

ON SOURCE TERMS AND BOUNDARY CONDITIONS USING ARBITRARY HIGH ORDER DISCONTINUOUS GALERKIN SCHEMES

MICHAEL DUMBSER, CLAUS-DIETER MUNZ

Institut für Aerodynamik und Gasdynamik, Pfaffenwaldring 21, D-70550 Stuttgart, Germany
 e-mail: {michael.dumbser, munz}@iag.uni-stuttgart.de

This article is devoted to the discretization of source terms and boundary conditions using discontinuous Galerkin schemes with an arbitrary high order of accuracy in space and time for the solution of hyperbolic conservation laws on unstructured triangular meshes. The building block of the method is a particular numerical flux function at the element interfaces based on the solution of Generalized Riemann Problems (GRPs) with piecewise polynomial initial data. The solution of the generalized Riemann problem, originally introduced by Toro and Titarev in a finite volume context, provides simultaneously a numerical flux function as well as a time integration method. The resulting scheme is extremely local since it integrates the PDE from one time step to the successive one in a single step using only information from the direct side neighbors. Since source terms are directly incorporated into the numerical flux via the solution of the GRP, our very high order accurate method is also able to maintain very well smooth steady-state solutions of PDEs with source terms, similar to the so-called well-balanced schemes which are usually specially designed for this purpose. Boundary conditions are imposed solving inverse generalized Riemann problems. Furthermore, we show numerical evidence proving that by using very high order schemes together with high order polynomial representations of curved boundaries, high quality solutions can be obtained on very coarse meshes.

Keywords: discontinuous Galerkin schemes, ADER approach, source terms, boundary conditions, unstructured meshes

1. Discretization of Source Terms with Arbitrary High Order DG Schemes

We consider the non-homogeneous system of conservation laws

$$u_t + \operatorname{div} F(u) = S(u, \vec{x}, t), \quad (1)$$

with the flux $F(u) = [f(u), g(u)]$ in a two-dimensional domain $\Omega \in \mathbb{R}^2$. S is a source term that depends on the state vector u and that may also explicitly depend on space $\vec{x} = (x, y)$ and time t . For (1), proper initial conditions at $t = 0$ have to be provided. On the boundary $\partial\Omega$ of the domain, appropriate boundary conditions must be imposed on u or on the fluxes.

The computational domain Ω is divided into conforming triangles $T^{(m)}$ addressed by a unique index (m) . The numerical solution u_h is sought in the space V of piecewise polynomials up to degree N . It is hence approximated inside each triangle $T^{(m)}$ by a linear combination of some time independent polynomial basis functions $\Phi_l(\vec{\xi})$ of degree N defined on $T^{(m)}$ and with time

dependent degrees of freedom $\hat{u}^{(m)}(t)$:

$$u_h^{(m)}(\vec{\xi}, t) = \sum_{l=1}^{N_d} \hat{u}_l^{(m)}(t) \Phi_l(\vec{\xi}). \quad (2)$$

We note that the coordinates $\vec{\xi}$ refer to a reference element T_E , see Fig. 2. The number of degrees of freedom per element N_d in two dimensions is $N_d = \frac{1}{2}(N+1)(N+2)$.

For the discretization of the inhomogeneous equation (1), we multiply it by test functions Φ_k from the same space V of piecewise polynomials of order N . After integration by parts, the semi-discrete DG scheme for the numerical solution u_h is

$$\int_{T^{(m)}} \Phi_k \frac{\partial}{\partial t} u_h \, dV + \int_{\partial T^{(m)}} \Phi_k F_h \vec{n} \, dS - \int_{T^{(m)}} F(u_h) \operatorname{grad}(\Phi_k) \, dV = \int_{T^{(m)}} \Phi_k S(u_h, \vec{x}, t) \, dV. \quad (3)$$

In the nonlinear case, the first integral in (3) can still be computed analytically, but all the other integral

terms are approximated by Gaussian quadrature rules of suitable order, see (Cockburn and Shu, 1989; Cockburn and Shu, 1998), where N_S^{GP} is the number of Gaussian quadrature points needed for the boundary integral and N_V^{GP} denotes the number of Gaussian points needed for the volume integral. The quadrature points are χ_n on the boundary and ξ_n in the volume, with the weights ω_n^S on the boundary and ω_n^V in the volume, respectively. For general multidimensional integration formulas, see (Stroud, 1971). We obtain

$$\int_{\partial T^{(m)}} \Phi_k F_h \vec{n} dS \approx \sum_{j=1}^3 |S_j| \sum_{n=1}^{N_S^{\text{GP}}} \omega_n^S \Phi_k(\vec{\xi}(\chi_n)) \times F_h(u_h^{(m)}(\vec{\xi}(\chi_n), t), u_h^{(k_j)}(\vec{\xi}(\chi_n), t)) \vec{n}_j, \quad (4)$$

$$\int_{T^{(m)}} F(u_h) \nabla_x(\Phi_k) dV \approx \sum_{n=1}^{N_V^{\text{GP}}} \omega_n^V |J(\vec{\xi}_n)| F(u_h(\vec{\xi}_n, t)) \times J^{-T}(\vec{\xi}_n) \nabla_\xi \Phi_k(\vec{\xi}_n), \quad (5)$$

$$\int_{T^{(m)}} \Phi_k S(u_h, \vec{x}, t) dV \approx \sum_{n=1}^{N_V^{\text{GP}}} \omega_n^V |J(\vec{\xi}_n)| \Phi_k(\vec{\xi}_n) S(u_h(\vec{\xi}_n, t), \vec{\xi}_n, t). \quad (6)$$

Here, $|S_j|$ and \vec{n}_j are the edge length and the normal vector of edge j , respectively. ∇_x and ∇_ξ denote the gradient operators in the physical x - y system and in the ξ - η reference system, respectively, and $|J(\vec{\xi})|$ is the determinant of the Jacobian matrix $J(\vec{\xi})$ of the transformation between both coordinate systems. In general, it depends on $\vec{\xi} = (\xi, \eta)$, but J is a constant for classical triangles. For curved triangles, see Section 2.2 and Eqn. (42).

In the discontinuous Galerkin framework the numerical solution is represented by piecewise polynomials which may be discontinuous at the element boundaries. This particular data representation leads naturally to the so-called generalized Riemann problems, denoted by GRP_N in the following, at the element interfaces. The initial condition of a generalized Riemann problem consists of two polynomials of degree N that are in general separated by a discontinuity at the interface. So, a quite natural choice for the numerical flux F_h over the element boundaries seems to be the application of the ADER (arbitrary high order schemes using derivatives) approach of (Titarev and Toro, 2002; Titarev

and Toro, 2005; Toro, 2001; Toro and Titarev, 2002), which constructs an approximate solution to this generalized Riemann problem for small $\tau > 0$, where $\tau = t - t^n$ is the time t relative to the current time level t^n . This provides simultaneously a numerical flux and an explicit time integration procedure. For further details, especially concerning the theory of the generalized Riemann problem, see (Ben-Artzi and Falcovitz, 1984; Le Floch and Raviart, 1988; Bourgeade *et al.*, 1989).

The generalized Riemann problem GRP_N is given by the polynomials $u_h^{(m)}(x, y)$ in cell (m) and $u_h^{(k_j)}(x, y)$ in its neighbor (k_j) on the common cell interface j at time $\tau = 0$ at each Gaussian quadrature point χ_n :

$$\begin{aligned} \text{PDE:} \quad & u_t + \text{div } F(u) = S(u, \vec{x}, t), \\ \text{IC:} \quad & u(\vec{x}, 0) = \begin{cases} u_h^{(m)}(\vec{x}, 0) & \text{if } \vec{x} \in T^{(m)}, \\ u_h^{(k_j)}(\vec{x}, 0) & \text{if } \vec{x} \in T^{(k_j)}. \end{cases} \quad (7) \end{aligned}$$

The solution of GRP_N is first determined by the solution $u_h^{\text{RP}_0}(\chi_n)$ of a conventional (piecewise constant data) homogeneous nonlinear Riemann problem RP_0 of the boundary extrapolated values $u_h^{(m)}(\chi_n, 0) = u_h^{(m)}(\vec{\xi}(\chi_n), 0)$ and $u_h^{(k_j)}(\chi_n, 0) = u_h^{(k_j)}(\vec{\xi}(\chi_n), 0)$ on the left and right hand sides, respectively, at the quadrature point χ_n ,

$$\begin{aligned} \text{PDE:} \quad & u_t + \text{div } F(u) = 0, \\ \text{IC:} \quad & u(\vec{x}, 0) = \begin{cases} u_h^{(m)}(\chi_n, 0) & \text{if } \vec{x} \in T^{(m)}, \\ u_h^{(k_j)}(\chi_n, 0) & \text{if } \vec{x} \in T^{(k_j)}, \end{cases} \quad (8) \end{aligned}$$

and second, it is determined by the set of solutions $u_h^{\text{RP}_{(q,r)}}(\chi_n)$ of a sequence of linearized conventional homogeneous Riemann problems $\text{RP}_{(q,r)}$ for all space derivatives of u_h . Linearization is performed about the solution $u_h^{\text{RP}_0}(\chi_n)$,

$$\begin{aligned} \text{PDE:} \quad & u_t^{(q,r)} + A_0 u_x^{(q,r)} + B_0 u_y^{(q,r)} = 0, \\ \text{IC:} \quad & u^{(q,r)}(\vec{x}, 0) = \begin{cases} \frac{\partial^p u_h^{(m)}}{\partial x^q \partial y^r}(\chi_n, 0) & \text{if } \vec{x} \in T^{(m)}, \\ \frac{\partial^p u_h^{(k_j)}}{\partial x^q \partial y^r}(\chi_n, 0) & \text{if } \vec{x} \in T^{(k_j)}, \end{cases} \quad (9) \end{aligned}$$

with the Jacobians $A_0 = \partial f / \partial u$ and $B_0 = \partial g / \partial u$ evaluated at the solution $u_h^{\text{RP}_0}(\chi_n)$ of the leading order Riemann problem. This sequence of linearized Riemann problems $\text{RP}_{(q,r)}$ can be solved in a very efficient way since the PDE and therefore also the Jacobians as well as their left and right eigenvectors remain the same for all space derivatives, and thus need to be calculated only once. We emphasize that the source term does *not* appear in the Riemann problems (8) and (9), which have to

be solved only in order to define the state and the derivatives at the interface, see (Toro and Titarev, 2002). However, the source term has to be taken into account in the Cauchy-Kovalewski procedure, which will yield the time evolution of the solution at the element interface.

The solution of the generalized Riemann problem as a function of relative time τ is finally given by a Taylor series where time derivatives have been replaced by space derivatives using the so-called Cauchy-Kovalewski procedure. With the Cauchy-Kovalewski procedure, in the general case, the k -th time derivative can be expressed as a nonlinear function C_k of all (mixed) space derivatives up to order k , using successive differentiation of the governing equation (1). We emphasize in particular that the function C_k also contains information on the source term. Hence, we write

$$\frac{\partial^k u}{\partial t^k} = C_k \left(\frac{\partial^p u}{\partial x^q \partial y^r} \right), \quad \forall 0 \leq p = q + r \leq k. \quad (10)$$

The solution of GRP_N can thus be written as

$$u_h^{\text{GRP}_N}(\chi_n, \tau) = u_h^{\text{RP}_0}(\chi_n) + \sum_{k=1}^N \frac{\tau^k}{k!} C_k \left(u_h^{\text{RP}_{(q,r)}}(\chi_n) \right). \quad (11)$$

The time dependent approximate solution of GRP_N as given by Eqn. (11) can now be plugged into the physical flux in order to obtain a numerical flux for the DG scheme in Eqn. (4).

For the temporal approximation of the volume integrals (5) and (6), the space derivatives can directly be obtained by differentiating Eqn. (2) at the Gaussian points $\vec{\xi}_n$ since the polynomials are differentiable inside each triangle $T^{(m)}$ and thus the Cauchy-Kovalewski procedure (10) directly provides the time derivatives for the temporal Taylor series,

$$u_h^{\text{GAUSS}}(\vec{\xi}_n, \tau) = \sum_{k=0}^N \frac{\tau^k}{k!} C_k \left(\frac{\partial^p u_h^{(m)}(\vec{\xi}_n, 0)}{\partial x^q \partial y^r} \right), \quad (12)$$

$$\forall 0 \leq p \leq k, \quad \forall q + r = p.$$

With these ingredients, Eqns. (4)–(6) can be integrated in time using a Gaussian quadrature rule of appropriate order with N_T^{GP} integration points τ_l and the associated weights α_l in the interval $[0, \Delta t]$:

$$\begin{aligned} & \int_0^{\Delta t} \int_{\partial T^{(m)}} \Phi_k F_h \vec{n} \, dS \, d\tau \\ & \approx \sum_{l=1}^{N_T^{\text{GP}}} \alpha_l \sum_{j=1}^3 |S_j| \sum_{n=1}^{N_S^{\text{GP}}} \omega_n^S \Phi_k(\chi_n) \\ & \quad \times F \left(u_h^{\text{GRP}_N}(\chi_n, \tau_l) \right) \vec{n}_j, \end{aligned} \quad (13)$$

$$\begin{aligned} & \int_0^{\Delta t} \int_{T^{(m)}} F(u_h) \nabla_x \Phi_k \, dV \, d\tau \\ & \approx \sum_{l=1}^{N_T^{\text{GP}}} \alpha_l \sum_{n=1}^{N_V^{\text{GP}}} \omega_n^V \left| J(\vec{\xi}_n) \right| F \left(u_h^{\text{GAUSS}}(\vec{\xi}_n, \tau_l) \right) \\ & \quad \times J^{-T}(\vec{\xi}_n) \nabla_{\xi} \Phi_k(\vec{\xi}_n), \end{aligned} \quad (14)$$

$$\begin{aligned} & \int_0^{\Delta t} \int_{T^{(m)}} \Phi_k S(u_h, x, \tau) \, dV \, d\tau \\ & \approx \sum_{l=1}^{N_T^{\text{GP}}} \alpha_l \sum_{n=1}^{N_V^{\text{GP}}} \omega_n^V \left| J(\vec{\xi}_n) \right| \Phi_k(\vec{\xi}_n) \\ & \quad \times S \left(u_h^{\text{GAUSS}}(\vec{\xi}_n, \tau_l), \vec{\xi}_n, \tau_l \right). \end{aligned} \quad (15)$$

After introducing Eqns. (13)–(15), the semi-discrete scheme (3) can be integrated in time and one obtains the fully discrete ADER-DG scheme including the source terms.

We emphasize that in Eqn. (13) the flux function F is the *physical* flux which is evaluated by using the solution of GRP_N at time τ_l . It is therefore a high order generalization of Godunov's scheme if the exact Riemann solver is used for the solution of RP_0 . Any other approximate Riemann solver might be used as well, if it is able to deliver the approximate solution $u_h^{\text{RP}_0}(\chi_n)$ of the conventional nonlinear Riemann problem at the interface. For an exhaustive overview of Riemann solvers, see (Toro, 1999).

In the following, we present a special algorithm for the Cauchy-Kovalewski procedure for the system of the two-dimensional Euler equations with gravitational force. For this particular inhomogeneous system, the Cauchy-Kovalewski procedure turns out to be a particularly simple extension of the algorithm presented in (Dumbser and Munz, 2006). The equations are

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}_t + \begin{pmatrix} (\rho u) \\ (\rho u^2) + p \\ (\rho uv) \\ u(\rho E + p) \end{pmatrix}_x + \begin{pmatrix} (\rho v) \\ (\rho uv) \\ (\rho v^2) + p \\ v(\rho E + p) \end{pmatrix}_y = \begin{pmatrix} 0 \\ 0 \\ -\rho g \\ -\rho g v \end{pmatrix} \quad (16)$$

with the state equation

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} (\rho u^2 + \rho v^2) \right). \quad (17)$$

Here $\gamma = 1.4$ is the ratio of specific heats and in (16) g denotes the gravitation constant.

The resulting algorithm for the Cauchy-Kovalewski procedure with gravitational source is given in Fig. 1 and exhibits only very few changes compared with the algorithm presented in (Dumbser, 2005; Dumbser and Munz, 2006) for the homogeneous Euler equations. As in (Dumbser, 2005; Dumbser and Munz, 2006), we use

DO $c = 0, N - 1$

DO $b = 0, N - c; \quad \text{DO } a = 0, N - c - b$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} u = \mathcal{L}_{**}^{(a,b,c)}((\rho u), \rho, u)$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} v = \mathcal{L}_{**}^{(a,b,c)}((\rho v), \rho, v)$$

ENDDO; ENDDO

DO $b = 0, N - c; \quad \text{DO } a = 0, N - c - b$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} \rho u^2 = \mathcal{L}^{(a,b,c)}((\rho u), u)$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} \rho uv = \mathcal{L}^{(a,b,c)}((\rho u), v)$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} \rho v^2 = \mathcal{L}^{(a,b,c)}((\rho v), v)$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} p = (\gamma - 1) \left[\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} (\rho E - \frac{1}{2} (\rho u^2 + \rho v^2)) \right]$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} u(\rho E + p) = \mathcal{L}^{(a,b,c)}(u, (\rho E + p))$$

$$\frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} v(\rho E + p) = \mathcal{L}^{(a,b,c)}(v, (\rho E + p))$$

ENDDO; ENDDO

DO $b = 0, N - 1 - c; \quad \text{DO } a = 0, N - 1 - c - b$

$$\frac{\partial^{a+b+c+1}}{\partial x^a \partial y^b \partial t^{c+1}} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} = - \frac{\partial^{a+1+b+c}}{\partial x^{a+1} \partial y^b \partial t^c} \begin{pmatrix} (\rho u) \\ (\rho u^2) + p \\ (\rho uv) \\ u(\rho E + p) \end{pmatrix}$$

$$- \frac{\partial^{a+b+1+c}}{\partial x^a \partial y^{b+1} \partial t^c} \begin{pmatrix} (\rho v) \\ (\rho uv) \\ (\rho v^2) + p \\ v(\rho E + p) \end{pmatrix} - g \frac{\partial^{a+b+c}}{\partial x^a \partial y^b \partial t^c} \begin{pmatrix} 0 \\ 0 \\ \rho \\ \rho v \end{pmatrix}$$

ENDDO; ENDDO

ENDDO

Fig. 1. Efficient algorithm for the Cauchy-Kovalewski procedure for the two-dimensional Euler equations with gravitational source terms.

the generalized Leibniz rule in order to perform automatic differentiation of a product of two functions f and g :

$$\begin{aligned} \mathcal{L}^{(a,b,c)}(f, g) &:= \frac{\partial^{a+b+c}(f(x, y, t)g(x, y, t))}{\partial x^a \partial y^b \partial t^c} \\ &= \sum_{i=0}^a \sum_{j=0}^b \sum_{k=0}^c \Pi_{bc} \frac{\partial^{(a-i)+(b-j)+(c-k)} f}{\partial x^{(a-i)} \partial y^{(b-j)} \partial t^{(c-k)}} \frac{\partial^{i+j+k} g}{\partial x^i \partial y^j \partial t^k}. \end{aligned} \quad (18)$$

We also need the modified Leibniz rule to compute the space-time derivatives of the primitive variables from those of the conservative ones,

$$\begin{aligned} \mathcal{L}_{**}^{(a,b,c)}(fg, f, g) &:= \frac{\partial^{a+b+c} g(x, y, t)}{\partial x^a \partial y^b \partial t^c} \\ &= \frac{1}{f} \left[\frac{\partial^{a+b+c}(f(x, y, t)g(x, y, t))}{\partial x^a \partial y^b \partial t^c} - \mathcal{L}_*^{(a,b,c)}(f, g) \right], \end{aligned} \quad (19)$$

with $f \neq 0$. The operator $\mathcal{L}_*^{(a,b,c)}(f, g)$ only contains lower order derivatives of g and is defined as the original Leibniz rule (18) except for the last term in the sum:

$$\begin{aligned} \mathcal{L}_*^{(a,b,c)}(f, g) &= \sum_{i=0}^a \sum_{j=0}^b \sum_{k=0}^c \underbrace{\Pi_{bc}}_{i+j+k \neq a+b+c} \\ &\times \frac{\partial^{(a-i)+(b-j)+(c-k)} f}{\partial x^{(a-i)} \partial y^{(b-j)} \partial t^{(c-k)}} \frac{\partial^{i+j+k} g}{\partial x^i \partial y^j \partial t^k}, \end{aligned} \quad (20a)$$

with

$$\mathcal{L}_*^{(0,0,0)}(f, g) := 0. \quad (20b)$$

In the equations above, Π_{bc} denotes the following product of the binomial coefficients:

$$\Pi_{bc} = \binom{a}{i} \binom{b}{j} \binom{c}{k}. \quad (21)$$

2. High Order Boundary Discretization

A major advantage of the discontinuous Galerkin finite element method is the fact that boundary conditions can be imposed just by choosing a suitable numerical flux at the domain boundary. This remains true even for high order schemes since no reconstruction is necessary. Particularly, in the DG framework there are no ambiguities in corners where different kinds of boundary conditions meet because the edges are always well defined with respect to their boundary condition. This is not the case for high order finite difference schemes, where it may become

cumbersome to correctly adjust the values of the ghost points to have certain boundary conditions satisfied, especially when treating corners where different boundary conditions are imposed on the two corner edges.

2.1. Inverse Generalized Riemann Problems. While in the Runge-Kutta DG framework boundary conditions are imposed by the numerical flux so that the desired condition is fulfilled in each Runge-Kutta substage, in the ADER-DG framework we have to think more carefully about boundary conditions. Because the ADER-DG scheme is a single step scheme, we must require that the appropriate boundary condition be satisfied *at all times* during one time step and so the generalized Riemann problem at the boundary must be posed in a suitable way. For first order finite volume and RK-DG schemes it is sufficient to solve an inverse Riemann problem so that it delivers the desired flux or the desired Godunov state at the interface. For ADER-DG schemes we will now proceed in a similar manner and will solve *Inverse Generalized Riemann Problems* so that the flux or the state at the boundary fulfils certain conditions. To simplify the notation, we present the strategy only for the scalar case in one space-dimension.

Definition 1. Let the PDE be a system of hyperbolic conservation laws equipped with an initial condition of piecewise polynomials of order N and the solution at the interface $x = 0$ be given by the Taylor series $u_h^{\text{GRP}_N}(0, \tau)$, $\tau > 0$,

$$\begin{aligned} \text{PDE:} \quad & u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \\ \text{IC:} \quad & u(x, 0) = \begin{cases} u_h^L(x, 0) & \text{if } x < 0, \\ u_h^R(x, 0) & \text{if } x > 0, \end{cases} \\ & u_h^{\text{GRP}_N}(0, \tau) = u_h^{\text{RP}_0}(0, 0) + \sum_{k=1}^N \frac{\tau^k}{k!} \frac{\partial^k}{\partial t^k} u_h^{\text{GRP}_N}(0, 0). \end{aligned} \tag{22}$$

If the polynomial $u_h^L(x, 0)$ and the solution $u_h^{\text{GRP}_N}(0, \tau)$ at the interface are known and the polynomial $u_h^R(x, 0)$ is unknown, then we say that (22) defines an Inverse Generalized Riemann problem (GRP_N^{-1}).

In Eqn. (22), $u_h^{\text{RP}_0}(0)$ is the known solution of the conventional nonlinear Riemann problem of the leading order and $\frac{\partial^k}{\partial t^k} u_h^{\text{GRP}_N}$ are the known time derivatives of the Taylor series of the solution of the GRP.

Because we do not have yet any theoretical knowledge about the structure, nor even about the existence of solutions to such a kind of problems, and because the analysis becomes extremely cumbersome since already the straightforward solution of a GRP involves a complicated function C_k and the solution of a nonlinear conventional Riemann problem, as well as the solution of a sequence of linearized Riemann problems, we will proceed

in the following in a numerical way to construct a solution $u_h^R(x, 0)$ to GRP_N^{-1} even if we do not know anything about the existence and uniqueness.

We recall from Section 1 that the solution of the GRP is firstly determined by the solution $u_h^{\text{RP}_0}(0, 0)$ of a conventional (piecewise constant data) nonlinear Riemann problem RP_0 of the boundary extrapolated values $u_h^L(0^-, 0)$ and $u_h^R(0^+, 0)$ on the left and right hand sides, respectively,

$$\begin{aligned} \text{PDE:} \quad & u_t + f(u)_x = 0, \quad x \in \mathbb{R}, \\ \text{IC:} \quad & u(x, 0) = \begin{cases} u_h^L(0^-, 0) & \text{if } x < 0, \\ u_h^R(0^+, 0) & \text{if } x > 0, \end{cases} \end{aligned} \tag{23}$$

and, secondly, it is determined by the set of solutions $u_h^{\text{RP}(p)}(0, 0)$ of a sequence of linearized conventional Riemann problems for all space derivatives. Linearization is performed about the solution $u_h^{\text{RP}_0}(0, 0)$ with the Jacobian $A_0 = \partial f / \partial u$,

$$\begin{aligned} \text{PDE:} \quad & u_t^{(p)} + A_0 u_x^{(p)} = 0, \quad x \in \mathbb{R}, \\ \text{IC:} \quad & u^{(p)}(x, 0) = \begin{cases} \frac{\partial^p}{\partial x^p} u_h^L(0^-, 0) & \text{if } x < 0, \\ \frac{\partial^p}{\partial x^p} u_h^R(0^+, 0) & \text{if } x > 0, \end{cases} \end{aligned} \tag{24}$$

where all $\frac{\partial^p}{\partial x^p} u_h^R(0^+, 0)$ with $0 \leq p \leq N$ are unknowns.

The clue for solving GRP_N^{-1} numerically consists in exploiting the structure of the solution of the GRP and the underlying Cauchy-Kovalewski procedure. We know that the solution of the GRP is given by

$$u_h^{\text{GRP}_N}(0, \tau) = u_h^{\text{RP}_0}(0, 0) + \sum_{k=1}^N \frac{\tau^k}{k!} C_k \left(u_h^{\text{RP}(p)}(0, 0) \right), \quad \forall 0 \leq p \leq k. \tag{25}$$

This implies that the k -th time derivative of the GRP solution only depends on all unknown space derivatives of order p less than or equal to k and not on the higher ones. With this very important piece of information, we can solve the problem hierarchically, starting from the zeroth derivative and proceeding successively towards the highest derivative of order N . The problem now consists in finding the roots $\frac{\partial^p}{\partial x^p} u_h^R(0^+, 0)$ of the system of $N + 1$ nonlinear equations

$$C_k \left[u_h^{\text{RP}(p)} \left(\frac{\partial^p}{\partial x^p} u_h^R(0^+, 0) \right) \right] = \frac{\partial^k}{\partial t^k} u_h^{\text{GRP}_N}, \quad \forall 0 \leq p \leq k, \quad \forall 0 \leq k \leq N, \tag{26}$$

where the right hand side is known. If we proceed with solving the equations in (26) from $k = 0$ to $k = N$ successively in order to decouple them, then for the k -th equation

in (26) all roots of order lower than k are already known from previous steps and one only has to find a solution for $\frac{\partial^k}{\partial x^k} u_h^R(0^+, 0)$. Please note that (26) implicitly includes also the solution of Eqns. (23) and (24) to obtain the arguments of the Cauchy-Kovalewski procedure. Proceeding in such a way considerably reduces the complexity of the problem since one does not need to solve (26) for all unknowns at the same time but it suffices to seek one level of roots after the other.

The necessity to solve such inverse GRPs arises, e.g., at wall boundaries for the Euler equations with a gravitational source term. At the boundary, the normal velocity component with respect to the wall must vanish at all times. However, we have no information about the temporal pressure evolution. This is provided by the solution of the inverse GRP, where we search for a polynomial *outside* the computational domain such that the GRP solution at the boundary interface yields exactly the desired condition on the normal velocity.

2.2. Curved Elements. In the previous section we discussed the high order discretization of boundary conditions in time. However, we did not yet take into account the real *shape* of the domain boundaries $\partial\Omega$.

However, in real-life engineering problems, walls are most often *curved*. One could imagine that the discretization by triangles is sufficiently accurate for most of the problems, but there are some striking examples where this strategy fails. In (Bassi and Rebay, 1997; Cockburn *et al.*, 2000), it was shown that when computing the inviscid flowfield around a circular cylinder at a moderate Mach number using unstructured grids and classical triangles at the boundary, one gets a highly unsteady solution which exhibits even separation and vortex-shedding although this is not at all supposed to happen in the inviscid case. In the framework of the Euler equations, i.e., without viscosity, one would rather expect to obtain a steady potential flow field around the cylinder. Bassi and Rebay have shown that they indeed are able to obtain the correct solution if they use *curved* triangles at the cylinder to take into account the correct physical shape of the wall.

The finite element framework is a powerful tool that also allows for this kind of discretization in a straightforward manner. The only major change is the integration domain in Eqn. (3), which now no longer is a classical triangle $T^{(m)}$ but a curved triangle $T_c^{(m)}$. Since integration is always performed in the ξ - η coordinate system of the canonical reference element T_E in which also the basis functions are defined, the only change when going from classical to curved triangles is a different mapping, which we will present in the following. In general, we will also lose the nice property of the Jacobian of the transformation to be constant within each triangle. Instead, it will now rather become a function of $\vec{\xi}$.

In the finite element literature, curved elements are classified according to the polynomial order N_B of the boundary discretization relative to the order N of the basis functions. One speaks of subparametric elements if the boundary accuracy is less than the order of the basis functions ($N_B < N$), isoparametric elements if they are equal ($N_B = N$) and superparametric elements if the boundary is discretized with polynomials of order higher than the one of the basis functions ($N_B > N$). We implement the scheme in such a way that N_B and N can be chosen independently of each other so that any of the above-mentioned cases can be realized.

Before we can construct the mapping formulas, we must first give a definition of the curved triangle $T_c^{(m)}$ in physical space. Since we allow for any regular curved shape of the three edges, we define the triangle by its edges. The j -th edge $\partial T_{c,j}^{(m)}$ of $T_c^{(m)}$ is parameterized with the edge parameter χ by

$$\partial T_{c,j}^{(m)} = \left\{ \vec{x} \in \mathbb{R}^2 \mid \vec{x} = \begin{pmatrix} X_j(\chi) \\ Y_j(\chi) \end{pmatrix}, \right. \\ \left. 0 \leq \chi \leq 1, \quad j \in \{1, 2, 3\}, \right. \quad (27)$$

where X_j and Y_j are two sets of continuously differentiable functions, which are either given analytically or interpolated from a table using cubic spline interpolation. The boundary of the curved triangle is then defined to be the union of the edges:

$$\partial T_c^{(m)} = \partial T_{c,1}^{(m)} \cup \partial T_{c,2}^{(m)} \cup \partial T_{c,3}^{(m)}. \quad (28)$$

The set of functions X_j and Y_j has to fulfil the requirements that $\partial T_c^{(m)}$ is closed,

$$\begin{pmatrix} X_j(0) \\ Y_j(0) \end{pmatrix} = \begin{pmatrix} X_{j-1}(1) \\ Y_{j-1}(1) \end{pmatrix}, \quad j \in \{1, 2, 3\}, \quad (29)$$

that all edges are parameterized counter-clockwise,

$$X_j(0) Y_j(1) - X_j(1) Y_j(0) > 0, \\ \forall j \in \{1, 2, 3\}, \quad (30)$$

and that $T_c^{(m)}$ does not degenerate:

$$|J(\vec{\xi})| > 0 \quad \forall \xi \in T_E. \quad (31)$$

As a special case, we can thus parametrize the reference element T_E in the ξ - η coordinate system as follows:

$$\begin{pmatrix} \xi_1^R(\chi) \\ \eta_1^R(\chi) \end{pmatrix} = \begin{pmatrix} \chi \\ 0 \end{pmatrix}, \\ \begin{pmatrix} \xi_2^R(\chi) \\ \eta_2^R(\chi) \end{pmatrix} = \begin{pmatrix} 1 - \chi \\ \chi \end{pmatrix}, \\ \begin{pmatrix} \xi_3^R(\chi) \\ \eta_3^R(\chi) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 - \chi \end{pmatrix}. \quad (32)$$

The mapping from the ξ - η reference system to the physical x - y coordinate system is approximated by polynomials of order $N_B \geq 1$ of the form

$$x(\xi, \eta) = \sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \gamma_{mn} \xi^m \eta^n, \quad (33)$$

$$y(\xi, \eta) = \sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \delta_{mn} \xi^m \eta^n. \quad (34)$$

In Eqn. (34), the term δ_{mn} is not the Kronecker symbol but denotes the coefficients of a polynomial. If the boundary is sufficiently regular, we also approximate the inverse mapping by polynomials of order $N_B \geq 1$, which is, of course, a simplification:

$$\xi(x, y) = \sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \alpha_{mn} x^m y^n, \quad (35)$$

$$\eta(x, y) = \sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \beta_{mn} x^m y^n. \quad (36)$$

We must now determine the coefficients α_{mn} , β_{mn} , γ_{mn} and δ_{mn} . This can be achieved by requiring that certain points in the x - y system be mapped to well-defined positions in the reference element, see Fig. 2. In detail, we require that

$$\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \gamma_{mn} (\xi_j^R(\chi_k))^m (\eta_j^R(\chi_k))^n = X_j(\chi_k), \quad (37)$$

$$\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \delta_{mn} (\xi_j^R(\chi_k))^m (\eta_j^R(\chi_k))^n = Y_j(\chi_k), \quad (38)$$

and

$$\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \alpha_{mn} (X_j(\chi_k))^m (Y_j(\chi_k))^n = \xi_j^R(\chi_k), \quad (39)$$

$$\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \beta_{mn} (X_j(\chi_k))^m (Y_j(\chi_k))^n = \eta_j^R(\chi_k), \quad (40)$$

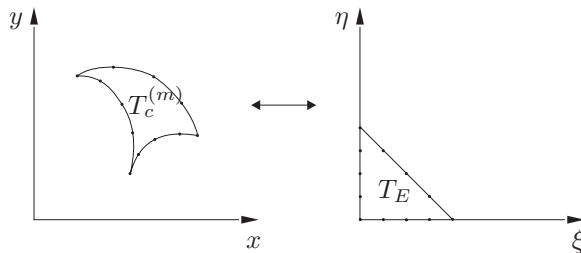


Fig. 2. Mapping points and transformation of a curvilinear triangle from the physical x - y system into the ξ - η reference system.

be fulfilled for all edge parameters

$$\chi_k = \frac{k-1}{N_B}, \quad \forall 1 \leq k \leq N_B + 1, \quad (41)$$

and for all edges $1 \leq j \leq 3$. Each of the four equation systems (37)–(40) is solved by using a singular value decomposition algorithm (Press *et al.*, 1996). This is necessary because the total number of unknowns $\frac{1}{2}(N_B + 1)(N_B + 2)$ is in general different from the number of independent equations $3N_B$ (note that in Eqns. (37)–(40) we add the three vertices of the triangle several times). For the special case of $N_B = 2$ we have 6 independent equations and 6 unknowns. Once the coefficients are determined, one can map all points in the curved triangle $T_C^{(m)}$ onto the reference triangle T_E and vice-versa using (33)–(36). The Jacobian matrix of the transformation is then given analytically as a function of ξ and η as

$$J(\xi, \eta) = \sum_{i=0}^{N_B} \sum_{j=0}^{N_B-i} \begin{pmatrix} \frac{\partial}{\partial \xi} (\gamma_{ij} \xi^i \eta^j) & \frac{\partial}{\partial \xi} (\delta_{ij} \xi^i \eta^j) \\ \frac{\partial}{\partial \eta} (\gamma_{ij} \xi^i \eta^j) & \frac{\partial}{\partial \eta} (\delta_{ij} \xi^i \eta^j) \end{pmatrix}. \quad (42)$$

This is sufficient information to compute the volume space-time integrals in (14) and (15). In the surface space-time integral (13), the normal vector \vec{n}_j is no longer constant per edge but must be replaced by a normal vector $\vec{n}_j(\chi)$ depending on the edge parameter χ as follows:

$$\vec{n}_j(\chi) = J(\xi_j^R(\chi), \eta_j^R(\chi)) \frac{\partial}{\partial \chi} \begin{pmatrix} \eta_j^R(\chi) \\ -\xi_j^R(\chi) \end{pmatrix}. \quad (43)$$

Note that according to the algorithm given in Fig. 1 the Cauchy-Kovalewski procedure is written in the physical x - y coordinate system, but all basis functions and thus the space-derivatives of u_h are *a priori* defined in the reference triangle. So one first has to transform the numerical solution u_h into the physical x - y coordinates. Let the i -th polynomial of an N -th order DG basis be

$$\phi_i(\xi, \eta) = \sum_{k=0}^N \sum_{l=0}^{N-k} \lambda_{kl}^i \xi^k \eta^l. \quad (44)$$

Then it can be rewritten as a polynomial in the x - y system as

$$\begin{aligned} \phi_i(x, y) &= \sum_{k=0}^N \sum_{l=0}^{N-k} \mu_{kl} x^k y^l \\ &= \sum_{k=0}^N \sum_{l=0}^{N-k} \lambda_{kl}^i \left(\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \alpha_{mn} x^m y^n \right)^k \\ &\quad \times \left(\sum_{m=0}^{N_B} \sum_{n=0}^{N_B-m} \beta_{mn} x^m y^n \right)^l. \end{aligned} \quad (45)$$

The unknown coefficients μ_{kl} can be found by identifying equal powers of x and y on both sides of Eqn. (45).

3. Numerical Examples

3.1. Conservation of Steady State Solutions. For the simulation of PDEs with source terms it is particularly important that the numerical scheme be able to conserve steady-state solutions of the governing PDE where flux and source terms cancel out so that no temporal evolution is excited. In the framework of finite volume schemes, the so-called well balanced methods have been constructed in the literature (Botta *et al.*, 2004; Greenberg and Le Roux, 1996; LeVeque, 1998) to achieve this goal since it is well known that finite volume schemes without this special property are not able to conserve these steady-state solutions when slight perturbations are introduced. Usually, the numerical flux does not know about the presence of the source terms and is discretized as in the homogeneous case. Since in the ADER-DG scheme the source term is explicitly integrated into the numerical flux via the Cauchy-Kovalevski procedure, we expect the scheme to conserve steady-state solutions at least to some extent. The two numerical examples presented in the following are simulations of a steady atmosphere at rest over a steep topography in a computational domain of extent $[-8000, 8000] \times [0, 8000]$. The governing equations are the two-dimensional Euler equations with gravitation (16). The bottom topography $y_0(x)$ is given by a Gaussian-type distribution as

$$y_0(x) = h e^{-x^2/b^2}. \tag{46}$$

We choose $h = 2000$ and $b = 1000$. For both examples we use a coarse unstructured grid with 456 triangles as depicted in Fig. 3. We use wall boundary conditions on *all* boundaries to prevent arising perturbations from exiting the computational domain.

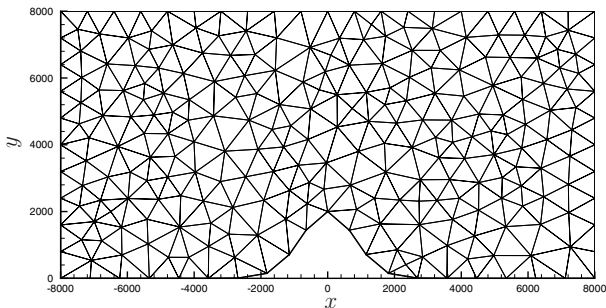


Fig. 3. Computational domain and grid for the simulation of a hydrostatic steady atmosphere at rest over a steep topography.

3.1.1. Isothermal Atmosphere at Rest. Our first example is an isothermal atmosphere at rest given by the

steady-state solution $u = v = 0$ and

$$\rho(y) = \rho_0 e^{-\frac{\rho_0}{p_0} g y}, \quad p(y) = p_0 e^{-\frac{\rho_0}{p_0} g y}, \tag{47}$$

with the gravitation constant $g = 10$. We set the density and the pressure at the sea level ($y = 0$) to $\rho_0 = 1.21$ and $p_0 = 10^5$, respectively. Several simulations are now performed using second to fifth order ADER-DG schemes, i.e., using P1 to P4 elements. The maximum error of vertical velocity is depicted in Fig. 4 as a function of time. We clearly see that the conservation of the steady-state solution is considerably improved if the order of the numerical scheme is increased. While the second order ADER-DG scheme still produces velocity errors of the order of magnitude of 10^{-1} , the fifth order scheme only shows perturbations of the order 10^{-5} . For comparison, with an unbalanced second order finite volume scheme we obtained vertical velocity errors of the order of 10^{+1} .

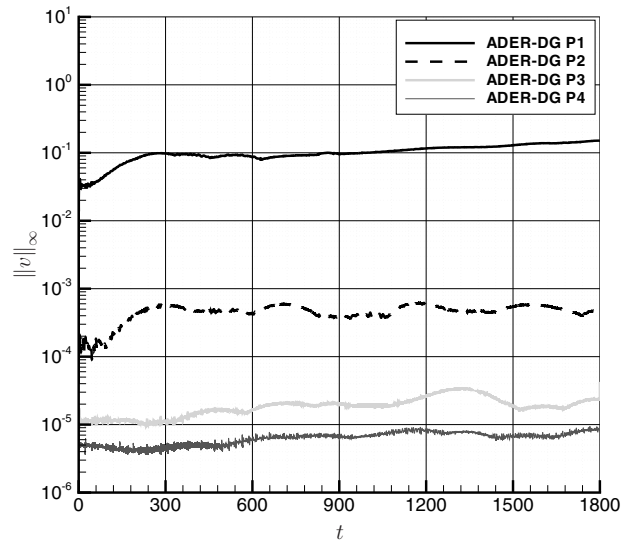


Fig. 4. Maximum error of vertical velocity as a function of time for ADER-DG $\mathcal{O}2$ - $\mathcal{O}5$ schemes. Isothermal atmosphere.

3.1.2. Strongly Stratified Atmosphere at Rest. The second example is a strongly stratified atmosphere at rest, taken from (Botta *et al.*, 2004). This test case corresponds to a thermal inversion situation that can occur in winter in alpine valleys, where the temperature *increases* with the altitude. Usually, the temperature in the Earth's atmosphere is *decreasing* with the altitude. According to (Botta *et al.*, 2004), the steady-state solution is $u = v = 0$ and

$$\rho(y) = \rho_0 \left(\frac{p(y)}{p_0} \frac{1}{1 + \sigma y} \right)^{\frac{1}{\gamma}},$$

$$p(y) = p_0^{\frac{-1}{\gamma-1}} \left(p_0 - \frac{g}{\sigma} \rho_0 (1 + \sigma y)^{\frac{\gamma-1}{\gamma}} - 1 \right)^{\frac{\gamma}{\gamma-1}}, \tag{48}$$

with $\gamma = 1.4$ and $\sigma = 1.2 \times 10^{-4}$. We set again $\rho_0 = 1.21$, $p_0 = 10^5$ and $g = 10$. The maximum error of vertical velocity is depicted in Fig. 5 as a function of time

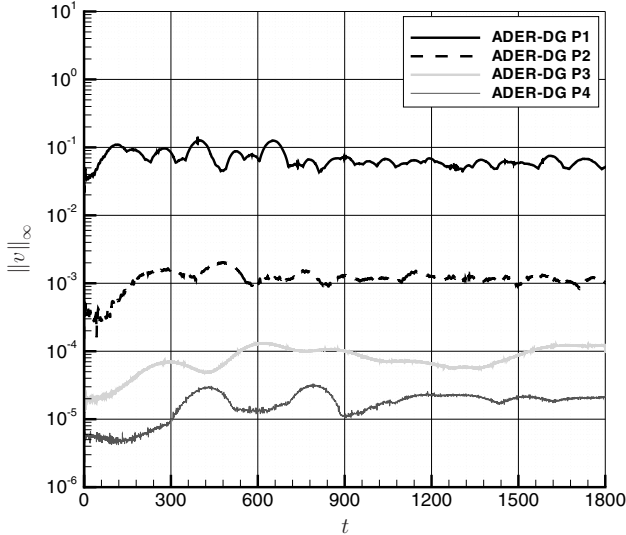


Fig. 5. Maximum error of vertical velocity as a function of time for ADER-DG $\mathcal{O}2$ - $\mathcal{O}5$ schemes. Stratified atmosphere.

for second to fifth order ADER-DG schemes. For comparison, the second order well balanced scheme of Botta *et al.* (2004) produced for this example vertical velocity errors between 10^{-2} and 10^{-1} on a structured grid with 64×32 elements. This is comparable to the results of our second order ADER-DG scheme. However, the ADER approach allows the construction of higher order schemes, which also in this numerical example clearly perform better than second order methods concerning the conservation of steady-state solutions of (16).

3.2. Inviscid Low Mach Number Flow over a Cylinder. We now show the capability of the ADER-DG scheme to correctly compute a subsonic inviscid steady cylinder flow on a coarse unstructured mesh when a high order boundary discretization is employed, as presented, e.g., in (Bassi and Rebay, 1997; Cockburn *et al.*, 2000) for Runge-Kutta DG schemes. We demonstrate that wrong results are obtained if the wall boundaries are only approximated with first order polynomials, i.e., with classical triangles, thus confirming the results of Bassi and Rebay (1997). Furthermore, the behaviour of our arbitrary high order DG approach is analyzed in the low Mach number limit.

The problem is to compute the flow field around a circular cylinder located at the origin with radius $R = 1$ in a computational domain with extent $[-10, 10] \times [-10, 10]$, solving (16) without gravitation. Hence, the viscosity is neglected. The boundary condition at infinity is defined as

$$\rho_\infty = 1, \quad M_\infty = 0.1, \quad p_\infty = 1, \quad \gamma = 1.4. \quad (49)$$

On the cylinder wall, slip wall boundary conditions are imposed, see (Dumbser, 2005). The potential solution of this *steady* flow problem will serve us as a reference and is briefly derived in the following.

The complex velocity potential w for this problem is

$$w(z) = V_\infty \left(z + \frac{R^2}{z} \right), \quad w \in \mathbb{C}, \quad z = x + Iy \in \mathbb{C}, \quad (50)$$

where $V_\infty = M_\infty \sqrt{\gamma \frac{p_\infty}{\rho_\infty}}$ is the velocity at infinity and $I = \sqrt{-1}$ is the imaginary unit. The velocity components in polar coordinates r and θ can be computed as

$$v_r(r, \theta) = \frac{\partial \Phi}{\partial r}, \quad v_\theta(r, \theta) = -\frac{\partial \Psi}{\partial r}, \quad z = re^{I\theta}, \quad (51)$$

where the potential $\Phi \in \mathbb{R}$ and the stream function $\Psi \in \mathbb{R}$ are related to the complex potential w by

$$w = \Phi + I\Psi. \quad (52)$$

One finally obtains

$$v_r(r, \theta) = V_\infty \left(1 - \frac{R^2}{r^2} \right) \cos \theta, \quad (53)$$

$$v_\theta(r, \theta) = -V_\infty \left(1 + \frac{R^2}{r^2} \right) \sin \theta, \quad (54)$$

and with the Bernoulli equation the pressure can be computed as

$$p(r, \theta) = p_\infty + \frac{1}{2} \gamma p_\infty M_\infty^2 - \frac{1}{2} \left[v_r(r, \theta)^2 + v_\theta(r, \theta)^2 \right]. \quad (55)$$

We emphasize that the potential solution is symmetric with respect to the x - and y -axes and that it has *two* stagnation points where $u_r = u_\theta = 0$. One is located at $(1, 0)$ and the other at $(-1, 0)$. At the stagnation points, the pressure coefficient c_p is

$$c_p = \frac{p - p_\infty}{\frac{1}{2} \gamma p_\infty M_\infty^2} = 1. \quad (56)$$

Two numerical simulations of this problem are now performed using fourth order ADER-DG schemes (P3 elements) on the same grid, which is quite coarse. The only difference between them is that in the first case the boundary is discretized with classical triangles, $N_B = 1$, and in the second case we use superparametric elements with $N_B = 4$ on the cylinder wall.

In the case of classical triangles we get a highly unsteady solution exhibiting flow separation and vortex-shedding. The scheme does not converge at all to a steady state even after thousands of iterations. A possible explanation for this behavior may be that with usual triangles the boundary is approximated by a polygon which naturally exhibits corners. The potential flow, however, is

singular in such corners where it would produce infinite velocities. The high order DG scheme is able to resolve this singular potential flow up to a certain extent, producing high velocities in the corners. However, the velocity growth once becomes limited by the numerical viscosity, which then causes separation and thus vortex shedding. We underline the fact that this is a purely numerical artifact, which must not arise in the inviscid Euler equations. It has nothing to do with vortex-shedding commonly observed in real air flows around cylinders, where it is due to real viscous effects of the molecular viscosity. The appropriate equations to describe this kind of problem would rather be the Navier-Stokes equations than the Euler equations, which are under consideration here. Unlike in the case of the classical triangles, superparametric ADER-DG elements quickly converge to a steady state.

To investigate the numerical solution quantitatively, we show the pressure coefficient c_p obtained at the points $(-1, 0)$ and $(1, 0)$ as a function of time. The two points are the stagnation points, and therefore the pressure coefficient should converge to a steady state value of $c_p = 1$ at *both* points. Figures 6(a) and (b) show the signals of $c_p(t)$ obtained in the simulation with classical triangles and with superparametric elements on the cylinder wall, respectively. We see that no convergence is obtained in the first case. The pressure coefficient at the point $(1, 0)$ is highly oscillatory and far from the exact value of $c_p = 1$. Although at the point $(-1, 0)$ the pressure coefficient is quite close to the exact one, some spurious oscillations are, however, visible. This may be due to the acoustic waves that are caused by the vortex-shedding and which propagate through the computational domain. In the second case, using superparametric elements, very fast convergence to the exact value is obtained at both points.

The exact potential solution for the pressure coefficient, see Eqn. (55), is depicted in Fig. 7. An instantaneous view of the pressure coefficient obtained in our numerical experiments at $t = 100$ is shown in Figs. 8(a) and (b) for the classical triangles and the superparametric elements, respectively. In Fig. 8(a) one can see a vortex shedding from the cylinder. The simulation carried out with the superparametric elements clearly possesses the symmetry properties of the potential solution as well as the two stagnation points, see Fig. 8(b). Note that the visualization of the DG solution is based only on the cell averages.

We now decrease the inflow Mach number to $M = 10^{-2}$ and $M = 10^{-3}$ to demonstrate that high order DG schemes are still functional in the low Mach number range. It is well known that first and second order finite volume schemes have problems computing low Mach number flows, especially in the region around stagnation points, where the local Mach number is almost zero, because numerical errors dominate the physical behavior of the hydrodynamic pressure which is of the order of mag-

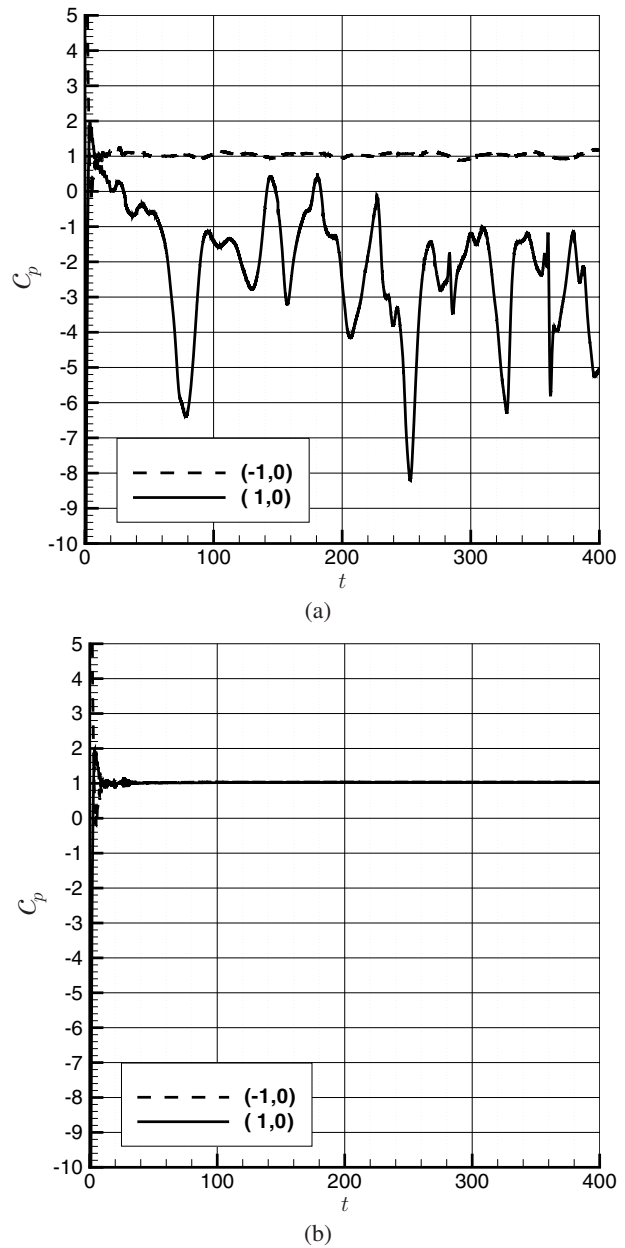


Fig. 6. Time signal of c_p in the stagnation points $(-1, 0)$ and $(1, 0)$: (a) classical triangles ($N = 3, N_B = 1$), (b) superparametric elements ($N = 3, N_B = 4$).

nitude $\mathcal{O}(M^2)$. This was shown in (Klein, 1995; Meister, 1999) using asymptotic analysis. Computations are performed on the same grid as in the previous example and also the parameters of the scheme ($N = 3$ and $N_B = 4$) remain unchanged. The distributions of the pressure coefficient at $t = 100$ are shown in Figs. 9(a) and (b) for $M = 10^{-2}$ and $M = 10^{-3}$, respectively. The fourth order scheme converges even in this low Mach number range and the numerical solution still remains symmetric. Note that the explicit time step for the ADER-DG scheme is restricted by the speed of sound, which is some

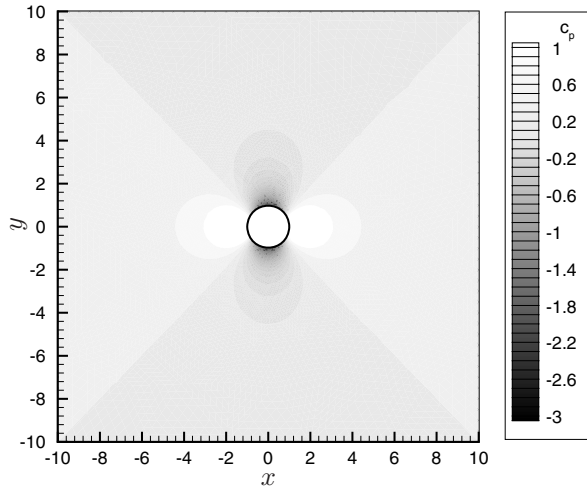
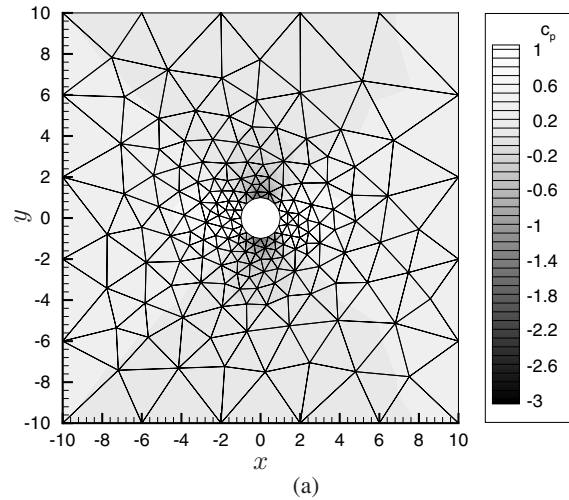
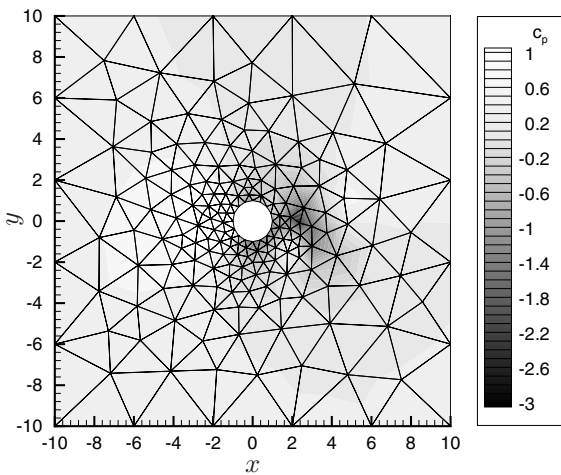


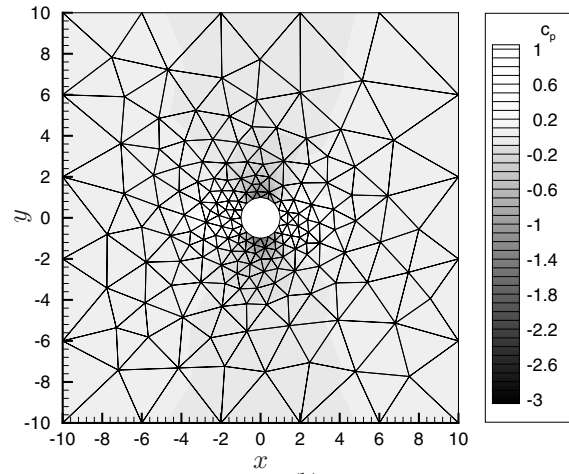
Fig. 7. Reference solution for the pressure coefficient c_p of the potential flow around the cylinder.



(a)

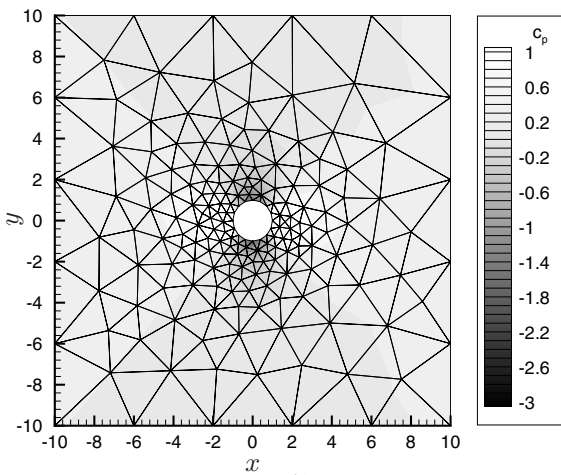


(a)



(b)

Fig. 9. Pressure coefficient c_p at $t = 100$ using superparametric ADER-DG $\mathcal{O}4$ schemes ($N = 3, N_B = 4$): (a) $M = 10^{-2}$, (b) $M = 10^{-3}$.



(b)

Fig. 8. Pressure coefficient c_p at $t = 100$ using ADER-DG $\mathcal{O}4$ schemes at $M = 10^{-1}$: (a) classical triangles, (b) superparametric elements.

orders of magnitude larger than the flow speed. Therefore it is computationally very expensive and so it is only justified to use this class of schemes for unsteady low Mach number flow problems, where one is also interested in the acoustics generated by the problem.

If we further decrease the Mach number to $M = 10^{-4}$, we do not get convergence any more on this grid and with this numerical scheme. There, one would either have to use a preconditioner (Meister, 2003) or directly switch to an appropriate low Mach number solver, as developed by Roller and Munz (2000).

3.3. Subsonic Flow around an NACA0012 Airfoil.

Previous works on arbitrary high order DG schemes (Dumbser, 2005; Dumbser and Munz, 2005; Qiu *et al.*, 2005) demonstrated that very high order discontinuous Galerkin schemes may be much more efficient in terms

of the total number of degrees of freedom and CPU time than low order schemes when the same precision is required. However, with our numerical studies on the inviscid cylinder flow we confirmed the results of Bassi and Rebay (1997) that the use of very high order methods also necessitates a very careful treatment of the boundaries, since otherwise a globally high order of accuracy cannot be attained and wrong numerical results may arise if the boundaries are not discretized appropriately. Yet one may benefit from the very coarse grids which high order schemes allow us to use if also the boundaries are high order accurate. To demonstrate the power of curved discontinuous Galerkin finite elements on a practical example, we compute the flow around an NACA0012 airfoil on an extremely coarse grid containing only 120 triangles in the whole computational domain with extent $[-5, 6] \times [-5, 5]$, see Fig. 10. The governing equations are (16) without gravitation. The upper part of the symmetric NACA0012 profile is defined by the following analytical function:

$$y(x) = \frac{3}{5} (a_0\sqrt{x} + a_1x + a_2x^2 + a_3x^3 + a_4x^4), \tag{57}$$

in the range $0 \leq x \leq 1$. We assume that the coordinates y and x are normalized by the chord length. The coefficients are $a_0 = 0.2969$, $a_1 = -0.1260$, $a_2 = -0.3516$, $a_3 = 0.2843$ and $a_4 = -0.1036$. The lower part is symmetric to the upper part with respect to the x -axis. The profile is defined by Eqn. (57) and the given inflow parameters at infinity are the Mach number $M_\infty = 0.3$, the pressure $p_\infty = 1$ and the angle of attack $\alpha = 4^\circ$.

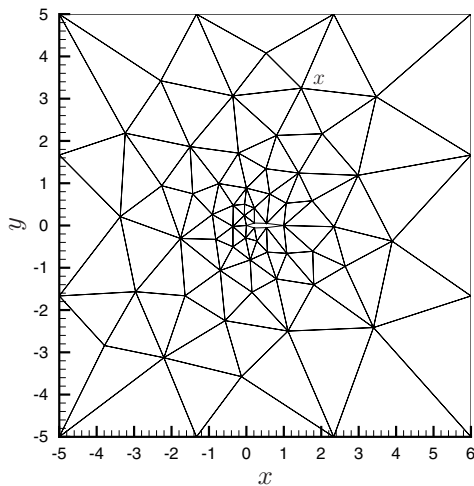


Fig. 10. Very coarse grid (120 triangles) for the computation of the subsonic flow around a NACA0012 profile ($M_\infty = 0.3$, $\alpha = 4^\circ$).

We use only *three* elements to discretize each side of the airfoil and now perform two computations on this

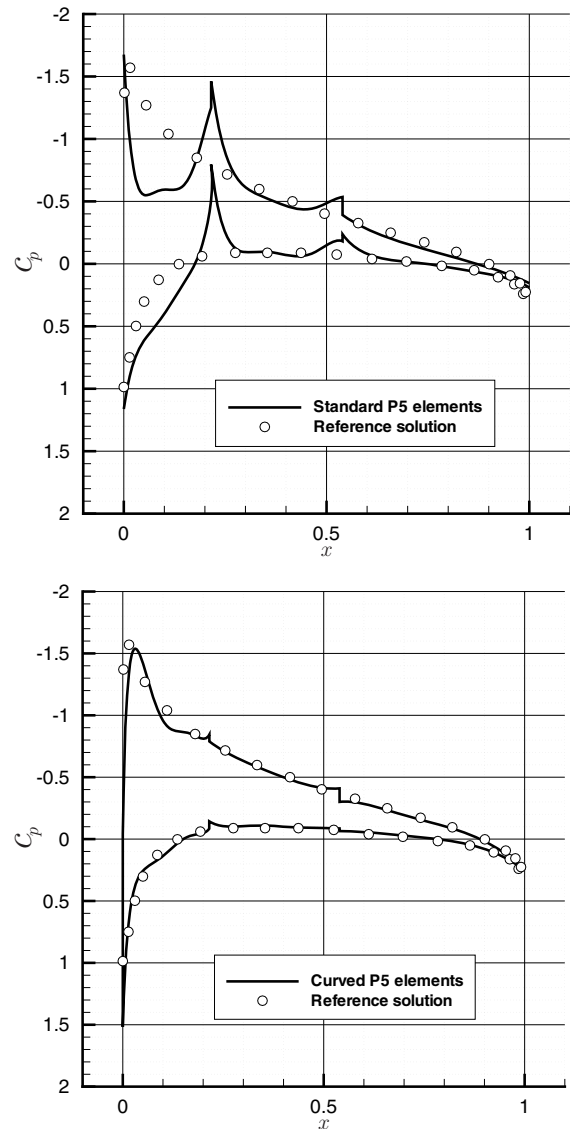


Fig. 11. Comparison of standard P5 elements and curved P5 elements on the NACA0012 airfoil flow ($M_\infty = 0.3$, $\alpha = 4^\circ$) using the grid shown in Fig 10.

grid using P5 elements on classical and curved triangles ($N_B = 5$). The reference solution was taken from the literature (Milholen, 2000). Looking at the pressure coefficient depicted in Fig. 11 one clearly sees that P5 elements with standard triangles produce a wrong solution in parts where the airfoil exhibits high curvature. This is especially the case in the region of the leading edge. In moderately curved parts of the airfoil the solution is quite acceptable. The peaks in the numerical solution at $x = 0.22$ and $x = 0.54$ indicate the presence of corner singularities that arise with standard triangles, leading to a considerable increase in the velocity in the corner and to an associated decrease in the pressure. Switching from standard to curved triangles we get a very significant improvement of the solution quality, see Fig. 11. The corner

singularities almost disappear and we globally get a very good agreement with the reference solution. This example is to encourage the reader to use coarse grids if he or she employs very high order discontinuous Galerkin finite element schemes together with a very high order boundary discretization.

4. Conclusions

In this paper we have discussed the discretization of nonlinear hyperbolic systems with source terms on unstructured meshes, underlining the importance of a careful and consistent discretization of the computational boundary. In the first section of the article, we derived semidiscrete and fully discrete versions of the scheme. Furthermore, we proposed an efficient algorithm to perform the Cauchy-Kovalewski procedure for the nonlinear Euler equations with gravitational source terms. The second section was devoted to a discussion of the correct implementation of the boundary conditions. To satisfy the boundary condition at all times, the so-called inverse generalized Riemann problems must be solved at the boundary interfaces. This becomes necessary because usually not all values of the state or of the flux are known at the boundary. For wall boundaries in the Euler equations, we have, e.g., only a condition on the velocity that has to be satisfied, whereas the pressure at the boundary is unknown *a priori*.

Our approach of numerically solving inverse generalized Riemann problems for the state vector at the boundary provides a solution at the boundary interfaces which fulfills the velocity condition on the wall for all times and simultaneously yields a time-dependent pressure at the boundary interface. In the third section, we provided several applications of our method. The example of the stratified steady atmosphere at rest was taken from the literature on well-balanced schemes (Botta *et al.*, 2004). We note that well-balanced schemes must be explicitly constructed for maintaining particular steady-state solutions of hyperbolic conservation laws with source terms, whereas our high order accurate method is not constructed explicitly for this purpose.

Our numerical experiments with the steady atmosphere at rest have shown that very high order ADER-DG schemes automatically maintain this particular steady state solution and that increasing the order dramatically improves the solution quality. Inviscid low Mach number flows around a cylinder were computed successfully with our method using the high order boundary discretization described in Section 2. Our method produces very good results even down to quite low Mach numbers. The last numerical example with the NACA0012 airfoil demonstrated that high order boundary discretization together with a very high order scheme in space and time allows the use of very coarse grids.

Acknowledgement

The work presented in this article was supported by *Deutsche Forschungsgemeinschaft* (DFG). The authors would also like to thank the two referees for their constructive remarks and suggestions, which helped to improve the clarity and the quality of this article.

References

- Bassi F. and Rebay S. (1997): *High-order accurate discontinuous finite element solution of the 2D Euler equations*. Journal of Computational Physics, Vol. 138, pp. 251–285.
- Ben-Artzi M. and Falcovitz J. (1984): *A second-order Godunov-type scheme for compressible fluid dynamics*. Journal of Computational Physics, Vol. 55, pp. 1–32.
- Botta N., Klein R., Langenberg S. and Lützenkirchen S. (2004): *Well balanced finite volume methods for nearly hydrostatic flows*. Journal of Computational Physics, Vol. 196, pp. 539–565.
- Bourgeade A., LeFloch P., and Raviart P.A. (1989): *An asymptotic expansion for the solution of the generalized Riemann problem. Part II: Application to the gas dynamics equations*. Annales de l’Institut Henri Poincaré (C) Analyse non linéaire, Vol. 6, pp. 437–480.
- Cockburn B., Karniadakis G. E. and Shu C.W. (2000): *Discontinuous Galerkin Methods*. Springer.
- Cockburn B. and Shu C.W. (1989): *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II: General framework*. Mathematics of Computation, Vol. 52, pp. 411–435.
- Cockburn B. and Shu C. W. (1998): *The Runge-Kutta discontinuous Galerkin method for conservation laws. V: Multidimensional systems*. Journal of Computational Physics, Vol. 141, pp. 199–224.
- Dumbser M. (2005): *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*. Aachen: Shaker Verlag.
- Dumbser M. and Munz C.D. (2005): *Arbitrary high order Discontinuous Galerkin schemes*, In: Numerical Methods for Hyperbolic and Kinetic Problems (S. Cordier, T. Goudon, M. Gutnic and E. Sonnendrucker, Eds.). EMS Publishing House, pp. 295–333.
- Dumbser M. and Munz C.D. (2006): *Building blocks for arbitrary high order discontinuous Galerkin schemes*. Journal of Scientific Computing, Vol. 27, pp. 215–230.
- Greenberg J.M. and Le Roux A.Y. (1996): *A well-balanced scheme for the numerical processing of source terms in hyperbolic equations*. SIAM Journal on Numerical Analysis, Vol. 33, pp. 1–16.
- Klein R. (1995): *Semi-implicit extension of a Godunov-type scheme based on low mach number asymptotics. I: One-dimensional flow*. Journal of Computational Physics, Vol. 121, pp. 213–237.

- Le Floch P. and Raviart P.A. (1988): *An asymptotic expansion for the solution of the generalized riemann problem. Part I: General theory*. Annales de l'Institut Henri Poincaré (C) Analyse non linéaire, Vol. 5, pp. 179–207.
- LeVeque R.J. (1998): *Balancing source terms and flux gradients in high resolution Godunov methods*. Journal of Computational Physics, Vol. 146, pp. 346–365.
- Meister A. (1999): *Asymptotic single and multiple scale expansions in the low Mach number limit*. SIAM Journal on Applied Mathematics, Vol. 60, No. 1, pp. 256–271.
- Meister A. (2003): *Asymptotic based preconditioning technique for low mach number flows*. Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM), Vol. 83, pp. 3–25.
- Milholen W.E. (2000): *An efficient inverse aerodynamic design method for subsonic flows*. Technical Report No. 2000–0780, American Institute of Aeronautics and Astronautics, Reno, NV.
- Press W.H., Teukolsky S.A., Vetterling W.T. and Flannery B.P. (1996): *Numerical Recipes in Fortran 77*. Cambridge: Cambridge University Press.
- Qiu J., Dumbser M. and Shu C.W. (2005): *The discontinuous Galerkin method with Lax-Wendroff type time discretizations*. Computer Methods in Applied Mechanics and Engineering, Vol. 194, pp. 4528–4543.
- Roller S. and Munz C.D. (2000): *A low mach number scheme based on multi-scale asymptotics*. Computing and Visualization in Science, Vol. 3, pp. 85–91.
- Stroud A.H. (1971): *Approximate Calculation of Multiple Integrals*. Englewood Cliffs, NJ: Prentice-Hall.
- Titarev V.A. and Toro E.F. (2002): *ADER: Arbitrary high order Godunov approach*. Journal of Scientific Computing, Vol. 17, No. 1–4, pp. 609–618.
- Titarev V.A. and Toro E.F. (2005): *ADER schemes for three-dimensional nonlinear hyperbolic systems*. Journal of Computational Physics, Vol. 204, pp. 715–736.
- Toro E.F. (1999): *Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd Ed*. Springer.
- Toro E.F., Millington R.C. and Nejad L.A.M (2001): *Towards very high order Godunov schemes*, In: Godunov Methods. Theory and Applications (E.F. Toro, Ed.). Kluwer/Plenum Academic Publishers, pp. 905–938.
- Toro E.F. and Titarev V.A. (2002): *Solution of the generalized Riemann problem for advection-reaction equations*. Proceedings of the Royal Society A, Vol. 458, pp. 271–281.