32] overwrite the boundary integral by a lumped mass approximation, while Shapiro [30] and Lyra [23] directly change the boundary fluxes to incorporate the boundary conditions. A comparative study of solid wall boundary conditions for the Euler equations is given in [1] and a detailed description of boundary conditions for different hyperbolic systems is reported in [6]. Neumann-type boundary conditions were also implemented in the discontinuous Galerkin framework [5, 3].

In the following sections, we address the design of high-resolution finite element schemes for the Euler equations. In particular, the treatment of boundary conditions based on a boundary Riemann solver is described, and the implicit solver is presented. Furthermore, we illustrate the steady-state convergence of the proposed algorithm by numerical results and perform grid convergence studies to examine the spatial accuracy. Finally, the numerical performance and accuracy of the proposed boundary conditions are analyzed.

2. GOVERNING EQUATIONS

Compressible flows are usually modeled by the governing principles of mass, momentum, and energy conservation. Based on these assumptions one derives the system of compressible Navier-Stokes equations. At high speeds the impact of viscosity is rather small and the Euler equations are a suitable approximation. Neglecting viscous terms, heat sources, and body forces we obtain the compressible Euler equations

$$\partial_t \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I} P \\ \mathbf{v}(\rho E + P) \end{bmatrix} = 0, \tag{1}$$

where ρ, $\mathbf{v} = (u, v)^T$, E, P are the density, velocity, total energy, and pressure of the fluid. We close the system of Euler equations by the ideal gas law and add the equation of state

$$P = (\gamma - 1)\rho \left(E - \frac{|\mathbf{v}|^2}{2} \right), \qquad \gamma = 1.4. \tag{2}$$

To simplify notation we abbreviate (1) by

$$\partial_t U + \nabla \cdot \mathbf{F} = 0 \tag{3}$$

with the vector of conservative variables U and the flux tensor

$$\mathbf{F} = (F^{(x)}, F^{(y)})^T. \tag{4}$$

System (3) is in conservative form. For our purposes, the quasi-linear form may also be of interest. Applying the chain rule one transforms the Euler equations to

$$\partial_t U + \mathbf{A} \cdot \nabla U = 0 \tag{5}$$

with the Jacobian tensor

$$\mathbf{A} = \left(\frac{\partial F^{(x)}}{\partial U}, \frac{\partial F^{(x)}}{\partial U} \right). \tag{6}$$

Both Jacobians $\frac{\partial F^{(x)}}{\partial U}$ and $\frac{\partial F^{(y)}}{\partial U}$ are diagonalizable with a complete set of eigenvectors, which makes the Euler equations hyperbolic. The eigenvectors, eigenvalues and Jacobians can be found in [29].

DOI: 10.1002/fld

3. DISCRETIZATION

The construction of the high-resolution scheme involves three steps [18]. First, system (3) is discretized in space by the standard Galerkin method, which serves as a high-order scheme. Since the Galerkin discretization is oscillatory and unstable, we add a suitably defined artificial diffusion operator D to preserve the physical properties of the solution. The error induced by this manipulation is proportional to the mesh size, which reduces the order of approximation to one. For this reason the resulting discretization is called low-order scheme. Finally, we increase the order of approximation by reinserting a limited fraction of nonlinear antidiffusion F^*, which results in the desired high-resolution scheme. The amount of antidiffusion is controlled by TVD-type flux limiters based on the local smoothness of the solution. This road map is closely related to the scalar case [16, 18].

3.1. High-Order Scheme

In this paper, we discretize (3) in space using continuous finite elements. Let $\{\varphi_i : i = 1, \ldots, NVT\}$ be a set of piecewise linear or bilinear basis functions associated with the vertices of the mesh. The group finite element approximation

$$U_h = \sum_j U_j \varphi_j, \quad \mathbf{F}_h = \sum_j \mathbf{F}_j \varphi_j$$

yields a system of differential-algebraic equations for the nodal values of U_h [19]

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x} \right] \frac{dU_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x} \right] \cdot \mathbf{F}_j = 0 \quad \forall i. \tag{7}$$

The above Galerkin discretization is second-order accurate and globally conservative. It can be written as

$$\sum_j m_{ij} \frac{dU_j}{dt} = -\sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j \tag{8}$$

or in the compact matrix form

$$M_C \frac{dU}{dt} = KU \tag{9}$$

with the discrete transport operator

$$K = \{K_{ij}\}, \qquad K_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{A}_j = {K_{ij}}^{(x)} + {K_{ij}}^{(y)} \tag{10}$$

due to the homogeneity property [7]. The coefficients of this finite element scheme are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, d\mathbf{x} \qquad \text{and} \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}. \tag{11}$$

The $NVT \times NVT$-matrix

$$M_C = \{M_{ij}\} = \{m_{ij} I\} \tag{12}$$

($NVT =$ number of vertices) denotes the block consistent mass matrix , where I is the 4×4 identity matrix. Since the basis functions sum to unity at every point of the domain one easily verifies

$$\sum_j \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \left(\sum_j \varphi_j \right) d\mathbf{x} = 0. \tag{13}$$

This transforms the semi-discretized Euler equations (8) equivalently into the flux difference formulation [19]

$$\sum_j M_{ij} \frac{dU_j}{dt} = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i). \tag{14}$$

DOI: 10.1002/fld

To impose boundary conditions in a weak sense, we integrate equation (8) by parts, which gives rise to a surface integral in the weak formulation

$$\sum_j M_{ij}\frac{dU_j}{dt} = \sum_j \mathbf{c}_{ji}\cdot\mathbf{F}_j - \int_{\partial\Omega}\varphi_i\mathbf{F}_h\cdot\mathbf{n}\,ds, \tag{15}$$

where $\mathbf{n}$ is the unit outward normal to the boundary $\partial\Omega$ of the computational domain Ω. The matrix form of the semi-discrete problem remains unchanged. The boundary integrals allow the application of weak Neumann-type flux boundary conditions, which are investigated in this paper. Without taking the boundary conditions into account equation (15) is equivalent to (8) and (14). On the other hand, this weak formulation is improper for computing the stabilization since it does not allow the decomposition into suitable flux differences. Consequently, it is preferable to derive the stabilization operator for (14) and apply it to the high-order discretization (15).

3.2. Algebraic Flux Correction of TVD type

The standard Galerkin method is known to violate the discrete maximum principle whenever small scale features cannot be resolved properly on a given mesh. To prevent the formation of spurious oscillations and numerical instabilities, we constrain (8) using algebraic flux correction of TVD type. The result is a nonlinear blend of high- and low-order approximations. Flux limiters are designed for scalar equations and they are not directly applicable to systems of hyperbolic equations. To constrain the high-order discretization of hyperbolic systems, the equations must be decoupled by a diagonalization of the Jacobians, which is only locally possible in the nonlinear case.

The numerical solution is free of oscillations provided the numerical scheme is local extremum diminishing (LED):

Theorem 3.1 (Characteristic LED criterion [19])
The discretization of a linear hyperbolic system of conservation laws of the form

$$M_i\frac{dU_i}{dt} = \sum_{j\neq i} L_{ij}(U_j - U_i) \tag{16}$$

is local extremum diminishing for a suitable set of characteristic variables, provided the off-diagonal blocks L_{ij} are positive semi-definite.

It was discovered by Roe [28], that the flux differences in formula (14)

$$\begin{aligned}(KU)_i &= -\sum_{j\neq i}\mathbf{c}_{ij}\cdot(\mathbf{F}_j-\mathbf{F}_i)\\ &= -\sum_{j\neq i}\left[c_{ij}^{(x)}(F_j^{(x)}-F_i^{(x)})+c_{ij}^{(y)}(F_j^{(y)}-F_i^{(y)})\right]\end{aligned} \tag{17}$$

can be linearized exactly in terms of the uni-directional Roe matrices ${A_{ij}}^{(x)}$ and ${A_{ij}}^{(y)}$ satisfying

$${F_j}^{(x)} - {F_i}^{(x)} = {A_{ij}}^{(x)}(U_j - U_i) \quad \text{and} \quad {F_j}^{(y)} - {F_i}^{(y)} = {A_{ij}}^{(y)}(U_j - U_i). \tag{18}$$

Substitution of (18) into (17) yields the desired locally linearized formulation

$$(KU)_i = -\sum_{j\neq i}\left[c_{ij}^{(x)}A_{ij}^{(x)} + c_{ij}^{(y)}A_{ij}^{(y)}\right](U_j - U_i), \tag{19}$$

where the matrix $c_{ij}^{(x)}A_{ij}^{(x)} + c_{ij}^{(y)}A_{ij}^{(y)}$ is called the cumulative Roe matrix [19].

DOI: 10.1002/fld

3.2.1. Low-Order Scheme The above formulation is of the form (16). It is local extremum diminishing for a set of suitably defined characteristic variables if the matrix $-\left(c_{ij}^{(x)}A_{ij}^{(x)}+c_{ij}^{(y)}A_{ij}^{(y)}\right)$ does not contain negative eigenvalues according to theorem 3.1. Kuzmin and Möller [19] applied theorem 3.1 to the Euler equations and enforced the LED criterion by eliminating negative eigenvalues on the algebraic level by an addition of a suitably defined diffusion tensor. This strategy, which is free of user-defined parameters, is applicable to arbitrary dimensions and unstructured meshes. It is therefore adopted in the present work. Since the definition of characteristic variables requires a dimensional splitting, which was introduced by Yee et al. [37], the LED criterion is imposed on the Roe matrices for both coordinate directions separately, i. e. $-c_{ij}^{(x)}A_{ij}^{(x)}$ and $-c_{ij}^{(y)}A_{ij}^{(y)}$.

Due to the hyperbolicity of the governing equations, the uni-directional Roe matrices of every edge $\{ij\}$ admit a diagonalization with real eigenvalues

$$A_{ij}^{(d)}=R_{ij}^{(d)}\Lambda_{ij}^{(d)}{R_{ij}^{(d)}}^{-1},\qquad d=x,y, \tag{20}$$

where ${R_{ij}^{(d)}}^{-1}$, $R_{ij}^{(d)}$ are the matrices of left and right eigenvectors and

$$\Lambda_{ij}^{(d)}=diag\{\lambda_1^{(d)},\dots,\lambda_4^{(d)}\}_{ij} \tag{21}$$

is a diagonal matrix of the eigenvalues.

The Galerkin coefficients (11) satisfy

$$\mathbf{c}_{ij}=-\mathbf{c}_{ji}+\int_{\partial\Omega}\varphi_i\varphi_j\,\mathbf{n}\,ds \tag{22}$$

and can be decomposed into a skew-symmetric and a symmetric part

$$\mathbf{c}_{ij}=\mathbf{a}_{ij}+\mathbf{b}_{ij},\quad \mathbf{a}_{ij}=\frac{\mathbf{c}_{ij}-\mathbf{c}_{ji}}{2}=-\mathbf{a}_{ji},\quad \mathbf{b}_{ij}=\frac{\mathbf{c}_{ij}+\mathbf{c}_{ji}}{2}=\mathbf{b}_{ji}. \tag{23}$$

For the design of the diffusion tensor, the symmetric part corresponding to the boundary integral is neglected. One defines the 4×4 diffusion blocks D_{ij} in terms of their uni-directional counterparts $D_{ij}^{(d)}$

$$D_{ij}=D_{ij}^{(x)}+D_{ij}^{(y)},\qquad D_{ij}^{(d)}=|a_{ij}^{(d)}||A_{ij}^{(d)}|,\qquad |A_{ij}^{(d)}|=R_{ij}^{(d)}|\Lambda_{ij}^{(d)}|{R_{ij}^{(d)}}^{-1} \tag{24}$$

for the space dimension d, to eliminate negative eigenvalues from the discrete transport operator. To ensure conservation, the diffusion tensor has to feature block-symmetry. This is enforced by

$$D_{ji}^{(d)}=-D_{ij}^{(d)}\quad\forall j\neq i \tag{25}$$

and the diffusion tensor has vanishing row and column sums. The application of the discrete diffusion operator to node i results in the low-order operator

$$(LU)_i=\sum_{j\neq i}L_{ij}(U_j-U_i)=\sum_{j\neq i}(K_{ij}+D_{ij})(U_j-U_i), \tag{26}$$

which has positive semi-definite off-diagonal blocks L_{ij} [9].

The consistent mass matrix may also generate undershoots and overshoots and affect the LED criterion due to non-zero off-diagonal blocks. Therefore, it is replaced by its lumped counterpart

$$M_L=diag\{M_i\},\quad M_i=m_iI,\quad\text{and}\quad m_i=\sum_j m_{ij}. \tag{27}$$

DOI: 10.1002/fld

Note that in the stationary case mass lumping does not affect the accuracy since the time derivative vanishes in the steady-state limit. It improves the matrix properties and the performance of iterative linear solvers.

Finally, the low order scheme is defined by

$$M_L \frac{dU}{dt} = LU, \qquad L = K + D. \tag{28}$$

3.2.2. High-Resolution Scheme The above low-order scheme does not incorporate information about the local smoothness of the solution. Since there is no need for adding artificial viscosity in smooth flow regions, a limited antidiffusive correction F^* is applied to the modified Galerkin operator. By construction, this scheme is nonoscillatory and more accurate than the low-order scheme. On the other hand, the antidiffusive part F^* is highly nonlinear, and the local order of approximation depends on the properties of the unknown solution.

In the present paper equations (14) with equivalent linearization (19) are transformed to a set of characteristic variables (29) on every edge and for every space dimension separately as suggested in [19]. In the case of the multidimensional nonlinear Euler equations, characteristic variables are defined in terms of characteristic differences for the locally linearized equations (19) [19]

$$\Delta W_{ij}^{(d)} = {R_{ij}^{(d)}}^{-1}(U_i - U_j). \tag{29}$$

The goal of this section is to define an artificial antidiffusion operator F^* on the algebraic level using a nodal TVD type flux limiter. The application of the operator F^* results in the high-resolution scheme

$$M_L \frac{dU}{dt} = K^*U = LU + F^*U. \tag{30}$$

The algorithmic steps are as follows:

1. Remove all antidiffusive fluxes from the high-order Galerkin discretization by applying the numerical diffusion tensor.
2. Compute the sums of positive/negative antidiffusive/diffusive fluxes to individual nodes in terms of characteristic variables.
3. Compute correction factors in characteristic variables.
4. Add the limited antidiffusion in conservative variables.

Since the construction of the low-order operator was described in section 3.2.1, it remains to derive the computation of the nodal correction factors, which control the limited antidiffusive fluxes applied to the low-order scheme. We may initialize the high-resolution operator by the low-order one. The transformation to characteristic variables is performed to determine how much artificial diffusion may be safely removed, while the artificial limited antidiffusion is inserted into the residual in conservative variables. The proposed algorithm for the computation of the antidiffusive correction is based on [19, 17, 9] and has to be applied for each space dimension d:

- In a loop over edges $\{ij\}$:
  1. Compute the Roe averaged eigenvalues and matrices of left and right eigenvectors of the uni-directional Roe matrices.
  2. Transform the solution differences on the edge to the characteristic variables.
  3. Determine the upwind node I and the downwind node J for each characteristic field k:
     - If $a_{ij}^{(d)} \lambda_k^{(d)} \geq 0$:

$$I = i, \qquad J = j, \qquad \Delta W_{IJ}^{(k)} = \Delta W_{ij}^{(k)}. \tag{31}$$

DOI: 10.1002/fld

– If $a_{ij}^{(d)}\lambda_k^{(d)} < 0$:

$$I = j, \qquad J = i, \qquad \Delta W_{IJ}^{(k)} = -\Delta W_{ij}^{(k)}. \tag{32}$$

4. Compute the raw diffusive/antidiffusive fluxes on the current edge

$$F_{IJ}^{(k)} = |a_{ij}^{(d)}\lambda_k^{(d)}|\Delta W_{IJ}^{(k)}. \tag{33}$$

5. For each characteristic field k update the sums of positive and negative edge contributions

$$P_{I,k}^{+} = P_{I,k}^{+} + \max\{0, F_{IJ}^{(k)}\} \qquad P_{I,k}^{-} = P_{I,k}^{-} + \min\{0, F_{IJ}^{(k)}\}.$$

Reverse the sign of the flux and add it to the upper/lower bounds

$$Q_{I,k}^{+} = Q_{I,k}^{+} + \max\{0, -F_{IJ}^{(k)}\} \qquad Q_{J,k}^{+} = Q_{J,k}^{+} + \max\{0, F_{IJ}^{(k)}\}$$
$$Q_{I,k}^{-} = Q_{I,k}^{-} + \min\{0, -F_{IJ}^{(k)}\} \qquad Q_{J,k}^{-} = Q_{J,k}^{-} + \min\{0, F_{IJ}^{(k)}\}.$$

- In a loop over nodes:
  1. Evaluate the nodal correction factors

$$R_{i,k}^{\pm} = \min\{1, \frac{Q_{i,k}^{\pm}}{P_{i,k}^{\pm}}\}. \tag{34}$$

- In a loop over edges $\{ij\}$:
  1. Repeat the first four steps of the first loop over edges.
  2. Compute the limited amount of antidiffusion for each characteristic direction k:
     – Determine the correction factors

$$\alpha_{ij}^{(k)} = \begin{cases} R_{I,k}^{+} & \text{if} \quad F_{IJ}^{(k)} \geq 0 \\ R_{I,k}^{-} & \text{if} \quad F_{IJ}^{(k)} < 0 \end{cases}. \tag{35}$$

     – Apply the correction factors

$$F_{IJ}^{(k)} = \alpha_{ij}^{(k)} F_{IJ}^{(k)}. \tag{36}$$

  3. Transform back to conservative variables

$$F_{IJ} = R_{ij} F_{IJ}. \tag{37}$$

  4. Add the antidiffusive fluxes into the upwind and downwind nodes

$$(K^*U)_I = (K^*U)_I + F_{IJ}, \qquad (K^*U)_J = (K^*U)_J - F_{IJ}. \tag{38}$$

Obviously, the whole stabilization is free of parameters and can be generalized to arbitrary hyperbolic systems of conservation laws which allow for the determination of Roe averages. If this is not the case, the Roe matrices can be replaced by other types of averages, e. g. arithmetic averages. One also obtains a conservative scheme due to the construction of the diffusion and antidiffusion tensors.

4. BOUNDARY CONDITIONS

The implementation of boundary conditions can be generally divided into the strong enforcement and its weak counterpart. In a strong implementation, the boundary conditions are prescribed in a

DOI: 10.1002/fld

nodal sense on the algebraic level. Assume that the discretization yields a linear algebraic system of the form

$$Au = b \tag{39}$$

in each nonlinear iteration step. If a node i belongs to the boundary and a Dirichlet boundary condition $u_i = g_i$ has to be prescribed, the off-diagonal entries of the i-th row of the matrix A are deleted and the diagonal one is replaced by one. Finally, one overwrites the i-th entry of the right hand side b_i by g_i. Obviously, the solution of the linear system satisfies $u_i = g_i$. On the other hand, such ad-hoc changes of matrix entries influence the internodal mass exchange in the vicinity of the boundary and result in a deterioration of matrix properties and convergence rates or even divergence particularly in steady-state computations. Moreover, time step restrictions arise due to the explicit nature of that strategy. The explicitness is caused mainly by the deletion of off-diagonal matrix entries.

In the case of the Euler equations g_i typically depends on the solution, which requires a predictor-corrector algorithm in an implicit formulation [19, 25]. However, the predictor-corrector algorithm is semi-explicit as well. Trépanier et al. [35] prescribe fully implicit boundary conditions in the finite volume framework, where the boundary conditions are enforced by additional equations superimposed on the algebraic system, and avoid the explicitness arising by the deletion of matrix entries. The derivation and implementation of such boundary conditions is complicated and the authors observe a deterioration of convergence rates if the CFL number exceeds some upper bound. This upper bound may be problem-dependent and need adjusting, which is an unfavorable property and may be a consequence of the strong nature of boundary conditions.

In contrast to finite volumes, finite element methods are node-based schemes and fluxes over the boundaries of the elements are not directly useful. Although it is possible to modify the boundary fluxes of the strong Galerkin formulation (7) by the imposed boundary conditions, it is more flexible and robust to simply overwrite the boundary integral of the weak Galerkin formulation (15) in the Neumann-sense:

$$\int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \mathbf{F}_h \, ds \qquad \longrightarrow \qquad \int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds \tag{40}$$

We sometimes observed numerical problems at boundary states in terms of negative pressures and densities, if the boundary fluxes in the boundary integrals were directly overwritten by the the characteristic boundary conditions. To prevent such unphysical phenomena in the developed code, a Riemann problem is solved, either by the approximate Riemann solver of Roe or the exact Riemann solver of Toro [34], which is applicable to solid wall boundary conditions. For a strong enforcement of boundary conditions it was observed in [34, 25] that Toro's exact Riemann solver may offer the opportunity of convergence at very low CFL numbers and yields accurate results, even when other methods fail.

4.1. Ghost Nodes

Since the Euler equations may admit waves leaving and entering the domain in the normal direction simultaneously it is preferable to solve the Riemann problem on the boundary instead of simply overwriting the boundary fluxes. In order to define a Riemann problem on the boundary a ghost node is added. Note that this ghost node is only used to solve the boundary Riemann problem, while no additional variables are introduced to the nonlinear system.

The context of ghost nodes can be interpreted as the finite element counterpart of ghost cells, which was quite successfully employed in the finite volume framework [22]. Typically in finite volume schemes a virtual cell is assumed on the outer side of the boundary and its state is defined so as to impose the boundary condition. In this work it is proposed to introduce a virtual node or ghost node corresponding to each point located at the boundary, where a boundary condition has to be

DOI: 10.1002/fld

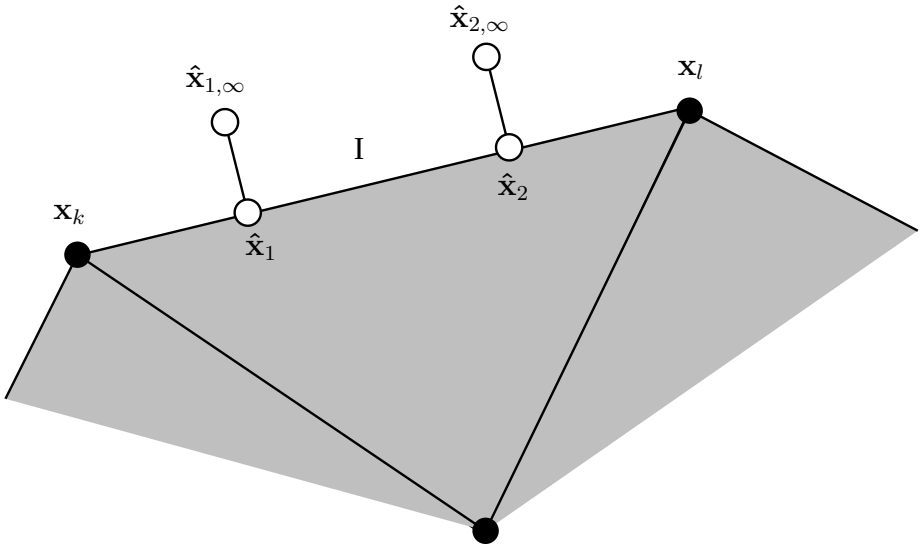


Figure 1. Quadrature points and ghost nodes

imposed. It is placed in the normal direction on the outer side of the boundary. The corresponding ghost state is supposed to incorporate the boundary condition and it is defined in such a way that unphysical wave reflections at the boundary are avoided. Last but not least, the concept of ghost nodes is independent of the finite element space. For convenience the ghost node will be denoted by $\mathbf{x}_\infty$ and the point at the boundary, which is treated as an interior one, by $\mathbf{x}_i$. The corresponding states are referred to as U_∞ and U_i, respectively. Note that the framework of ghost nodes applies to arbitrary points at the boundary including quadrature points or boundary nodes. In this work a ghost node is typically associated with the quadrature points of the boundary integrals. Figure 1 illustrates the setting for a quadrature rule incorporating two quadrature points.

4.2. Riemann Invariants

To account for incoming and outgoing information, boundary conditions should be imposed only on the incoming waves to leave the boundary transparent to outgoing information. Therefore, it is recommended to prescribe boundary conditions in terms of the Riemann invariants [30]

$$W = \{W_1, W_2, W_3, W_4\}, \tag{41}$$

which are given by

$$W_1 = v_n - \frac{2c}{\gamma - 1}, \quad W_2 = c_v \log\left(\frac{P}{\rho^\gamma}\right), \quad W_3 = v_\tau, \qquad W_4 = v_n + \frac{2c}{\gamma - 1}, \tag{42}$$

for the corresponding eigenvalues

$$\lambda_1 = v_n - c, \qquad \lambda_2 = \lambda_3 = v_n, \qquad \lambda_4 = v_n + c \tag{43}$$

of the projected Jacobian tensor $\mathbf{n} \cdot \mathbf{A}$. The tangential and normal velocities are

$$v_n = \mathbf{n} \cdot \mathbf{v}, \qquad \text{and} \qquad v_\tau = \boldsymbol{\tau} \cdot \mathbf{v}, \tag{44}$$

respectively. For the sake of simplicity and without loss of generality W_2 is replaced by [30]

$$W_2 = \frac{P}{\rho^\gamma} \tag{45}$$

DOI: 10.1002/fld

in the context of boundary conditions. A decision based on the sign of the corresponding eigenvalue of each characteristic variable is made to prescribe boundary conditions. If the eigenvalue is positive, then the wave leaves the domain and the Riemann invariant remains unchanged. Otherwise, in the case of a negative eigenvalue, a boundary condition has to be imposed on the wave.

4.3. Inflow and Outflow Boundary conditions

Inlet and outlet boundary conditions are often prescribed based on the free stream conditions. It is also possible to fix other variables, like the pressure at a subsonic outlet or density, tangential velocity, and pressure at a subsonic inlet. The reader is referred to [10] for a detailed analysis.

4.3.1. Supersonic Inflow and Outflow Boundary Conditions A supersonic inflow/outflow is characterized by $|v_n| > c$. If additionally $v_n > 0$, all eigenvalues are positive, or in other words all waves leave the domain. Therefore, no boundary condition is needed. This corresponds to a supersonic outlet and is the simplest case of boundary condition. Otherwise, in the case of $v_n < 0$, all eigenvalues have negative sign and all waves enter the domain, which characterizes a supersonic inlet. Hence, boundary conditions have to be imposed on all waves. Alternatively in this case one may prescribe boundary conditions in terms of conservative variables.

4.3.2. Subsonic Inflow and Outflow Boundary Conditions If $|v_n| < c$, the flow is of subsonic nature. Consequently, either the first three eigenvalues are negative ($-c < v_n < 0$) and the fourth is positive, which corresponds to a subsonic inlet, or the first eigenvalue ($0 < v_n < c$) is the only one that has a negative sign. The latter case is related to a subsonic outlet and the former corresponds to a subsonic inlet. In the case of a subsonic inlet the first three Riemann invariants must be prescribed, while at a subsonic outlet only the first Riemann invariant has to be imposed due to the negative sign of the corresponding eigenvalue. Nevertheless, in both cases the flow behavior depends on both the incoming and outgoing characteristics and it is therefore impossible to prescribe boundary conditions in terms of conservative variables.

4.3.3. Free Stream Boundary Conditions Subsonic boundary conditions are much more complicated then their supersonic counterparts since one has to compute the boundary flux that depends on four variables. However, there are less than four Riemann invariants defined by the boundary conditions. The remaining Riemann invariants correspond to outgoing waves and must be extracted from the solution.

Free stream boundary conditions can be implemented as follows [30, 9]. Assume, that a set of free stream invariants

$$W_\infty = \{W_{\infty,1}, W_{\infty,2}, W_{\infty,3}, W_{\infty,4}\} \tag{46}$$

is specified. In the case of subsonic normal flow conditions, the boundary values depend on the free stream invariants as well as the outgoing Riemann invariants given by the current solution. Let the superscript * denote the state taken from the solver that is the current solution without taking into account the boundary conditions. The superscript ** refers to the final state. Free stream boundary conditions can be computed from the following steps:

1. Compute the Riemann invariants of the current solution

$$W^* = \{W_1^*, W_2^*, W_3^*, W_4^*\}. \tag{47}$$

2. Define the final set of invariants W^{**} either by the predicted invariants or the free stream ones, depending on the signs of the corresponding eigenvalues λ_k

$$W_k^{**} = \begin{cases} W_k^* & \text{if} \quad \lambda_k \geq 0 \\ W_{\infty,k} & \text{if} \quad \lambda_k < 0 \end{cases} \tag{48}$$

for each characteristic field k.

Prepared using fldauth.cls
DOI: 10.1002/fld

3. Transform W^{**} back to conservative variables:
   The speed of sound
$$c^{**} = \frac{\gamma - 1}{4}(W_4^{**} - W_1^{**}) \tag{49}$$
   is given in terms of the difference of the first and the fourth Riemann invariant, and hence the primitive variables pressure, density, and velocity are determined by
$$\rho^{**} = \left(\frac{c^{**2}}{\gamma W_2^{**}}\right)^{\frac{1}{\gamma-1}} \tag{50}$$
$$P^{**} = \frac{c^{**2}\rho^{**}}{\gamma} \tag{51}$$
$$\mathbf{v}^{**} = \frac{W_1^{**} + W_4^{**}}{2}\,\mathbf{n} + W_3^{**}\boldsymbol{\tau}. \tag{52}$$
   A few algebraic manipulations transform the boundary state to the final conservative variables
$$\rho^{**} = \left(\frac{c^{**2}}{\gamma W_2^{**}}\right)^{\frac{1}{\gamma-1}} \tag{53}$$
$$(\rho\mathbf{v})^{**} = \rho^{**}\left[\frac{W_1^{**} + W_4^{**}}{2}\,\mathbf{n} + W_3^{**}\boldsymbol{\tau}\right] \tag{54}$$
$$(\rho E)^{**} = \frac{P^{**}}{\gamma - 1} + \rho\frac{|\mathbf{v}^{**}|^2}{2}. \tag{55}$$

4. Prescribe boundary conditions.

In a practical implementation one may prescribe free stream invariants at each quadrature point of the inlets and outlets a priori in a preprocessing step. Then the algorithm computes the characteristic direction and determines the desired state automatically during the simulation for the subsonic and supersonic inlets/outlets. Such kind of boundary conditions is transparent to outgoing waves since the algorithm uses the solution-dependent invariants for outgoing waves instead of the a priory defined ones, which are taken to specify incoming waves.

Note that it is also possible and sometimes more physical to prescribe primitive variables at subsonic boundaries [10] instead of free stream values. In the case of a subsonic outlet, the outlet pressure may be prescribed, while at a subsonic inlet e. g. density, tangential velocity, and pressure may be fixed [9].

4.3.4. Pressure Outlet Due to physical reasons, it may be preferable to prescribe the pressure P_{out} at a subsonic outlet instead of the first Riemann invariant. In this case boundary conditions have to be imposed on the first Riemann invariant only. The exit pressure P_{out} can be fixed by replacing the first Riemann invariant by [30]
$$W_{\infty,1} = -\frac{4}{\gamma - 1}\sqrt{\frac{\gamma P_{out}}{\rho^*}\left(\frac{P^*}{P_{out}}\right)^{\frac{1}{\gamma}}} + W_4^* \tag{56}$$
in step two of the algorithm in section 4.3.3.

4.4. Wall Boundary Conditions

In an inviscid flow mass and energy fluxes through the boundary are zero, while the fluxes parallel to the wall are not affected. In other words, the normal velocity vanishes
$$\mathbf{n} \cdot \mathbf{v} = 0. \tag{57}$$

DOI: 10.1002/fld

Since the solid wall boundary condition prevents the flow from penetrating through the wall, it is called free-slip or no-penetration condition. This is different from viscous flow dynamics, where a no-slip condition with $\mathbf{v} = 0$ is satisfied due to the viscous stresses. The implementation of solid wall boundary conditions is a challenging task in computational fluid dynamics and carried out in a weak sense in this study. In [1] Balakrishnan and Fernandez compare different types of solid wall boundary implementations. It turns out that strong, weak and mixed implementations yield almost the same results.

The easiest way to constrain the normal velocity to zero, is to overwrite the flux in the boundary integral by

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_h \Big|_{wall} = \begin{bmatrix} 0 \\ n^{(x)} P \\ n^{(y)} P \\ 0 \end{bmatrix}. \tag{58}$$

This treatment does not involve the solution of a Riemann problem and is refered to as zero-flux condition. Moreover, in some cases it provides increased robustness in comparison with the more physical mirror condition to be described below. Nevertheless, this boundary treatment was successfully applied e. g. in [32, 3, 1].

A more physical constraint is the so-called mirror or reflection condition since it mimics a continuous model of the ideal collision of a spherical particle with a solid wall. Moreover, this condition requires an additional ghost node (compare section 4.1) and enables the application of a Riemann solver at the boundary, which takes into account the characteristic direction of the flow. It yields very accurate results [34, 25, 9] and is consistent with the interior discretization and the inlet/outlet boundary conditions. To implement the mirror condition the normal velocity is reflected at the boundary, while the tangential velocity, density and pressure or total energy remain unchanged. Adopting the notation introduced in section 4.1 the ghost state is defined by

$$\rho_\infty = \rho_i, \qquad v_{n\infty} = -v_{ni}, \qquad v_{\tau\infty} = v_{\tau i}, \qquad E_\infty = E_i. \tag{59}$$

This can be transformed to conservative variables

$$\begin{aligned} \rho_\infty &= \rho_i \\ (\rho \mathbf{v})_\infty &= \rho_\infty(-v_{ni}\mathbf{n} + v_{\tau i}\boldsymbol{\tau}) \\ (\rho E)_\infty &= (\rho E)_i. \end{aligned} \tag{60}$$

Note that the ghost state may also be defined in terms of the primitive variables

$$\rho_\infty = \rho_i, \qquad v_{n\infty} = -v_{ni}, \qquad v_{\tau\infty} = v_{\tau i}, \qquad P_\infty = P_i, \tag{61}$$

which yields almost the same solution. The results computed by the mirror condition and (58) are similar. It is shown in [9] that (58) sometimes provides increased robustness, while the mirror condition represents a slightly more physical boundary condition. The mirror condition is applied unless otherwise stated and both possibilities are compared in a numerical study. Note that other implementations of the wall boundary condition are possible , see e. g. [1, 19].

4.5. Solution of the Boundary Riemann Problem

In this study the Riemann problem is solved approximately by means of Roe's Riemann solver although other choices are also possible. This treatment of the boundary flux is consistent with the inner discretization and makes it possible to compute a Jacobian matrix, which is important for the computation of stationary solutions by implicit schemes. This issue is addressed later. For the wall boundary conditions an exact Riemann solver presented by Toro [34] will be employed for comparison. The Riemann solver takes into account the direction of information propagation

DOI: 10.1002/fld

and avoids artificial non-physical effects like negative pressures or densities in the vicinity of the boundary particularly at high CFL numbers. This sometimes prevents the algorithm from divergence in comparison with an ad-hoc specification of boundary conditions, where the flux associated with the ghost node is directly substituted into the boundary term. To employ the approximate Riemann solver, the flux formula of Roe

$$\mathbf{n}\cdot\tilde{\mathbf{F}}_h = F(U_i, U_\infty) = \frac{1}{2}\mathbf{n}\cdot(\mathbf{F}(U_i)+\mathbf{F}(U_\infty)) - \frac{1}{2}|A^n_{i\infty}|(U_\infty - U_i) \tag{62}$$

is simply substituted into the boundary integral, where

$$U_\infty = U^{**} \tag{63}$$

is defined by the algorithm in section 4.3.3. In the equation above,

$$A^n_{i\infty} = \mathbf{n}\cdot\mathbf{A}_{i\infty} \tag{64}$$

is the Jacobian tensor evaluated in the normal direction and

$$|A^n_{i\infty}| = R^n_{i\infty}|\Lambda^n_{i\infty}|\,[R^n_{i\infty}]^{-1} \tag{65}$$

corresponds to (24). Obviously, (62) is Roe's 1D flux formula projected onto the outer normal.

5. THE IMPLICIT SOLVER

It is common practice to solve the arising nonlinear systems by a (pseudo-) time stepping scheme and march the solution to steady-state. In this case, the time step can be interpreted as an underrelaxation, where an infinite time step corresponds to a direct solution of the stationary problem without underrelaxation.

In this study we employ the backward Euler scheme. To avoid computationally expensive nonlinear iterations, a linearized version is used.

5.1. Linearized Backward Euler Scheme

The spatially discretized Euler equations can be expressed in the condensed form

$$M_L\frac{dU}{dt} = F, \tag{66}$$

where U is the vector of conservative variables and F is the flux vector. This system of ordinary differential equations can be integrated in time by the backward Euler scheme

$$M_L\frac{U^{n+1}-U^n}{\Delta t} = F^{n+1} \tag{67}$$

with a time step of length Δt. The superscript n refers to time level n and the result is a nonlinear system of algebraic equations, which calls for nonlinear iterations in each (pseudo-) time step. Nonlinear iterations are computationally very expensive. To circumvent this problem, one assumes sufficient smoothness and linearizes the equations around the current solution U^n by a Taylor series expansion of the fluxes

$$F^{n+1} = F^n + \left(\frac{\partial F}{\partial U}\right)^n (U^{n+1}-U^n) + \mathcal{O}(\|U^{n+1}-U^n\|^2). \tag{68}$$

Substitution of equation (68) into the nonlinear equations (67) leads to a linear algebraic system

$$\left[\frac{M_L}{\Delta t} - \left(\frac{\partial F}{\partial U}\right)^n\right](U^{n+1}-U^n) = F^n. \tag{69}$$

DOI: 10.1002/fld

At first glance, this scheme seems to be conditionally stable since it is semi-implicit. On the other hand, there is strong numerical evidence that the semi-implicit scheme based on the backward Euler method is unconditionally stable if the initial values are sufficiently accurate. In the limit of infinite CFL numbers we recover Newton's method

$$-\left(\frac{\partial F}{\partial U}\right)^n (U^{n+1}-U^n) = F^n, \tag{70}$$

which is known to be second order convergent if the fluxes and source terms are sufficiently smooth. Since this is not the case here, we call the scheme Newton-like.

5.2. The Approximate Jacobian

The solution of the nonlinear algebraic system resulting from

$$F(U) = 0 \tag{71}$$

involves the computation of the Jacobian of the nonlinear numerical fluxes $F(U)$ with respect to the conservative variables U. This function lacks differentiability, which prevents Newton's scheme from achieving second order convergence and we call the scheme Newton-like to indicate this fact. There are some approaches to compute an approximation of the numerical flux Jacobian reported in the literature. The easiest option to compute such an approximation is finite differencing of first or second order [26, 27]

$$\frac{\partial F}{\partial U} \approx \frac{F(U+\varepsilon)-F(U)}{|\varepsilon|} \qquad \text{or} \qquad \frac{\partial F}{\partial U} \approx \frac{F(U+\varepsilon)-F(U-\varepsilon)}{2|\varepsilon|}. \tag{72}$$

The approximation can be further improved by extrapolation [25]. Finite differencing is a convenient way to differentiate the fluxes. On the other hand, the convergence behavior depends on the choice of the vector ε. There is an optimal value of ε, which differs for each problem and needs tuning to be determined by trial and error. This rules out the use of finite differencing in a general purpose code, which should be applicable more or less as a black box tool to a wide range of real life problems.

A more appropriate approach is to replace the non-differentiable functions of maximum, minimum and the absolute value by smooth approximations [24], which enables the analytic derivation of the Jacobian. On the other hand, the analytical determination of the Jacobian of Roe's flux formula is subject to a rather complicated algebra and programming. It can be found in [2]. Another approach to Jacobian approximation, which has been applied quite successfully, is automatic differentiation. It is available as commercial software and it is able to compute the derivative of arithmetic operations using the chain rule. However, it relies on a differentiable residual function. Therefore, the non-differentiable maximum and minimum as well as the absolute value have to be replaced by smooth approximations. The use of a limiter increases the stencil of the scheme and results in a considerable amount of additional fill-in compared to the Galerkin flux Jacobian [25] and may result in a deterioration of matrix properties.

In this paper a different approach compared to the ones reported above is used to approximate the Jacobian. The original Jacobian in (70) and (69) is replaced by an edge-based approximation of the low-order Jacobian, where the correction factors are neglected. The approximate Jacobian (or preconditioner) constructed in this way is free of additional problem-dependent parameters and enjoys several advantages in comparison with the previously discussed approximations. No additional fill-in is created since the low-order fluxes only depend on the direct neighbors of a current node. For scheme (69) the resultant matrix is of M-matrix type, which is related to positivity preservation of each iterate subject to a suitable time step (at least on the characteristic level). This is an important fact since it rules out unphysical effects, which may cause divergence (e. g. zero density). The low-order approximation therefore provides increased robustness and is subject to

DOI: 10.1002/fld

lower memory requirements. Moreover, its computation is much less expensive than that of the approximation of the full Jacobian.

The kind of Jacobian approximation proposed here, can be determined analytically by a derivation of the low-order fluxes under certain simplifying assumptions. The sum of fluxes into a node i, which is related to the i-th row of the residual vector, is given by

$$F_i^{low} = \sum_j \mathbf{c}_{ji} \cdot \mathbf{F}_j - \underbrace{\int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \mathbf{F}_h \, ds}_{=B_i} + \sum_{j \neq i} D_{ij}(U_j - U_i). \tag{73}$$

On the right hand side the first two terms represent the Galerkin discretization (15). The diffusion blocks D_{ij} are defined by equation (24). For the sake of simplicity the derivatives of D_{ij} with respect to the conservative variables are neglected. The first term of equation (73) involves the Galerkin coefficients $\mathbf{c}_{ji}$. They are independent of the conservative variables. Under the above simplification, the derivatives of the low-order nodal flux F_i^{low} with respect to the conservative variables at nodes i and j are given by

$$\frac{\partial F_i^{low}}{\partial U_j} = \mathbf{c}_{ji} \cdot \mathbf{A}_j - \frac{\partial B_i}{\partial U_j} + D_{ij}, \qquad \frac{\partial F_i^{low}}{\partial U_i} = \mathbf{c}_{ii} \cdot \mathbf{A}_i - \frac{\partial B_i}{\partial U_i} - D_{ij} \tag{74}$$

where $\mathbf{A}_j$ and $\mathbf{A}_i$ is the Jacobian tensor (6) evaluated at nodes j and i, respectively. The approximate Jacobian proposed so far is exact with respect to the Galerkin discretization if the derivative of the boundary part can be determined exactly.

The differentiation of the boundary integral B_i is complicated since the solution of the boundary Riemann problem is involved and the ghost state may even depend on its related interior counterpart. We refer the interested reader to [9], where a boundary flux Jacobian is derived by multiple applications of the chain rule.

6. NUMERICAL RESULTS

The last part of this paper is concerned with numerical studies of the presented methods. The accuracy is examined by grid convergence studies and the numerical performance at large and infinite CFL numbers is investigated. Special emphasis is laid on the analysis of the proposed boundary conditions. Further validation can be found in [9].

For the purpose of accuracy assessment we measure the relative L_2-error. The analytical solution to the test cases to be presented in this work is typically unknown. Consequently, the error analysis is based on a sufficiently accurate numerical reference solution. For the results below, the solutions computed on the finest grid serve as the reference solutions. This enables us to determine the empirical order of convergence [33]. Let the numerical solutions of either the low-order or the high-resolution scheme, associated with the mesh sizes h and $\frac{h}{2}$, be given by U_h respectively $U_{\frac{h}{2}}$. Assume that the error satisfies approximately

$$\|U_h - U_{ref}\|_2 \approx Ch^p, \qquad \|U_{\frac{h}{2}} - U_{ref}\|_2 \approx C\left(\frac{h}{2}\right)^p, \tag{75}$$

where U_{ref} is the reference solution and p indicates the order of convergence. Under these conditions the order of approximation can be determined by

$$p \approx \frac{\log\left(\frac{\|U_h - U_{ref}\|_2}{\|U_{\frac{h}{2}} - U_{ref}\|_2}\right)}{\log(2)}. \tag{76}$$

DOI: 10.1002/fld

In what follows the relative L_2-error is calculated by

$$E_2 = \frac{\|U_h - U_{ref}\|_2}{\|U_{ref}\|_2}. \tag{77}$$

6.1. 10% GAMM Channel

In this section the well known GAMM channel is investigated. Transonic flow at free stream Mach number $M_\infty = 0.67$ impinges on a 10% circular bump in a channel. The flow is reflected on the bump, accelerated, and a shock arises on the bump. In this test, the Mach number varies between approximately 0.22 and 1.41. Subsonic, transonic, and supersonic regions as well as the sonic point at Mach number $M = 1$ occur. Characteristic stiffness associated with the sonic point, the curved boundary of the bump, and the relatively wide range of Mach numbers make this test case a challenging task for the numerical scheme. Additionally, the subsonic nature of the inlet and outlet requires a suitable boundary treatment to achieve convergence to steady-state.

The unstructured triangular coarse mesh is illustrated in figure 2 (b). To derive higher mesh levels, the grid is globally refined by subdividing each edge into two equal parts. Mesh data for the different refinement levels is listed in table I, where NVT denotes the number of vertices and NEL refers to the number of elements involved. A numerical solution on mesh level six computed by the high-resolution scheme is presented in figure 2 (a), where the Mach number is illustrated. This serves as the reference solution for the grid convergence analysis reported below. It can be observed that the solution computed by the high-resolution scheme is well resolved and free of unphysical effects like numerical oscillations. All simulations by the high-resolution scheme are initialized by the solutions of the low-order problem, which provide sufficiently accurate linearizations to employ infinite CFL numbers. The free stream values serve as initial values for the low-order computations. During the low-order startup phase one employs a moderate CFL number of 100. After the relative residual falls to 10^{-2} the linearization is sufficiently accurate and the CFL number is increased to the desired level. Note that even $CFL = \infty$ is possible and divergence was never observed. In figure 3 nonlinear convergence histories for both the high-resolution scheme and the low-order scheme are depicted in logarithmic scale. The relative error is measured in the L_2-norm.

It follows from the numerical results that the nonlinear convergence rates of the semi-implicit pseudo time stepping scheme improve as the CFL number increases. In particular for the low-order scheme, an infinite CFL number results in the best convergence rates. This illustrates the performance of the proposed weak Neumann-type boundary conditions. After ten iterations the residual falls below 10^{-12}, while the solution approaches steady-state very slowly in the case of small CFL numbers. Small CFL numbers correspond to large underrelaxation, which is unnecessary for the linear low-order scheme. Moreover, an almost monotone decreasing error can be observed for sufficiently large CFL numbers.

A similar behavior of the high-resolution scheme is indicated by the convergence history, although more nonlinear iterations are required for convergence. When the CFL number is increased, the convergence rates improve until the threshold $CFL = 100$ is reached. At higher values of the CFL number, the convergence rates remain approximately the same. In contrast to [35] the rate of convergence does not deteriorate if the CFL number is too large. The high-resolution scheme is highly nonlinear and the correction factors lack differentiability. In fact, the correction factors oscillate. The lack of smothness of correction factors is a reason for the slower convergence as reported in [24]. Moreover, the employed preconditioner does not incorporate the correction factors. Convergence is nevertheless reached. Therefore, the performance of the developed boundary techniques can be better illustrated by the low-order computations. On the other hand, the additional computational effort is the price to pay for a superior solution. A preconditioner equipped with a differentiable limiter is likely to deliver better convergence rates [26, 24].

Last but not least let us focus our attention on the grid convergence analysis. Since, no analytical

DOI: 10.1002/fld

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	176	292	$5.47 \cdot 10^{-2}$	0.59	$3.05 \cdot 10^{-2}$	0.56
2	643	1168	$3.64 \cdot 10^{-2}$	0.64	$2.07 \cdot 10^{-2}$	1.04
3	2453	4672	$2.34 \cdot 10^{-2}$	0.59	$1.01 \cdot 10^{-2}$	0.99
4	9577	18688	$1.55 \cdot 10^{-2}$	0.61	$5.07 \cdot 10^{-3}$	1.45
5	37841	18688	$1.01 \cdot 10^{-2}$		$1.85 \cdot 10^{-3}$	
6	150433	299008				

Table I. GAMM channel: Mesh properties and error analysis

solution is known, a numerical counterpart is computed on mesh level six and serves as the reference solution. The relative L_2-error is shown in table I. The empirical order of convergence can be determined, if one assumes that the reference solution is exact. For the low-order scheme the empirical order of convergence is approximately 0.6. As expected, the high-resolution scheme converges faster. In this example, the order of approximation is about one. Note that the correction factors depend on the solution. Hence, the order of convergence of the high-resolution scheme is problem-dependent. For this particular problem a higher order of convergence is impossible due to the lack of smoothness.

Sokolichin compared several limiters for the linear 1D continuity equation discretized using finite volumes on a uniform mesh. He found out [33] that the order of convergence is less than or equal to one when the solution is discontinuous. In fact, in our results the numerical error of the reference solution may also play a role and the rate of convergence to the exact solution may be slightly worse. Nevertheless, grid convergence takes place, as shown by the results above.

6.2. NACA 0012 Airfoil

The second test case is the flow around an airfoil at free stream Mach number $M_\infty = 0.8$. The symmetric profile of the airfoil surface is given by the function [19]

$$f^{\pm}(x) = \pm 0.6 \left(0.2969\sqrt{x} - 0.126x - 0.3516x^2 - 0.1015x^4\right) \tag{78}$$

with $x \in [0, 1.00893]$. The computational domain consists of a circle with radius 10, which is centered at the tip of the airfoil and covered by an unstructured triangular grid. The coarse grid is illustrated in figure 4 (b). For the numerical solution an inclination angle of $\alpha = 1.25^\circ$ is prescribed and a hierarchy of four meshes is investigated. The flow takes place at Mach numbers ranging between about $M = 0.02$ at the tip and $M = 1.36$ on the top of the airfoil. This indicates a characteristic stiffness related to the low Mach number regime as well as the sonic point. Moreover, the solution contains a shock and there are curved boundaries. The mesh data is summarized in table II. Steady-state low-order solutions act as initial values for the high-order computations. We initialized the low-order solutions by the free-stream values and a few pre-iterations with $CFL = 10$ are performed before the simulation is switched to a higher CFL number.

Similar to the GAMM channel test case we examine the grid convergence and the related empirical order of convergence based on a reference solution, which is computed on mesh level four. A zoom of the reference solution computed by the high-resolution scheme is depicted in figure 4 (a). One observes that the Mach number contours compare well with the results reported in [4, 19, 26, 11]. Table II lists the grid convergence study for the current test case. We obtain convergence rates comparable to those of the GAMM channel test case. Convergence is of order 0.5 in the case of the low-order scheme. At the same time the high-resolution scheme exhibits faster convergence of order one.

The convergence behavior of the nonlinear iteration is also comparable to the GAMM channel test

DOI: 10.1002/fld

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	2577	4963	$4.08 \cdot 10^{-2}$	0.51	$1.68 \cdot 10^{-3}$	1.02
2	10117	19852	$2.86 \cdot 10^{-2}$	0.46	$8.27 \cdot 10^{-4}$	1.02
3	40086	79408	$2.08 \cdot 10^{-2}$		$2.68 \cdot 10^{-4}$	
4	159580	317632				

Table II. NACA airfoil: Mesh properties and error analysis

case (compare figure 5). The error of the low-order scheme falls below 10^{-15} in 20 iterations for an infinite CFL number and the rate of convergence deteriorates with decreasing CFL numbers. On the other hand, the relative error hardly decreases for small CFL numbers. The high-resolution scheme exhibits a qualitatively similar convergence history, although the number of iterations required for convergence is much larger due to the reasons pointed out above. After about 200 iterations the relative error falls below 10^{-8} and the solution can be considered to be stationary. There is an upper bound for the convergence rate. In other words, an increase of CFL numbers in the range above $CFL = 100$ yields only slightly better convergence rates. The convergence histories corresponding to $CFL = 100$, $CFL = 1000$, and $CFL = \infty$ are almost equal. In fact the relative error does not decrease significantly for $CFL = 1$ and $CFL = 10$. The error curves of both simulations are similar.

However, the very fast convergence of the low-order scheme gives evidence for the performance and robustness of the proposed boundary treatment for the semi-implicit FEM discretization. It is shown by the numerical results that a satisfactory error decay by several orders of magnitude can be achieved despite oscillating correction factors, provided sufficient care is taken in the implementation of the boundary conditions and of the implicit time stepping scheme. Moreover, it turns out that large CFL numbers are required for a satisfactory convergence to steady-state.

6.3. Converging-Diverging Nozzle Flow

Let us focus our attention on a converging-diverging nozzle. The upper and lower walls are defined by the function

$$g^{\pm}(x) = \begin{cases} \pm 1 & \text{if} \quad -2 \le x \le 0 \\ \pm \frac{\cos\left(\frac{\pi x}{2}\right)+3}{4} & \text{if} \quad 0 < x \le 4 \\ \pm 1 & \text{if} \quad 4 < x \le 8 \end{cases}, \tag{79}$$

which is adopted from [11]. A subsonic/supersonic flow with a shock is investigated. At the inflow boundary subsonic flow at free stream Mach number $M_\infty = 0.3$ is prescribed. For the purpose of comparison we prescribe the same free stream conditions as Hartmann [12]. That is, we impose a free stream pressure $P_\infty = 1$. At the outlet the free stream boundary conditions are replaced by a pressure outlet, which is presented in section 4.3.4. The prescribed pressure is $P_{out} = \frac{2}{3}$. This definition of boundary conditions yields subsonic flow of Mach number $M = 0.27$, which is accelerated in the diverging part of the nozzle up to Mach number $M = 2$. The flow suddenly decelerates at the end of the diverging part of the nozzle as it passes through a shock. The shock arises due to the pressure outlet boundary condition. Computational results obtained on mesh level seven are depicted in figure 6. They are almost indistinguishable from those obtained in [12]. A more detailed analysis of the boundary conditions concerning this test case can be found in section 6.4.

6.4. Numerical Verification of Weak Boundary Conditions

A numerical treatment of boundary conditions that provides enhanced stability and robustness is one of the main goals of this study. A proper implementation of boundary conditions is crucial for the overall accuracy of the scheme since the errors arising from an inaccurate boundary discretization

DOI: 10.1002/fld

Level	NVT	NEL	NVT_{out}	E_2^{out}	p^{out}
5	5313	5120	33	$2.50 \cdot 10^{-3}$	1.32
6	20865	20480	65	$1.00 \cdot 10^{-3}$	1.12
7	82689	81920	129	$4.62 \cdot 10^{-4}$	

Table III. Mesh properties and outlet boundary error analysis

propagate into the interior of the domain. This section is therefore concerned with the numerical analysis of the proposed boundary techniques. The accuracy and competitiveness of the weak treatment of solid wall boundary conditions as compared to its strong counterpart was already analyzed in the finite volume framework [1]. Our numerical results reported in the sections above indicate that the proposed weak treatment of boundary conditions is unconditionally stable. A comparison of the weak wall boundary conditions, proposed in section 4.4, is drawn in the sections below. Weakly imposed boundary conditions are satisfied in an integral and possibly not pointwise sense. Recall that finite element basis functions have small support. Therefore, a weak treatment enforces boundary conditions locally. Last but not least, a comparison of weak and strong boundary conditions is performed below.

6.4.1. Inlet and Outlet Boundary Conditions The accuracy of inlet and outlet boundary conditions is verified in this section. We examine the pressure outlet of the converging-diverging nozzle flow (compare section 6.3) as a representative test case of subsonic boundary conditions. The analysis of supersonic boundary conditions is reported in [9]. For the former test case, the prescribed outlet pressure is $P_{out} = \frac{2}{3}$. To verify the convergence of the outlet boundary condition, three mesh levels of bilinear elements are involved. Snapshots of the pressure at the outlet boundary for mesh levels five, six, and seven are reported in figure 7 (a). The relative error in the L_2-norm

$$E_2^{out} = \frac{\|P - P_{out}\|_{2,\Gamma_{out}}}{\|P_{out}\|_{2,\Gamma_{out}}} \tag{80}$$

as well as the empirical rate of convergence p^{out} are listed in table III, where NVT_{out} denotes the number of nodes at the outlet. Obviously, the boundary discretization error is very small, even on a relatively coarse grid. Similarly to the discretization error of the whole domain, the boundary discretization error vanishes with vanishing mesh size, which is important for grid convergence. To indicate the importance and influence of the outlet boundary condition on the whole flow, the pressure at the centerline of the nozzle is displayed in figure 7 (b).

6.4.2. Wall Boundary Conditions In section 4.4 two different weak treatments of solid walls are mentioned. On the one hand, one may enforce the free-slip condition by cancelling the normal velocity components (compare equation (58)), which turns out to be an easy but robust and accurate way. On the other hand, a more physical constraint in terms of the reflection or mirror condition may be imposed (compare equation (60)) and the related Riemann problem should be solved by either an approximate Riemann solver or its exact counterpart. These two basic approaches are compared for the GAMM channel test case already considered in section 6.1.

It is difficult to verify the free-slip condition on a curved boundary since the computational domain is an approximation of the physical one and the normal vector does not exist at the nodes. To circumvent this trouble it is possible to consider a wall boundary formed by a straight line. Therefore, the normal vector is defined at every point of the boundary. This simplifies the evaluation significantly. For the analysis we choose the upper wall boundary of the GAMM channel domain. Obviously, the normal velocity corresponding to the upper wall is simply the second velocity component. The characteristics of the involved mesh are reported in table I. The mesh levels four, five, and six are used. First of all, we compare the zero-flux condition (58) with the mirror condition. Snapshots of the normal velocity computed by both schemes are depicted in figure 8 (a).

DOI: 10.1002/fld

Level	NVT$_{out}$	E_2^{Lim}	p^{Lim}
4	145	$3.95 \cdot 10^{-4}$	0.99
5	289	$1.99 \cdot 10^{-4}$	1.00
6	577	$9.94 \cdot 10^{-5}$	

Table IV. Wall boundary error analysis

The mirror condition is embedded either into the approximate Riemann solver of Roe or the exact one of Toro [34]. It turns out that the mirror as well as the zero-flux solid wall boundary condition yield comparable results with very small errors, while the mirror condition gives rise to slightly better approximations. Moreover, the solutions computed by the approximate Riemann solver and the exact solver of Toro are almost the same. The difference between the exact and the approximate Riemann solvers are illustrated in figure 8 (b) in terms of $|v_n^{Toro} - v_n^{Roe}|$. In the following it remains to consider the mirror condition. A grid convergence analysis of the above configuration can be found in figure 8 (c) and the relative errors are listed in table IV. Obviously, the error in the normal velocity decreases with decreasing mesh size and the empirical order of convergence is about 1.0. This result is almost the same as the empirical rate of convergence associated with the whole domain, compare table I.

Summarizing the results, we conclude that the accuracy of the wall boundary treatment is comparable to that of the interior discretization, which is essential for the overall accuracy. The wall boundary discretizations, proposed so far, yield comparable results, while the mirror condition turns out to be slightly more accurate. At the same time it is much more complicated and sometimes slightly less robust than the zero-flux condition. Moreover, there are practically no differences between the approximate Riemann solver of Roe and the exact Riemann solver of Toro applied to the mirror condition. Of course one can identify and analyze the small errors and differences in the solutions computed by the different wall boundary conditions. In a practical application they are negligible.

6.5. Strong vs. Weak Boundary Conditions

Alternatively to the proposed weak Neumann-type boundary treatment, boundary conditions may be implemented in a strong sense. For the comparison to be drawn in this section the predictor-corrector algorithm described in [19, 25], which has proven to be an accurate and reliable technique in implicit schemes, is implemented in the semi-implicit time stepping scheme developed so far. The solid wall boundary condition is replaced by the Riemann solver of Toro to equalize the wall boundary treatment. Both codes are applied to the GAMM channel test case reported in section 6.1 and the computations are performed on mesh level four.

In figure 9 the convergence histories of both the weak and the strong type of boundary conditions are reported. The CFL number of the simulation, where weak boundary conditions are used, is $CFL = \infty$ since it was already shown that there are no time step restrictions. In contrast, the strong treatment of boundary conditions hampers convergence to steady-state. In fact, the scheme diverges for $CFL \geq 1$. In the reported computation, the CFL number is set to $CFL = 0.8$. It follows from the convergence histories of both boundary treatments depicted in figure 9 that the strong version is less competitive since the CFL number is bounded from above by one. Therefore, several thousand iterations are required to achieve convergence if there is any. The comparison of both types of boundary conditions is best suited for the low-order schemes (compare figure 9 (b)) since the nonlinearity of the limiter, which degrades the rate of convergence, is not involved in the solutions.

The computational observations can also be justified in a more mathematical way. While the weak type of boundary conditions can be incorporated into the matrix in a natural way, the strong enforcement of boundary conditions requires that off-diagonal matrix entries be eliminated. This is similar (but not completely equivalent) to dropping the Jacobian in (69) at boundary nodes. It

DOI: 10.1002/fld

makes the scheme explicit at boundary nodes and therefore gives rise to the well known stability restrictions. In a fully implicit code this does not completely hold true and the stability bound may be slightly larger since the solution is updated in an outer fixed point iteration loop. Nevertheless, the predictor-corrector algorithm remains semi-explicit. On the other hand, off-diagonal matrix entries also have to be deleted. The elimination of off-diagonal matrix entries as well as solution estimates, involved in the predictor-corrector algorithm, introduce some explicitness. Moreover, many iterations are required in each pseudo time step to compute the solution update, which makes the fully implicit approach computationally expensive. In summary, it is shown in this section that the weak imposition of characteristic boundary conditions yields accurate results and far superior stability and convergence compared to the strong form described in [19].

7. CONCLUSIONS AND OUTLOOK

A high-resolution finite element scheme was proposed for the compressible Euler equations. The main highlight is a robust iterative solver that delivers converged steady-state solutions despite oscillatory correction factors. The semi-implicit pseudo-time-stepping appears to be unconditionally stable and reduces to a Newton-like method in the limit of infinite CFL numbers. We replaced the original Jacobian by a low-order approximation. With increasing time steps the convergence rates improve and they do not deteriorate if the CFL number exceeds some upper bound. The Galerkin finite element discretization was equipped with weak boundary conditions of Neumann type based on a boundary Riemann solver. Their accuracy was demonstrated by numerical convergence studies. It was shown that the proposed boundary conditions provide improved robustness and stability compared to their strong counterparts. The high-resolution scheme based on a multidimensional flux limiter of TVD type was shown to converge twice as fast as its low-order counterpart, even in the case of discontinuous weak solutions.

The convergence to steady state can be further accelerated within the framework of a nonlinear (FAS-FMG) multigrid method. Further tasks to be accomplished include the implementation of the new algorithm in 3D, its combination with adaptive mesh refinement techniques, and application to real-life problems.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support by the German Research Foundation (DFG) within the framework of the Collaborative Research Center SFB 708.

8. CONCLUSIONS AND OUTLOOK

A high-resolution finite element scheme was proposed for the compressible Euler equations. The main highlight is a robust iterative solver that delivers converged steady-state solutions despite oscillatory correction factors. The semi-implicit pseudo-time-stepping appears to be unconditionally stable and reduces to a Newton-like method in the limit of infinite CFL numbers. We replaced the original Jacobian by a low-order approximation. With increasing time steps the convergence rates improve and they do not deteriorate if the CFL number exceeds some upper bound. The Galerkin finite element discretization was equipped with weak boundary conditions of Neumann type based on a boundary Riemann solver. Their accuracy was demonstrated by numerical convergence studies. It was shown that the proposed boundary conditions provide improved robustness and stability compared to their strong counterparts. The high-resolution scheme based on a multidimensional flux limiter of TVD type was shown to converge twice as fast as its low-order counterpart, even in the case of discontinuous weak solutions.

The convergence to steady state can be further accelerated within the framework of a nonlinear (FAS-FMG) multigrid method. Further tasks to be accomplished include the implementation of the new algorithm in 3D, its combination with adaptive mesh refinement techniques, and application to real-life problems.

DOI: 10.1002/fld

ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support by the German Research Foundation (DFG) within the framework of the Collaborative Research Center SFB 708.

REFERENCES

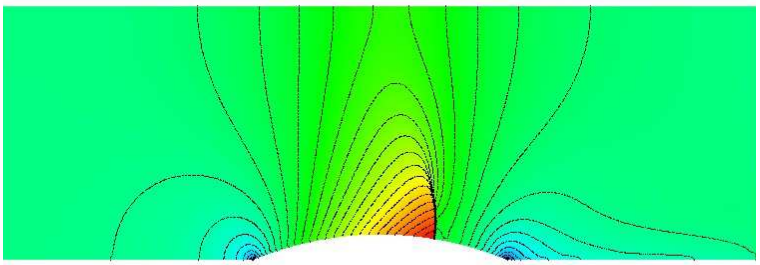
1. N. Balakrishnan, G. Fernandez: *Wall boundary conditions for inviscid compressible flows on unstructured meshes*, Int. J. Num. Meth. in Fluids, Vol. 28, 1998, pp. 1481-1501.
2. T. J. Barth: *Analysis of implicit local linearization techniques for upwind and TVD algorithms*, AIAA Paper, 1987, Article No. 87-0595.
3. V. Dolejší, M. Feistauer: *A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow*, J. Comp. Phys., Vol. 198, 2004, pp. 727-746.
4. M. Feistauer, J. Felcman, I. Straškraba: *Mathematical and computational methods for compressible flow*, Clarendon Press, Oxford, 2003.
5. M. Feistauer, V. Kučera: *On a robust discontinuous Galerkin technique for the solution of compressible flow*, J. Comp. Phys., Vol. 224, 2007, pp. 208-231.
6. J.-M. Ghidaglia, F. Pascal: *On boundary conditions for multidimensional hyperbolic systems of conservation laws in the finite volume framework*, Technical Report, ENS de Cachan, 2002.
7. E. Godlewski, P. A. Raviart: *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer Verlag, 1995.
8. K. Michalak, C. Ollivier Gooch: *Differentiability of Slope Limiters on Unstructured Grids*, In: Proceedings of Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada, 2006.
9. M. Gurris: *Implicit Finite Element Schemes for Compressible Gas and Particle-Laden Gas Flows*, PhD Thesis, Dortmund, 2009.
10. C. Hirsch: *Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows*, Vol. II, John Wiley and Sons, New York, 1990.
11. R. Hartmann, P. Houston: *Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations*, J. Comp. Phys., Vol. 183 (2), 2002, pp. 508-532.
12. R. Hartmann: *http://www.numerik.uni-hd.de/ hartmann/*, Homepage.
13. V. John, E. Schmeyer: *Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion*, Comput. Meth. Appl. Mech. Eng., Vol 198, 2008, pp. 475-494.
14. M. Feistauer, V. Kučera: *On a robust discontinuous Galerkin technique for the solution of compressible flow*, J. Comput. Phys., Vol. 224, 2007, pp. 208–221.
15. D.Kuzmin: *Explicit and implicit FEM-FCT algorithms with flux linearization*, J. Comput. Phys., Vol. 228, 2009, pp. 2517–2534.
16. D. Kuzmin: *On the design of general purpose flux limiters for finite element schemes. I. Scalar convection*, J. Comp. Phys., Vol. 219, 2006, pp. 513-531.
17. D. Kuzmin: *Algebraic flux correction for finite element discretizations of coupled systems*, In E. Onate, M. Papadrakakis, B. Schrefler, editors, Comp. Meth. for Coupled Prob. in Sc. and. Eng. II, CIMNE, 2007, pp. 653-656.
18. D. Kuzmin, M. Möller: *Algebraic flux correction I. Scalar conservation laws*, In D. Kuzmin, R. Löhner, S. Turek, editors, Flux-Corrected Transport, Principles, Algorithms, and Applications, pp. 155-206, Springer, 2005.
19. D. Kuzmin, M. Möller: *Algebraic flux correction II. Compressible Euler equations*, In D. Kuzmin, R. Löhner, S. Turek, editors, Flux-Corrected Transport, Principles, Algorithms, and Applications, pp. 207-250, Springer, 2005.
20. D. Kuzmin, S. Turek: *High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter*, J. Comp. Phys., Vol. 198 (1), 2004, pp. 131-158.
21. D. Kuzmin, M. Möller, J. Shadid, M. Shashkov: *Failsafe flux limiting and constrained data projections for systems of conservation laws*, submitted to J. Comp. Phys., 2010.
22. R. J. LeVeque: *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
23. P. R. M. Lyra: *Unstructured Grid Adaptive Algorithms for Fluid Dynamics*, PhD Thesis, Swansea, 1994.
24. K. Michalak, C. Ollivier Gooch: *Differentiability of Slope Limiters on Unstructured Grids*, In: Proceedings of Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada, 2006.
25. M. Möller: *Adaptive High-Resolution Finite Element Schemes*, PhD Thesis, TU Dortmund, 2008.
26. A. Nejat: *A Higher-Order Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*, PhD Thesis, Vancouver, 2007.
27. A. Pueyo, D. W. Zingg: *An efficient Newton-GMRES solver for aerodynamic computations*, AIAA Paper, 1997, Article No. 97-1955.
28. P. L. Roe: *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comp. Phys., Vol. 43, 1981, pp. 357-372.
29. A. Rohde: *Eigenvalues and eigenvectors of the Euler equations in general geometries*, AIAA Paper, 2001, Article No. 2001-2609.
30. R. A. Shapiro: *An Adaptive Finite Element Solution Algorithm for the Compressible Euler Equations*, PhD Thesis, Massachusetts, 1988.
31. V. Selmin: *The node-centered finite volume approach: Bridge between finite differences and finite elements*, Comp. Meth. Appl. Mech. Eng., Vol. 102, 1993, pp. 107-138.
32. V. Selmin, L. Formaggia: *Unified construction of finite element and finite volume discretizations for compressible flows*, Int. J. Num. Meth. Eng., Vol. 39, 1996, pp. 1-32.

DOI: 10.1002/fld

33. A. Sokolichin: *Mathematische Modellbildung und Numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*, (German), Habilitation Thesis, Stuttgart, 2002.
34. E. F. Toro: *Riemann Solvers and Numerical Methods for Fluid Dynamics*, A Practical Introduction, Springer, 1999.
35. J.-Y. Trépanier, M. Reggio, D. Ait-Ali-Yahia: *An implicit flux-difference splitting method for solving the Euler equations on adaptive triangular grids*, Int. J. Num. Meth. Heat Fluid Flow, Vol. 3, 1993, pp. 63-77.
36. D. L. Whitaker, D. C. Slack, R. W. Walters: *Solution algorithms for the two-dimensional Euler equations on unstructured meshes*, AIAA Paper, 1990, Article No. 90-0697.
37. H. C. Yee, R. F. Warming, A. Harten: *Implicit total variation diminishing (TVD) schemes for steady-state calculations*, J. Comp. Phys., Vol. 57, 1985, pp. 327-360.

DOI: 10.1002/fld

(a) High-resolution scheme (blue=0.22, red=1.41)



(b) Coarse mesh

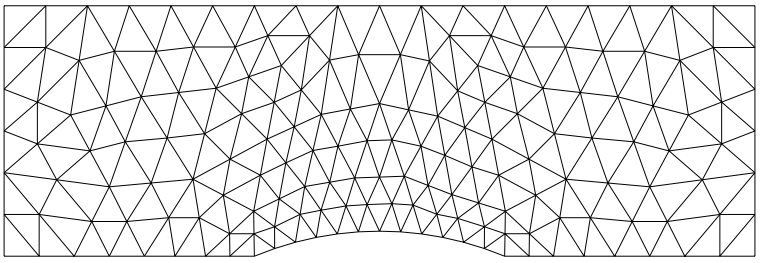
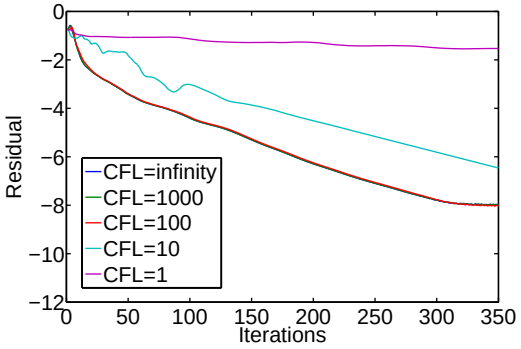


Figure 2. GAMM channel: Mach number (30 contours) and coarse grid

(a) High-resolution scheme



(b) Low-order Scheme

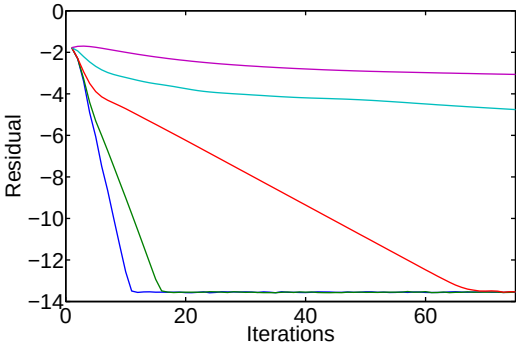
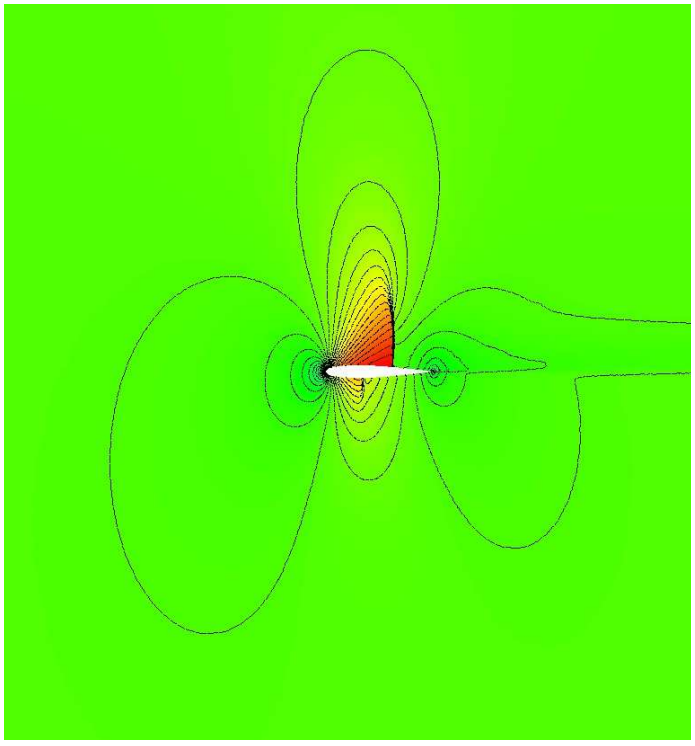


Figure 3. GAMM channel: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level four

DOI: 10.1002/fld

(a) High-resolution scheme (blue$= 0.02$, red$= 1.36$)



(b) Coarse mesh

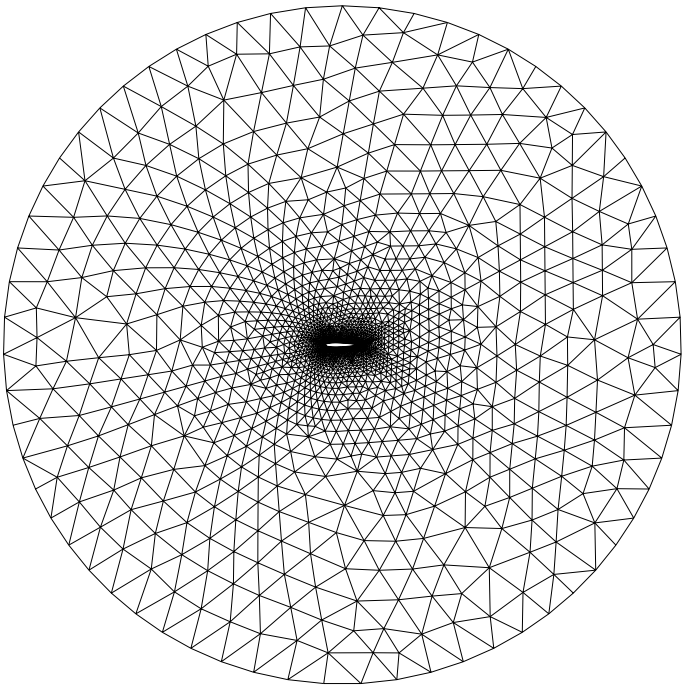


Figure 4. NACA airfoil: Mach number zoom (40 contours) and coarse mesh

DOI: 10.1002/fld

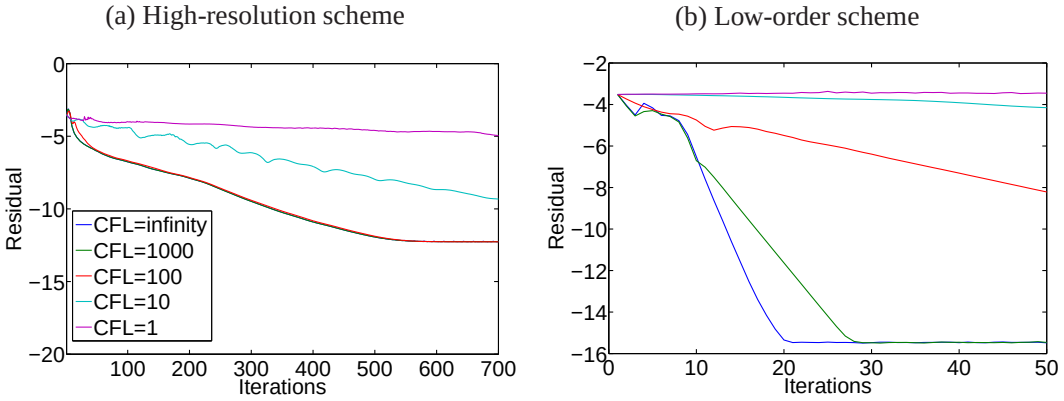
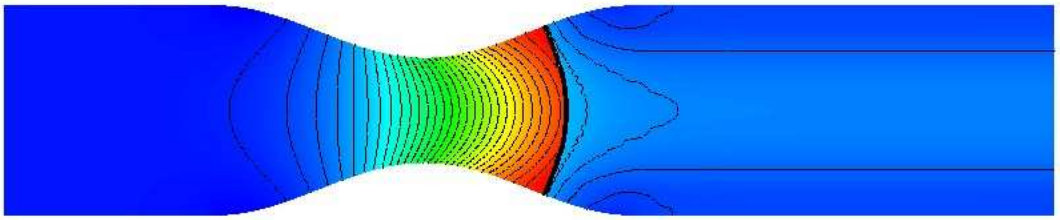


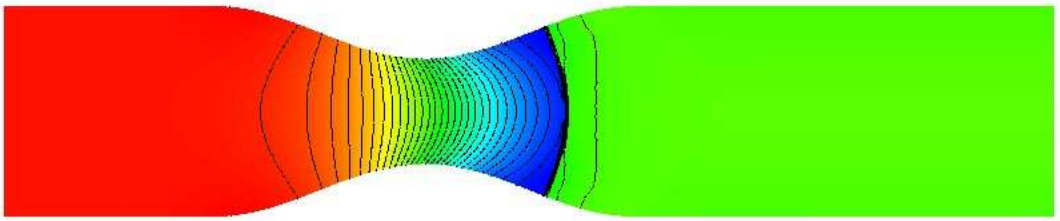
Figure 5. NACA airfoil: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level two

DOI: 10.1002/fld

(a) Mach number (blue$= 0.27$, red$= 2.03$)



(b) Density (blue$= 0.23$, red$= 1.01$)



(c) Pressure (blue$= 0.13$, red$= 1.01$)

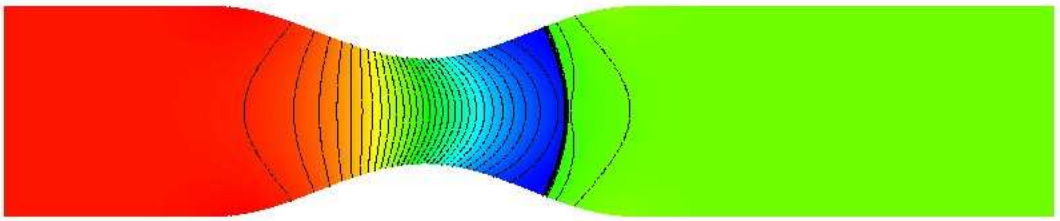
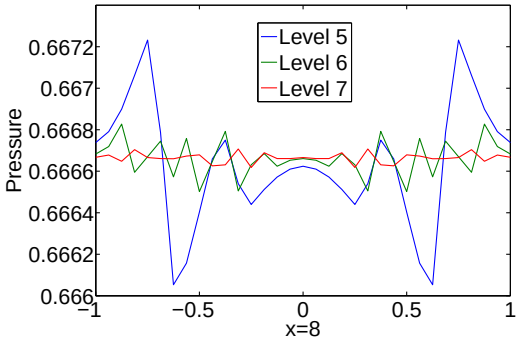


Figure 6. Nozzle flow with pressure outlet: Flux-corrected solution (30 contours)

DOI: 10.1002/fld

(a) Outlet pressure



(b) Pressure at the centerline

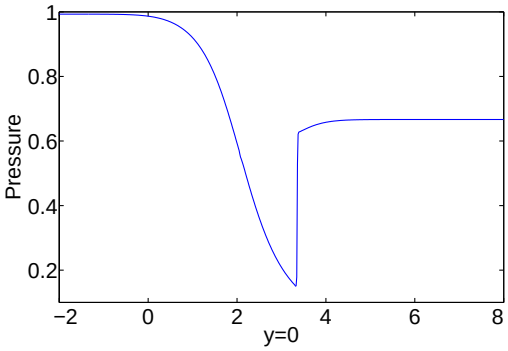
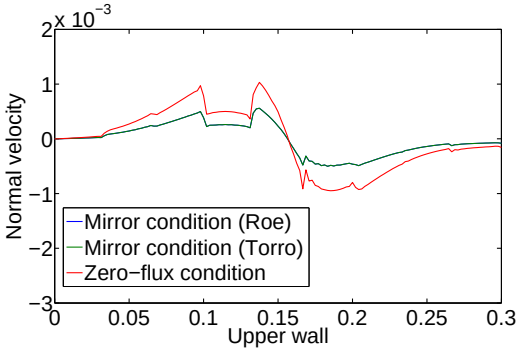
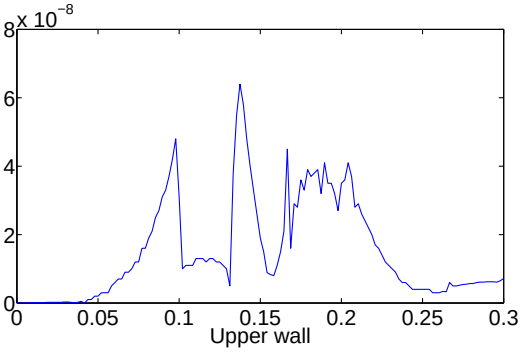


Figure 7. Investigation of the pressure outlet boundary condition

(a) Comparison of different wall boundary conditions (mesh level five)



(b) Difference of approximate/exact Riemann solver (mesh level five)



(c) Convergence analysis

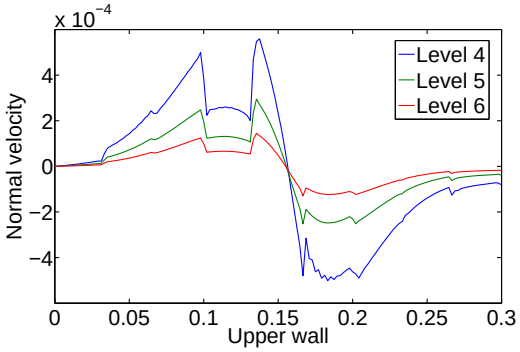


Figure 8. Analysis of wall boundary conditions at the upper wall boundary

DOI: 10.1002/fld

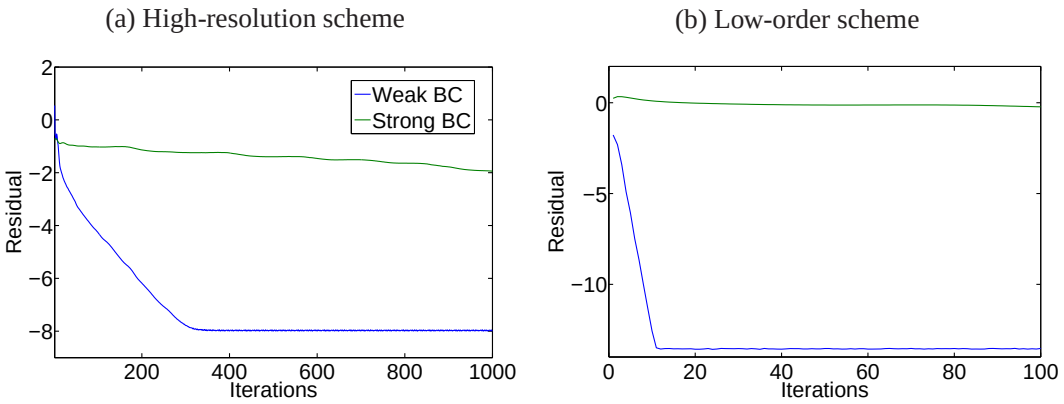


Figure 9. Nonlinear convergence history for weak and strong boundary conditions in logarithmic scale

DOI: 10.1002/fld