Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 5426–5446

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A discontinuous Galerkin finite element discretization of the Euler equations for compressible and incompressible fluids

L. Pesch *, J.J.W. van der Vegt

University of Twente, Department of Applied Mathematics, P.O. Box 217, 7500 AE Enschede, The Netherlands

Received 11 April 2007; received in revised form 17 November 2007; accepted 30 January 2008
Available online 8 February 2008

Abstract

Using the generalized variable formulation of the Euler equations of fluid dynamics, we develop a numerical method that is capable of simulating the flow of fluids with widely differing thermodynamic behavior: ideal and real gases can be treated with the same method as an incompressible fluid. The well-defined incompressible limit relies on using pressure primitive or entropy variables. In particular entropy variables can provide numerical methods with attractive properties, e.g. fulfillment of the second law of thermodynamics. We show how a discontinuous Galerkin finite element discretization previously used for compressible flow with an ideal gas equation of state can be extended for general fluids. We also examine which components of the numerical method have to be changed or adapted. Especially, we investigate different possibilities of solving the nonlinear algebraic system with a pseudo-time iteration. Numerical results highlight the applicability of the method for various fluids.

PACS: 65M60; 76M10; 76N10

Keywords: Discontinuous Galerkin finite element methods; Euler equations; Entropy variables; Pseudo-time integration methods; General equations of state; Compressible flow; Incompressible flow

1. Introduction

Many numerical methods for fluid dynamics are suitable only for a single, idealized type of fluid. Most prominently, algorithms for compressible flow are often tailored to ideal gasses and another class of schemes is designed for incompressible media. Both underlying fluid models are important as they capture well the essential behavior of many fluid flows. At the same time the restriction to one or the other case means that the numerical method cannot be applied for other media (e.g. real gasses) or for situations in which several fluids with different

* Corresponding author.
E-mail addresses: L.Pesch@math.utwente.nl (L. Pesch), J.J.W.vanderVegt@math.utwente.nl (J.J.W. van der Vegt).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.046

properties populate the domain in the form of a fluid composite (e.g. bubbly flow). Furthermore, many compressible flow schemes suffer from convergence and accuracy problems in the low Mach number (M) limit [1,2], which hampers their application on domains where high and low Mach numbers coexist.

In this article, we develop a numerical method that can be applied to different media (i.e. different equations of state) and for high as well as low Mach numbers. We use the compressible Euler equations but notice the approach to conservation laws that was introduced by Godunov [3]: by performing a change of variables, the system matrix of a (quasi-) linear system of partial differential equations (PDE) is symmetrized if the special set of *entropy variables* is used. The symmetrized form of the governing equations has several interesting properties [4], in the case of fluid mechanics especially that it yields a formulation suitable for a wide range of physical conditions. Aside from the broad applicability of the numerical method, another advantage is the satisfaction of the second law of thermodynamics [5]. Dependent on the numerical method, this property may rely on the choice of a numerical flux [6]. Note that entropy stability is a global stability statement, just like energy stability, and it does not ensure that locally no numerical oscillations can occur.

The application of the generalized variable approach – in which various sets of variables may be used instead of the conservative base set – has been at the heart of a number of finite element discretizations for the Euler and Navier–Stokes equations [5–7]. An important observation is that for pressure primitive and entropy variables the Euler equations have a well-defined incompressible limit. Provided all components of the numerical method maintain this property, stable and efficient algorithms can be designed that apply to both compressible and incompressible flow. This route was taken in [4,8], which contain numerical examples for both cases based on a least-squares continuous finite element method (FEM). The theoretical advantages of entropy variable formulations are opposed by the complexity of using them, which may also be the reason why they have not been exploited more often. For background on the symmetrization of hyperbolic systems and the derivation of the entropy variables we refer to the work of Barth [6] and references therein.

Our numerical method combines the general variable formulation (with entropy variables constituting a particular choice) with a discontinuous Galerkin (DG) discretization [9,10]. The advantages of this type of discretization include increased locality – both in the sense of data dependencies and regarding the possibility of more accurate solution representation: local *hp*-adaptation is accommodated naturally, i.e., *h*-adaptation for refining the mesh in space regions where (possibly discontinuous) small scale flow structures need to be resolved [9,11], and *p*-refinement for representing the solution with higher order smooth basis functions per element [12]. By using a space–time weak formulation and (discontinuous) finite-element basis functions in space and time, the method can be readily combined with an arbitrarily Lagrangian–Eulerian formulation [9] to tackle problems with moving and deforming meshes and boundaries. Barth [6] has analyzed the standard DG formulation and has proven (nonlinear) entropy stability with entropy variables for several choices of the numerical flux.

In his implementation, Barth uses a Newton iteration for the solution of the system of equations arising from the discretization. While the global linearization of the entropy variable formulation refers back to the well-behaved Jacobians of the pseudo-linear form, its implementation is cumbersome and the computation costly. In the algorithm developed in this article, we avoid the linearization and use a pseudo-time iteration for solving the nonlinear system of equations [11,13]. The technique consists of augmenting the (time-dependent) equations with an additional artifical time coordinate, for which the computed solution at a previous time level is the initial condition, and for which a steady state solution is sought. The steady-state in pseudo-time is approached by integrating with Runge–Kutta methods, often adapted to this special purpose [9,13–15]. The advantage of these methods is that the local character of the DG discretization is – as far as possible – conserved as no global operations or data structures are required. Pseudo-time methods have been analyzed by Klaij et al. [14] for the ideal gas Navier–Stokes equations and found beneficial in particular for advection-dominated flow. An important question is, however, how to combine the pseudo-time method with the general variable approach. The analysis by Klaij et al. cannot answer this question as it considers a simplified problem – the scalar advection–diffusion equation – which does not accommodate the transformation that is interposed in the generalized variable formulation. Our findings demonstrate how the pseudo-time iteration can be combined with the generalized variable approach and which measures have to be taken when the incompressible limit is attained. Note, however, that for large scale computations the pseudo-time integration method needs to be combined with multigrid convergence acceleration. We have developed this technique both for viscous and inviscid compressible flows [9,34], but in this article we focus on the investigation of the space–time discretization and pseudo-time integration method for the generalized variables approach including general equations of state.

The outline of this article is as follows: the Euler equations of fluid dynamics and the variable sets which can be used to obtain a formulation with well-defined incompressible limit are summarized in Section 2. Here we also provide thermodynamical equations of state for different media. These relations are necessary to close the Euler equations and to incorporate the behavior of different fluids. The main focus of this article is the finite element discretization detailed in Section 3; we extend a space–time discontinuous Galerkin technique for an ideal gas [9] to a method suitable for both compressible and incompressible flows. Stability aspects regarding the solution of the nonlinear system of equations with a pseudo-time method are analyzed in Section 4. We show how the pseudo-time iteration can be joined with the generalized variable formulation and what the impact of different fluid models on the algebraic equation system is. We present numerical examples in Section 5 for both compressible and incompressible fluids. Finally, we draw conclusions in Section 6.

2. Governing equations and variable sets

2.1. The Euler equations for conservation variables

We consider a fluid during the time interval $\mathtt{T} := [t_\mathrm{s}, t_\mathrm{e}]$ in the d-dimensional space domain $\overline{\mathcal{E}}$. The physical conservation statements for the conserved quantities mass, momentum, and energy lead to the introduction of the *conservation variables* U and the flux matrix $\overline{F}$ given by

$$U = \begin{pmatrix} \rho \\ \rho v_i \\ \rho e^{\mathrm{tot}} \end{pmatrix}, \quad \overline{F}_k = \begin{pmatrix} \rho v_k \\ \rho v_i v_k + p\delta_{ik} \\ v_k(\rho e^{\mathrm{tot}} + p) \end{pmatrix}, \quad \text{with } i, k = 1, \ldots, d. \tag{1}$$

Here ρ denotes density, v_i are the velocity components with respect to a Cartesian coordinate system, and p is the pressure. The total energy e^{tot} in this case is the sum of internal and kinetic energy,

$$e^{\mathrm{tot}} = e + k = e + \frac{1}{2} v_j^2, \tag{2}$$

and the summation convention is implied on repeated indices. The Euler equations of fluid dynamics written in conservation form are

$$\frac{\partial U_s}{\partial t} + \frac{\partial \overline{F}_{sj}}{\partial x_j} = 0, \quad s = 1, \ldots, d+2, \ \forall (\boldsymbol{x}, t) \in \overline{\mathcal{E}} \times \mathtt{T}, \tag{3}$$

combined with suitable boundary and initial conditions of the form

$$\begin{aligned} U(\boldsymbol{x}, t) &= B(U, U_w), \quad \forall (\boldsymbol{x}, t) \in \partial\overline{\mathcal{E}} \times \mathtt{T}, \\ U(\boldsymbol{x}, t_\mathrm{s}) &= U_0(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \overline{\mathcal{E}}, \end{aligned}$$

with U_w a given set of boundary values and B a boundary operator depending on the type of boundary conditions imposed. For the discretization discussed in this article, the time dimension is treated equally to the space dimensions, hence we extend the flux matrix $\overline{F}$ by a column with the time flux $F_0(U) = U$ on the left, to receive the space–time flux F. Making use of this definition, the Euler equations (3) can be stated as

$$\frac{\partial F_{sj}}{\partial x_j} = 0, \quad s = 1, \ldots, d+2, \tag{4}$$

where the summation on j runs from 0 to d, i.e. over time and space dimensions.

2.2. Thermodynamic properties and equations of state

The non-linear system of partial differential equations (3) for the conservation variables and flux (1) contains two more unknowns than equations, hence it is not closed. One additional relationship has been given by

(2), but to complete the description, we have to specify how the internal energy e and pressure p depend on the other variables.

Remark. In the more general context of the Navier–Stokes equations, an additional quantity, the temperature T, is needed to describe the diffusive effect of internal energy: heat conduction. To remain within the framework of thermodynamics that relates state variables of divariant fluids (i.e. those whose state is determined by two independent variables, e.g. p and T), we include temperature in our formulation.

Equations that diagnostically relate state variables of a fluid, for instance ρ, p, e, T, are called equations of state (EOS). The system consisting of (3) and (1) can be closed by specifying, for example, the functions $e = e(\rho, T)$ and $p = p(\rho, T)$. These relations depend on the medium considered; countless variations have been proposed to adequately describe different materials, see for example [16].

Another way to characterize different fluids – especially regarding their compressibility – is given by two derived parameters, namely the volume expansivity α_p and the isothermal compressibility β_T, which are defined as the relative changes of the specific volume $\alpha = 1/\rho$ as

$$\alpha_p := \frac{1}{\alpha}\left(\frac{\partial \alpha}{\partial T}\right)_p, \quad \beta_T := -\frac{1}{\alpha}\left(\frac{\partial \alpha}{\partial p}\right)_T. \tag{5}$$

Knowledge of these two quantities in terms of the thermodynamical state allows to work out the previously mentioned equations of state, too [17].

In the sequel we summarize the mentioned thermodynamical relations and parameters for three media: the ideal gas, as the most frequently used idealization, the van der Waals gas, as a prototype of a real gas, and the incompressible fluid.

2.2.1. Ideal gas

For an ideal gas with constant specific heat (at constant volume) c_v and the gas constant R, the equations of state are given by

$$e(T) = e_0 + c_v(T - T_0), \tag{6a}$$

$$p(\rho, T) = \rho R T, \tag{6b}$$

where the subscript 0 refers to a suitable reference state. For the transformation to entropy variables (cf. Section 2.3) we also need to know the entropy of an ideal gas at a given state:

$$s(\alpha, T) = s_0 + R \ln\frac{\alpha}{\alpha_0} + c_v \ln\frac{T}{T_0}. \tag{7}$$

The sound speed a can be computed as $a(p, \rho) = \sqrt{\gamma p/\rho}$ with the adiabatic coefficient $\gamma = c_p/c_v$, i.e., the ratio of the specific heats at constant pressure and volume. The compressibility parameters defined in (5) evaluate to $\alpha_p = 1/T$ and $\beta_T = 1/p$.

2.2.2. van der Waals gas

At high densities the interaction of the gas molecules modifies the thermodynamic behavior of the material, which is neglected by the ideal gas approximation. The van der Waals (vdW) equations of state represent a model that takes into account the finite volume b occupied by the molecules in the total specific volume α, and their attraction, governed by another material parameter a, see e.g. [18]. The equations of state read

$$e(\alpha, T) = e_0 + c_v(T - T_0) + a\left(\frac{1}{\alpha_0} - \frac{1}{\alpha}\right), \tag{8a}$$

$$p(\alpha, T) = \frac{RT}{\alpha - b} - \frac{a}{\alpha^2}, \tag{8b}$$

where – analogously to the ideal gas – c_v has been assumed constant. Unlike the ideal gas, it follows that the specific heat at constant pressure of the vdW gas cannot be constant, since

$$c_p - c_v = \frac{R^2\alpha^3 T}{R\alpha^3 T - 2a(\alpha - b)^2}. \tag{9}$$

The derivation of the previous relationship and of the entropy of the vdW gas,

$$s(\alpha, T) = s_0 + c_v \ln\frac{T}{T_0} + R\ln\frac{\alpha - b}{\alpha_0 - b}, \tag{10}$$

can be found in, e.g., [17].

Although the vdW EOS reproduce the ideal gas EOS for $a = b = 0$, the vdW gas is substantially more complicated because the pressure equation of state (8b) is a rational function in p, α, and T, and computing arbitrary states is made difficult by the nonlinearity. For temperatures below a critical temperature T_c, the (α,p)-graph is non-monotone in an environment of the critical point (α_c,p_c), as can be seen from the expressions for the compressibility parameters,

$$\alpha_p = \frac{1}{\frac{\alpha}{\alpha-b}T - 2a\frac{\alpha-b}{R\alpha^2}}, \quad \beta_T = \frac{1}{\frac{\alpha}{\alpha-b}p - \frac{a}{\alpha^2}\left(2 - \frac{\alpha}{\alpha-b}\right)}, \tag{11}$$

which can change sign at sufficiently small values of T, p. Under these conditions sensible state evaluations are impossible. Consequently the (iterative) computation of thermodynamical parameters fails close to the critical point. This deficiency can be remedied, for example with the Maxwell construction. In this article only cases with monotonic state functions are treated, which avoids non-uniqueness and guarantees the hyperbolicity of the Euler equations with the used thermodynamical models [19,20].

2.2.3. Incompressible fluid

A different idealization, the incompressible fluid, represents the limit in which the density ρ is independent of the thermodynamic state. In this case, pressure becomes a purely mechanical variable and is not determined by an EOS like (6b). The internal energy depends only on temperature, and, assuming again a constant specific heat at constant volume, c_v, the energy is given by

$$e(T) = e_0 + c_v(T - T_0). \tag{12}$$

For incompressible fluids the specific heats at constant volume and at constant pressure are necessarily equal, $c_v = c_p$. The entropy is given by

$$s(\alpha, T) = s_0 + c_v \ln\frac{T}{T_0}. \tag{13}$$

For an incompressible fluid, both compressibility parameters are zero, $\alpha_p = 0$, $\beta_T = 0$.

At a later stage of this article the topic of different media will reappear and we will see an influence on the choice of parts of the numerical algorithm. But notably, given the right choice, a simulation tool can be set up that accommodates the different media mentioned above. Also in our software implementation the equations of state are encapsulated in separate classes that can be exchanged independently from the other code. Having provided the necessary equations of state we take a look at the Euler equations from a different angle now.

2.3. Introduction of the pressure primitive and entropy variable sets

We return to our consideration of the Euler equations (3). One way to derive numerical methods that are suitable both for compressible and incompressible flows is to solve for other unknowns than the set of conservation variables U, for which the equations are originally derived. Two sets of variables that yield well-defined formulations in the incompressible limit are pressure primitive variables Y and entropy variables V, cf. [4]. These sets are defined as

$$Y = \begin{pmatrix} p \\ v_i \\ T \end{pmatrix}, \quad V = \frac{1}{T}\begin{pmatrix} \mu - \frac{1}{2}v_j^2 \\ v_i \\ -1 \end{pmatrix}, \tag{14}$$

respectively, with μ the chemical potential. We now interpret the conservation variables U as dependent on one of these variable sets. In formulae we generally use V, but the derivation of the discretization and algorithm equally holds for pressure primitive variables Y. When results are specific for one or the other set we explicitly differentiate between the two. By rewriting (3) as quasi-linear system with respect to the variables V, we obtain

$$A_0^V \frac{\partial V}{\partial t} + A_j^V \frac{\partial V}{\partial x_j} = 0, \tag{15}$$

with the Jacobians $A_0^V = \partial U / \partial V$, $A_i^V = \partial \overline{F}_i / \partial V$. Having stated the different variable sets the benefits of using one or the other set have to be examined. Analysis shows that [4]:

- For conservation variables U, the flux Jacobians A_i^U are not well defined in the incompressible limit α_p, $\beta_T \to 0$ (i.e. they contain entries that diverge or take the form 0/0). This is connected to the fact that the equation for density in (3) loses its prognostic character for $\rho = \text{const}$ – and causes the breakdown of many numerical methods using conservation variables when simulating (almost) incompressible fluids.
- The Jacobians A_i^Y, A_i^V are well defined in the incompressible limit. This corresponds to the suitability of numerical methods based on these sets of variables to compute both compressible *and* incompressible flows. In particular, many methods for incompressible flow are based on the pressure variables.
- The matrix A_0^V (for entropy variables) is symmetric and positive definite (in the incompressible case this weakens to semi-definiteness). The matrices A_i^V are symmetric.

These properties led us to adapt a discontinuous Galerkin discretization [9] to use the pressure primitive and in particular the entropy variables, in place of conservation variables. Note that our method is not built on the quasi-linearized form (15), but discretizes the original conservation equations (3) with an additional dependence of U on either Y or V. The method is elementwise conservative, even in the incompressible limit. The entropy variables have the additional advantage that they symmetrize the equations and link to nonlinear stability, cf. [6,21].

3. Space–time discontinuous Galerkin discretization

3.1. Definitions

3.1.1. Space–time geometry and formulation

In this article we consider a flow problem on a fixed d-dimensional space domain $\overline{\mathcal{E}}$. The extension to moving domains and meshes with an arbitrary Lagrangian–Eulerian (ALE) formulation is easily accomplished but outside the focus of the current work and we refer to [9] in this respect. We incorporate the time dimension in the domain so that $\mathcal{E} \subset \mathbb{R}^{1+d}$ is the space–time domain of the problem. As in the domain notation, symbols with an overbar generally refer to space dimensions only, while those without include the time dimension, too. A point in the space–time domain is denoted as $\boldsymbol{x}$ and, depending on the context, it is represented by its coordinates with respect to an underlying Cartesian coordinate system as $\boldsymbol{x} = (x_0, \ldots, x_d) = (x_0, \bar{\boldsymbol{x}}) = (t, \bar{\boldsymbol{x}})$.

The boundary of the space–time domain $\partial\mathcal{E}$ can be decomposed into the spatial domain at the start and end time, $\overline{\mathcal{E}}(t_\text{s})$ and $\overline{\mathcal{E}}(t_\text{e})$, respectively, and the hypersurface spanned by the spatial domain boundary $\partial\overline{\mathcal{E}}$ and time, $\mathcal{Q} := \{\boldsymbol{x} \in \partial\mathcal{E} | t_\text{s} < x_0 < t_\text{e}\}$. An example of this notation for the spatial dimension one is given in Fig. 1.

To make the space–time domain $\mathcal{E}$ available to finite-element methods, it has to be discretized by a suitable mesh. First, the considered time span $\mathtt{T} = [t_\text{s}, t_\text{e}]$ is partitioned into N_t intervals $\mathtt{T}^n := [t_{n-1}, t_n]$, with $t_\text{s} = t_0 < t_1 < \cdots < t_{N_t} = t_\text{e}$. The space–time domain $\mathcal{E}$ is partitioned by the hyperplanes $H^n := \{\boldsymbol{x} \in \mathbb{R}^{1+d} | x_0 = t_n\}$ into space–time slabs $\mathcal{E}^n := \{\boldsymbol{x} \in \mathcal{E} | x_0 \in \mathtt{T}^n\}$, $1 \leqslant n \leqslant N_t$.

At each time t_{n-1}, a tessellation $\overline{\mathcal{T}}_h^{n-1,+}$ of the physical space domain $\overline{\mathcal{E}}(t_{n-1})$ into N_el^n elements $\overline{\mathcal{K}}_e^{n-1,+}$, $e = 1, \ldots, N_\text{el}^n$, is given. We consider elements with a shape that can be related to the reference d-cube $\hat{\overline{\mathcal{K}}} = [-1; 1]^d$ by the invertible mapping $\overline{G}_e^{n-1,+} : \hat{\overline{\mathcal{K}}} \to \overline{\mathcal{K}}_e^{n-1,+}$. By extending the elements $\overline{\mathcal{K}}_e^{n-1,+}$ orthogonally into the time dimension we obtain the space–time elements $\mathcal{K}_e^n$ in the slab $\mathcal{E}^n$, mappable to the reference ele-

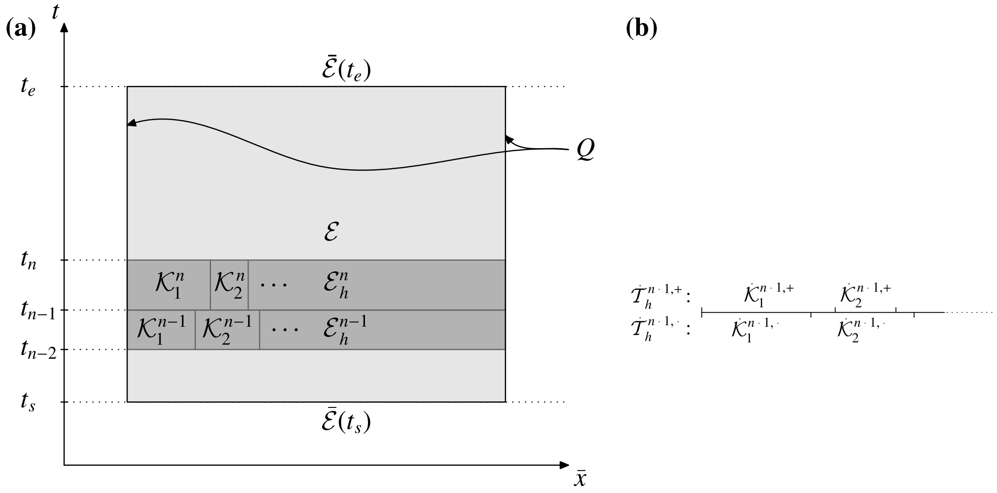

Fig. 1. Illustration of the notation for a one-dimensional space domain: (a) space–time domain and mesh; (b) space-tesselations at an inter-timeslab boundary.

ment $\widehat{\mathcal{K}} = [-1;1]^{1+d}$ by $G_e^n : \widehat{\mathcal{K}} \to \mathcal{K}_e^n$. The set of all space–time elements in the nth space–time slab constitutes the space–time tessellation $\mathcal{T}_h^n = \{\mathcal{K}_e^n | e = 1, \ldots, N_{\text{el}}^n\}$, and the domain of the discretized space–time slab is $\mathcal{E}_h^n := \bigcup_{e=1}^{N_{\text{el}}^n} \mathcal{K}_e^n$, with h some representative measure of the cell diameter.

Remark: In the above framework, different space tessellations $\overline{\mathcal{T}}_h^{n,-}$ and $\overline{\mathcal{T}}_h^{n,+}$ may exist on either side of the hyperplane H^n, allowing a re-meshing procedure to be used in case the physical space tessellation does not fulfill the quality requirements given the current solution. The re-meshing is fully conservative as the integrals over the time faces $\overline{\mathcal{K}}_e^{n,-}$ and $\overline{\mathcal{K}}_e^{n,+}$ from the old and new space–time slab are equal even under inexact quadrature.

In the following, the entities of codimension one (with regard to the space dimension) that bound the space element $\overline{\mathcal{K}}_e^{n-1,+}$ are called the element faces $\overline{\mathcal{S}}_{e,j}^{n-1,+}$, $j = 1, \ldots, 2d$. The element face $\overline{\mathcal{S}}_{e,j}^{n-1,+}$ is extended in the same way into space–time as the elements, which defines the space–time element face $\mathcal{S}_{e,j}^n$. The space–time element $\mathcal{K}_e^n$ then has the boundary

$$\partial\mathcal{K}_e^n = \overline{\mathcal{K}}_e^{n-1,+} \cup \overline{\mathcal{K}}_e^{n,-} \cup \bigcup_{j=1}^{2d} \mathcal{S}_{e,j}^n. \tag{16}$$

For a function w defined on the discrete space–time slab $\mathcal{E}_h^n$, the traces at a point $\boldsymbol{x}$ of the boundary $\partial\mathcal{K}_e^n$ of element $\mathcal{K}_e^n$ are defined as

$$w_{\mathcal{K}_e^n}^{\pm} := \lim_{\epsilon \downarrow 0} w(\boldsymbol{x} \pm \epsilon \boldsymbol{n}_{\mathcal{K}_e^n}). \tag{17}$$

Given that the normal vector $\boldsymbol{n}_{\mathcal{K}_e^n}$ is outward with respect to element $\mathcal{K}_e^n$, the superscript '+' defines the exterior trace and ' − ' the interior trace. If the point $\boldsymbol{x} \in \partial\mathcal{K}_e^n$ is situated on the boundary $\partial\mathcal{E}_h^n \setminus (H^{n-1} \cup H^n)$ of the space–time slab then only the interior trace $w_{\mathcal{K}_e^n}^-$ exists.

For the discontinuous Galerkin method it is beneficial to see the faces not as induced by the elements but as independent constituents of the mesh. Hence we define different kinds of faces, according to their position in the space–time slab $\mathcal{E}_h^n$. Based on the element face $\mathcal{S}_{e,j}^n$ we define an internal face $\mathcal{F}_{\{e,e'\}}^{\text{int},n}$ if $\exists e, e' \in \{1, \ldots, N_{\text{el}}^n\},\ e \neq e',\ j, j' \in \{1, \ldots, 2d\} : \mathcal{S}_{e,j}^n = \mathcal{S}_{e',j'}^n$, and a boundary face $\mathcal{F}_{e,j}^{\text{bnd},n}$ otherwise. The internal faces are subsumed in the set $\mathcal{F}^{\text{int},n}$ and the boundary faces in $\mathcal{F}^{\text{bnd},n}$, and finally $\mathcal{F}^n = \mathcal{F}^{\ \text{int},n} \cup \mathcal{F}^{\text{bnd},n}$.

3.1.2. Function spaces

The test and trial spaces for our DG method contain functions that are elements of the space of tensor product polynomials in space, augmented with monomial basis functions in time, $\mathcal{Q}_{(p_t,p_s)}(\widehat{\mathcal{K}})$, on the space–time reference element $\widehat{\mathcal{K}}$. Their maximum order in time and space is given by p_t and p_s, respectively. These properties are reflected in the definition of the function space for test and trial functions on the space–time slab n,

$$\mathcal{P}_h^{n,(p_t,p_s)} := \{w \in (L^2(\mathcal{E}_h^n))^{d+2} | w|_{\mathcal{K}_e^n} \circ G_e^n \in (\mathcal{Q}_{(p_t,p_s)}(\widehat{\mathcal{K}}))^{d+2}, \quad \forall \mathcal{K}_e^n \in \mathcal{T}_h^n\}. \tag{18}$$

On one element, all variables are expanded to the same order. The order of the temporal and spatial expansion could vary between elements, but we do not exploit this advantage of DG methods here. At the internal faces, the polynomial representations of a variable V on the two adjacent elements may lead to different values on the two sides of the face.

3.2. Weak form

In (4), the Euler equations appear in (space–time) divergence form and thus lend themselves well to the standard procedure to derive the weak form of a hyperbolic PDE system: scalar multiplication with a test function vector $W \in \mathcal{P}_h^{n,(p_t,p_s)}$ and integration over the space–time domain. The latter is represented as the sum of the integrals over space–time elements $\mathcal{K}_e^n$ of the tessellation $\mathcal{T}_h^n$. Integration by parts yields

$$\sum_{n=1}^{N_t} \sum_{\mathcal{K}_e^n \in \mathcal{T}_h^n} \left\{ -\int_{\mathcal{K}_e^n} \frac{\partial W_r}{\partial x_s} F_{rs} \mathrm{d}\mathcal{K} + \int_{\partial \mathcal{K}_e^n} W_r^- \widetilde{F}_{rs} n_s \mathrm{d}(\partial \mathcal{K}) \right\} = 0, \tag{19}$$

with $\boldsymbol{n}$ the unit outward normal vector with respect to element $\mathcal{K}_e^n$. The repeated index r implicitly sums over the equations, $r = 1, \ldots, d+2$ and the sum over s covers the space–time dimensions, $s = 0, \ldots, d$. In the integral over the element boundary, the Euler flux is denoted $\widetilde{F}$. The tilde denotes a numerical flux function, whose introduction is necessary because the previously defined function spaces allow discontinuities at the inter-element boundaries. To make the discretization conservative, $\widetilde{F}(U^{\mathrm{L}}, U^{\mathrm{R}}; \boldsymbol{n})$ determines a unique value of the flux in the direction $\boldsymbol{n}$ given two states U^{L} and U^{R}. Necessary conditions for numerical flux functions for correct coupling and consistency are (i) $\widetilde{F}(U^{\mathrm{L}}, U^{\mathrm{R}}; \boldsymbol{n}) = -\widetilde{F}(U^{\mathrm{R}}, U^{\mathrm{L}}; -\boldsymbol{n})$, and (ii) $\widetilde{F}(U, U; \boldsymbol{n}) = F(U) \cdot \boldsymbol{n}$. The choice of the spatial numerical flux function for the Euler equations is the subject of Section 3.3. Having introduced the numerical flux, the range of the summation in (19) can be conceptionally changed: instead of integrating over the element boundaries $\partial \mathcal{K}_e^n$ one rather uses the faces $\mathcal{F}^n$ as summation base. In this way the flux is defined uniquely per face and – in computational regard – needs to be evaluated only once per face, as opposed to once per side of the face (i.e. element boundary). Obviously the numerical flux function may have to be evaluated in several points to approximate the face integral based on a numerical quadrature rule.

In the space–time face integrals, the traces (17) are no longer uniquely defined because we lose the connection to an element. Instead, for each internal face we arbitrarily declare one element to be the left one (L) and the other one to be the right one (R). Boundary faces have only a left neighbor and an external state V^{R} has to be determined from the boundary condition, for example an inflow state, or a ‘ghost state’. In any case, the normal vector $\boldsymbol{n}$ is outward from the left element; to compensate for this, the internal limit W^- in (19) has to be replaced by the difference $W^{\mathrm{L}} - W^{\mathrm{R}}$ in the face integral, see [9].

The sum over the space–time slabs in (19) can be dropped based on the introduction of a suitable numerical flux in the time direction. The definition

$$\widetilde{U}(V) := \begin{cases} U(V_{e'}^{n-1}|_{\overline{\mathcal{K}}_{e'}^{n-1,-}}) & \text{on } \overline{\mathcal{K}}_e^{n-1,+}, \\ U(V_e^n|_{\overline{\mathcal{K}}_e^{n+1,-}}) & \text{on } \overline{\mathcal{K}}_e^{n+1,-}, \end{cases} \tag{20}$$

ensures causality and decouples a space–time slab from all others except its immediate predecessor. Here e' denotes the number of the element in $\mathcal{E}_h^{n-1}$ that connects to $\mathcal{K}_e^n$ at $\overline{\mathcal{K}}_e^{n-1,+}$. The limited explicit dependence on the previous slab has been exploited already in the definition of the function space $\mathcal{P}_h^{n,(p_t,p_s)}$ in (18). Further it is now used to give the weak form based on a single space–time slab as:

Find $V \in \mathcal{P}_h^{n,(p_t,p_s)}$ such that for all $W \in \mathcal{P}_h^{n,(p_t,p_s)}$ holds

$$-\sum_{\mathcal{K}_e^n \in \mathcal{T}_h^n} \int_{\mathcal{K}_e^n} \frac{\partial W_r}{\partial x_s} F_{rs}(U(V))\mathrm{d}\mathcal{K} + \sum_{\mathcal{F} \in \mathcal{F}^n} \int_{\mathcal{F}} (W_r^{\mathrm{L}} - W_r^{\mathrm{R}})\widetilde{F}_r(U(V^{\mathrm{L}}), U(V^{\mathrm{R}}); \boldsymbol{n})\mathrm{d}\mathcal{F}$$
$$-\sum_{\mathcal{K}_e^n \in \mathcal{T}_h^n} \int_{\overline{\mathcal{K}}_e^{n-1,+}} W_r^- U_r(V_{\mathcal{K}_e^n}^+)\mathrm{d}\overline{\mathcal{K}} + \sum_{\mathcal{K}_e^n \in \mathcal{T}_h^n} \int_{\overline{\mathcal{K}}_e^{n,-}} W_r^- U_r(V_{\mathcal{K}_e^n}^-)\mathrm{d}\overline{\mathcal{K}} = 0. \tag{21}$$

3.3. Numerical fluxes

In Eq. (21), integrals of the flux are computed on the faces. To give a meaning to the ambiguity of the state on inter-element boundaries, the two states from the data on the neighboring elements may be considered a (hypothetic) initial value problem with discontinuous data, a so-called Riemann problem. This interpretation is frequently adopted in gas dynamics. The solution of the Riemann problem requires additional knowledge about the considered medium and – depending on the amount of detail invested in the solution – different compromises between accuracy and computational cost are possible, cf. [22]. Alternatively, the numerical flux can be defined purely based on mathematical considerations, without reference to underlying physics. Two possible choices for the spatial numerical flux function $\widetilde{\overline{F}}$ for the DG method are presented next, along with a discussion of their suitability for the goals set out for the current work.

3.3.1. HLLC numerical flux

The HLLC numerical flux, cf. [22,23] for its derivation, is an approximate Riemann solver originally developed in the context of Godunov finite volume methods. It has been applied many times in DG FEMs, e.g. in [10,11], and in [9], which also details the computation of the flux in space–time.

The HLLC flux is a good choice for computing ideal gas flow, and can be applied to real gasses (see the application with the covolume EOS in [23] and Section 5.3), but ultimately its derivation restricts it to gaseous media. The same holds for many numerical fluxes, which often do not scale correctly in the low Mach-number/incompressible limit [1,2]. This may manifest itself in the deterioration of numerical accuracy and convergence. Consequently such flux functions are not suitable for a numerical method that aims at being applicable for different media and flow conditions. Hence, when general applicability is the main concern, our simulation framework uses the local Lax–Friedrichs flux, which applies to all fluids since it does not exploit specific information, for example about the wave structure of a Riemann problem.

3.3.2. Local Lax–Friedrichs flux

The Lax–Friedrichs (LF) flux for conservation variables in the direction of a normal $\boldsymbol{n}$ is given by

$$\widetilde{\overline{F}}(U^{\mathrm{L}}, U^{\mathrm{R}}; \boldsymbol{n}) := \frac{1}{2}(\overline{F}_s(U^{\mathrm{L}}) + \overline{F}_s(U^{\mathrm{R}}))n_s - \frac{1}{2}\lambda_{\max}(U_\star; \boldsymbol{n})(U^{\mathrm{R}} - U^{\mathrm{L}}). \tag{22}$$

with $\lambda_{\max}(U_\star;\boldsymbol{n})$ an estimate of the maximum eigenvalue of $A^U(U;\boldsymbol{n}) = n_s\partial\overline{F}_s/\partial U$ for the states U on a face. The computation is facilitated by the knowledge of the eigenvalues of the Euler flux Jacobians for conserved variables, cf. [22].

The LF flux adds a diffusion-like term to the (unstable) mean flux to damp numerical instabilities. It is actually overly diffusive, but thanks to its very general nature and independence of the EOS it is often preferred when (mixtures of) general fluids are treated. For the local version (LLF), the eigenvalue computation is based on the mean states on the adjacent element(s) and hence needs to be done only once per element. The global version maximizes the eigenvalue non-locally, which is computationally more involved. Cockburn and Shu [24] report that they found the LLF flux to be well-suited for their Runge–Kutta DG method, leading to less dissipation, but also decreased robustness, compared with the global Lax–Friedrichs flux.

For entropy variables, Barth [6] defines the flux as

$$\widetilde{\overline{F}}(V^{\mathrm{L}}, V^{\mathrm{R}}; \boldsymbol{n}) := \frac{1}{2}(\overline{F}_s(V^{\mathrm{L}}) + \overline{F}_s(V^{\mathrm{R}}))n_s - \frac{1}{2}\lambda_{\max}(V_\star; \boldsymbol{n})A_0^V(V_\star)(V^{\mathrm{R}} - V^{\mathrm{L}}), \tag{23}$$

and proves nonlinear entropy stability. The matrix $A_0^V(V_\star)$ is evaluated for the state $V_\star$, for which the above eigenvalue maximization occurs.

3.4. The nonlinear set of equations and its solution

Based on the weak form (21), a set of equations is obtained by choosing the basis and test functions. We use the basis functions $\hat{\psi}$ directly as test functions and define the basis as tensor product of the monomials with the aforementioned order p_s in space combined with the monomials up to order p_t in time. In two-dimensional space, for example, we choose as a basis of $\mathcal{Q}_{(p_t,p_s)}(\widehat{\mathcal{K}})$ the following sequence of monomials in reference coordinates $\hat{x}_i$: $[\hat{\psi}_0 = 1,\ \hat{\psi}_1 = \hat{x}_1,\ \hat{\psi}_2 = \hat{x}_2,\ \hat{\psi}_3 = \hat{x}_1\hat{x}_2,\ \hat{\psi}_4 = \hat{x}_0]$, where it has been assumed that the reference time $\hat{x}_0$ maps to $x_0 = t$. Limiting the number of basis functions to 1, 4, or 5, respectively, includes the standard cases: (i) constant representation per element ($p_t = p_s = 0$), resulting in a first order convergent numerical method, (ii) constant in time and bilinear in space ($p_t = 0, p_s = 1$), which makes the spatial discretization second order accurate, and (iii) second order description in both space and time ($p_t = 1, p_s = 1$). Corresponding bases for other dimensions and the generalization to higher order are straightforward to procure.

The basis functions exist on each element and hence we enumerate the physical space basis functions as $\psi_{(e,k,m)}$, where e represents the element number, $k \in \{1, \ldots, d+2\}$ gives the equation number, and m indexes the local basis functions per element. For a variable vector $V \in \mathcal{P}_h^{n,(p_t,p_s)}$ we denote the expansion coefficient matrix on element $\mathcal{K}_e^n$ as $\check{V}_e^n$ and subsume all expansion coefficients related to the nth space–time slab as $\check{V}^n$.

Based on (21), we define for $V \in \mathcal{P}_h^{n,(p_t,p_s)}$ (no summation on double index k):

$$A^n_{(e,k,m)}(V) := -\int_{\mathcal{K}_e^n} \frac{\partial \psi^n_{(e,k,m)}}{\partial x_s} F_{ks}(V) \mathrm{d}\mathcal{K}, \tag{24a}$$

$$B^n_{(e,k,m)}(V) := -\int_{\overline{\mathcal{K}}_e^{n-1,+}} (\psi^n_{(e,k,m)})^- U_k(V^+_{\mathcal{K}_e^n}) \mathrm{d}\overline{\mathcal{K}} + \int_{\overline{\mathcal{K}}_e^{n,-}} (\psi^n_{(e,k,m)})^- U_k(V^-_{\mathcal{K}_e^n}) \mathrm{d}\overline{\mathcal{K}}, \tag{24b}$$

$$C^n_{(e,k,m)}(V) := \sum_{\mathcal{F} \in \mathcal{F}^n} \int_{\mathcal{F}} (\psi^{n,\mathrm{L}}_{(e,k,m)} - \psi^{n,\mathrm{R}}_{(e,k,m)}) \widetilde{\widetilde{F}}_k(U(V^{\mathrm{L}}), U(V^{\mathrm{R}}); \boldsymbol{n}) \mathrm{d}\mathcal{F}. \tag{24c}$$

In the definition of $A^n_{(e,k,m)}$ and $B^n_{(e,k,m)}$ we could immediately make use of the fact that the support of $\psi^n_{(e,k,m)}$ is limited to a single element $\mathcal{K}_e^n$. In the same way it is clear that $C^n_{(e,k,m)}$ only yields contributions from faces $\mathcal{F}$ that neighbor $\mathcal{K}_e^n$.

For a single choice of the test function W as $\psi_{(e,k,m)}$, Eq. (21) becomes

$$\mathcal{L}^n_{(e,k,m)}(\check{V}^n; \check{V}^{n-1}) := A^n_{(e,k,m)}(\check{V}^n) + B^n_{(e,k,m)}(\check{V}^n; \check{V}^{n-1}) + C^n_{(e,k,m)}(\check{V}^n) = 0, \tag{25}$$

and the set of all these equations for the nth space–time slab is denoted as

$$\mathcal{L}^n(\check{V}^n; \check{V}^{n-1}) = 0, \tag{26}$$

including the unknowns $\check{V}^n$ and the known solution $\check{V}^{n-1}$ from the previous space–time slab, which enters through the integral over the ‘past’ time faces in (24b). Note that the method developed in this article is implicit, hence no Courant–Friedrichs–Lewy (CFL) criterion applies, but just the accuracy requirement limits the time step. The evaluation of all integrals in Eq. (24) is based on Gauss quadrature rules of order $2(p+1)$ on the faces and $2p+1$ on the elements, with p the maximum order of the polynomial representation.

Having derived the FE discretization of the Euler equations, a crucial part of the numerical method is how to solve the algebraic system of Eq. (26) for the expansion coefficients $\check{V}^n$. In the generalized variable context, Newton’s method has frequently been used [4,6], since the solution of the linear system in each iteration can benefit from the symmetrization property of the entropy variables. However, the linearization needs the flux Jacobians A_i^V, which are nontrivial to derive, implement and evaluate. In the DG context also the linearization of the numerical flux would be required. Based on our previous work [9,10], we have chosen to pursue a different strategy, namely pseudo-time iteration. To apply this technique, Eq. (26) is considered dependent on an additional (hypothetical) time variable, τ, in the form

$$|\overline{\mathcal{K}}_e^n|P\frac{\partial V}{\partial \tau} = -\frac{1}{\Delta t}\mathcal{L}^n(V, \check{V}^{n-1}), \tag{27}$$

where the volume of the space element $|\overline{\mathcal{K}}_e^n|$ approximates the mass matrix and the division by the time step Δt facilitates the computation of steady state solutions and restores dimensional consistency [9]. The presence of the matrix P will be explained later.

The solution $\check{V}^n$ of (26) is obtained as the steady state of (27). The steady state is computed by advancing in pseudo-time τ with a Runge–Kutta (RK) method. For this purpose, specialized RK methods are available that reduce the amount of storage and computations by computing each RK stage based solely on the previous one. In this article, we use a 5th order method (RK5) [9] together with the Melson correction [13]. For low values of the CFL number $\sigma_{\Delta t} = \Delta t|\boldsymbol{v}|/\Delta x$, the Melson correction stretches the stability domain of the RK method further into the imaginary half plane to cover the eigenvalues of the Euler operator $\mathcal{L}$. For high CFL numbers, the eigenvalues are concentrated close to the imaginary axis and the Melson correction vanishes. These effects are achieved by making the RK scheme semi-implicit, depending on the CFL number. The computation of the RK5 stages $\check{V}^{(i)}$, $i = 1, \ldots, 5$, using the coefficients $[\alpha_1; \ldots; \alpha_5] = [0.0791451; 0.163551; 0.283663; 0.5; 1.0]$ proceeds as

$$\begin{aligned}
\check{V}^{(0)} &= \check{V}^{n-1},\\
(1+\alpha_s\lambda)\check{V}^{(i)} &= \check{V}^{(0)} + \alpha_s\lambda\left(\check{V}^{(i-1)} - \frac{1}{|\overline{\mathcal{K}}_e^n|}P^{-1}\mathcal{L}(\check{V}^{(i-1)}, \check{V}^{n-1})\right), \quad i = 1, \ldots, 5,
\end{aligned} \tag{28}$$

where $\lambda = \Delta\tau/\Delta t = \sigma_{\Delta t}/\sigma_{\Delta\tau}$ incorporates the influence of the physical and pseudo-time CFL numbers, $\sigma_{\Delta t}$ and $\sigma_{\Delta\tau} = \Delta\tau|\boldsymbol{v}|/\Delta x$, respectively. In practice, a pseudo-time CFL number $\sigma_{\Delta\tau}$ is prescribed and used to locally deduce the pseudo time step $\Delta\tau$.

The success of the pseudo-time iteration depends on the choice of the pseudo-time step $\Delta\tau$. Making it too small renders the method inefficient, while a too large value of $\Delta\tau$ leads to numerical instabilities. The analysis of Section 4 aims at finding suitable bounds for the pseudo-time CFL number and ways and means to allow large pseudo-time steps in different circumstances.

It remains to explain the premultiplication of the pseudo-time derivative $\partial\check{V}/\partial\tau$ with the matrix P in Eq. (27). Although the augmentation of the system (26) with the pseudo-time derivative is an artificial step and one could argue that it should work for any variable set directly, we found out that this is not necessarily true. In the original pseudo-time method, the conservation variables U are related to the residual $\mathcal{L}^n(U)$ in the ordinary differential equation (ODE) system (27). If, however, the left hand side of the equations contains a different variable set, the system can become very stiff and thereby hardly solvable, unless extremely small pseudo-time steps are used. This effect – occurring for $P = \mathcal{I}$, with $\mathcal{I}$ the identity matrix – can be undone by transforming to the conserved variable metric with $P = A_0^V$. As we proceed to show in Section 4, for regular A_0^V the resulting system has the same properties as the one for conservation variables. Unfortunately, the choice $P = A_0^V$ is not well-defined in the incompressible limit because then A_0^V is singular, independent of which variable set we use. To be able to solve the system also for the case of incompressible fluids, we apply Chorin's idea of artificial compressibility [25] in the matrix A_0^V, which has the following functional form for entropy variables,

$$A_0^V = \rho^2 T\begin{pmatrix} \beta_T & \beta_T v_i & \beta_T(h+k) - \alpha\alpha_p T\\ & \beta_T v_i v_j + \alpha\mathcal{I} & [\beta_T(h+k) - \alpha(\alpha_p T - 1)]v_j\\ \text{sym} & & (h+k)[\beta_T(h+k) - 2\alpha\alpha_p T] + \alpha(c_p T + 2k)\end{pmatrix}, \tag{29}$$

with $i,j = 1, \ldots, d$, [17]. It is easily seen that for an incompressible medium ($\alpha_p = 0$, $\beta_T = 0$) the first row and column contain only zeroes. To keep A_0^V regular, we set its top left entry to a positive value δ in case of (near-) incompressibility. Since we cannot build on available analysis for the magnitude of this artificial compressibility-like parameter, we have evaluated its impact by numerical experiment, cf. Section 4.3. It should be emphasized that the artificial compressibility is used only for the pseudo-time iteration. Unlike the direct application of the idea in the physical time derivative, the combination with the pseudo-time variable allows to obtain a time accurate solution [26]. In practice, the incorporation of P in (27) requires a premultiplication of the residual per element with P^{-1} or the solution of a $(d+2) \times (d+2)$ linear system with as many right hand sides as the number of basis functions used on the element.

4. Stability analysis

4.1. Description

One consequence of both the usage of a generalized variable set and the different equations of state is that the stability properties of the discretization change compared to the standard conservation variable/ideal gas case that is usually considered. Further investigation in the stability proved necessary, but the approach previously applied by Klaij et al. [14] – considering a scalar linear advection–diffusion equation and derive pertinent stability bounds – could not be applied here. The reason is that the numerical method investigated here contains the extra transformation between the entropy or pressure primitive variables and the conservative base set, which cannot be included in a single advection–diffusion equation. Therefore, we have numerically linearized the complete discrete Euler operator $\mathcal{L}^n$ from Eq. (26) with respect to the expansion coefficients $\check{V}^n$. The base state is a two-dimensional subsonic homogeneous flow (with constant density and temperature) diagonally over the domain $\overline{\mathcal{E}} = [0;1]^2$ with periodic boundary conditions in both space directions. All computations were carried out on a mesh of 20×20 equal-sized elements.

The analysis of the spectrum of the matrix $\partial\mathcal{L}^n/\partial\check{V}$ discloses two problems: firstly the use of other variable sets than the conservative variables generally affects the stiffness of the ODE system. This is demonstrated in Section 4.2 based on the ideal gas. Secondly, incompressibility causes the spectrum to concentrate along the imaginary axis, and we address this in Section 4.3.

4.2. Compressible media

4.2.1. Conservative variables

We first consider the spectrum of $\mathcal{L}^n$ for an ideal gas, using conservative variables U at the physical CFL numbers $\sigma_{\Delta t} = 1$ and 100. In Fig. 2, the eigenvalues are plotted in the complex plane together with the stability domain of the Runge–Kutta method RK5 defined in (28). The pseudo-time CFL number $\sigma_{\Delta\tau}$ is chosen such that the eigenvalues remain within the linear stability domain of RK5. Note the stretching of the stability domain along the negative real axis in the low CFL number case due to the Melson correction. This initial

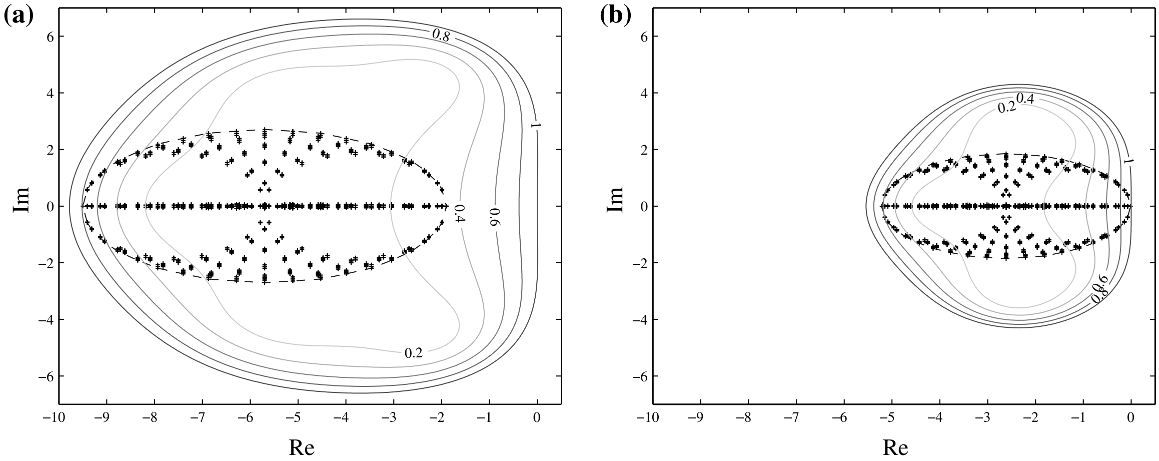

Fig. 2. Eigenvalues ('+') of the linearized Euler operator $\mathcal{L}^n$ for two different physical CFL numbers $\sigma_{\Delta t}$. The convex hull of the spectra appears as dashed line; the stability domain of the RK5 method is plotted with solid lines indicating the damping factor. The pseudo-time CFL number $\sigma_{\Delta\tau}$ has been adapted to locate all eigenvalues inside the stability domain of the RK5 method: (a) $\sigma_{\Delta t} = 1$, $\sigma_{\Delta\tau} = 1.9$; (b) $\sigma_{\Delta t} = 100$, $\sigma_{\Delta\tau} = 1.3$.

test serves to verify the results of Klaij et al. [14] about the stability of the pseudo-time iteration. Note that Klaij et al. obtained their results by applying a DG discretization to the advection–diffusion equation and inserting Fourier-modes, which is quite different from the approach taken here. Nevertheless the results are in the same range: where Klaij et al. gave the maximal pseudo-time CFL numbers $\sigma_{\Delta\tau} = 1.6$ and 1.8 for the physical CFL numbers $\sigma_{\Delta t} = 1$ and 100, respectively, the current method yields $\sigma_{\Delta\tau} = 1.9$ and 1.3. For practical cases, the pseudo-time step has to be limited more strictly owing to stronger nonlinearity and the presence of boundary conditions; for applications as those in Section 5, we use $\sigma_{\Delta\tau} = 0.8$–1.0. From now on, we discuss the effects of introducing a different variable set for the time-accurate case, $\sigma_{\Delta t} = 1$. The overall effects are the same for $\sigma_{\Delta t} = 100$. Mainly we consider constant basis functions, $p_t = p_s = 0$, but a comparison with higher order is given in Section 4.2.3.

4.2.2. Pressure primitive and entropy variables with $P = \mathcal{I}$

As the goal of our work is to use a variable set that has a well-defined incompressible limit we continue by examining the use of pressure primitive and entropy variables. Initially, we treat the nonlinear system as if we were solving directly for the concerning set of unknowns, hence $P = \mathcal{I}$ in Eq. (27). The spectra of the resulting Euler operators are shown in Fig. 3. Apparently, these are of little use for computational purposes: for Y-variables the spectrum extends into the (unstable) right half-plane. For V-variables, on the other hand, the eigenvalues stay in the left half plane, but extend extremely far away from the origin, outside the stability domain of Runge–Kutta methods for any reasonable pseudo-time step.

Hereby the need for restoring the metric of the conservation variables in the pseudo-time iteration becomes evident. This is equivalent to preconditioning with the inverse of the Jacobian of the transformation from entropy to conservation variables.

4.2.3. Pressure primitive and entropy variables with $P = A_0^V$

By using the Jacobian $A_0^V = \partial U / \partial V$ of the transformation $U \to V$ as the ‘preconditioner’ matrix P, we recover the same operator as in the conservation variable case. This is confirmed by the spectrum plotted in Fig. 4a. It was computed using entropy variables with the residual transformation by $(A_0^V)^{-1}$, and it coincides with the eigenvalue structure for U-variables in Fig. 2a. For pressure primitive variables the introduction of $P = A_0^Y$ leads to the same result.

Incidentally, we show a result for the eigenvalues when using bilinear basis functions in space ($p_s = 1$). Fig. 4 supports that the same bound for the pseudo-time CFL number $\sigma_{\Delta\tau} = 1.9$ that originated from the analysis for constant representation per element also holds for such a higher order computation.

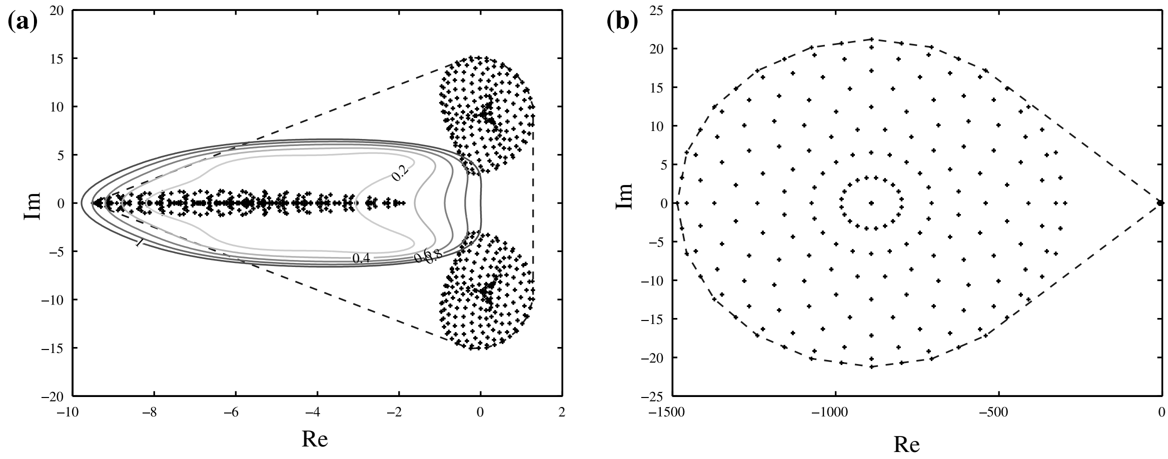

Fig. 3. Eigenvalues for pressure primitive and entropy variables with $P = \mathcal{I}$. Note the different scales on both real and imaginary axis: (a) Y-variables. The spectrum extends to the positive real axis. (b) V-variables. The problem becomes extremely stiff.

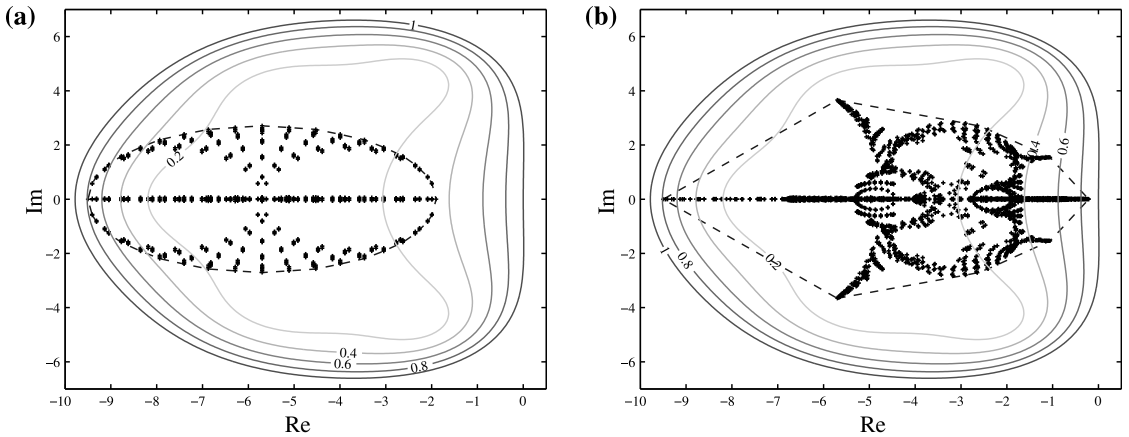

Fig. 4. Eigenvalue spectra for the method using entropy variables V but with restored conservation variable metric, $P = A_0^V$. Comparison for different spatial representation order: (a) $p_s = 0$, compare with Fig. 2a; (b) $p_s = 1$, i.e. four basis functions per element.

4.3. Incompressible media

In Section 3.4 we have already discussed that A_0^V is singular in the incompressible limit and how to avoid the related problems by the introduction of an artificial compressibility-like parameter δ in the matrix. Generally, the eigenvalues of the incompressible operator lie very close to the imaginary axis so that the pseudo-time step $\Delta\tau$ is limited primarily by the imaginary extent of the spectrum. When this is the case, the introduction of the artificial compressibility parameter δ in A_0^V can help alleviate the stability restriction. In Fig. 5a the convex hulls of the eigenvalue sets computed with different values of δ are plotted. The pseudo-time CFL number $\sigma_{\Delta\tau} = 0.016$ is the same for all shown sets. The figure substantiates that the extent in the imaginary direction is decreased by approximately a factor of 3 when setting $\delta = 1$ instead of $\delta = 0.001$. For $\delta > 1$ the effect saturates and no further advantage is gained. For smaller values of δ the eigenvalues spread out further along the imaginary axis. The introduction of δ does not degrade the (time) accuracy, because it affects only the pseudo-time process and the equations solved in physical time are still the original Euler equations.

Fig. 5b shows an alternative Runge–Kutta method, RK44 [27], which may help to overcome some of the trouble caused by the proximity of the eigenvalues to the imaginary axis. The stability domain of RK5 only touches the imaginary axis in the origin and then turns away from the axis, which may be a problem in the incompressible case. In contrast, the stability domain of RK44 runs along the imaginary axis and even con-

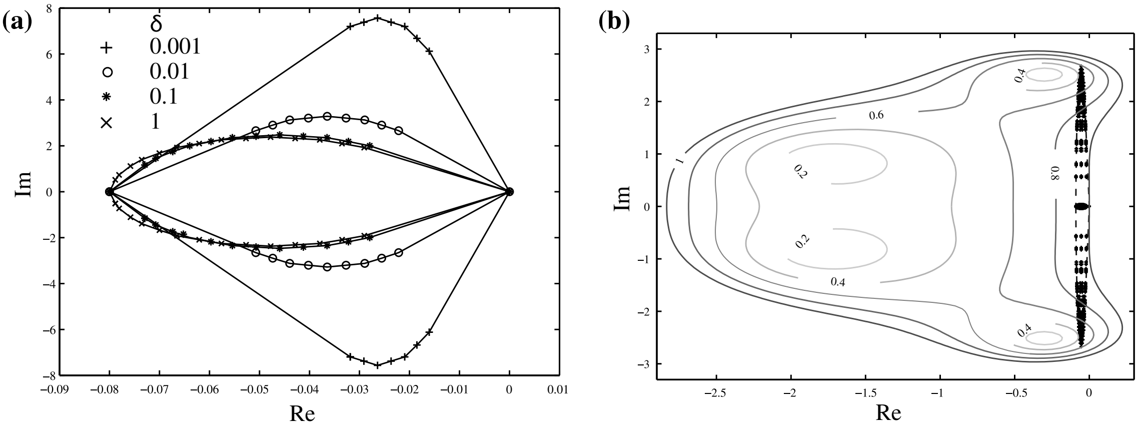

Fig. 5. Examination of the eigenvalue spectrum of the Euler operator $\mathcal{L}$ for entropy variables with $P = A_0^V$ and incompressible flow: (a) convex hulls of the eigenvalue spectra for different values of δ; (b) $\sigma_{\Delta t} = 1$, $\sigma_{\Delta\tau} = 0.018$; stability domain of the RK44 method.

tains a small part of the positive half plain. Thereby the eigenvalues are enclosed even if they are very close to the imaginary axis. The problem that the time step is limited by the extent along the imaginary axis, however, remains. Runge–Kutta methods that stretch out further in this direction typically have more stages and are more expensive to compute, so that no runtime advantage can be expected. Improvements hence should concern the structure of the spectrum, but further research is necessary on this topic. The numerical examples in Section 5 were all computed using the RK5 method.

5. Numerical examples

To complete the presentation of the numerical method, we present results of computations for several flow problems. The main emphasis is to demonstrate applicability for different media, e.g. compressible and incompressible fluids, when using entropy or primitive variables. The discretization uses linear in space and constant in time basis functions for the stationary solutions, unless stated otherwise. We start with a low Mach-number flow through a channel with a bump in Section 5.1. Supersonic flow is the topic of the following two test cases, of which one uses an ideal gas, cf. Section 5.2, and the other one a non-ideal gas, see Section 5.3. In Section 5.4, we provide results for compressible and incompressible flow around a circular cylinder. Finally, in Section 5.5, we add some remarks about the relative computational cost of using the different variable sets. For the supersonic gas flows we use the HLLC flux. In the compressible/incompressible comparison, for the described reasons, HLLC is replaced by the LLF flux. All computations are based on entropy variables unless explicitly mentioned otherwise. Our implementation is based on the general-purpose DG FEM software framework hpGEM [28].

5.1. Ideal gas: channel with a bump

We consider subsonic flow in a channel of unit width with a 10% circular bump in the central third of the simulated length [29]. The grid is boundary fitted with 63 × 22 cells. Results are given for the inflow Mach numbers $M_\infty = 1.0 \times 10^{-6}$, and $M_\infty = 0.5$. At the inflow, entropy, stagnation enthalpy and the flow angle are prescribed, while at the outflow only the pressure is specified [30]. Along the upper and lower channel wall, a slip flow boundary condition is used. All ideal gas tests assume the adiabatic index to be $\gamma = 1.4$.

For subsonic flow, the Mach number fields should be symmetric with respect to the middle of the channel ($x_1 = 1.5$). This is confirmed by the results in Fig. 6, with a small deviation for the higher inflow Mach number in the region downstream of the bump. The plots show the same contour levels as used by Bijl and Wesseling [29] for this test case, and very good agreement with their results is observed.

5.2. Ideal gas: oblique shock

To give evidence of the wide range of applicability of the developed method, we add a result for a compressible inviscid test case also presented in [4]: an oblique shock forming at an angle of 29.3° due to a Mach 2 inflow at an angle of 10° with respect to a slip wall at the lower edge of the unit square computational domain. The left and top boundaries of the domain are inflow boundaries. The right boundary is the outflow and along the base only $v_2 = 0$ is enforced.

The pressure field including the shock is plotted in Fig. 7a. Note that we are not using any form of discontinuity stabilization, hence the overshoot close to the shock. In this way, it is also reasonable to compare the results for different sets of variables, cf. Fig. 7b. Only minor differences can be observed, after all the equations that are solved are always the same set of conservation equations (3). The small differences that are visible arise because of the differing discrete solutions when expanding in terms of the various variable sets and due to the transformations between the conservative or entropy variables and the plotted variable p. Independent of the variable set the location of the shock is correctly represented.

5.3. Real gas: shock-tube problem

We add a test case for a real gas as described in Section 2.2.2. Toro [31] posed a shock-tube problem with the following initial data: left state $\rho_L = 1$, $u_L = 0$, $p_L = 1$, right state $\rho_R = 0.125$, $u_R = 0$, $p_R = 0.1$, separated

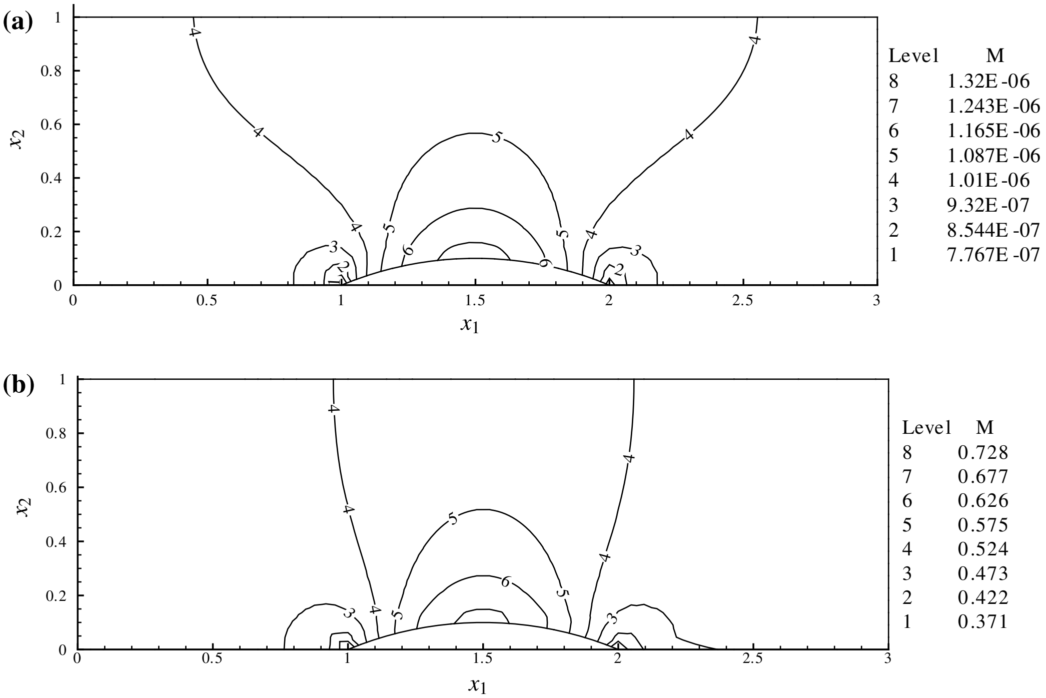

Fig. 6. Mach number fields in the channel with 10% circular bump, computed on a grid with 63 × 22 elements and $p_s = 1$: (a) $M_\infty = 1.0 \times 10^{-6}$; (b) $M_\infty = 0.5$.

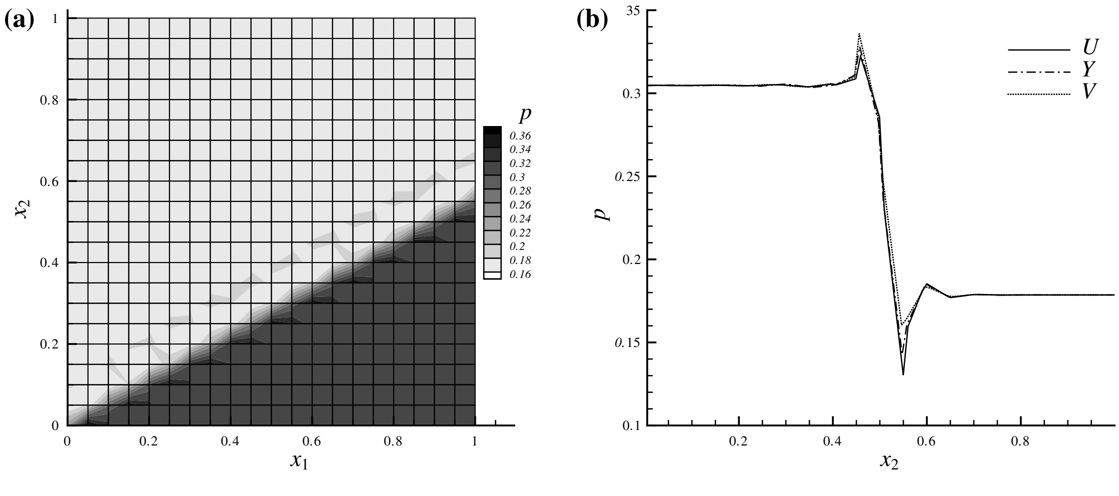

Fig. 7. Numerical results for the oblique shock test case, computed on a 20 × 20 mesh with $p_s = 1$: (a) pressure field computed using entropy variables; (b) comparison of the computed pressure fields along $x_1 = 0.9$ using different variable sets.

at $x = 0.5$, for a gas with $a = 0$, cf. Eq. (8b), but the (unrealistically high) covolume $b = 0.8$. The adiabatic index is $\gamma = 1.4$. As it is well-known that the performance of the LLF flux is inferior to Riemann-problem based methods on this kind of problems, we use the HLLC flux to obtain a result of comparable quality as with more specialized methods.

This test highlights the wide applicability of the generalized variable formulation of the Euler equations (here using V-variables) and our implementation. The solution of the shock-tube problem for a covolume gas differs substantially from the ideal gas case, cf. Fig. 8, which is reflected well by the numerical results.

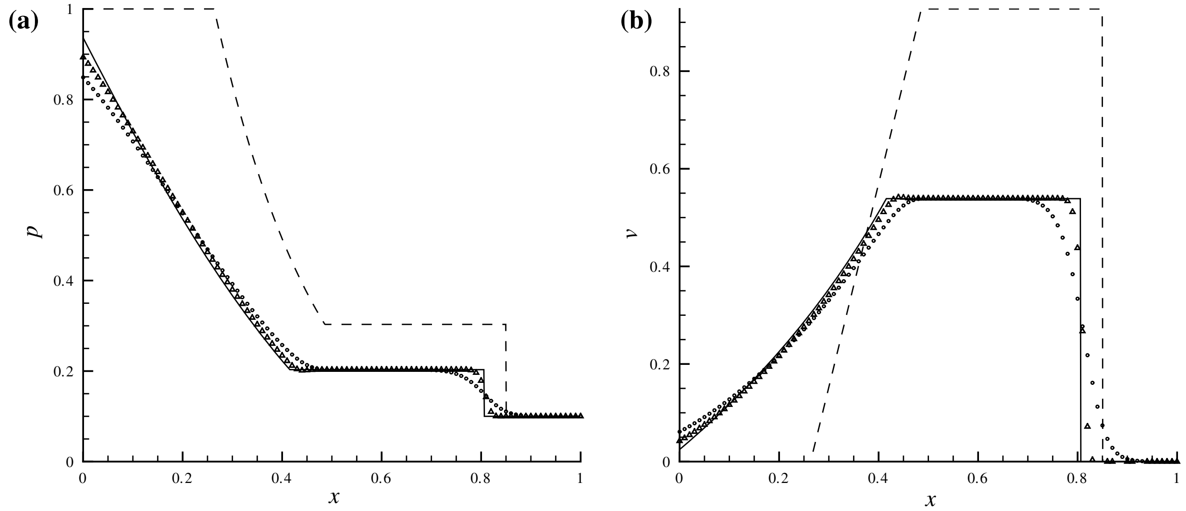

Fig. 8. Shock-tube problem with covolume gas. Symbol ○: computation on a mesh of 100 elements with $p_t = p_s = 0$, △: computed solution with linear basis functions, solid line: exact solution, dashed line: exact solution without covolume, i.e. for an ideal gas: (a) pressure p; (b) velocity v.

5.4. Compressible and incompressible flow around a circular cylinder

We consider the flow of an ideal gas (subsonic) and an incompressible fluid around a fixed circular cylinder with unit radius. The far-field flow is aligned with the x_1-axis, constant, and homogeneous: $\boldsymbol{v}_\infty = u_\infty \boldsymbol{e}_1$. In the limit of vanishing Mach number, $M_\infty \to 0$, the flow field is given by the classical potential flow solution. To conform to a previous study [32,33], we choose the free-stream Mach number of the ideal gas flow as $M_\infty = 0.38$. At the outer boundary, we prescribe the far-field state $p_\infty = 4.945$, $\boldsymbol{v} = \boldsymbol{v}_\infty$, $T = 1$, as the external state of the numerical flux. Due to the absence of viscosity in the Euler equations, a slip flow condition is applied on the inner cylinder surface: $\boldsymbol{v} \cdot \boldsymbol{n} = 0$. This test indicates the magnitude of the error of the numerical method in the compressible simulation by evaluating the total pressure loss π, defined as

$$\pi = 1 - \frac{p}{p_\infty}\left(\frac{1+\frac{1}{2}(\gamma-1)M^2}{1+\frac{1}{2}(\gamma-1)M_\infty^2}\right)^{\frac{\gamma}{\gamma-1}}. \tag{30}$$

The total pressure loss should be zero everywhere [32], deviations occur only due to the discrete approximation, which leads to entropy production at the surface of the cylinder.

Flow around a cylinder is also a test case for numerical methods for incompressible flow. The results are very close to those of low Mach number compressible computations.

The pressure field (normalized by the far-field pressure p_∞) and Mach contours for $M = 0.38$ according to a computation with entropy variables using linear basis functions per element on a mesh with 48 × 36 elements is shown in Fig. 9. Note that we plot a continuous solution by averaging the solution from neighboring elements in each node. The result should be contrasted with those presented by Bassi and Rebay in [33]: without superparametric elements, higher order discretization, and on a mesh with lower resolution than their 64 × 16 grid, we are able to compute a numerical solution without the substantial unsteady wake behind the cylinder. This is confirmed by a computation with doubled resolution, cf. Fig. 10.

Fig. 11 shows the total pressure loss along the cylinder surface for entropy variables using two different orders of the spatial representation (since this is a steady state case, we use $p_t = 0$ on all elements). The result using bilinear polynomials per element compares very well with the one obtained in [32] on a linear isoparametric mesh of comparable resolution. Results for different variable sets are hardly distinguishable in this test, so we do not present a comparison.

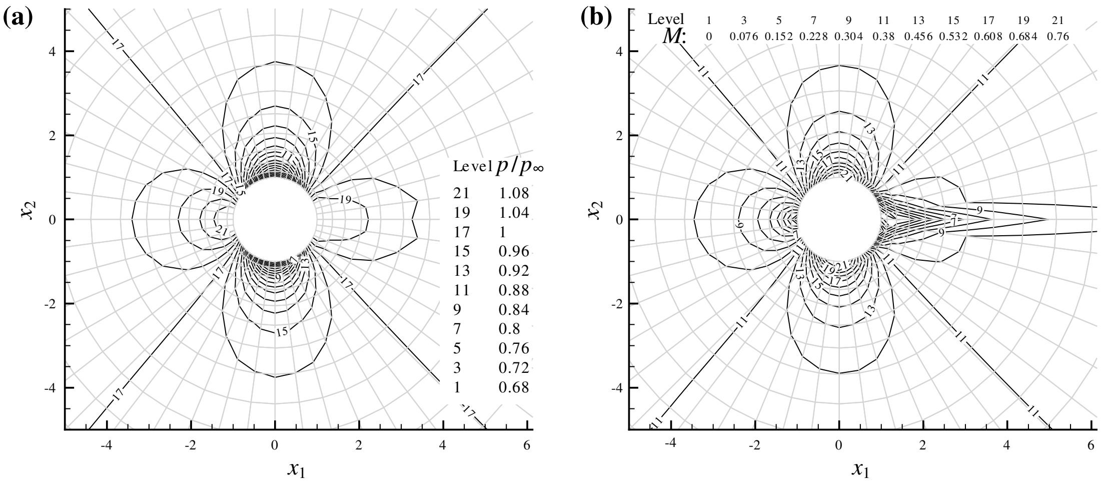

Fig. 9. Numerical results for the flow of an ideal gas at $M = 0.38$ around a circular cylinder of unit radius. The computation uses entropy variables on a 48×36 mesh that extends to the radius $r = 56.5$: (a) normalized pressure field; (b) local Mach number M.

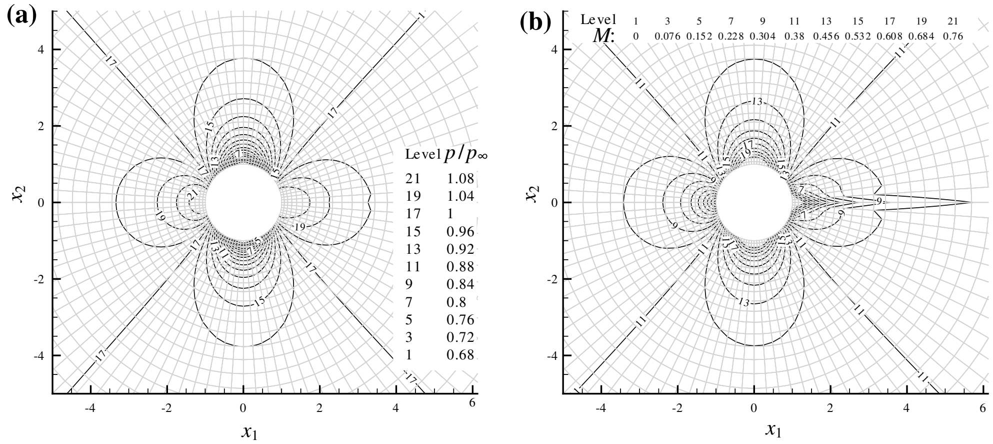

Fig. 10. Same test case as in Fig. 9 but computed on a 96×72 mesh: (a) normalized pressure field; (b) local Mach number M.

Repeating the same test with an incompressible fluid yields almost symmetric fields about $x_1 = 0$, cf. Fig. 12.

5.5. Runtime comparison

Finally, we give an indication of the relative cost of using the different variable sets for simulations of the Euler equations. It is quite difficult to compare on the level of a complete simulation as the magnitude and convergence of the residual $\mathcal{L}^n(V)$, which is used as stopping criterion for the pseudo-time iteration, differs between variable sets and it is not clear how to compare residual levels. Errors may be amplified by the discontinuous discretization, see Fig. 7b, so that again an exact comparison is hard to realize. What we can compare quite easily, however, is the cost per pseudo-time step, unrelated to its effect on the convergence. If we define the cost of a pseudo-time step using U-variables as 1.0, then the relative cost of one step with pressure primitive variables Y is approximately 1.0–1.1 and for entropy variables V it is in the range 1.6–1.7. These are indications that depend on several details of the numerical method, such as the numerical flux, the Runge–Kutta method, the order of the polynomial expansion per element, and the thermodynamical relations of

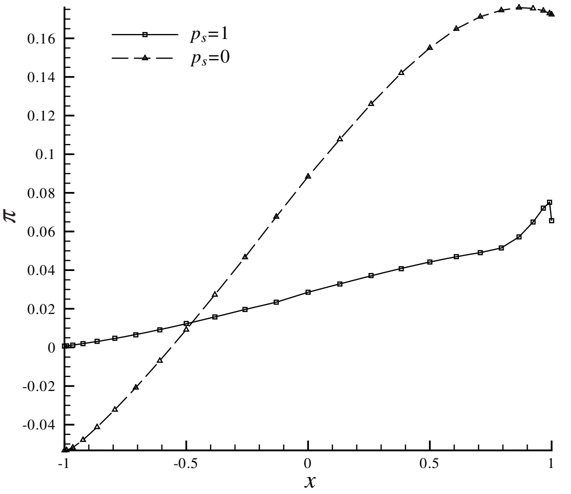

Fig. 11. Total pressure loss along the cylinder surface for two orders of polynomial expansion per element.

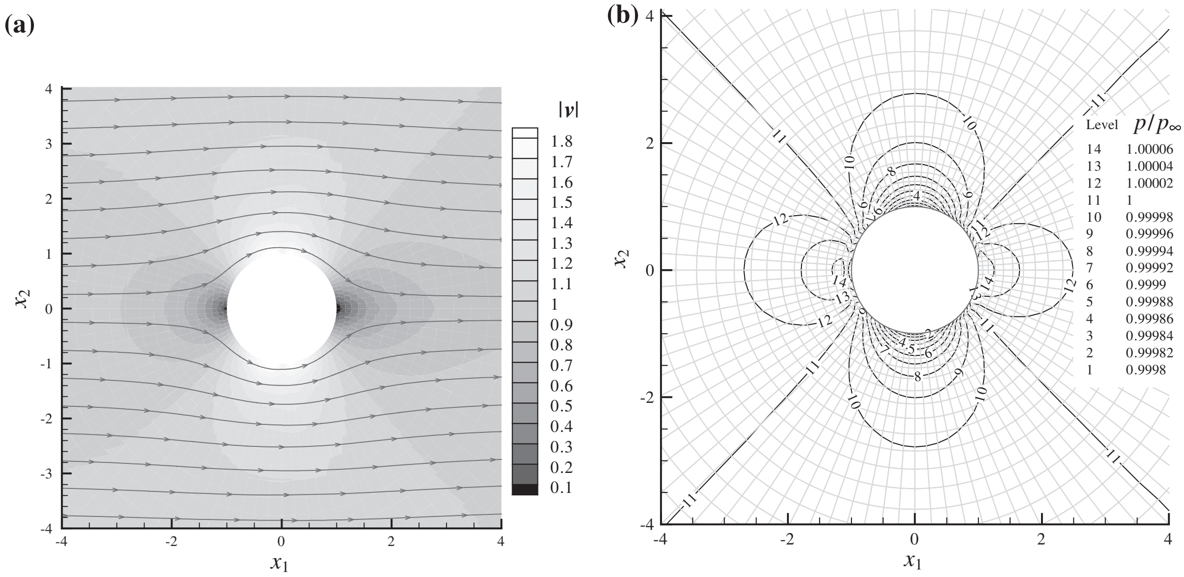

Fig. 12. Incompressible and low Mach number flow around a circular cylinder: (a) absolute value of the velocity and streamlines for incompressible flow around a circular cylinder; (b) normalized pressure field for $M = 0.01$ computed on a 96×72 grid.

the medium considered. Two contributions account for the extra cost: firstly the transformation that has to take place each time a certain set of quantities is used, most notably to conservation variables in the flux evaluations. Secondly, the residual transformation with $P = A_0^V$ introduces additional effort.

The cost of pressure primitive variable computations is surprisingly low, a pseudo-time step takes approximately the same time as one with conservative variables. Presumably, this effect is rooted in the extra computations that are necessary also when conservative variables are used (e.g. to obtain the pressure for the flux evaluation) and that the transformations for pressure primitive variables are comparably simple, mostly rational functions. For entropy variables, by contrast, the variable transformations typically use more complex functions (this depends on the thermodynamics of the medium of course). Furthermore, in our software the residual transformation is implemented as a linear system solution for entropy variables, but as a direct multiplication of the residuals with $(A_0^Y)^{-1}$ for pressure primitive variables. This difference is rooted in the simpler structure of A_0^Y and presumably accounts for a considerable part of the difference.

Ultimately, the consideration of the computational cost has to be combined with an appraisal of the benefits of different variable sets, cf. Sections 1 and 2.3. Entropy variables constitute an interesting choice because of their theoretical properties while a consideration solely based on computational efficiency suggests preferring pressure primitive variables.

6. Conclusions

We have described the extension of a discontinuous Galerkin finite element discretization of the Euler equations to allow using different sets of variables. In particular we have considered entropy variables, because of their theoretically appealing properties. Using entropy variables allows to treat compressible and incompressible flows with the same numerical method. Some parts of the solution algorithm have to be adapted, though. We have investigated the working of the pseudo-time stepping method used to solve the nonlinear algebraic system of equations. By transforming the residual and exploiting an artificial compressibility-like idea we find the method suitable for entropy and pressure primitive variables. We have presented several numerical test cases originating from different fields in compressible and incompressible fluid dynamics. Further work will aim at improving the efficiency of the nonlinear solution step using a multigrid algorithm [9,34] and the extension to viscous flows.

Acknowledgments

L. Pesch acknowledges support by the Technologiestichting STW through project TSF 6102 “Two phase flows with free surfaces”. J.J.W. van der Vegt acknowledges support through the national program BSIK in the ICT project BRICKS (http://www.bsik-bricks.nl), theme MSV1.

References

[1] H. Guillard, C. Viozat, On the behaviour of upwind schemes in the low Mach number limit, Comput. Fluids 28 (1) (1999) 63–86.

[2] H. Guillard, A. Murrone, On the behavior of upwind schemes in the low Mach number limit: II. Godunov type schemes, Comput. Fluids 33 (4) (2004) 655–675.

[3] S.K. Godunov, The problem of a generalized solution in the theory of quasilinear equations and in gas dynamics, Russ. Math. Surv. 17 (3) (1962) 145–156.

[4] G. Hauke, T.J.R. Hughes, A comparative study of different sets of variables for solving compressible and incompressible flows, Comput. Methods Appl. Mech. Eng. 153 (1998) 1–44.

[5] T.J.R. Hughes, L.P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics, Comput. Methods Appl. Mech. Eng. 54 (2) (1986) 223–234.

[6] T.J. Barth, An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Lecture Notes in Computational Science and Engineering, Ch. Numerical Methods for Gasdynamic Systems on Unstructured Meshes, Springer, 1999, pp. 195–285.

[7] F. Shakib, T.J.R. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 89 (1–3) (1991) 141–219.

[8] M. Polner, L. Pesch, J.J.W. van der Vegt, Construction of stabilization operators for Galerkin least-squares discretizations of compressible and incompressible flows, Comput. Methods Appl. Mech. Eng. 196 (21–24) (2007) 2431–2448.

[9] J.J.W. 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, J. Comput. Phys. 182 (2) (2002) 546–585.

[10] H. van der Ven, J.J.W. van der Vegt, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows II. Efficient flux quadrature, Comput. Methods Appl. Mech. Eng. 191 (41–42) (2002) 4747–4780.

[11] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations, J. Comput. Phys. 217 (2) (2006) 589–611.

[12] P. Houston, D. Schötzau, T.P. Wihler, An hp-adaptive mixed discontinuous Galerkin FEM for nearly incompressible linear elasticity, Comput. Methods Appl. Mech. Eng. 195 (25–28) (2006) 3224–3246.

[13] N.D. Melson, H.L. Atkins, M.D. Sanetrik, Time-accurate Navier–Stokes calculations with multigrid acceleration, Tech. Rep., NASA Langley Research Center, 1993.

[14] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Pseudo-time stepping methods for space–time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 219 (2) (2006) 622–643.

[15] W.L. Kleb, W.A. Wood, B. van Leer, Efficient multi-stage time marching for viscous flows via local preconditioning, in: Collection of Technical Papers, no. AIAA 99–3267, in: 14th Computational Fluid Dynamics Conference, Norfolk, VA, June 28–July 1, 1999, American Institute of Aeronautics and Astronautics (AIAA), 1999, pp. 1–14.

[16] J.V. Sengers, Equations of State for Fluids and Fluid Mixtures, Elsevier, 2000.

[17] M. Polner, Galerkin least-squares stabilization operators for the Navier–Stokes equations – a unified approach, Ph.D. Thesis, University of Twente (November 2005). URL: http://doc.utwente.nl/50848.

[18] P.W. Atkins, J. de Paula, Physical Chemistry, sixth ed., Oxford University Press, 2002.

[19] P. Colella, H.M. Glaz, Efficient solution algorithms for the Riemann problem for real gases, J. Comput. Phys. 59 (2) (1985) 264–289.

[20] C. Chalons, P.G. LeFloch, High-order entropy-conservative schemes and kinetic relations for van der Waals fluids, J. Comput. Phys. 168 (1) (2001) 184–206.

[21] G. Hauke, A. Landaberea, I. Garmendia, J. Canales, On the thermodynamics, stability and hierarchy of entropy functions in fluid flow, Comput. Methods Appl. Mech. Eng. 195 (33–36) (2006) 4473–4489.

[22] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, second ed., Springer, 1999.

[23] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1) (1994) 25–34.

[24] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, J. Comput. Phys. 141 (2) (1998) 199–224.

[25] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1) (1967) 12–26.

[26] W.Y. Soh, J.W. Goodrich, Unsteady solution of incompressible Navier–Stokes equations, J. Comput. Phys. 79 (1988) 113–134.

[27] J.R. Dormand, Numerical Methods for Differential Equations, CRC Press, 1996.

[28] L. Pesch, A. Bell, W.E.H. Sollie, V.R. Ambati, O. Bokhove, J.J.W. van der Vegt, hpGEM – a software framework for discontinuous Galerkin finite element methods, ACM Trans. Math. Software 33 (4) (2007).

[29] H. Bijl, P. Wesseling, A unified method for computing incompressible and compressible flows in boundary-fitted coordinates, J. Comput. Phys. 141 (2) (1998) 153–173.

[30] D.L. Darmofal, P. Moinier, M.B. Giles, Eigenmode analysis of boundary conditions for the one-dimensional preconditioned Euler equations, J. Comput. Phys. 160 (1) (2000) 369–384.

[31] E.F. Toro, A fast Riemann solver with constant covolume applied to the random choice method, Internat. J. Numer. Methods Fluids 9 (1989) 1145–1164.

[32] J.J.W. van der Vegt, H. van der Ven, Slip flow boundary conditions in discontinuous Galerkin discretizations of the Euler equations of gas dynamics, in: H.A. Mang, F. Rammerstorfer, J. Eberhardsteiner (Eds.), Proceedings Fifth World Congress on Computational Mechanics, 2002, pp. 1–16.

[33] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (2) (1997) 251–285.

[34] C.M. Klaij, M.H. van Raalte, J.J.W. van der Vegt, H. van der Ven, *h*-Multigrid for space–time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. in print.