Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 198 (2004) 727–746
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow ☆

V. Dolejší *, M. Feistauer

Charles University Prague, Faculty of Mathematics and Physics, Sokolovská 83, 186 75 Praha, Czech Republic

Received 3 September 2003; received in revised form 27 January 2004; accepted 28 January 2004
Available online 26 February 2004

Abstract

The paper is concerned with the numerical solution of an inviscid compressible flow with the aid of the discontinuous Galerkin finite element method. Since the explicit time discretization requires a high restriction of the time step, we propose semi-implicit numerical schemes based on the homogeneity of inviscid fluxes, allowing a simple linearization of the Euler equations which leads to a linear algebraic system on each time level. Numerical experiments performed for the Ringleb flow problem verify a higher order of accuracy of the presented method and demonstrate lower CPU-time costs in comparison with an explicit method. Then the method is tested on more complex unsteady Euler flows.

AMS: 65M60; 76N10

PACS: 02.60.Cb; 02.70.Dh

Keywords: Compressible Euler equations; Homogeneity of inviscid fluxes; Discontinuous Galerkin finite element method; Implicit backward Euler method; First- and second-order time discretization; Vijayasundaram numerical flux; Semi-implicit linearized numerical scheme; CFL-stability condition; Ringleb test problem; Experimental order of accuracy; Complex Euler flows

1. Introduction

Our goal is to develop a sufficiently accurate, efficient and robust numerical method for the solution of an inviscid compressible flow, which is described by the system of the *Euler equations*. These nonlinear conservation laws have solutions with discontinuities and their approximations by conforming finite

☆ This work was supported under the Grant Nos. 201/02/0684 and 201/00/D116 of the Czech Grant Agency and the Grant No. MSM 113200007 of the Ministry of Education of the Czech Republic.
* Corresponding author. Tel.: +420-2-2191-3373.
E-mail addresses: dolejsi@karlin.mff.cuni.cz (V. Dolejší), feist@karlin.mff.cuni.cz (M. Feistauer).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.01.023

elements (FE) suffer from the Gibbs phenomenon. From this point of view, it seems that for conservation laws with discontinuous solutions, the finite volume (FV) method [19,31] using piecewise constant approximations is more suitable, because the FV approximations are discontinuous on interelement interfaces, which allows better resolution of shock waves and contact discontinuities. On the other hand, the increase of accuracy in finite volume schemes applied on unstructured and/or anisotropic meshes seems to be problematic.

A combination of ideas and techniques of the FV and FE methods yields the *discontinuous Galerkin finite element method* (DGFEM) using advantages of both approaches and allowing to obtain schemes with a higher order accuracy in a natural way. DGFEM is based on the approximation of the solution of an initial-boundary value problem by piecewise polynomial functions over a finite element mesh without any requirement on interelement continuity. DGFEM was applied to nonlinear conservation laws already in 1989 by Cockburn and Shu [10]. It was used for the numerical simulation of the compressible Euler equations later by Bassi and Rebay in [2], where the space DG discretization is combined with explicit Runge–Kutta time discretization. In [4] Baumann and Oden describe an *hp* version of the space DG discretization with explicit time stepping to compressible flow. Van der Vegt and van der Ven apply space–time discontinuous Galerkin method to the solution of the Euler equations in [39,40]. Here the discrete problem is solved with the aid of a multigrid accelerated pseudo-time-integration. During several recent years the DGFE schemes have been extensively developed and become more and more popular. Some aspects of the DGFEM and applications to gas dynamics are discussed in [1,7,14–18,21]. For a survey see, e.g. [9,11].

In all cited works except [39,40] explicit Euler or Runge–Kutta time discretization is used. Explicit time stepping for the solution of the Euler equations is very popular particularly in the framework of the finite difference and finite volume schemes. Its advantage is a simple algorithmization. However, it requires to satisfy rather restrictive CFL-stability conditions, which is quite inconvenient over nonuniform unstructured anisotropic meshes. Therefore, it is suitable to consider implicit methods for the numerical solution of the Euler equations as well. It is well known that the use of implicit methods contributes to an improvement of the efficiency of numerical schemes for solving the Euler equations in some cases, because implicit methods permit to use longer time steps.

In the framework of the finite volume methods implicit schemes were used, for example in [26,32,37]. The drawback of the implicit schemes is the necessity to solve a large nonlinear algebraic system on each time level. To this end, the Newton method is often applied leading to a sequence of linear discrete problems. One variant of this approach is a well-known Δ-scheme by Beam and Warming [5,6] (see also [29]). This approach is often combined with multigrid techniques (see, e.g., [12,28,30]). The application of the Newton-like schemes requires of course the differentiability of the numerical flux and the computation of its partial derivatives, which is usually rather complicated. This is the reason that some authors use artificial pseudo-time-integration as was applied together with multigrid in [39,40] for the DG discrete problem. The multigrid techniques require, of course, the use of structured meshes and, in the case of the mesh refinement, a sequence of nested meshes. This is not the case when the anisotropic mesh adaptation (AMA) method is used. Then algebraic multigrid would have to be applied, but its efficiency is not so high. Therefore, one often uses the Krylov subspace methods for the solution of linear systems in linearized schemes for the Euler equations (cf., e.g., [32]).

The goal of this paper is to develop a sufficiently accurate, efficient and robust method for the numerical solution of the nonstationary Euler equations applicable on unstructured meshes obtained with the aid of the AMA technique. We propose semi-implicit DGFE schemes, based on the homogeneity of the inviscid fluxes and properties of the Vijayasundaram numerical flux, leading in a natural way to a linear system on each time level. This approach is combined with a first- or second-order time stepping. The linear algebraic systems are solved by the GMRES method. The described method is not based on the Newton linearization, does not require to differentiate the numerical flux and can be applied on arbitrary meshes. It is

practically unconditionally stable. This fact allows us to use the method also for the solution of the stationary Euler equations via time stabilization for "$t \to \infty$", using very large time steps.

The contents of the paper is the following. In Section 2, the initial-boundary value problem for the Euler equations is formulated and some properties of the Euler equations are mentioned. In Section 3, we carry out the discretization of the problem with the aid of a semi-implicit DGFEM and a first- or second-order time discretization. Section 4 contains a detailed description of the implementation of the method. Numerical examples demonstrating the accuracy and efficiency of the scheme are given in Section 5. The summary of results and outlook are contained in Section 6.

2. Governing equations

The system of the Euler equations describing 2D inviscid flow can be written in the form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = 0 \quad \text{in } Q_T = \Omega \times (0,T), \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain occupied by gas, $T > 0$ is the length of a time interval,

$$\boldsymbol{w} = (w_1, \ldots, w_4)^{\mathrm{T}} = (\rho, \rho v_1, \rho v_2, e)^{\mathrm{T}} \tag{2}$$

is the *state vector* and

$$\boldsymbol{f}_s(\boldsymbol{w}) = (f_s^1(w), \ldots, f_s^4(w)) = (\rho v_s, \rho v_s v_1 + \delta_{s1} p, \rho v_s v_2 + \delta_{s2} p, (e+p)v_s)^{\mathrm{T}}, \quad s = 1,2, \tag{3}$$

are the *inviscid* (*Euler*) *fluxes*. We use the following notation: ρ – density, p – pressure, e – total energy, $\boldsymbol{v} = (v_1, v_2)$ – velocity, δ_{sk} – Kronecker symbol (if $s = k$, then $\delta_{sk} = 1$, else $\delta_{sk} = 0$). The equation of state implies that

$$p = (\gamma - 1)(e - \rho \mid \boldsymbol{v} \mid^2 /2). \tag{4}$$

Here $\gamma > 1$ is the Poisson adiabatic constant.The system (1)–(4) is *hyperbolic*. It is equipped with the initial condition

$$\boldsymbol{w}(\boldsymbol{x}, 0) = \boldsymbol{w}^0(x), \quad x \in \Omega, \tag{5}$$

and the boundary conditions

$$B(\boldsymbol{w}) = 0 \quad \text{on } \partial\Omega \times (0,T), \tag{6}$$

chosen in such a way that problem (1)–(6) is linearly well–posed (see, e.g. [24, Section 3.3.6]). To this end, the boundary $\partial\Omega$ is formed by disjoint parts Γ_{IO} and Γ_{W} representing the inflow/outflow and impermeable walls, respectively. On Γ_{W} we prescribe the impermeability condition

$$\boldsymbol{v} \cdot \boldsymbol{n} = 0 \quad \text{on } \Gamma_{\mathrm{W}}, \tag{7}$$

where $\boldsymbol{n}$ denotes the unit outer normal to $\partial\Omega$. In order to determine boundary conditions on Γ_{IO}, we define the matrix

$$\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n}) := \sum_{s=1}^{2} \boldsymbol{A}_s(\boldsymbol{w}) n_s, \tag{8}$$

where $\boldsymbol{n} = (n_1, n_2) \in \mathbb{R}^2$, $n_1^2 + n_2^2 = 1$ and

$$\boldsymbol{A}_s(\boldsymbol{w}) = \frac{\mathrm{D}\boldsymbol{f}_s(\boldsymbol{w})}{\mathrm{D}\boldsymbol{w}}, \quad s = 1, 2, \tag{9}$$

are the Jaccobi matrices of the mappings $\boldsymbol{f}_s$. Then we prescribe m_n quantities characterizing the state vector $\boldsymbol{w}$, where m_n is the number of negative eigenvalues of the matrix $\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n})$ and extrapolate m_p quantities of $\boldsymbol{w}$ from interior of Ω, where $m_p = 4 - m_n$ is the number of nonnegative eigenvalues of $\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n})$. For details, see, e.g., [20] or [24].

Using relations (2)–(4), we express the fluxes $\boldsymbol{f}_s$, $s = 1, 2$, in terms of the variables $w_1, \ldots, w_4$ in the form

$$\boldsymbol{f}_s(\boldsymbol{w}) = \begin{pmatrix} w_{s+1} \\ \frac{w_{s+1}w_2}{w_1} + \delta_{s1}(\gamma - 1)\left(w_4 - \frac{w_2^2 + w_3^2}{2w_1}\right) \\ \frac{w_{s+1}w_3}{w_1} + \delta_{s2}(\gamma - 1)\left(w_4 - \frac{w_2^2 + w_3^2}{2w_1}\right) \\ \frac{w_{s+1}}{w_1}\left(\gamma w_4 - (\gamma - 1)\frac{w_2^2 + w_3^2}{2w_1}\right) \end{pmatrix}, \quad s = 1, 2. \tag{10}$$

Obviously, $\boldsymbol{f}_s$, $s = 1, 2$, are homogeneous mappings of order one, i.e.,

$$\boldsymbol{f}_s(\alpha \boldsymbol{w}) = \alpha \boldsymbol{f}_s(\boldsymbol{w}), \quad \alpha \in \mathbb{R}, \ \alpha \neq 0, \ i = 1, 2. \tag{11}$$

Then it is easy to show (see [20], p. 432) that

$$\boldsymbol{f}_s(\boldsymbol{w}) = \boldsymbol{A}_s(\boldsymbol{w})\boldsymbol{w}, \quad s = 1, 2. \tag{12}$$

3. Discretization

3.1. Broken Sobolev space

In what follows we use the standard notation for function spaces: $H^k(\Omega)$ is the Sobolev space and $C^1(0, T; X)$ is the space of continuously differentiable mappings on $[0, T]$ with values in X.

Let Ω_h be a polygonal approximation of Ω. In order to derive the discrete problem, we consider a mesh T_h of Ω_h consisting of various types of convex elements $K_i \in T_h$, $i \in I$ ($I \subset Z^+ = \{0, 1, 2, \ldots\}$ is a suitable index set), e.g., triangles, quadrilaterals or in general convex polygons.

By Γ_{ij} we denote a common edge between two neighbouring elements K_i and K_j. The symbol $\boldsymbol{n}_{ij} = ((n_{ij})_1, (n_{ij})_2)$ denotes the unit outer normal to ∂K_i on the side Γ_{ij}. Moreover, we set $s(i) = \{j \in I; K_j \text{ is a neighbour of } K_i\}$. The boundary $\partial\Omega_h$ is formed by a finite number of faces of elements K_i adjacent to $\partial\Omega_h$. We denote all these boundary faces by S_j, where $j \in I_b \subset Z^- = \{-1, -2, \ldots\}$. Now we set $\gamma(i) = \{j \in I_b; S_j \text{ is a face of } K_i \in T_h\}$ and $\Gamma_{ij} = S_j$ for $K_i \in T_h$ such that $S_j \subset \partial K_i$, $j \in I_b$. For K_i not containing any boundary face S_j we set $\gamma(i) = \emptyset$. Obviously, $s(i) \cap \gamma(i) = \emptyset$ for all $i \in I$. Now, if we write $S(i) = s(i) \cup \gamma(i)$, we have

$$\partial K_i = \bigcup_{j \in S(i)} \Gamma_{ij}, \quad \partial K_i \cap \partial\Omega_h = \bigcup_{j \in \gamma(i)} \Gamma_{ij}. \tag{13}$$

Moreover, we define the subsets $\gamma_{\mathrm{IO}}(i)$ and $\gamma_{\mathrm{W}}(i)$ of $\gamma(i)$ such that faces Γ_{ij}, $j \in \gamma_{\mathrm{IO}}(i)$, approximate Γ_{IO} and Γ_{ij}, $j \in \gamma_{\mathrm{W}}(i)$, approximate Γ_{W}.

We define the so-called *broken Sobolev space*

$$H^k(\Omega, T_h) = \{v; v\,|_K \in H^k(K) \ \forall K \in T_h\}. \tag{14}$$

If $v \in H^k(\Omega, T_h)$, then

$$v|_{\Gamma_{ij}} \neq v|_{\Gamma_{ji}}, \tag{15}$$

in general, where $v|_{\Gamma_{ij}}$ and $v|_{\Gamma_{ji}}$ denote the values of v on Γ_{ij} considered from the interior and the exterior of K_i, respectively.

In order to derive the discrete problem, we multiply (1) by a test function $\boldsymbol{\varphi} \in [H^1(\Omega, T_h)]^4$, integrate over any element $K_i, i \in I$, apply Green's theorem and sum over all $i \in I$. In this way we obtain the integral identity

$$\frac{\partial}{\partial t} \sum_{K_i \in T_h} \int_{K_i} \boldsymbol{w} \cdot \boldsymbol{\varphi} \, \mathrm{d}x = \sum_{K_i \in T_h} \int_{K_i} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x_s} \mathrm{d}x - \sum_{K_i \in T_h} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}) \cdot \boldsymbol{\varphi}(n_{ij})_s \, \mathrm{d}S, \tag{16}$$

which represents a weak form of the Euler equations in the sense of the broken Sobolev space $H^1(\Omega, T_h)$.

3.2. Numerical solution

Now we shall introduce the discrete problem approximating identity (16) with the aid of the discontinuous Galerkin finite element method.

To evaluate the boundary integrals in (16) we use the approximation

$$\int_{\Gamma_{ij}} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}(t))(n_{ij})_s \cdot \boldsymbol{\varphi} \, \mathrm{d}S \approx \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}(t)|_{\Gamma_{ij}}, \boldsymbol{w}(t)|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi} \, \mathrm{d}S, \tag{17}$$

where $\boldsymbol{H}$ is a *numerical flux*, $\boldsymbol{w}(t)|_{\Gamma_{ij}}$ and $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ are the values of $\boldsymbol{w}$ on Γ_{ij} considered from the interior and the exterior of K_i, respectively, and at time t. It is necessary to specify the meaning of $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ for $j \in \gamma(i)$. Taking into account the boundary conditions on Γ_{IO} described above and following [20, Section 7.3.43] or [24, Section 3.6.6], we prescribe m_n components of $\boldsymbol{w}$ on Γ_{ij} and extrapolate m_p components from the interior of K_i to Γ_{ij}. Thus, we define $\boldsymbol{w}|_{\Gamma_{ji}} = (w_1, w_2, w_3, w_4)|_{\Gamma_{ji}}$ by

$$w_l|_{\Gamma_{ji}} = \begin{cases} g_l & \text{if the } l\text{th component of } \boldsymbol{w} \text{ is prescribed,} \\ w_l|_{\Gamma_{ij}} & \text{if the } l\text{th component of } \boldsymbol{w} \text{ is extrapolated,} \end{cases} \quad l = 1, \ldots, 4, \;\; j \in \gamma_{\mathrm{IO}}(i), \tag{18}$$

where $\boldsymbol{g} = (g_1, g_2, g_3, g_4)$ is a given state vector. For details, see, e.g. [20] or [24].

If $j \in \gamma_{\mathrm{W}}(i)$, then we use the impermeability condition (7) and replace (17) by the approximation

$$\int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}(t)|_{\Gamma_{ij}}, \boldsymbol{w}(t)|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi} \, \mathrm{d}S := \int_{\Gamma_{ij}} \boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}(t), \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi} \, \mathrm{d}S, \quad j \in \gamma_{\mathrm{W}}(i), \tag{19}$$

where

$$\boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}, \boldsymbol{n}) \equiv (0, pn_1, pn_2, 0)^{\mathrm{T}}. \tag{20}$$

The pressure p is expressed in the form

$$p = (\gamma - 1)(w_4 - (w_2^2 + w_3^2)/(2w_1)), \tag{21}$$

following from (4) and (2) and extrapolated on Γ_{ij} from K_i and $\boldsymbol{n} = (n_1, n_2) = \boldsymbol{n}_{ij}$.

Let us note that the choice of a suitable numerical flux plays an important role, particularly in finite volume schemes. We can mention, e.g. the well-known Steger-Warming, Van Leer, Roe and Vijayasundaram numerical fluxes or the numerical fluxes based on the direct Riemann solver of the Euler equations (for more details and references, see [24] or [38]). As for efficiency and robustness, we can recommend the

Osher–Solomon numerical flux (see [25,33,36] or [24, Section 3.4]), which we applied with a great success, e.g. in [22,23].

An approximate solution is sought at each time instant t as an element of the space of discontinuous piecewise polynomial functions

$$S_h \equiv S^{p,-1}(\Omega_h, T_h) = \{v; v\,|_K \in P_p(K)\ \forall K \in T_h\}, \tag{22}$$

where $p \geqslant 0$ is an integer and $P_p(K)$ denotes the space of all polynomials on K of degree $\leqslant p$. For $\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in [S_h]^4$ we introduce the forms

$$(\boldsymbol{w}_h, \boldsymbol{\varphi}_h)_h = \int_{\Omega_h} \boldsymbol{w}_h(\boldsymbol{x}) \cdot \boldsymbol{\varphi}_h(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

$$\tilde{b}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) = -\sum_{K\in T_h} \int_K \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}_h(\boldsymbol{x})) \cdot \frac{\partial \boldsymbol{\varphi}_h(\boldsymbol{x})}{\partial x_s}\,\mathrm{d}\boldsymbol{x} + \sum_{K_i\in T_h} \sum_{j\in S(i)} \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}(t)|_{\Gamma_{ij}}, \boldsymbol{w}(\boldsymbol{x})|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}_h \,\mathrm{d}S. \tag{23}$$

We say that $\boldsymbol{w}_h$ is the *approximate solution* of (1), if it satisfies the conditions

$$\begin{aligned} &\text{(a)}\ \boldsymbol{w}_h \in C^1([0,T],[S_h]^4),\\ &\text{(b)}\ \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h)_h + \tilde{b}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0 \quad \forall \boldsymbol{\varphi}_h \in [S_h]^4\ \ \forall t \in (0,T),\\ &\text{(c)}\ \boldsymbol{w}_h(0) = \Pi_h \boldsymbol{w}^0, \end{aligned} \tag{24}$$

where $\Pi_h \boldsymbol{w}^0$ is the L^2-projection of $\boldsymbol{w}^0$ from the initial condition (5) on the space $[S_h]^4$. If we set $p = 0$, then we obviously obtain the finite volume method.

Relations (24), (b) represent a system of ordinary differential equations which can be solved by a suitable numerical method. Usually, Runge–Kutta schemes are applied. Then we get conditionally stable methods applicable under a severe restriction of the length of the time step due to the CFL-stability condition. Since we are interested in numerical schemes not suffering from this drawback, we shall start from the time discretization by the implicit *backward Euler method*. To this end, we consider a partition $0 = t_0 < t_1 < t_2 \cdots$ of the time interval $(0,T)$ and set $\tau_k = t_{k+1} - t_k$. We use the notation $\boldsymbol{w}_h^k$ for the approximation of $\boldsymbol{w}_h(t_k)$. Then the *discrete problem* reads: for each $k \geqslant 0$ find $\boldsymbol{w}_h^{k+1}$ such that

$$\begin{aligned} &\text{(a)}\ \boldsymbol{w}_h^{k+1} \in [S_h]^4,\\ &\text{(b)}\ \left(\frac{\boldsymbol{w}_h^{k+1} - \boldsymbol{w}_h^k}{\tau_k}, \boldsymbol{\varphi}_h\right)_h + \tilde{b}_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = 0 \quad \forall \boldsymbol{\varphi}_h \in [S_h]^4,\ \ k = 0,1,\ldots,\\ &\text{(c)}\ \boldsymbol{w}_h^0 = \Pi_h \boldsymbol{w}^0. \end{aligned} \tag{25}$$

Scheme (25) leads to a system of highly nonlinear algebraic equations whose numerical solution is rather complicated. In order to simplify the problem, in the following we shall linearize relation (24), (b) and obtain a linear system.

3.3. Linearization

By (23), for $\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h \in [S_h]^4$ we have

$$\tilde{b}_h(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = \underbrace{-\sum_{K\in T_h} \int_K \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}_h^{k+1}(\boldsymbol{x})) \cdot \frac{\partial \boldsymbol{\varphi}_h(\boldsymbol{x})}{\partial x_s}\,\mathrm{d}\boldsymbol{x}}_{=:\tilde{\sigma}_1} + \underbrace{\sum_{K_i\in T_h} \sum_{j\in S(i)} \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}}, \boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}_h \,\mathrm{d}S}_{=:\tilde{\sigma}_2}. \tag{26}$$

The individual terms $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ will be linearized separately. For $\tilde{\sigma}_1$, we use the property (12) of the Euler fluxes and use the approximation

$$\tilde{\sigma}_1 \approx \sigma_1 = \sum_{K\in T_h} \int_K \sum_{s=1}^{2} \boldsymbol{A}_s(\boldsymbol{w}_h^k(\boldsymbol{x}))\boldsymbol{w}_h^{k+1}(\boldsymbol{x}) \cdot \frac{\partial \boldsymbol{\varphi}_h(\boldsymbol{x})}{\partial x_s}\,\mathrm{d}\boldsymbol{x}. \tag{27}$$

The linearization of the term $\tilde{\sigma}_2$ can be carried out in a simple way, when $\boldsymbol{H}$ in (26) is chosen, for example, as the Vijayasundaram numerical flux, see [41,20, Section 7.3] or [24, Section 3.3.4]. The matrix $\boldsymbol{P}(\boldsymbol{w},\boldsymbol{n})$ defined by (8) is diagonalizable: there exist matrices $\boldsymbol{D}$ and $\boldsymbol{T}$ such that

$$\boldsymbol{P}(\boldsymbol{w},\boldsymbol{n}) = \boldsymbol{T}\boldsymbol{D}\boldsymbol{T}^{-1}, \quad \boldsymbol{D} = \mathrm{diag}(\lambda_1,\ldots,\lambda_4), \tag{28}$$

where $\lambda_1,\ldots,\lambda_4$ are the eigenvalues of $\boldsymbol{P}$. We define the "positive" and "negative" part of $\boldsymbol{P}$ by

$$\boldsymbol{P}^{\pm}(\boldsymbol{w},\boldsymbol{n}) = \boldsymbol{T}\boldsymbol{D}^{\pm}\boldsymbol{T}^{-1}, \quad \boldsymbol{D}^{\pm} = \mathrm{diag}(\lambda_1^{\pm},\ldots,\lambda_4^{\pm}). \tag{29}$$

Then the Vijayasundaram numerical flux reads

$$\boldsymbol{H}_{\mathrm{VS}}(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{n}) = \boldsymbol{P}^{+}\Big(\frac{\boldsymbol{w}_1+\boldsymbol{w}_2}{2},\boldsymbol{n}\Big)\boldsymbol{w}_1 + \boldsymbol{P}^{-}\Big(\frac{\boldsymbol{w}_1+\boldsymbol{w}_2}{2},\boldsymbol{n}\Big)\boldsymbol{w}_2. \tag{30}$$

The form of $\boldsymbol{H}_{\mathrm{VS}}$ offers the linearized approximation

$$\tilde{\sigma}_2 \approx \sum_{K_i\in T_h}\sum_{j\in S(i)} \int_{\Gamma_{ij}} \Big[\boldsymbol{P}^{+}\Big(\langle \boldsymbol{w}_h^k\rangle_{ij},\boldsymbol{n}_{ij}\Big)\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}} + \boldsymbol{P}^{-}\Big(\langle \boldsymbol{w}_h^k\rangle_{ij},\boldsymbol{n}_{ij}\Big)\boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}}\Big]\cdot\boldsymbol{\varphi}_h\,\mathrm{d}S, \tag{31}$$

where

$$\langle \boldsymbol{w}_h^k\rangle_{ij} \equiv \tfrac{1}{2}\Big(\boldsymbol{w}_h^k|_{\Gamma_{ij}} + \boldsymbol{w}_h^k|_{\Gamma_{ji}}\Big). \tag{32}$$

It is necessary to specify the meaning of $\boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}}$ for $\Gamma_{ij}\subset\partial\Omega_h$. If $j\in\gamma_{\mathrm{IO}}(i)$, then we replace $\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}}$ in (31) by the state $\boldsymbol{w}_h^k|_{\Gamma_{ij}}$ determined in (18). For $j\in\gamma_{\mathrm{W}}(i)$, in virtue of (19), we use the approximation

$$\boldsymbol{H}(\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}},\boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}},\boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}_h\,\mathrm{d}S \approx \int_{\Gamma_{ij}} \boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^{k+1},\boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}\,\mathrm{d}S, \quad j\in\gamma_{\mathrm{W}}(i), \tag{33}$$

where $\boldsymbol{F}_{\mathrm{W}}$ is given by (20). The vector $\boldsymbol{F}_{\mathrm{W}}$ is a nonlinear function of $\boldsymbol{w}$ and its linearization can be carried out in two ways.

(a) *Explicit way*: we simply put

$$\boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^{k+1},\boldsymbol{n}) \approx \tilde{\boldsymbol{F}}_{\mathrm{W}}(\boldsymbol{w}_h^k,\boldsymbol{w}_h^{k+1}) := \boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^k|_{\Gamma_{ij}},\boldsymbol{n}). \tag{34}$$

This term appears on the right hand side of the resulting system of algebraic equations.

(b) *Implicit way*: we use a linearization with the aid of the Taylor expansion as

$$\boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^{k+1},\boldsymbol{n}) \approx \tilde{\boldsymbol{F}}_{\mathrm{W}}(\boldsymbol{w}_h^k,\boldsymbol{w}_h^{k+1}) := \boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^k,\boldsymbol{n}) + D\boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w}_h^k,\boldsymbol{n})\big(\boldsymbol{w}_h^{k+1}-\boldsymbol{w}_h^k\big), \tag{35}$$

where

$$D\boldsymbol{F}_{\mathrm{W}}(\boldsymbol{w},\boldsymbol{n}) \equiv (\gamma-1)\begin{pmatrix} 0 & 0 & 0 & 0\\ (v_1^2+v_2^2)n_1/2 & -v_1n_1 & -v_2n_1 & n_1\\ (v_1^2+v_2^2)n_2/2 & -v_1n_2 & -v_2n_2 & n_2\\ 0 & 0 & 0 & 0\end{pmatrix} \tag{36}$$

is obtained by the differentiation of function $\boldsymbol{F}_{\mathrm{W}}$ given by (20) with respect to $\boldsymbol{w} = (w_1, \ldots, w_4)$. Here $\boldsymbol{n} = (n_1, n_2), v_j = w_{j+1}/w_1, j = 1, 2$.

In the case (a), $\tilde{\boldsymbol{F}}_{\mathrm{W}}$ is independent of $\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}}$, whereas in the case (b) it depends linearly on $\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}}$. The approach (a) is simpler for implementation, but the possibility (b) allows to use a significantly higher CFL-number in the stability condition specified in (50). In both cases (a) and (b) we put

$$\begin{aligned}\sigma_2 = &\sum_{K_i\in T_h}\sum_{j\in s(i)}\int_{\Gamma_{ij}}\left[\boldsymbol{P}^+\left(\langle \boldsymbol{w}_h^k\rangle_{ij}, \boldsymbol{n}_{ij}\right)\boldsymbol{w}_h^{k+1}|_{\Gamma_{ij}} + \boldsymbol{P}^-\left(\langle \boldsymbol{w}_h^k\rangle_{ij}, \boldsymbol{n}_{ij}\right)\boldsymbol{w}_h^{k+1}|_{\Gamma_{ji}}\right]\cdot\boldsymbol{\varphi}_h\,\mathrm{d}S\\ &+\sum_{K_i\in T_h}\sum_{j\in\gamma_{\mathrm{IO}}(i)}\int_{\Gamma_{ij}}\left[\boldsymbol{P}^+\left(\langle \boldsymbol{w}_h^k\rangle_{ij}, \boldsymbol{n}_{ij}\right)\boldsymbol{w}_h^{k}|_{\Gamma_{ij}} + \boldsymbol{P}^-\left(\langle \boldsymbol{w}_h^k\rangle_{ij}, \boldsymbol{n}_{ij}\right)\boldsymbol{w}_h^{k}|_{\Gamma_{ji}}\right]\cdot\boldsymbol{\varphi}_h\,\mathrm{d}S\\ &+\sum_{K_i\in T_h}\sum_{j\in\gamma_{\mathrm{W}}(i)}\int_{\Gamma_{ij}}\tilde{\boldsymbol{F}}_{\mathrm{W}}(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}\,\mathrm{d}S.\end{aligned}\tag{37}$$

Finally, we define the form

$$b_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = -\sigma_1 + \sigma_2,\tag{38}$$

where σ_1 and σ_2 are given by (27) and (37), respectively. The form b_h is linear with respect to the second and third variable. Using (25) and (38) we arrive at the following *semi-implicit linearized numerical scheme*: for each $k \geqslant 0$ find $\boldsymbol{w}_h^{k+1}$ such that

$$\begin{aligned}&\text{(a) } \boldsymbol{w}_h^{k+1}\in[S_h]^4,\\ &\text{(b) } \left(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h\right)_h + \tau_k b_h(\boldsymbol{w}_h^k, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h) = \left(\boldsymbol{w}_h^k, \boldsymbol{\varphi}_h\right)_h \quad \forall\boldsymbol{\varphi}_h\in[S_h]^4,\ \ k = 0, 1, \ldots,\\ &\text{(c) } \boldsymbol{w}_h^0 = \Pi_h\boldsymbol{w}^0.\end{aligned}\tag{39}$$

Scheme (39) is formally first-order accurate in time. In order to increase the accuracy of the time discretization, the following *two-step second-order version* of scheme (39) can be used: for each $k \geqslant 0$ find $\boldsymbol{w}_h^{k+1}$ such that

$$\begin{aligned}&\text{(a) } \boldsymbol{w}_h^{k+1}\in[S_h]^4,\\ &\text{(b) } \frac{2\tau_k+\tau_{k-1}}{\tau_k(\tau_k+\tau_{k-1})}\left(\boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h\right)_h + b_h(\tilde{\boldsymbol{w}}_h^{k+1}, \boldsymbol{w}_h^{k+1}, \boldsymbol{\varphi}_h)\\ &\qquad = \frac{\tau_k+\tau_{k-1}}{\tau_k\tau_{k-1}}\left(\boldsymbol{w}_h^{k}, \boldsymbol{\varphi}_h\right)_h - \frac{\tau_k}{\tau_k(\tau_k+\tau_{k-1})}\left(\boldsymbol{w}_h^{k-1}, \boldsymbol{\varphi}_h\right)_h \quad \forall\boldsymbol{\varphi}_h\in[S_h]^4,\ \ k = 0, 1, \ldots,\\ &\text{(c) } \boldsymbol{w}_h^0 = \Pi_h\boldsymbol{w}^0,\ \ \boldsymbol{w}_h^{-1} = \boldsymbol{w}_h^0,\end{aligned}\tag{40}$$

where

$$\tilde{\boldsymbol{w}}_h^{k+1} = \frac{\tau_k+\tau_{k-1}}{\tau_{k-1}}\boldsymbol{w}_h^k - \frac{\tau_k}{\tau_{k-1}}\boldsymbol{w}_h^{k-1}.\tag{41}$$

This scheme is obtained by using the approximation

$$\left.\frac{\partial\boldsymbol{w}}{\partial t}\right|_{t=t_{k+1}} \approx \frac{2\tau_k+\tau_{k-1}}{\tau_k(\tau_k+\tau_{k-1})}\boldsymbol{w}_h^{k+1} - \frac{\tau_k+\tau_{k-1}}{\tau_k\tau_{k-1}}\boldsymbol{w}_h^k + \frac{\tau_k}{\tau_k(\tau_k+\tau_{k-1})}\boldsymbol{w}_h^{k-1}\tag{42}$$

and the approximation (41) in the nonlinear part of the form b_h. The extension of the algorithmization from scheme (39) to (40) does not represents any difficulties and any significant increase of CPU time.

In case that the time step is constant, i.e. $\tau_k = \tau$ for all $k = 0, 1, \ldots$, formulae (42) and (41) reduce to standard approximations

$$\left.\frac{\partial \boldsymbol{w}}{\partial t}\right|_{t=t_{k+1}} \approx \frac{3\boldsymbol{w}_h^{k+1} - 4\boldsymbol{w}_h^k + \boldsymbol{w}_h^{k-1}}{2\tau_k} \tag{43}$$

and

$$\boldsymbol{w}_h^{k+1} \approx \tilde{\boldsymbol{w}}_h^{k+1} = 2\boldsymbol{w}_h^k - \boldsymbol{w}_h^{k-1}, \tag{44}$$

respectively. Then (40), (b) is simplified in an obvious way.

As we see, the above schemes (39) and (40) are not of the Δ-scheme type. In schemes combined with the realization (34) of the impermeable boundary conditions one does not need to express any Jacobian matrix. In the case (35) with a more accurate realization of the boundary conditions on an impermeable wall, it is necessary to evaluate the Jacobian matrix (36), which is quite simple.

3.4. Matrix representation

Let

$$\left\{\boldsymbol{\varphi}_{imn} \in [S_h]^4;\ n = 1, \ldots, 4, m = 1, \ldots, \mathrm{DOF}(p), i \in I\right\} \tag{45}$$

be a basis of $[S_h]^4$. Here $\mathrm{DOF}(p)$ denotes the number of degrees of freedom of functions $\phi \in S_h$ on one element $K \in T_h$. For example, $\mathrm{DOF}(0) = 1$ for a piecewise constant approximation (finite volume method), $\mathrm{DOF}(1) = 3$ for a piecewise linear approximation, $\mathrm{DOF}(2) = 6$ for a piecewise quadratic approximation, etc. Then the dimension of the space $[S_h]^4$ is $N = 4 \cdot \mathrm{DOF}(p) \cdot \#T_h$, where $\#T_h$ denotes the number of elements of T_h. The state vector $\boldsymbol{w}_h^k$ can be written in the form

$$\boldsymbol{w}_h^k(\boldsymbol{x}) = \sum_{i \in I} \sum_{m=1}^{\mathrm{DOF}(p)} \sum_{n=1}^{4} \alpha_{imn}^k \boldsymbol{\varphi}_{imn}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,\ \ k = 0, 1, \ldots, \tag{46}$$

where $\alpha_{imn}^k \in \mathbb{R}$. Then the numerical schemes (39) and (40) have the following matrix representations

$$\boldsymbol{M}(\boldsymbol{\alpha}^k)\boldsymbol{\alpha}^{k+1} = \boldsymbol{g}(\boldsymbol{\alpha}^k), \tag{47}$$

and

$$\overline{\boldsymbol{M}}(\boldsymbol{\alpha}^{k-1}, \boldsymbol{\alpha}^k)\boldsymbol{\alpha}^{k+1} = \overline{\boldsymbol{g}}(\boldsymbol{\alpha}^{k-1}, \boldsymbol{\alpha}^k), \tag{48}$$

respectively, where

$$\boldsymbol{\alpha}^k = \{\alpha_{imn}^k\}_{n=1,\ldots,4, m=1,\ldots,\mathrm{DOF}(p), i \in I} \in \mathbb{R}^{\mathbb{N}}, \tag{49}$$

$\boldsymbol{g} : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$, $\overline{\boldsymbol{g}} : \mathbb{R}^{2\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ and $\boldsymbol{M}(\boldsymbol{\alpha})$, $\overline{\boldsymbol{M}}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are $N \times N$ matrices for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{\mathbb{N}}$. The solutions of (47) and (48) can be obtained by a suitable solver for sparse nonsymmetric systems. (For the application of various linear solvers to the solution of inviscid flow with the finite volume method, see [32].)

In order to guarantee the stability of schemes (39) and (40), we use the following CFL condition

$$6\tau_k \max_{K_i \in T_h} \frac{1}{|K_i|} \left(\max_{j \in S(i)} |\Gamma_{ij}| \lambda^{\max}_{\boldsymbol{P}(\boldsymbol{w}_h^k|_{\Gamma_{ij}}, \boldsymbol{n}_{ij})} \right) \leqslant \mathrm{CFL}, \tag{50}$$

where $|K_i|$ denotes the area of K_i, $|\Gamma_{ij}|$ the length of the edge Γ_{ij}, CFL a given constant and $\lambda^{\max}_{\boldsymbol{P}(\boldsymbol{w}_h^k|_{\Gamma_{ij}},\boldsymbol{n}_{ij})}$ is the maximal eigenvalue of the matrix $\boldsymbol{P}(\boldsymbol{w}_h^k|_{\Gamma_{ij}},\boldsymbol{n}_{ij})$ defined in (8), where the maximum is taken over Γ_{ij}. The condition (50) is similar to the stability condition widely used in the finite volume method, obtained on the basis of linearization and in analogy with a scalar problem, see [20, Section 7.3] or [24, Section 3.3.7]. For a detailed treatment of various approaches to the stability investigation, see also [42, Chapters 5 and 9]. In contrast to explicit schemes, for which we have to choose CFL < 1 [15], the semi-implicit scheme (39) allows us to use CFL constant up to CFL ≈ 6 for the linearization (34) and CFL ≈ 100 for the linearization (35), as is established by numerical experiments, see Section 5. Hence, the semi-implicit linearized schemes allow us to choose much longer time step than the explicit scheme. Of course, in the solution of an unsteady flow, it is necessary to make a compromise between the large CFL (allowing a long time step) and the accuracy of the time discretization (cf. Section 5.2).

4. Implementation

The computations presented in this paper were performed on triangular grids with the aid of piecewise linear approximations, i.e. with $p = 1$ in (22). (Algorithmization of higher-degree approximations, i.e. $p \geqslant 2$, will be treated separately in the framework of a package in preparation.)

The volume integrals in (27) are evaluated by the three point integration rule

$$\int_{K_i} z(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \frac{1}{3}|K_i| \sum_{j\in S(i)} z(Q_{ij}), \tag{51}$$

where $|K_i|$ denotes the area of K_i and $Q_{ij}, j \in S(i)$, denotes the midpoint of the edge Γ_{ij}. This integration rule is exact for second degree polynomials. The integrals in (37) are evaluated by the two points Gauss quadrature rule

$$\int_{\Gamma_{ij}} z\,\mathrm{d}S \approx \frac{|\Gamma_{ij}|}{2}\left(z(Q_{ij}^1) + z(Q_{ij}^2)\right), \tag{52}$$

where $|\Gamma_{ij}|$ denotes the length of the edge Γ_{ij} and Q_{ij}^1 and Q_{ij}^2 are integration points lying on Γ_{ij}. Let Γ_{ij} be parameterized in the form

$$\Gamma_{ij} = \left\{\boldsymbol{x} \in \mathbb{R}^2, \boldsymbol{x} = \boldsymbol{c}_{ij} + s(\boldsymbol{d}_{ij} - \boldsymbol{c}_{ij}), s \in [0,1]\right\}, \tag{53}$$

where $\boldsymbol{c}_{ij}$ and $\boldsymbol{d}_{ij}$ are endpoints of Γ_{ij}. Then

$$\begin{aligned} Q_{ij}^1 &= \boldsymbol{c}_{ij} + \left(1 - \sqrt{3}/3\right)(\boldsymbol{d}_{ij} - \boldsymbol{c}_{ij})/2, \\ Q_{ij}^2 &= \boldsymbol{c}_{ij} + \left(1 + \sqrt{3}/3\right)(\boldsymbol{d}_{ij} - \boldsymbol{c}_{ij})/2. \end{aligned} \tag{54}$$

The approximation (52) is exact for polynomials of the third degree. As we see, for each edge $\Gamma_{ij}, j \in S(i), i \in I$, we have to evaluate the state vector $\boldsymbol{w}$ at three integration nodes: Q_{ij}, Q_{ij}^1 and Q_{ij}^2, see Fig. 1.

There are two natural possibilities how to choose the basis (45) of the space $[S_h]^4$. First, let us consider basis functions associated with *vertices of the triangulation* T_h. Then we can write

$$\begin{aligned} &\boldsymbol{\varphi}_{imn} = (\phi_{im}\delta_{n1}, \ldots, \phi_{im}\delta_{n4}), \quad \phi_{im} \in S_h, \quad \phi_{im}(P_{i'}^{m'}) = \delta_{ii'}\delta_{mm'}, \quad m' = 1,2,3,\ i' \in I, \\ &\quad n = 1,\ldots,4,\ \ m = 1,2,3,\ \ i \in I, \end{aligned} \tag{55}$$

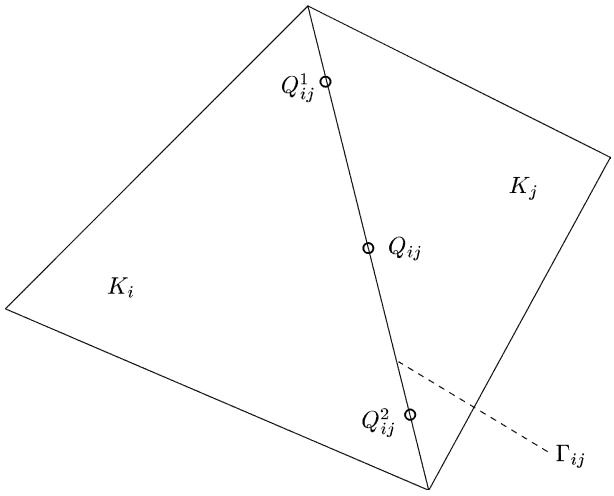

Fig. 1. Integration nodes Q_{ij}, Q_{ij}^1 and Q_{ij}^2 of the edge Γ_{ij}.

where $P_i^m, m = 1, 2, 3$, are vertices of the triangle K_i. The use of the basis (55) leads to a 7-point stencil for every test function and hence, the matrix $\boldsymbol{M}$ has at most 28 nonzero elements in each row, see Fig. 2.

The second possibility is to consider basis functions, whose components are piecewise linear functions associated with *midpoints of edges of triangles*:

$$\begin{aligned}&\boldsymbol{\varphi}_{imn} = (\phi_{im}\delta_{n1}, \ldots, \phi_{im}\delta_{n4}), \quad \phi_{im} \in S_h, \quad \phi_{im}(Q_{i'}^{m'}) = \delta_{ii'}\delta_{mm'}, \quad m' = 1, 2, 3, \ i' \in I,\\ &\quad n = 1, \ldots, 4, \ \ m = 1, 2, 3, \ \ i \in I,\end{aligned} \tag{56}$$

where $Q_i^m, m = 1, 2, 3$, are midpoints of edges of element K_i (of course, $Q_i^m = Q_{ij}$ for some $j \in S(i)$). The choice (56) leads to a 12-points stencil for any test function and the matrix $\boldsymbol{M}$ has at most 48 nonzero elements in each row, see Fig. 2.

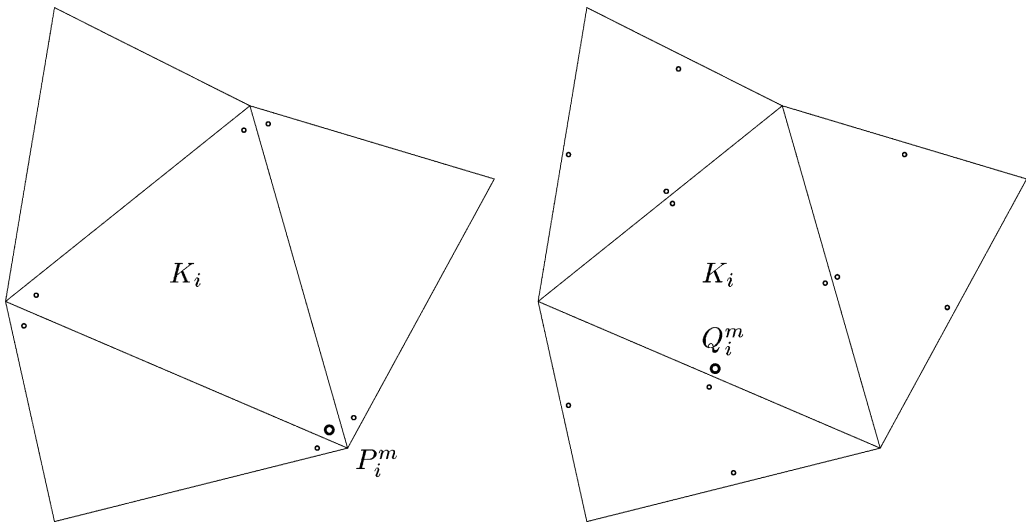

Fig. 2. Seven point stencil of a node P_i^m (left) and twelve point stencil of a node Q_i^m.

As we see, the choice (55) produces significantly smaller number (about 57%) of nonzero elements than the choice (56). This is caused by the fact that each function ϕ_{im} from (55) has a zero trace along one edge of K_i and, moreover, for the evaluation of $\boldsymbol{w}$ at Q_{ij}^1 and Q_{ij}^2 only the values of $\boldsymbol{w}$ at endpoints of Γ_{ij} are used (not the value at the third vertex of K_i, see Fig. 2).

The linear system (47) is solved by the GMRES method [34], in which the vector $\boldsymbol{\alpha}^k$ is taken as an initial approximation of $\boldsymbol{\alpha}^{k+1}$. Since we solve a time dependent problem, the vectors $\boldsymbol{\alpha}^k$ and $\boldsymbol{\alpha}^{k+1}$ are close to each other and, therefore, only few GMRES iterations at each time step are necessary (see Table 3). This is the reason that GMRES method can be used without any preconditioning. The computational cost for the preparation of matrices $\boldsymbol{M}$ and $\overline{\boldsymbol{M}}$ (i.e. evaluation of boundary and volume integrals) are approximately the same as the computational cost for the solution of (47) and (48), respectively.

Condition (50) guarantees the stability of the scheme with $\mathrm{CFL} \gg 1$. On the other hand, the computations start often from nonphysical data in the initial condition (5) (for example, we choose, $\boldsymbol{w}^0(\boldsymbol{x})$ as a constant vector for all $\boldsymbol{x} \in \Omega$). Consequently, it is suitable to start the computational process with a smaller CFL-number (in order to avoid physically unacceptable situations, as, for example, negative pressure) and then, step by step to increase CFL. Namely, in our computations we put

$$\mathrm{CFL}(t) \approx \mathrm{CFL} - (\mathrm{CFL} - 1)\exp(-ct), \tag{57}$$

where $\mathrm{CFL}(t)$ is the CFL-number used in (50) at time t, $\mathrm{CFL} \gg 1$ is a chosen fixed value and $c = 0.2$. Relation (57) implies that we start with $\mathrm{CFL}(0) = 1$ and $\mathrm{CFL}(t)$ grows up to CFL fast.

Numerical experiments show (see, e.g. [2,3,15,27]) that in order to obtain a physically admissible numerical solution, the use of superparametric finite elements is required. This means that the elements K adjacent to a curved boundary have to be approximated by elements $\tilde{K}$ which are images of a reference element $\hat{K}$ in a polynomial mapping $\zeta : \hat{K} \overset{\text{onto}}{\rightarrow} K$ having a higher degree than the degree of functions from the space S_h. Namely, in our case of linear finite elements, it is sufficient to employ a bilinear mapping ζ, see [15].

If the sought solution contains discontinuities (shock waves or contact discontinuities), then overshoots and undershoots appear in the DGFE solution near the discontinuities. To avoid this phenomenon, it is necessary to decrease the order of accuracy of the method choosing $p = 0$ in the vicinity of the discontinuities. One possibility, which is quite reliable, is to use the automatic adaptive limiting of order of accuracy proposed in [18]. Its implementation to schemes (39) and (40) is quite easy.

5. Numerical examples

In this section we present the solution of some test problems demonstrating the accuracy and efficiency of the proposed methods. We use the Euler equations written in the dimensionless form which is formally the same as (1).

5.1. Ringleb flow

In order to illustrate the accuracy of the schemes and to show that large CFL-numbers can be used, we test the schemes on the Ringleb flow problem. In this test case an analytical smooth steady-state solution of the Euler equations was obtained with the aid of the "hodograph method", see [8]. This problem represents a transonic channel flow, which is mostly subsonic, with a small supersonic region near the right-hand side wall, as shown in Fig. 3. Our aim is to obtain the approximate DG steady-state solution and compare it with the analytical one. On both walls we prescribed the impermeability condition $\boldsymbol{v} \cdot \boldsymbol{n} = 0$ where $\boldsymbol{n}$ denotes

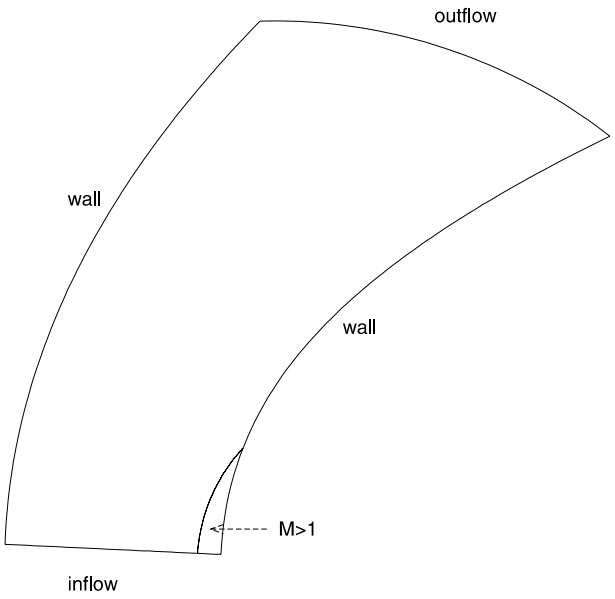

Fig. 3. Geometry for Ringleb flow problem with marked supersonic region.

the outer normal to $\partial\Omega$. On the inflow and outflow parts of the boundary we prescribe the exact solution. As an initial condition (5) we have chosen a constant vector $\boldsymbol{w}^0 = (0.5, 0, 0, 0.5)$.

The steady-state solution is achieved by a time stabilization method for "$t \to \infty$" with the aid of the forward Euler time discretization and the semi-implicit method (39) using linearizations (34) and (35) of fluxes. The stopping criterion is given by

$$\max\left(1, \frac{1}{\tau_k}\right) \left\| \rho_h^{k+1} - \rho_h^k \right\|_{L^1(\Omega)} \leqslant \mathrm{TOL}, \tag{58}$$

where ρ_h^{k+1} and ρ_h^k denote the density at time levels t_{k+1} and t_k, respectively, $\tau_k = t_{k+1} - t_k$ and TOL is a given tolerance. We put $\mathrm{TOL} = 10^{-6}$ for computations presented in this paper. For this case the final physical time was about $t \approx 70$. Since we seek only steady-state solution it is sufficient to use scheme (39).

The goal of our numerical experiments is to establish the order of accuracy and to compare the efficiency (CPU time) of the presented methods. The computations are performed for four triangular grids $T_{h_l}, l = 1, \ldots, 4$, having 5×10, 10×20, 20×40 and 40×80 vertices, see Fig. 4. Since the corresponding isolines of Mach number obtained by all mentioned methods are identical, we show only results obtained by the semi-implicit scheme with the linearization (35), which are viewed on Fig. 5. We observe a very smooth resolution although the discontinuous approximation was employed. Table 1 shows the computational errors e_h for explicit as well as semi-implicit methods, defined by

$$e_h \equiv \left\| \boldsymbol{w}_h - \boldsymbol{w} \right\|_{[L^2(\Omega_h)]^4}, \tag{59}$$

where $\boldsymbol{w}$ is the exact solution of the Ringleb flow problem and $\boldsymbol{w}_h$ is its numerical approximation on a mesh T_h. Table 1 also contains the experimental order of convergence given by

$$\alpha_l = \frac{\log\left(e_{h_l}/e_{h_{l-1}}\right)}{\log\left(h_l/h_{l-1}\right)}, \quad l = 2, \ldots, 4, \tag{60}$$

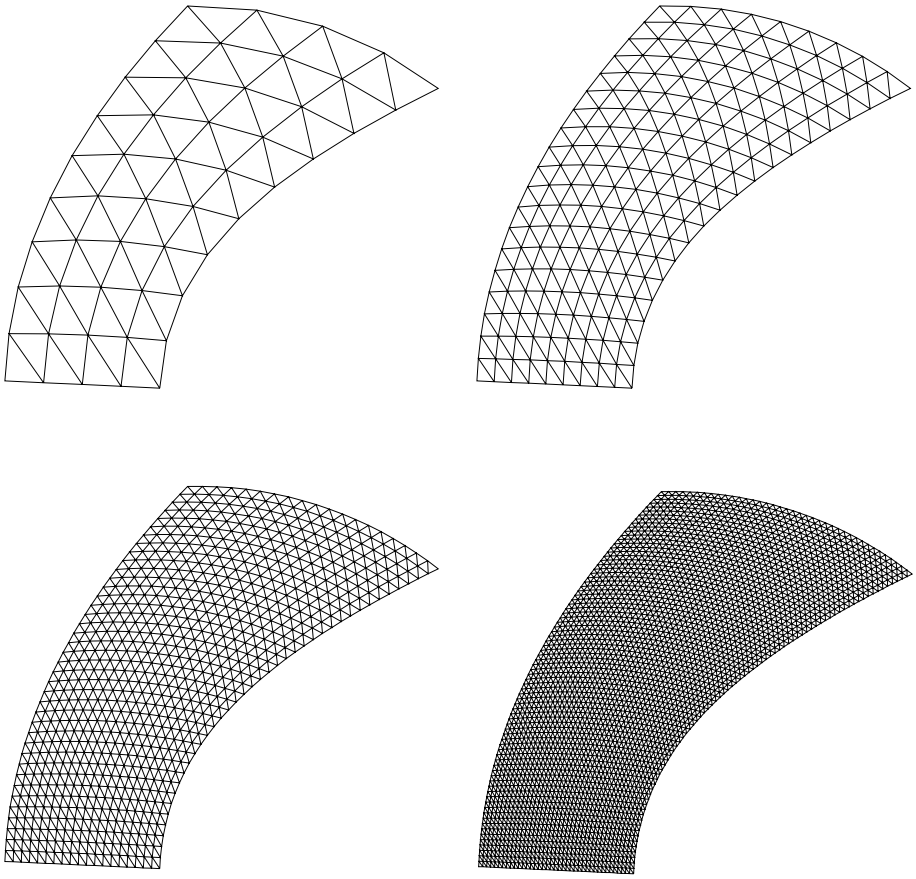

Fig. 4. Grids used for the Ringleb flow problem with 5×10, 10×20, 20×40 and 40×80 vertices.

where h_l is the mesh size of T_{h_l} and e_{h_l} is the corresponding computational error achieved on this mesh. All methods have the experimental order of convergence equal to 2, which corresponds to optimal approximation properties of linear finite elements.

Table 2 compares the number of time steps and CPU time measured in seconds for all three methods. The number of time steps necessary for the time stabilization satisfying the criterion (58) is about 10 times and 100 times smaller for semi-implicit method with (34) and (35) than for the explicit one, respectively. This corresponds to the choice of the CFL-number in the stability condition (50): $\mathrm{CFL} = 0.6$ for the forward Euler scheme, whereas $\mathrm{CFL} = 6$ and $\mathrm{CFL} = 100$ for the semi-implicit methods using (34) and (35), respectively, (see Section 3.4). The GMRES solution of system (47) needs some additional computational costs, but Table 2 shows that the total CPU-time is significantly smaller for semi-implicit schemes. (The computation was performed on PC Intel P4, 2.4 GHz.)

5.2. Vortex propagation

We consider the propagation of a vortex in compressible inviscid flow, analyzed numerically in [35]. The computational domain is taken as $[0, 10] \times [0, 10]$, extended periodically in both directions. The mean flow is $\bar{\rho} = 1$, $\bar{p} = 1$ and $\bar{\mathbf{v}} = (1, 1)$ (diagonal flow). To this mean flow we add an isentropic vortex, i.e. perturbation in $\boldsymbol{v}$ and the temperature $\theta = p/\rho$, but no perturbation in the entropy $\eta = p/\rho^{\gamma}$:

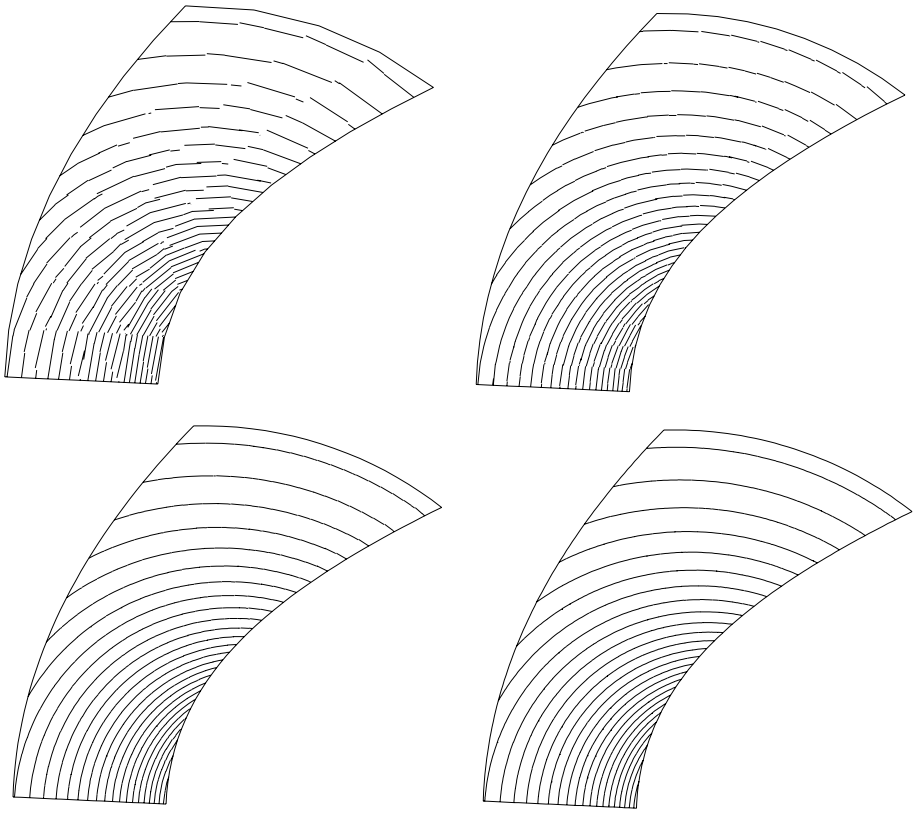

Fig. 5. Ringleb flow problem: computed isolines of Mach number obtained by the semi-implicit scheme on grids with 5×10, 10×20, 20×40 and 40×80 vertices.

Table 1
Computational errors in L^2-norm and corresponding experimental order of accuracy for explicit and semi-implicit methods

Mesh	Method					
	Explicit		Semi-implicit + (34)		Semi-implicit + (35)	
	e_h	α_l	e_h	α_l	e_h	α_l
5×10	7.85E−03	–	7.40E−03	–	7.81E−03	–
10×20	1.51E−03	2.05	1.45E−03	2.03	1.51E−03	2.05
20×40	3.41E−04	2.01	3.30E−04	2.00	3.39E−04	2.01
40×80	7.86E−05	2.04	7.71E−05	2.03	7.85E−05	2.04

$$\delta \boldsymbol{v} = \frac{\epsilon}{2\pi} \mathrm{e}^{(1-r^2)/2}(-\bar{x}_2, \bar{x}_1), \quad \delta\theta = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}\mathrm{e}^{1-r^2}, \quad \delta\eta = 0, \tag{61}$$

where $(-\bar{x}_2, \bar{x}_1) = (x_1 - 5, x_2 - 5)$, $r^2 = \bar{x}_1^2 + \bar{x}_2^2$, and the vortex strength $\epsilon = 5$. The perturbations $\delta\rho$ and δp are obtained from the above relations.

It is clear that the exact solution of the Euler equations with the above initial conditions

$$\rho(x,0) = \bar{\rho} + \delta\rho, \quad \boldsymbol{v}(x,0) = \bar{\boldsymbol{v}} + \delta\boldsymbol{v}, \quad p(x,0) = \bar{p} + \delta p, \tag{62}$$

Table 2
Number of time steps (# steps) and CPU time for explicit and semi-implicit methods

Mesh	Method					
	Explicit		Semi-implicit + (34)		Semi-implicit + (35)	
	# steps	CPU (s)	# steps	CPU (s)	# steps	CPU (s)
5×10	7855	17.3	722	3.2	74	0.6
10×20	22909	136.0	1820	36.9	166	6.7
20×40	51963	1046.3	4826	376.9	561	126.7
40×80	116364	8565.1	10872	3666.4	1262	905.6

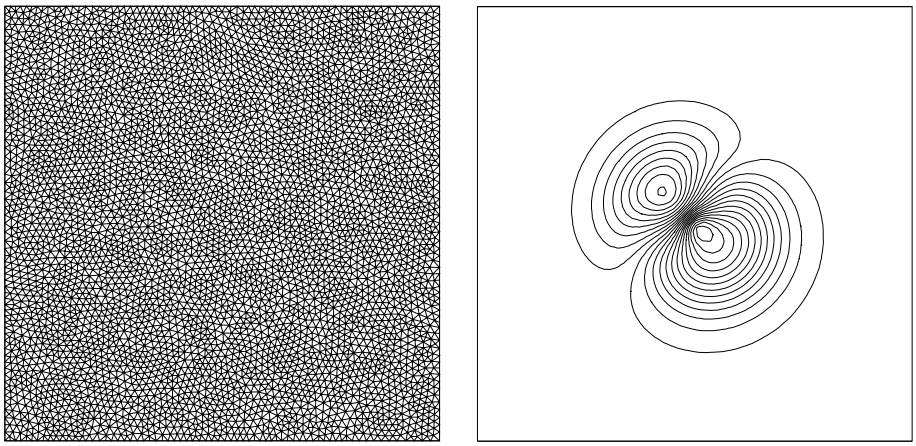

Fig. 6. Vortex propagation: triangular mesh having 9228 elements (left) and Mach number isolines at $t = 0$ (right).

and periodic boundary conditions is just the passive convection of the vortex with the mean velocity. An unstructured grid having 9228 triangles was used, see Fig. 6. The simulation was performed with the aid of the two-step scheme (40) until $t = 100$ (10 periods in time). If we compare the initial condition in Fig. 6 with the results from Fig. 7, we see that the form of the vortex is nearly unchanged after a very long time in-

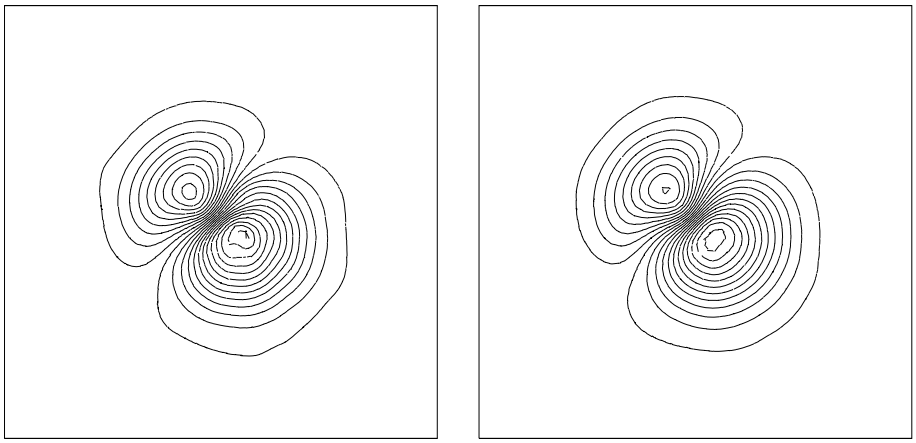

Fig. 7. Vortex propagation: Mach number isolines at $t = 50$ (left) and at $t = 100$ (right).

Table 3
Vortex propagation: number of GMRES iterations for some time levels

Time step	t	# of GMRES iterations
1	0.0021	17
25	0.0551	12
50	0.1088	11
500	1.1236	11
1000	2.4299	12
3264	10.000	14
5833	20.000	14

terval, which indicates a very good quality of the DGFE solution. Table 3 show the number of GMRES interations in some time steps.

In order to demonstrate the efficiency of the method, we also solved the vortex propagation with the aid of an explicit time discretization (second order Runge–Kutta scheme). The explicit method gives the identical solution as the semi-implicit one and therefore we do not present it here. Table 4 compares the chosen CFL number in (50), number of time steps and CPU time for the semi-implicit and explicit schemes necessary for reaching time $t = 100$. The CFL-number was chosen smaller than that one for the Ringleb flow problem, because now we solve a time dependent problem and have to guarantee not only the stability, but also the accuracy of the method. We observe that the length of the time steps is ten times larger for the semi-implicit scheme, but the computational cost for the realization of one semi-implicit time step is approximately five times higher than for the explicit time step. Therefore, the total CPU-time for semi-implicit scheme is two times smaller than for the explicit one.

5.3. GAMM channel – unsteady flow

Now let us consider a transonic flow through the GAMM channel (10% circular bump). This is a well-known benchmark for steady-state simulation with inlet Mach number $M_{\text{in}} = 0.67$. We take the steady-state solution as an initial condition and prescribe a periodical perturbation of the pressure on the outlet part of the channel given by

Table 4
Vortex propagation: comparison of the semi-implicit and explicit schemes

Method	CFL-number	# time steps	CPU time
Semi-implicit (40)	2.0	24796	23843
Explicit	0.2	247361	40895

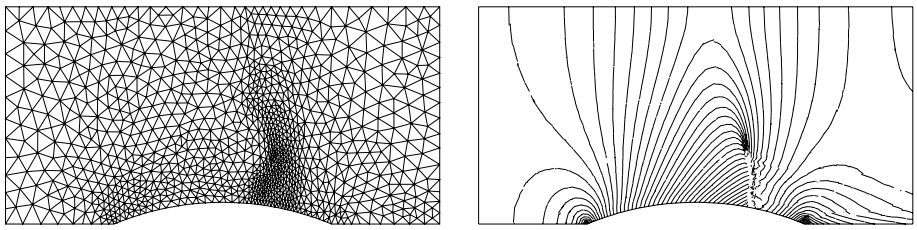

Fig. 8. GAMM channel: adaptively refined mesh having 1930 elements (left) and Mach number isolines at $t = 0$ (right).

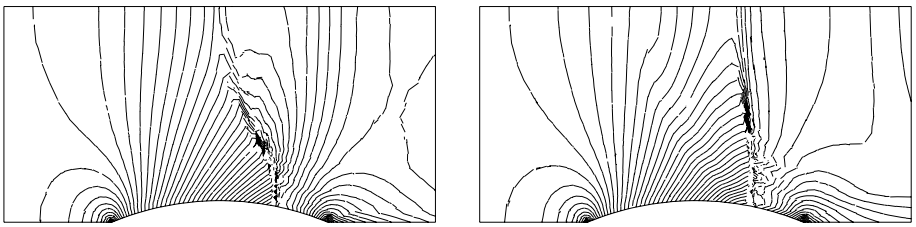

Fig. 9. GAMM channel: Mach number isolines at $t = 7$ (left) and $t = 8$ (right).

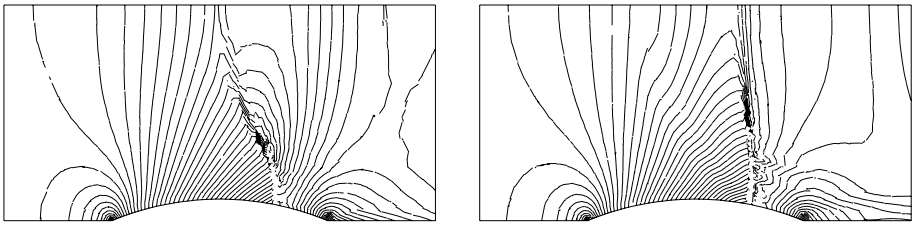

Fig. 10. GAMM channel: Mach number isolines at $t = 9$ (left) and $t = 10$ (right).

$$p = \bar{p}(1 + \varepsilon \sin(2\pi k t)), \tag{63}$$

where $\bar{p} = 1.59119$ is the value corresponding to the steady state, $\varepsilon = 0.02$ and $k = 1/2$.

The simulation was performed until $t = 10$ on an unstructured mesh (Fig. 8, left) adaptively refined by the anisotropic mesh adaptation algorithm, see [13]. We use the two-step scheme (40) and put CFL $= 5$ in (50). Fig. 8, (right) shows Mach number isolines for $t = 0$ (stationary flow). Figs. 9 and 10 show Mach number isolines at $t = 7, 8, 9$ and 10. We observe an interesting oscillatory behaviour of the solution.

6. Conclusion

We present efficient higher order numerical schemes for the solution of the compressible Euler equations using the discontinuous Galerkin finite element space discretization. In order to avoid a time step restriction known from explicit methods, semi-implicit numerical schemes were developed. The main tool is the linearization of the flux and numerical flux, leading to the solution of a linear algebraic system at each time level. In our schemes the Vijayasundaram numerical flux is used, which is suitable for a simple linearization. An important issue is the linearization of boundary conditions. We introduce here two versions – fully explicit and implicit. First and second order time discretization is applied. Our further goal was the investigation of the accuracy and efficiency of the developed schemes. Numerical experiments performed for the Ringleb flow problem confirm that the order of accuracy of discontinuous piecewise linear approximations (for semi-implicit as well as explicit schemes) is equal to 2. Thus, the order of accuracy is optimal. Moreover, the semi-implicit schemes allow us to use much larger CFL-number than the explicit one. This results among other in a strong reduction of the computer time necessary for obtaining a steady-state solution of the Euler equations with the use of the time stabilization for "$t \to \infty$" and also the reduction of the computer time in the solution of unsteady flows, as was shown on the solution of more challenging nonstationary Euler flows. Namely, the solution of the vortex propagation and the flow through the GAMM channel with prescribed unsteady pressure on the outlet is presented. These problems show the

applicability, efficiency and accuracy of the developed schemes. Future work will be concentrated on the development of a general package using higher-degree polynomial approximations and on the extension to 3D problems and the solution of compressible viscous flow, combining the DGFEM with anisotropic mesh adaptation.

Acknowledgements

We wish to express our sincere gratitude to Professor Tim Barth (NASA Ames Research Center) for providing the algorithm for generating the analytical solution to Ringleb flow and Paul Houston (University of Leicester) for providing the GMRES algorithm.

References

[1] S. Adjerid, D. Devine, J.E. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems, Comput. Methods Appl. Mech. Engrg. 191 (2002) 1097–1112.
[2] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[3] F. Bassi, S. Rebay, A high order discontinuous Galerkin method for compressible turbulent flow, in: B. Cockburn, G.E. Karniadakis, C.-W. Shu (Eds.), Discontinuous galerkin method: theory, computations and applications, Lecture Notes in Computational Science and Engineering, 11, Springer, Berlin, 2000, pp. 113–123.
[4] C.E. Baumann, J.T. Oden, A discontinuous *hp* finite element method for the Euler and Navier–Stokes equations, Int. J. Numer. Meth. Fluids 31 (1999) 79–95.
[5] R.M. Beam, R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, J. Comput. Phys. 22 (1976) 87–110.
[6] R.M. Beam, R.F. Warming, An implicit factored scheme for the compressible Navier–Stokes equations, AIAA J. 16 (1978) 393–402.
[7] M. Bejček, V. Dolejší, M. Feistauer, On discontinuous Galerkin method for numerical solution of conservation laws and convection–diffusion problems, in: Proceedings of the XIV Summer School Software and Algorithms of Numerical Mathematics, West-Bohemian University, Pilsen, 2002, pp. 7–32.
[8] G. Chiocchia, Exact solutions to transonic and superesonic flows, Technical Report AGARD-AR-211, Center for Aerospace Information, NASA, 1985.
[9] B. Cockburn, Discontinuous Galerkin methods for convection dominated problems, in: T.J. Barth, H. Deconinck (Eds.), High-order methods for computational physics, Lecture Notes in Computational Science and Engineering, 9, Springer, Berlin, 1999, pp. 69–224.
[10] B. Cockburn, S. Hou, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element for scalar conservation laws II: general framework, Math. Comp. 52 (1989) 411–435.
[11] B. Cockburn, G.E. Karniadakis, C.-W. Shu (Eds.), Discontinuous Galerkin methods, Lecture Notes in Computational Science and Engineering, 11, Springer, Berlin, 2000.
[12] E. Dick, Second-order formulation of a multigrid method for steady Euler equations through defect-correction, J. Comput. Appl. Math. 35 (1-3) (1991) 159–168.
[13] V. Dolejší, Anisotropic mesh adaptation technique for viscous flow simulation, East-West J. Numer. Math. 9 (1) (2001) 1–24.
[14] V. Dolejší, A higher order scheme based on the finite volume approach, in: R. Herbin, D. Kröner (Eds.), Finite Volumes for Complex Applications III (Problems and Perspectives), Hermes, London, 2002, pp. 333–340.
[15] V. Dolejší, M. Feistauer, On the discontinuous Galerkin method for the numerical solution of compressible high-speed flow, in: F. Brezzi, A. Buffa, S. Corsaro, A. Murli (Eds.), Numerical Mathematics and Advanced Applications, ENUMATH 2001, Springer-Verlag Italia, Milano, 2003, pp. 65–84.
[16] V. Dolejší, M. Feistauer, C. Schwab, A finite volume discontinuous Galerkin scheme for nonlinear convection-diffusion problems, Calcolo 39 (2002) 1–40.
[17] V. Dolejší, M. Feistauer, C. Schwab, On discontinuous Galerkin methods for nonlinear convection–diffusion problems and compressible flow, Mathematica Bohemica 127 (2) (2002) 163–179.
[18] V. Dolejší, M. Feistauer, C. Schwab, On some aspects of the discontinuous Galerkin finite element method for conservation laws, Math. Comput. Simul. 61 (2003) 333–346.

[19] R. Eymard, T. Gallouët, R. Herbin, Solution of equations in R^n (Part 3). Techniques of scientific computing (Part 3), in: Handbook of Numerical Analysis, vol. t, North-Holland/Elsevier, Amsterdam, 2000, Finite volume methods (chapter), p. 713-1-20.
[20] M. Feistauer, Mathematical Methods in Fluid Dynamics, Longman Scientific & Technical, Harlow, 1993.
[21] M. Feistauer, Discontinuous Galerkin method: compromise between FV and FE schemes, in: R. Herbin, D. Kröner (Eds.), Finite Volumes for Complex Applications III (Problems and Perspectives), Hermes, London, 2002, pp. 81–95.
[22] M. Feistauer, J. Felcman, Theory and applications of numerical schemes for nonlinear convection-diffusion problems and compressible Navier–Stokes equations, in: J.R. Whiteman (Ed.), The Mathematics of Finite Elements and Applications, John Wiley & Sons, Chichester, 1996, pp. 175–194.
[23] M. Feistauer, J. Felcman, V. Dolejší, Numerical simulation of compresssible viscous flow through cascades of profiles, ZAMM 76 (S4) (1996) 297–300.
[24] M. Feistauer, J. Felcman, I. Straškraba, Mathematical and Computational Methods for Compressible Flow, Clarendon Press, Oxford, 2003.
[25] J. Felcman, P. Šolín, On the construction of the Osher–Solomon scheme for 3D Euler equations, East-West J. Numer. Math. 6 (1) (1998) 43–64.
[26] L. Fezoui, B. Stoufflet, A class of implicit upwind schemes for Euler simulations with unstructured meshes, J. Comput. Phys. 84 (1) (1989) 174–206.
[27] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws, SIAM J. Sci. Comp. 24 (2002) 979–1004.
[28] P.W. Hemker, S.P. Spekreijse, Multiple grid and osher's scheme for the efficient solution of the steady Euler equations, Appl. Numer. Math. 2 (1986) 475–493.
[29] C. Hirsch, Numerical computation of internal and external flows, in: Fundamentals of numerical discretization, Wiley Series in Numerical Methods in Engineering, vol. 1, Wiley-Interscience Publication, Chichester, 1988.
[30] B. Koren, P.W. Hemker, Damped, direction-dependent multigrid for hypersonic flow computations, Appl. Numer. Math. 7 (4) (1991) 309–328.
[31] D. Kröner, Numerical Schemes for Conservation Laws, Wiley Teubner, Stuttgart, 1997.
[32] A. Meister, Comparison of different Krylov subspace methods embedded in an implicit finite volume scheme for the computation of viscous and inviscid flow fields on unstructured grids, J. Comput. Phys. 140 (1998) 311–345.
[33] S. Osher, F. Solomon, Upwind difference schemes for hyperbolic systems of conservation laws, Math. Comp. 38 (1982) 339–374.
[34] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[35] C.W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: A. Quarteroni et al. (Eds.), Advanced numerical approximation of nonlinear hyperbolic equations, Lect. Notes Math., 1697, Springer, Berlin, 1998, pp. 325–432.
[36] S.P. Spekreijse, Multigrid Solution of the Steady Euler Equations, Centre for Mathematics and Computer Science, Amsterdam, 1988.
[37] B. Stoufflet, Implicit finite element methods for the Euler equations. In Numerical methods for the Euler equations of fluid dynamics, Proc. INRIA Workshop, Rocquencourt/France 1983, 1985, pp. 409–434.
[38] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, Berlin, 1997.
[39] J.J.W. van der Vegt, H. van der Ven, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flow, J. Comput. Phys. 182 (2002) 546–585.
[40] H. vander Ven, J.J.W. vander Vegt, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. II. Efficient flux quadrature, Comput. Methods Appl. Mech. Engrg. 191 (2002) 4747–4780.
[41] G. Vijayasundaram, Transonic flow simulation using upstream centered scheme of Godunov type in finite elements, J. Comput. Phys. 63 (1986) 416–433.
[42] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, Berlin, 2001.