Preface

In these lecture notes, we will give a general introduction to the discontinuous Galerkin (DG) methods for solving time-dependent, convection-dominated partial differential equations (PDEs), including the hyperbolic conservation laws, convection-diffusion equations, and PDEs containing higher-order spatial derivatives such as the KdV equations and other nonlinear dispersive wave equations. We will discuss cell entropy inequalities, nonlinear stability, and error estimates. The important ingredient of the design of DG schemes, namely the adequate choice of numerical fluxes, will be explained in detail. Issues related to the implementation of the DG method will also be addressed.

Chapter 1

Introduction

Discontinuous Galerkin (DG) methods are a class of finite-element methods using completely discontinuous basis functions, which are usually chosen as piecewise polynomials. Since the basis functions can be completely discontinuous, these methods have the flexibility which is not shared by typical finite-element methods, such as the allowance of arbitrary triangulation with hanging nodes, complete freedom in changing the polynomial degrees in each element independent of that in the neighbors (p adaptivity), and extremely local data structure (elements only communicate with immediate neighbors regardless of the order of accuracy of the scheme) and the resulting embarrassingly high parallel efficiency (usually more than 99% for a fixed mesh, and more than 80% for a dynamic load balancing with adaptive meshes which change often during time evolution), see, e.g. [5]. A very good example to illustrate the capability of the discontinuous Galerkin method in h-p adaptivity, efficiency in parallel dynamic load balancing, and excellent resolution properties is the successful simulation of the Rayleigh-Taylor flow instabilities in [38].

The first discontinuous Galerkin method was introduced in 1973 by Reed and Hill [37], in the framework of neutron transport, i.e., a time-independent linear hyperbolic equation. A major development of the DG method is carried out by Cockburn et al. in a series of papers [14, 13, 12, 10, 15], in which they have established a framework to easily solve *nonlinear* time-dependent problems, such as the Euler equations of gas dynamics, using explicit, nonlinearly stable high-order Runge-Kutta time discretizations [44] and DG discretization in space with exact or approximate Riemann solvers as interface fluxes and total variation bounded (TVB) nonlinear limiters [41] to achieve non-oscillatory properties for strong shocks.

The DG method has found rapid applications in such diverse areas as aeroacoustics, electro-magnetism, gas dynamics, granular flows, magneto-hydrodynamics, meteorology, modeling of shallow water, oceanography, oil recovery simulation, semiconductor device simulation, transport of contaminant in

porous media, turbomachinery, turbulent flows, viscoelastic flows and weather forecasting, among many others. For more details, we refer to the survey paper [11], and other papers in that Springer volume, which contains the conference proceedings of the First International Symposium on Discontinuous Galerkin Methods held at Newport, Rhode Island in 1999. The lecture notes [8] is a good reference for many details, as well as the extensive review paper [17]. More recently, there are two special issues devoted to the discontinuous Galerkin method [18, 19], which contain many interesting papers in the development of the method in all aspects including algorithm design, analysis, implementation and applications.

Chapter 2

Time Discretization

In these lecture notes, we will concentrate on the method of lines DG methods, that is, we do not discretize the time variable. Therefore, we will briefly discuss the issue of time discretization at the beginning.

For hyperbolic problems or convection-dominated problems such as Navier-Stokes equations with high Reynolds numbers, we often use a class of high-order nonlinearly stable Runge-Kutta time discretizations. A distinctive feature of this class of time discretizations is that they are convex combinations of first-order forward Euler steps, hence they maintain strong stability properties in any semi-norm (total variation semi-norm, maximum norm, entropy condition, etc.) of the forward Euler step. Thus one only needs to prove nonlinear stability for the first-order forward Euler step, which is relatively easy in many situations (e.g., TVD schemes, see for example Section 3.2.2 below), and one automatically obtains the same strong stability property for the higher-order time discretizations in this class. These methods were first developed in [44] and [42], and later generalized in [20] and [21]. The most popular scheme in this class is the following third-order Runge-Kutta method for solving

$$u_t = L(u, t)$$

where $L(u, t)$ is a spatial discretization operator (it does not need to be, and often is not, linear!):

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n, t^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}, t^n + \Delta t), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}, t^n + \frac{1}{2}\Delta t). \end{aligned} \tag{2.1}$$

Schemes in this class which are higher order or are of low storage also exist. For details, see the survey paper [43] and the review paper [21].

If the PDEs contain high-order spatial derivatives with coefficients not very small, then explicit time marching methods such as the Runge-Kutta methods described above suffer from severe time-step restrictions. It is an important and active research subject to study efficient time discretization for such situations, while still maintaining the advantages of the DG methods, such as their local nature and parallel efficiency. See, e.g. [46] for a study of several time discretization techniques for such situations. We will not further discuss this important issue though in these lectures.

Chapter 3

Discontinuous Galerkin Method for Conservation Laws

The discontinuous Galerkin method was first designed as an effective numerical method for solving hyperbolic conservation laws, which may have discontinuous solutions. In this section we will discuss the algorithm formulation, stability analysis, and error estimates for the discontinuous Galerkin method solving hyperbolic conservation laws.

3.1 Two-dimensional Steady-State Linear Equations

We now present the details of the original DG method in [37] for the two-dimensional steady-state linear convection equation

$$au_x + bu_y = f(x, y), \qquad 0 \le x,\, y \le 1, \tag{3.1}$$

where a and b are constants. Without loss of generality we assume $a > 0$, $b > 0$. The equation (3.1) is well posed when equipped with the inflow boundary condition

$$u(x, 0) = g_1(x),\ \ 0 \le x \le 1 \qquad \text{and} \qquad u(0, y) = g_2(y),\ \ 0 \le y \le 1. \tag{3.2}$$

For simplicity, we assume a rectangular mesh to cover the computational domain $[0, 1]^2$, consisting of cells

$$I_{i,j} = \left\{(x, y) :\ x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}},\ y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}}\right\}$$

for $1 \le i \le N_x$ and $1 \le j \le N_y$, where

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x+\frac{1}{2}} = 1$$

and

$$0 = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y+\frac{1}{2}} = 1$$

are discretizations in x and y over $[0,1]$. We also denote

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \ \ 1 \le i \le N_x; \qquad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \ \ 1 \le j \le N_y;$$

and

$$h = \max\left(\max_{1\le i\le N_x} \Delta x_i, \ \max_{1\le j\le N_y} \Delta y_j \right).$$

We assume the mesh is regular, namely there is a constant $c > 0$ independent of h such that

$$\Delta x_i \ge ch, \ \ 1 \le i \le N_x; \qquad \Delta y_j \ge ch, \ \ 1 \le j \le N_y.$$

We define a finite-element space consisting of piecewise polynomials

$$V_h^k = \left\{ v : v|_{I_{i,j}} \in P^k(I_{i,j}); \ 1 \le i \le N_x, \, 1 \le j \le N_y \right\}, \tag{3.3}$$

where $P^k(I_{i,j})$ denotes the set of polynomials of degree up to k defined on the cell $I_{i,j}$. Notice that functions in V_h^k may be discontinuous across cell interfaces.

The discontinuous Galerkin (DG) method for solving (3.1) is defined as follows: find the unique function $u_h \in V_h^k$ such that, for all test functions $v_h \in V_h^k$ and all $1 \le i \le N_x$ and $1 \le j \le N_y$, we have

$$\begin{aligned} &- \int\int_{I_{i,j}} (a u_h(v_h)_x + b u_h(v_h)_y)\, dxdy + a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \widehat{u}_h(x_{i+\frac{1}{2}}, y) v_h(x^-_{i+\frac{1}{2}}, y) dy \\ &- a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \widehat{u}_h(x_{i-\frac{1}{2}}, y) v_h(x^+_{i-\frac{1}{2}}, y) dy + b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widehat{u}_h(x, y_{j+\frac{1}{2}}) v_h(x, y^-_{j+\frac{1}{2}}) dx \\ &- b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widehat{u}_h(x, y_{j-\frac{1}{2}}) v_h(x, y^+_{j-\frac{1}{2}}) dx = \int\int_{I_{i,j}} f\, v_h \, dxdy. \end{aligned} \tag{3.4}$$

Here, $\widehat{u}_h$ is the so-called "numerical flux", which is a single-valued function defined at the cell interfaces and in general depending on the values of the numerical solution u_h from both sides of the interface, since u_h is discontinuous there. For the simple linear convection PDE (3.1), the numerical flux can be chosen according to the upwind principle, namely

$$\widehat{u}_h(x_{i+\frac{1}{2}}, y) = u_h(x^-_{i+\frac{1}{2}}, y), \qquad \widehat{u}_h(x, y_{j+\frac{1}{2}}) = u_h(x, y^-_{j+\frac{1}{2}}).$$

Notice that, for the boundary cell $i = 1$, the numerical flux for the left edge is defined using the given boundary condition

$$\widehat{u}_h(x_{\frac{1}{2}}, y) = g_2(y).$$

Likewise, for the boundary cell $j = 1$, the numerical flux for the bottom edge is defined by

$$\widehat{u}_h(x, y_{\frac{1}{2}}) = g_1(x).$$

We now look at the implementation of the scheme (3.4). If a local basis of $P^k(I_{i,j})$ is chosen and denoted as $\varphi^\ell_{i,j}(x, y)$ for $\ell = 1, 2, \ldots, K = (k + 1)(k + 2)/2$, we can express the numerical solution as

$$u_h(x, y) = \sum_{\ell=1}^{K} u^\ell_{i,j} \varphi^\ell_{i,j}(x, y), \qquad (x, y) \in I_{i,j},$$

and we should solve for the coefficients

$$u_{i,j} = \begin{pmatrix} u^1_{i,j} \\ \vdots \\ u^K_{i,j} \end{pmatrix},$$

which, according to the scheme (3.4), satisfies the linear equation

$$A_{i,j} u_{i,j} = rhs \tag{3.5}$$

where $A_{i,j}$ is a $K \times K$ matrix whose (ℓ, m)-th entry is given by

$$\begin{aligned} a^{\ell,m}_{i,j} = &- \int\int_{I_{i,j}} \left(a\varphi^m_{i,j}(x, y)(\varphi^\ell_{i,j}(x, y))_x + b\varphi^m_{i,j}(x, y)(\varphi^\ell_{i,j}(x, y))_y \right) dxdy \\ &+ a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \varphi^m_{i,j}(x_{i+\frac{1}{2}}, y)\varphi^\ell_{i,j}(x_{i+\frac{1}{2}}, y)dy \\ &+ b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \varphi^m_{i,j}(x, y_{j+\frac{1}{2}})\varphi^\ell_{i,j}(x, y_{j+\frac{1}{2}})dx, \end{aligned} \tag{3.6}$$

and the ℓ-th entry of the right-hand side vector is given by

$$\begin{aligned} rhs^\ell = &\, a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u_h(x^-_{i-\frac{1}{2}}, y)\varphi^\ell_{i,j}(x_{i-\frac{1}{2}}, y)dy + b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_h(x, y^-_{j-\frac{1}{2}})\varphi^\ell_{i,j}(x, y_{j-\frac{1}{2}})dx \\ &+ \int_{I_{i,j}} f\, \varphi^\ell_{i,j}\, dxdy, \end{aligned}$$

which depends on the information of u_h in the left cell $I_{i-1,j}$ and the bottom cell $I_{i,j-1}$, if they are in the computational domain, or on the boundary condition, if one or both of these cells are outside the computational domain. It is easy to verify that the matrix $A_{i,j}$ in (3.5) with entries given by (3.6) is invertible, hence the numerical solution u_h in the cell $I_{i,j}$ can be easily obtained by solving the small linear system (3.5), once the solution at the left and bottom cells $I_{i-1,j}$

and $I_{i,j-1}$ are already known, or if one or both of these cells are outside the computational domain. Therefore, we can obtain the numerical solution u_h in the following ordering: first we obtain it in the cell $I_{1,1}$, since both its left and bottom boundaries are equipped with the prescribed boundary conditions (3.2). We then obtain the solution in the cells $I_{2,1}$ and $I_{1,2}$. For $I_{2,1}$, the numerical solution u_h in its left cell $I_{1,1}$ is already available, and its bottom boundary is equipped with the prescribed boundary condition (3.2). Similar argument goes for the cell $I_{1,2}$. The next group of cells to be solved are $I_{3,1}$, $I_{2,2}$, $I_{1,3}$. It is clear that we can obtain the solution u_h sequentially in this way for all cells in the computational domain.

Clearly, this method does not involve any large system solvers and is very easy to implement. In [25], Lesaint and Raviart proved that this method is convergent with the optimal order of accuracy, namely $O(h^{k+1})$, in L^2-norm, when piecewise tensor product polynomials of degree k are used as basis functions. Numerical experiments indicate that the convergence rate is also optimal when the usual piecewise polynomials of degree k given by (3.3) are used instead.

Notice that, even though the method (3.4) is designed for the steady-state problem (3.1), it can be easily used on initial-boundary value problems of linear time-dependent hyperbolic equations: we just need to identify the time variable t as one of the spatial variables. It is also easily generalizable to higher dimensions.

The method described above can be easily designed and efficiently implemented on arbitrary triangulations. L^2-error estimates of $O(h^{k+1/2})$ where k is again the polynomial degree and h is the mesh size can be obtained when the solution is sufficiently smooth, for arbitrary meshes, see, e.g., [24]. This estimate is actually sharp for the most general situation [33], however in many cases the optimal $O(h^{k+1})$ error bound can be proved [39, 9]. In actual numerical computations, one almost always observes the optimal $O(h^{k+1})$ accuracy.

Unfortunately, even though the method (3.4) is easy to implement, accurate, and efficient, it cannot be easily generalized to linear systems, where the characteristic information comes from different directions, or to nonlinear problems, where the characteristic wind direction depends on the solution itself.

3.2 One-dimensional Time-dependent Conservation Laws

The difficulties mentioned at the end of the last subsection can be by-passed when the DG discretization is only used for the spatial variables, and the time discretization is achieved by explicit Runge-Kutta methods such as (2.1). This is the approach of the so-called Runge-Kutta discontinuous Galerkin (RKDG) method [14, 13, 12, 10, 15].

We start our discussion with the one-dimensional conservation law

$$u_t + f(u)_x = 0. \tag{3.7}$$

As before, we assume the following mesh to cover the computational domain $[0,1]$, consisting of cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, for $1 \le i \le N$, where

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = 1.$$

We again denote

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \;\; 1 \le i \le N; \qquad h = \max_{1\le i\le N} \Delta x_i.$$

We assume the mesh is regular, namely there is a constant $c > 0$ independent of h such that

$$\Delta x_i \ge ch, \qquad 1 \le i \le N.$$

We define a finite-element space consisting of piecewise polynomials

$$V_h^k = \left\{ v : v|_{I_i} \in P^k(I_i);\; 1 \le i \le N \right\}, \tag{3.8}$$

where $P^k(I_i)$ denotes the set of polynomials of degree up to k defined on the cell I_i. The semi-discrete DG method for solving (3.7) is defined as follows: find the unique function $u_h = u_h(t) \in V_h^k$ such that, for all test functions $v_h \in V_h^k$ and all $1 \le i \le N$, we have

$$\int_{I_i} (u_h)_t (v_h) dx - \int_{I_i} f(u_h)(v_h)_x dx + \widehat{f}_{i+\frac{1}{2}} v_h(x_{i+\frac{1}{2}}^-) - \widehat{f}_{i-\frac{1}{2}} v_h(x_{i-\frac{1}{2}}^+) = 0. \tag{3.9}$$

Here, $\widehat{f}_{i+\frac{1}{2}}$ is again the numerical flux, which is a single-valued function defined at the cell interfaces and in general depends on the values of the numerical solution u_h from both sides of the interface

$$\widehat{f}_{i+\frac{1}{2}} = \widehat{f}(u_h(x_{i+\frac{1}{2}}^-, t), u_h(x_{i+\frac{1}{2}}^+, t)).$$

We use the so-called monotone fluxes from finite-difference and finite-volume schemes for solving conservation laws, which satisfy the following conditions:

- Consistency: $\widehat{f}(u,u) = f(u)$.
- Continuity: $\widehat{f}(u^-, u^+)$ is at least Lipschitz continuous with respect to both arguments u^- and u^+.
- Monotonicity: $\widehat{f}(u^-, u^+)$ is a non-decreasing function of its first argument u^- and a non-increasing function of its second argument u^+. Symbolically $\widehat{f}(\uparrow, \downarrow)$.

Well-known monotone fluxes include the Lax-Friedrichs flux

$$\widehat{f}^{LF}(u^-, u^+) = \frac{1}{2}\left(f(u^-) + f(u^+) - \alpha(u^+ - u^-) \right), \qquad \alpha = \max_u |f'(u)|;$$

the Godunov flux

$$\widehat{f}^{God}(u^-,u^+)=\begin{cases}\min_{u^-\le u\le u^+} f(u), & \text{if } u^-<u^+,\\ \max_{u^+\le u\le u^-} f(u), & \text{if } u^-\ge u^+;\end{cases}$$

and the Engquist-Osher flux

$$\widehat{f}^{EO}=\int_0^{u^-}\max(f'(u),0)du+\int_0^{u^+}\min(f'(u),0)du+f(0).$$

We refer to, e.g., [26] for more details about monotone fluxes.

3.2.1 Cell Entropy Inequality and L^2-Stability

It is well known that weak solutions of (3.7) may not be unique and the unique, physically relevant weak solution (the so-called entropy solution) satisfies the following entropy inequality

$$U(u)_t+F(u)_x\le 0 \tag{3.10}$$

in distribution sense, for any convex entropy $U(u)$ satisfying $U''(u)\ge 0$ and the corresponding entropy flux $F(u)=\int^u U'(u)f'(u)du$. It will be nice if a numerical approximation to (3.7) also shares a similar entropy inequality as (3.10). It is usually quite difficult to prove a discrete entropy inequality for finite-difference or finite-volume schemes, especially for high-order schemes and when the flux function $f(u)$ in (3.7) is not convex or concave, see, e.g., [28, 32]. However, it turns out that it is easy to prove that the DG scheme (3.9) satisfies a cell entropy inequality [23].

Proposition 3.1. *The solution u_h to the semi-discrete DG scheme* (3.9) *satisfies the following cell entropy inequality*

$$\frac{d}{dt}\int_{I_i}U(u_h)\,dx+\widehat{F}_{i+\frac12}-\widehat{F}_{i-\frac12}\le 0 \tag{3.11}$$

for the square entropy $U(u)=\frac{u^2}{2}$, for some consistent entropy flux

$$\widehat{F}_{i+\frac12}=\widehat{F}(u_h(x^-_{i+\frac12},t),u_h(x^+_{i+\frac12},t))$$

satisfying $\widehat{F}(u,u)=F(u)$.

Proof. We introduce a short-hand notation

$$B_i(u_h;v_h)=\int_{I_i}(u_h)_t(v_h)dx-\int_{I_i}f(u_h)(v_h)_x dx + \widehat{f}_{i+\frac12}v_h(x^-_{i+\frac12})-\widehat{f}_{i-\frac12}v_h(x^+_{i-\frac12}). \tag{3.12}$$

If we take $v_h = u_h$ in the scheme (3.9), we obtain

$$B_i(u_h;u_h)=\int_{I_i}(u_h)_t(u_h)dx-\int_{I_i}f(u_h)(u_h)_x dx + \widehat{f}_{i+\frac{1}{2}}u_h(x^-_{i+\frac{1}{2}})-\widehat{f}_{i-\frac{1}{2}}u_h(x^+_{i-\frac{1}{2}})=0. \quad (3.13)$$

If we denote $\widetilde{F}(u)=\int^u f(u)du$, then (3.13) becomes

$$B_i(u_h;u_h)=\int_{I_i}U(u_h)_t dx-\widetilde{F}(u_h(x^-_{i+\frac{1}{2}})) + \widetilde{F}(u_h(x^+_{i-\frac{1}{2}}))+\widehat{f}_{i+\frac{1}{2}}u_h(x^-_{i+\frac{1}{2}})-\widehat{f}_{i-\frac{1}{2}}u_h(x^+_{i-\frac{1}{2}})=0,$$

or

$$B_i(u_h;u_h)=\int_{I_i}U(u_h)_t dx+\widehat{F}_{i+\frac{1}{2}}-\widehat{F}_{i-\frac{1}{2}}+\Theta_{i-\frac{1}{2}}=0, \quad (3.14)$$

where

$$\widehat{F}_{i+\frac{1}{2}}=-\widetilde{F}(u_h(x^-_{i+\frac{1}{2}}))+\widehat{f}_{i+\frac{1}{2}}u_h(x^-_{i+\frac{1}{2}}), \quad (3.15)$$

and

$$\Theta_{i-\frac{1}{2}}=-\widetilde{F}(u_h(x^-_{i-\frac{1}{2}}))+\widehat{f}_{i-\frac{1}{2}}u_h(x^-_{i-\frac{1}{2}})+\widetilde{F}(u_h(x^+_{i-\frac{1}{2}}))-\widehat{f}_{i-\frac{1}{2}}u_h(x^+_{i-\frac{1}{2}}). \quad (3.16)$$

It is easy to verify that the numerical entropy flux $\widehat{F}$ defined by (3.15) is consistent with the entropy flux $F(u)=\int^u U'(u)f'(u)du$ for $U(u)=\frac{u^2}{2}$. It is also easy to verify

$$\Theta=-\widetilde{F}(u_h^-)+\widehat{f}u_h^-+\widetilde{F}(u_h^+)-\widehat{f}u_h^+=(u_h^+-u_h^-)(\widetilde{F}'(\xi)-\widehat{f})\geq 0,$$

where we have dropped the subscript $i-\frac{1}{2}$ since all quantities are evaluated there in $\Theta_{i-\frac{1}{2}}$. A mean value theorem is applied and ξ is a value between u^- and u^+, and we have used the fact $\widetilde{F}'(\xi)=f(\xi)$ and the monotonicity of the flux function $\widehat{f}$ to obtain the last inequality. This finishes the proof of the cell entropy inequality (3.11). □

We note that the proof does not depend on the accuracy of the scheme, namely it holds for the piecewise polynomial space (3.8) with any degree k. Also, the same proof can be given for the multi-dimensional DG scheme on any triangulation.

The cell entropy inequality trivially implies an L^2-stability of the numerical solution.

Proposition 3.2. *For periodic or compactly supported boundary conditions, the solution u_h to the semi-discrete* DG *scheme* (3.9) *satisfies the following L^2-stability*

$$\frac{d}{dt}\int_0^1(u_h)^2dx\leq 0, \quad (3.17)$$

or

$$\|u_h(\cdot,t)\| \le \|u_h(\cdot,0)\|. \tag{3.18}$$

Here and below, an unmarked norm is the usual L^2-norm.

Proof. We simply sum up the cell entropy inequality (3.11) over i. The flux terms telescope and there is no boundary term left because of the periodic or compact supported boundary condition. (3.17), and hence (3.18), are now immediate. □

Notice that both the cell entropy inequality (3.11) and the L^2-stability (3.17) are valid even when the exact solution of the conservation law (3.7) is discontinuous.

3.2.2 Limiters and Total Variation Stability

For discontinuous solutions, the cell entropy inequality (3.11) and the L^2-stability (3.17), although helpful, are not enough to control spurious numerical oscillations near discontinuities. In practice, especially for problems containing strong discontinuities, we often need to apply nonlinear limiters to control these oscillations and to obtain provable total variation stability.

For simplicity, we first consider the forward Euler time discretization of the semi-discrete DG scheme (3.9). Starting from a preliminary solution $u_h^{n,\mathrm{pre}} \in V_h^k$ at time level n (for the initial condition, $u_h^{0,\mathrm{pre}}$ is taken to be the L^2-projection of the analytical initial condition $u(\cdot,0)$ into V_h^k), we would like to "limit" or "preprocess" it to obtain a new function $u_h^n \in V_h^k$ before advancing it to the next time level: find $u_h^{n+1,\mathrm{pre}} \in V_h^k$ such that, for all test functions $v_h \in V_h^k$ and all $1 \le i \le N$, we have

$$\int_{I_i} \frac{u_h^{n+1,\mathrm{pre}} - u_h^n}{\Delta t} v_h dx - \int_{I_i} f(u_h^n)(v_h)_x dx + \widehat{f}^n_{i+\frac{1}{2}} v_h(x^-_{i+\frac{1}{2}}) - \widehat{f}^n_{i-\frac{1}{2}} v_h(x^+_{i-\frac{1}{2}}) = 0, \tag{3.19}$$

where $\Delta t = t^{n+1} - t^n$ is the time step. This limiting procedure to go from $u_h^{n,\mathrm{pre}}$ to u_h^n should satisfy the following two conditions:

- It should not change the cell averages of $u_h^{n,\mathrm{pre}}$. That is, the cell averages of u_h^n and $u_h^{n,\mathrm{pre}}$ are the same. This is for the conservation property of the DG method.
- It should not affect the accuracy of the scheme in smooth regions. That is, in the smooth regions this limiter does not change the solution, $u_h^n(x) = u_h^{n,\mathrm{pre}}(x)$.

There are many limiters discussed in the literature, and this is still an active research area, especially for multi-dimensional systems, see, e.g., [60]. We will only present an example [13] here.

We denote the cell average of the solution u_h as

$$\bar{u}_i = \frac{1}{\Delta x_i} \int_{I_i} u_h dx, \tag{3.20}$$

and we further denote

$$\widetilde{u}_i = u_h(x^-_{i+\frac{1}{2}}) - \bar{u}_i, \qquad \widetilde{\widetilde{u}}_i = \bar{u}_i - u_h(x^+_{i-\frac{1}{2}}). \tag{3.21}$$

The limiter should not change $\bar{u}_i$ but it may change $\widetilde{u}_i$ and/or $\widetilde{\widetilde{u}}_i$. In particular, the minmod limiter [13] changes $\widetilde{u}_i$ and $\widetilde{\widetilde{u}}_i$ into

$$\widetilde{u}^{(\mathrm{mod})}_i = m(\widetilde{u}_i, \Delta_+\bar{u}_i, \Delta_-\bar{u}_i), \qquad \widetilde{\widetilde{u}}^{(\mathrm{mod})}_i = m(\widetilde{\widetilde{u}}_i, \Delta_+\bar{u}_i, \Delta_-\bar{u}_i), \tag{3.22}$$

where

$$\Delta_+\bar{u}_i = \bar{u}_{i+1} - \bar{u}_i, \qquad \Delta_-\bar{u}_i = \bar{u}_i - \bar{u}_{i-1},$$

and the minmod function m is defined by

$$m(a_1, \cdots, a_\ell) = \begin{cases} s \, \min(|a_1|, \cdots, |a_\ell|), & \text{if } s = \mathrm{sign}(a_1) = \cdots \mathrm{sign}(a_\ell), \\ 0, & \text{otherwise.} \end{cases} \tag{3.23}$$

The limited function $u^{(\mathrm{mod})}_h$ is then recovered to maintain the old cell average (3.20) and the new point values given by (3.22), that is

$$u^{(\mathrm{mod})}_h(x^-_{i+\frac{1}{2}}) = \bar{u}_i + \widetilde{u}^{(\mathrm{mod})}_i, \qquad u^{(\mathrm{mod})}_h(x^+_{i-\frac{1}{2}}) = \bar{u}_i - \widetilde{\widetilde{u}}^{(\mathrm{mod})}_i, \tag{3.24}$$

by the definition (3.21). This recovery is unique for P^k polynomials with $k \le 2$. For $k > 2$, we have extra freedom in obtaining $u^{(\mathrm{mod})}_h$. We could for example choose $u^{(\mathrm{mod})}_h$ to be the unique P^2 polynomial satisfying (3.20) and (3.24).

Before discussing the total variation stability of the DG scheme (3.19) with the pre-processing, we first present a simple lemma due to Harten [22].

Lemma 3.1 (Harten). *If a scheme can be written in the form*

$$u^{n+1}_i = u^n_i + C_{i+\frac{1}{2}}\Delta_+ u^n_i - D_{i-\frac{1}{2}}\Delta_- u^n_i \tag{3.25}$$

with periodic or compactly supported boundary conditions, where $C_{i+\frac{1}{2}}$ and $D_{i-\frac{1}{2}}$ may be nonlinear functions of the grid values u^n_j for $j = i-p, \ldots, i+q$ with some $p, q \ge 0$, satisfying

$$C_{i+\frac{1}{2}} \ge 0, \qquad D_{i+\frac{1}{2}} \ge 0, \qquad C_{i+\frac{1}{2}} + D_{i+\frac{1}{2}} \le 1, \qquad \forall i, \tag{3.26}$$

then the scheme is TVD

$$TV(u^{n+1}) \le TV(u^n),$$

where the total variation seminorm is defined by

$$TV(u) = \sum_i |\Delta_+ u_i|.$$

Proof. Taking the forward difference operation on (3.25) yields

$$\begin{aligned}\Delta_+ u_i^{n+1} &= \Delta_+ u_i^n + C_{i+\frac{3}{2}}\Delta_+ u_{i+1}^n - C_{i+\frac{1}{2}}\Delta_+ u_i^n - D_{i+\frac{1}{2}}\Delta_+ u_i^n + D_{i-\frac{1}{2}}\Delta_- u_i^n \\ &= (1 - C_{i+\frac{1}{2}} - D_{i+\frac{1}{2}})\Delta_+ u_i^n + C_{i+\frac{3}{2}}\Delta_+ u_{i+1}^n + D_{i-\frac{1}{2}}\Delta_- u_i^n.\end{aligned}$$

Thanks to (3.26) and using the periodic or compactly supported boundary condition, we can take the absolute value on both sides of the above equality and sum up over i to obtain

$$\begin{aligned}\sum_i |\Delta_+ u_i^{n+1}| \le \sum_i (1 - C_{i+\frac{1}{2}} - D_{i+\frac{1}{2}})|\Delta_+ u_i^n| \\ + \sum_i C_{i+\frac{1}{2}}|\Delta_+ u_i^n| + \sum_i D_{i+\frac{1}{2}}|\Delta_+ u_i^n| = \sum_i |\Delta_+ u_i^n|.\end{aligned}$$

This finishes the proof. □

We define the "total variation in the means" semi-norm, or TVM, as

$$\mathrm{TVM}(u_h) = \sum_i |\Delta_+ \bar{u}_i|.$$

We then have the following stability result.

Proposition 3.3. *For periodic or compactly supported boundary conditions, the solution u_h^n of the DG scheme* (3.19)*, with the "pre-processing" by the limiter, is total variation diminishing in the means* (TVDM)*, that is*

$$\mathrm{TVM}(u_h^{n+1}) \le \mathrm{TVM}(u_h^n). \tag{3.27}$$

Proof. Taking $v_h = 1$ for $x \in I_i$ in (3.19) and dividing both sides by Δx_i, we obtain, by noticing (3.24),

$$\bar{u}_i^{n+1,\mathrm{pre}} = \bar{u}_i - \lambda_i \left(\widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_{i+1} - \widetilde{\bar{u}}_{i+1}) - \widehat{f}(\bar{u}_{i-1} + \widetilde{u}_{i-1}, \bar{u}_i - \widetilde{\bar{u}}_i) \right),$$

where $\lambda_i = \frac{\Delta t}{\Delta x_i}$, and all quantities on the right-hand side are at the time level n. We can write the right hand side of the above equality in the Harten form (3.25) if we define $C_{i+\frac{1}{2}}$ and $D_{i-\frac{1}{2}}$ as follows

$$\begin{aligned}C_{i+\frac{1}{2}} &= -\lambda_i \frac{\widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_{i+1} - \widetilde{\bar{u}}_{i+1}) - \widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_i - \widetilde{\bar{u}}_i)}{\Delta_+ \bar{u}_i}, \\ D_{i-\frac{1}{2}} &= \lambda_i \frac{\widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_i - \widetilde{\bar{u}}_i) - \widehat{f}(\bar{u}_{i-1} + \widetilde{u}_{i-1}, \bar{u}_i - \widetilde{\bar{u}}_i)}{\Delta_- \bar{u}_i}.\end{aligned} \tag{3.28}$$

We now need to verify that $C_{i+\frac{1}{2}}$ and $D_{i-\frac{1}{2}}$ defined in (3.28) satisfy (3.26). Indeed, we can write $C_{i+\frac{1}{2}}$ as

$$C_{i+\frac{1}{2}} = -\lambda_i \widehat{f}_2 \left[1 - \frac{\widetilde{\bar{u}}_{i+1}}{\Delta_+ \bar{u}_i} + \frac{\widetilde{\bar{u}}_i}{\Delta_+ \bar{u}_i} \right], \tag{3.29}$$

in which $\widehat{f}_2$ is defined as

$$\widehat{f}_2 = \frac{\widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_{i+1} - \widetilde{\widetilde{u}}_{i+1}) - \widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_i - \widetilde{\widetilde{u}}_i)}{(\bar{u}_{i+1} - \widetilde{\widetilde{u}}_{i+1}) - (\bar{u}_i - \widetilde{\widetilde{u}}_i)},$$

and hence

$$0 \le -\lambda_i \widehat{f}_2 = -\lambda_i \frac{\widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_{i+1} - \widetilde{\widetilde{u}}_{i+1}) - \widehat{f}(\bar{u}_i + \widetilde{u}_i, \bar{u}_i - \widetilde{\widetilde{u}}_i)}{(\bar{u}_{i+1} - \widetilde{\widetilde{u}}_{i+1}) - (\bar{u}_i - \widetilde{\widetilde{u}}_i)} \le \lambda_i L_2, \quad (3.30)$$

where we have used the monotonicity and Lipschitz continuity of $\widehat{f}$, and L_2 is the Lipschitz constant of $\widehat{f}$ with respect to its second argument. Also, since u_h^n is the pre-processed solution by the minmod limiter, $\widetilde{\widetilde{u}}_{i+1}$ and $\widetilde{\widetilde{u}}_i$ are the modified values defined by (3.22), hence

$$0 \le \frac{\widetilde{\widetilde{u}}_{i+1}}{\Delta_+ \bar{u}_i} \le 1, \qquad 0 \le \frac{\widetilde{\widetilde{u}}_i}{\Delta_+ \bar{u}_i} \le 1. \quad (3.31)$$

Therefore, we have, by (3.29), (3.30) and (3.31),

$$0 \le C_{i+\frac{1}{2}} \le 2\lambda_i L_2.$$

Similarly, we can show that

$$0 \le D_{i+\frac{1}{2}} \le 2\lambda_{i+1} L_1$$

where L_1 is the Lipschitz constant of $\widehat{f}$ with respect to its first argument. This proves (3.26) if we take the time step so that

$$\lambda \le \frac{1}{2(L_1 + L_2)}$$

where $\lambda = \max_i \lambda_i$. The TVDM property (3.27) then follows from the Harten Lemma and the fact that the limiter does not change cell averages, hence $\mathrm{TVM}(u_h^{n+1}) = \mathrm{TVM}(u_h^{n+1,pre})$. □

Even though the previous proposition is proved only for the first-order Euler forward time discretization, the special TVD (or strong stability preserving, SSP) Runge-Kutta time discretizations [44, 21] allow us to obtain the same stability result for the fully discretized RKDG schemes.

Proposition 3.4. *Under the same conditions as those in Proposition* 3.3*, the solution* u_h^n *of the* DG *scheme* (3.19)*, with the Euler forward time discretization replaced by any* SSP *Runge-Kutta time discretization* [21] *such as* (2.1)*, is* TVDM. □

We still need to verify that the limiter (3.22) does not affect accuracy in smooth regions. If u_h is an approximation to a (locally) smooth function u, then a simple Taylor expansion gives

$$\widetilde{u}_i = \frac{1}{2}u_x(x_i)\Delta x_i + O(h^2), \qquad \widetilde{\widetilde{u}}_i = \frac{1}{2}u_x(x_i)\Delta x_i + O(h^2),$$

while

$$\Delta_+\bar{u}_i = \frac{1}{2}u_x(x_i)(\Delta x_i+\Delta x_{i+1})+O(h^2), \quad \Delta_-\bar{u}_i = \frac{1}{2}u_x(x_i)(\Delta x_i+\Delta x_{i-1})+O(h^2).$$

Clearly, when we are in a smooth and monotone region, namely when $u_x(x_i)$ is away from zero, the first argument in the minmod function (3.22) is of the same sign as the second and third arguments and is smaller in magnitude (for a uniform mesh it is about half of their magnitude), when h is small. Therefore, since the minmod function (3.23) picks the smallest argument (in magnitude) when all the arguments are of the same sign, the modified values $\widetilde{u}_i^{(\mathrm{mod})}$ and $\widetilde{\widetilde{u}}_i^{(\mathrm{mod})}$ in (3.22) will take the unmodified values $\widetilde{u}_i$ and $\widetilde{\widetilde{u}}_i$, respectively. That is, the limiter does not affect accuracy in smooth, monotone regions.

On the other hand, the TVD limiter (3.22) does kill accuracy at smooth extrema. This is demonstrated by numerical results and is a consequence of the general results about TVD schemes, that they are at most second-order accurate for smooth but non-monotone solutions [31]. Therefore, in practice we often use a total variation bounded (TVB) corrected limiter

$$\widetilde{m}(a_1, \cdots, a_\ell) = \begin{cases} a_1, & \text{if } |a_1| \le Mh^2, \\ m(a_1, \ldots, a_\ell), & \text{otherwise,} \end{cases}$$

instead of the original minmod function (3.23), where the TVB parameter M has to be chosen adequately [13]. The DG scheme would then be total variation bounded in the means (TVBM) and uniformly high-order accurate for smooth solutions. We will not discuss more details here and refer the readers to [13].

We would like to remark that the limiters discussed in this subsection were first used for finite-volume schemes [30]. When discussing limiters, the DG methods and finite-volume schemes have many similarities.

3.2.3 Error Estimates for Smooth Solutions

If we assume the exact solution of (3.7) is smooth, we can obtain optimal L^2-error estimates. Such error estimates can be obtained for the general nonlinear conservation law (3.7) and for fully discretized RKDG methods, see [58]. However, for simplicity we will give here the proof only for the semi-discrete DG scheme and the linear version of (3.7):

$$u_t + u_x = 0, \tag{3.32}$$

for which the monotone flux is taken as the simple upwind flux $\widehat{f}(u^-, u^+) = u^-$. Of course the proof is the same for $u_t + au_x = 0$ with any constant a.

Proposition 3.5. *The solution u_h of the* DG *scheme* (3.9) *for the* PDE (3.32) *with a smooth solution u satisfies the error estimate*

$$\|u - u_h\| \leq Ch^{k+1} \tag{3.33}$$

where C depends on u and its derivatives but is independent of h.

Proof. The DG scheme (3.9), when using the notation in (3.12), can be written as

$$B_i(u_h; v_h) = 0, \tag{3.34}$$

for all $v_h \in V_h$ and for all i. It is easy to verify that the exact solution of the PDE (3.32) also satisfies

$$B_i(u; v_h) = 0, \tag{3.35}$$

for all $v_h \in V_h$ and for all i. Subtracting (3.34) from (3.35) and using the linearity of B_i with respect to its first argument, we obtain the error equation

$$B_i(u - u_h; v_h) = 0, \tag{3.36}$$

for all $v_h \in V_h$ and for all i.

We now define a special projection P into V_h. For a given smooth function w, the projection Pw is the unique function in V_h which satisfies, for each i,

$$\int_{I_i} (Pw(x) - w(x))v_h(x)dx = 0 \qquad \forall v_h \in P^{k-1}(I_i); \qquad Pw(x^-_{i+\frac{1}{2}}) = w(x_{i+\frac{1}{2}}). \tag{3.37}$$

Standard approximation theory [7] implies, for a smooth function w,

$$\|Pw(x) - w(x)\| \leq Ch^{k+1} \tag{3.38}$$

where here and below C is a generic constant depending on w and its derivatives but independent of h (which may not have the same value in different places). In particular, in (3.38), $C = \widetilde{C}\|w\|_{H^{k+1}}$ where $\|w\|_{H^{k+1}}$ is the standard Sobolev $(k+1)$ norm and $\widetilde{C}$ is a constant independent of w.

We now take:

$$v_h = Pu - u_h \tag{3.39}$$

in the error equation (3.36), and denote

$$e_h = Pu - u_h, \qquad \varepsilon_h = u - Pu \tag{3.40}$$

to obtain

$$B_i(e_h; e_h) = -B_i(\varepsilon_h; e_h). \tag{3.41}$$

For the left-hand side of (3.41), we use the cell entropy inequality (see (3.14)) to obtain

$$B_i(e_h; e_h) = \frac{1}{2}\frac{d}{dt}\int_{I_i} (e_h)^2 dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}}, \tag{3.42}$$

where $\Theta_{i-\frac{1}{2}} \geq 0$. As to the right-hand side of (3.41), we first write out all the terms

$$-B_i(\varepsilon_h; e_h) = -\int_{I_i} (\varepsilon_h)_t e_h dx + \int_{I_i} \varepsilon_h (e_h)_x dx - (\varepsilon_h)^-_{i+\frac{1}{2}} (e_h)^-_{i+\frac{1}{2}} + (\varepsilon_h)^-_{i-\frac{1}{2}} (e_h)^+_{i+\frac{1}{2}}.$$

Noticing the properties (3.37) of the projection P, we have

$$\int_{I_i} \varepsilon_h (e_h)_x dx = 0$$

because $(e_h)_x$ is a polynomial of degree at most $k-1$, and

$$(\varepsilon_h)^-_{i+\frac{1}{2}} = u_{i+\frac{1}{2}} - (Pu)^-_{i+\frac{1}{2}} = 0$$

for all i. Therefore, the right-hand side of (3.41) becomes

$$-B_i(\varepsilon_h; e_h) = -\int_{I_i} (\varepsilon_h)_t e_h dx \leq \frac{1}{2}\left(\int_{I_i} ((\varepsilon_h)_t)^2 dx + \int_{I_i} (e_h)^2 dx\right). \tag{3.43}$$

Plugging (3.42) and (3.43) into the equality (3.41), summing up over i, and using the approximation result (3.38), we obtain

$$\frac{d}{dt}\int_0^1 (e_h)^2 dx \leq \int_0^1 (e_h)^2 dx + Ch^{2k+2}.$$

A Gronwall's inequality, the fact that the initial error

$$\|u(\cdot, 0) - u_h(\cdot, 0)\| \leq Ch^{k+1}$$

(usually the initial condition $u_h(\cdot, 0)$ is taken as the L^2-projection of the analytical initial condition $u(\cdot, 0)$), and the approximation result (3.38) finally give us the error estimate (3.33). □

3.3 Comments for Multi-dimensional Cases

Even though we have only discussed the two-dimensional steady-state and one-dimensional time-dependent cases in previous subsections, most of the results also hold for multi-dimensional cases with arbitrary triangulations. For example, the semi-discrete DG method for the two-dimensional time-dependent conservation law

$$u_t + f(u)_x + g(u)_y = 0 \tag{3.44}$$

is defined as follows. The computational domain is partitioned into a collection of cells $\triangle_i$, which in 2D could be rectangles, triangles, etc., and the numerical solution is a polynomial of degree k in each cell $\triangle_i$. The degree k could change

with the cell, and there is no continuity requirement of the two polynomials along an interface of two cells. Thus, instead of only one degree of freedom per cell as in a finite-volume scheme, namely the cell average of the solution, there are now $K = \frac{(k+1)(k+2)}{2}$ degrees of freedom per cell for a DG method using piecewise k-th degree polynomials in 2D. These K degrees of freedom are chosen as the coefficients of the polynomial when expanded in a local basis. One could use a locally orthogonal basis to simplify the computation, but this is not essential.

The DG method is obtained by multiplying (3.44) by a test function $v(x, y)$ (which is also a polynomial of degree k in the cell), integrating over the cell $\triangle_j$, and integrating by parts:

$$\frac{d}{dt}\int_{\triangle_j} u(x,y,t)v(x,y)dxdy - \int_{\triangle_j} F(u)\cdot\nabla v\, dxdy + \int_{\partial\triangle_j} F(u)\cdot n\, v\, ds \;=\; 0, \quad (3.45)$$

where $F = (f, g)$, and n is the outward unit normal of the cell boundary $\partial\triangle_j$. The line integral in (3.45) is typically discretized by a Gaussian quadrature of sufficiently high order of accuracy,

$$\int_{\partial\triangle_j} F\cdot n\, v\, ds \;\approx\; |\partial\triangle_j| \sum_{k=1}^{q} \omega_k F(u(G_k,t))\cdot n\, v(G_k),$$

where $F(u(G_k, t)) \cdot n$ is replaced by a numerical flux (approximate or exact Riemann solvers). For scalar equations the numerical flux can be taken as any of the monotone fluxes discussed in Section 3.2 along the normal direction of the cell boundary. For example, one could use the simple Lax-Friedrichs flux, which is given by

$$F(u(G_k,t))\cdot n \approx \frac{1}{2}\left[\left(F(u^-(G_k,t)) + F(u^+(G_k,t))\right)\cdot n - \alpha\left(u^+(G_k,t) - u^-(G_k,t)\right)\right],$$

where α is taken as an upper bound for the eigenvalues of the Jacobian in the n direction, and u^- and u^+ are the values of u inside the cell $\triangle_j$ and outside the cell $\triangle_j$ (inside the neighboring cell) at the Gaussian point G_k. $v(G_k)$ is taken as $v^-(G_k)$, namely the value of v inside the cell $\triangle_j$ at the Gaussian point G_k. The volume integral term $\int_{\triangle_j} F(u)\cdot\nabla v\, dxdy$ can be computed either by a numerical quadrature or by a quadrature free implementation [2] for special systems such as the compressible Euler equations. Notice that if a locally orthogonal basis is chosen, the time derivative term $\frac{d}{dt}\int_{\triangle_j} u(x,y,t)v(x,y)dxdy$ would be explicit and there is no mass matrix to invert. However, even if the local basis is not orthogonal, one still only needs to invert a small $K \times K$ local mass matrix (by hand) and there is never a global mass matrix to invert as in a typical finite-element method.

For scalar equations (3.44), the cell entropy inequality described in Proposition 3.1 holds for arbitrary triangulation. The limiter described in Section 3.2.2 can also be defined for arbitrary triangulation, see [10]. Instead of the TVDM property given in Proposition 3.3, for multi-dimensional cases one can prove the

maximum norm stability of the limited scheme, see [10]. The optimal error estimate given in Proposition 3.5 can be proved for tensor product meshes and basis functions, and for certain specific triangulations when the usual piecewise k-th degree polynomial approximation spaces are used [39, 9]. For the most general cases, an L^2-error estimate of half an order lower $O(h^{k+\frac{1}{2}})$ can be proved [24], which is actually sharp [33].

For nonlinear hyperbolic equations including symmetrizable systems, if the solution of the PDE is smooth, L^2-error estimates of $O(h^{k+1/2} + \Delta t^2)$ where Δt is the time step can be obtained for the fully discrete Runge-Kutta discontinuous Galerkin method with second-order Runge-Kutta time discretization. For upwind fluxes the optimal $O(h^{k+1} + \Delta t^2)$ error estimate can be obtained. See [58, 59].

As an example of the excellent numerical performance of the RKDG scheme, we show in Figures 3.1 and 3.2 the solution of the second order (piecewise linear) and seventh order (piecewise polynomial of degree 6) DG methods for the linear transport equation

$$u_t + u_x = 0, \qquad \text{or} \qquad u_t + u_x + u_y = 0,$$

on the domain $(0, 2\pi) \times (0, T)$ or $(0, 2\pi)^2 \times (0, T)$ with the characteristic function of the interval $(\frac{\pi}{2}, \frac{3\pi}{2})$ or the square $(\frac{\pi}{2}, \frac{3\pi}{2})^2$ as initial condition and periodic boundary conditions [17]. Notice that the solution is for a *very long* time, $t = 100\pi$ (50 time periods), with a relatively coarse mesh. We can see that the second-order scheme smears the fronts, however the seventh-order scheme maintains the shape of the solution almost as well as the initial condition! The excellent performance can be achieved by the DG method on multi-dimensional linear systems using unstructured meshes, hence it is a very good method for solving, e.g. Maxwell equations of electromagnetism and linearized Euler equations of aeroacoustics.

To demonstrate that the DG method also works well for nonlinear systems, we show in Figure 3.3 the DG solution of the forward facing step problem by solving the compressible Euler equations of gas dynamics [15]. We can see that the roll-ups of the contact line caused by a physical instability are resolved well, especially by the third-order DG scheme.

In summary, we can say the following about the discontinuous Galerkin methods for conservation laws:

1. They can be used for arbitrary triangulation, including those with hanging nodes. Moreover, the degree of the polynomial, hence the order of accuracy, in each cell can be independently decided. Thus the method is ideally suited for *h-p* (mesh size and order of accuracy) refinements and adaptivity.
2. The methods have excellent parallel efficiency. Even with space time adaptivity and load balancing the parallel efficiency can still be over 80%, see [38].
3. They should be the methods of choice if geometry is complicated or if adaptivity is important, especially for problems with long time evolution of smooth solutions.

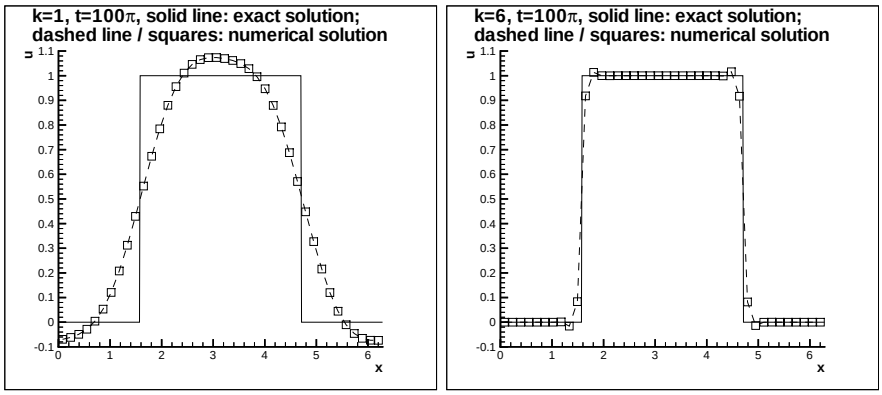


Figure 3.1: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. One-dimensional results with 40 cells, exact solution (solid line) and numerical solution (dashed line and symbols, one point per cell).

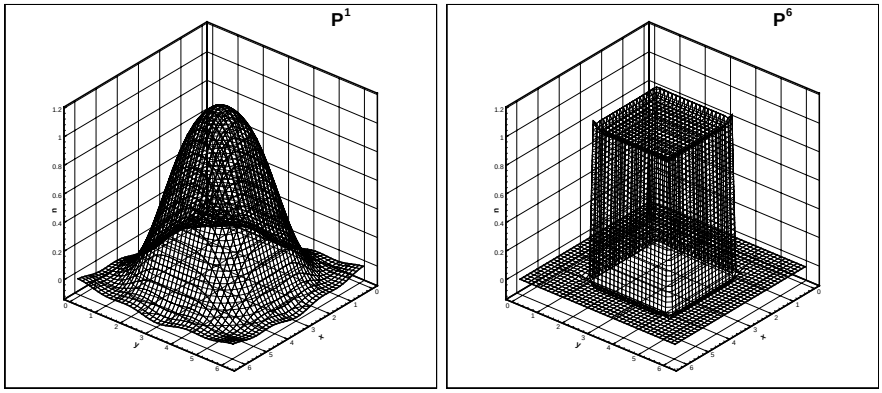


Figure 3.2: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. Two-dimensional results with 40×40 cells.

4. For problems containing strong shocks, the nonlinear limiters are still less robust than the advanced WENO philosophy. There is a parameter (the TVB constant) for the user to tune for each problem, see [13, 10, 15]. For rectangular meshes the limiters work better than for triangular ones. In recent years, WENO based limiters have been investigated [35, 34, 36].

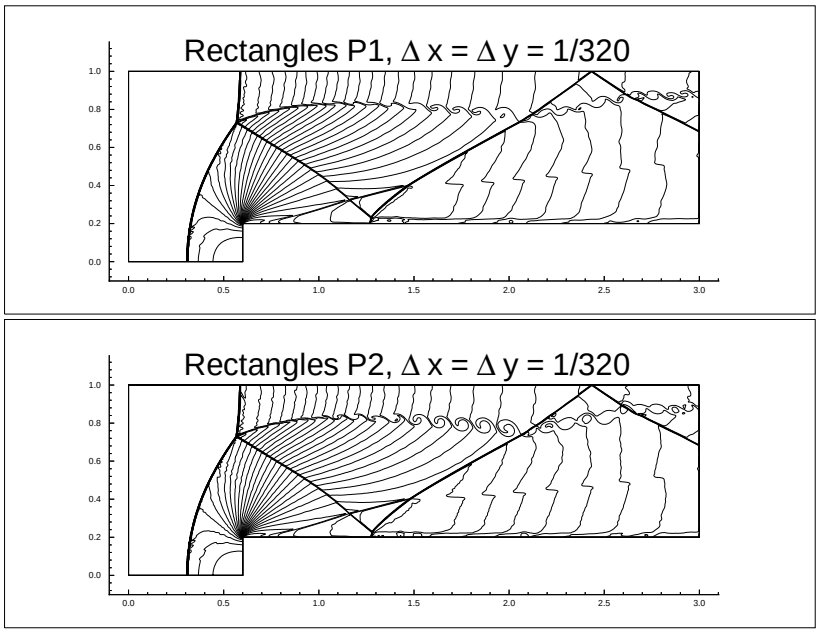


Figure 3.3: Forward facing step. Zoomed-in region. $\Delta x = \Delta y = \frac{1}{320}$. Top: P^1 elements; bottom: P^2 elements.

Chapter 4

Discontinuous Galerkin Method for Convection-Diffusion Equations

In this section we discuss the discontinuous Galerkin method for time-dependent convection-diffusion equations

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} - \sum_{i=1}^{d}\sum_{j=1}^{d} (a_{ij}(u)u_{x_j})_{x_i} = 0, \tag{4.1}$$

where $(a_{ij}(u))$ is a symmetric, semi-positive definite matrix. There are several different formulations of discontinuous Galerkin methods for solving such equations, e.g., [1, 4, 6, 29, 45], however in this section we will only discuss the local discontinuous Galerkin (LDG) method [16].

For equations containing higher-order spatial derivatives, such as the convection-diffusion equation (4.1), discontinuous Galerkin methods cannot be directly applied. This is because the solution space, which consists of piecewise polynomials discontinuous at the element interfaces, is not regular enough to handle higher derivatives. This is a typical "non-conforming" case in finite elements. A naive and careless application of the discontinuous Galerkin method directly to the heat equation containing second derivatives could yield a method which behaves nicely in the computation but is "inconsistent" with the original equation and has $O(1)$ errors to the exact solution [17, 57].

The idea of local discontinuous Galerkin methods for time-dependent partial differential equations with higher derivatives, such as the convection-diffusion equation (4.1), is to rewrite the equation into a first-order system, then apply the discontinuous Galerkin method on the system. A key ingredient for the success of such methods is the correct design of interface numerical fluxes. These fluxes must

be designed to guarantee stability and local solvability of all the auxiliary variables introduced to approximate the derivatives of the solution. The local solvability of all the auxiliary variables is why the method is called a "local" discontinuous Galerkin method in [16].

The first local discontinuous Galerkin method was developed by Cockburn and Shu [16], for the convection-diffusion equation (4.1) containing second derivatives. Their work was motivated by the successful numerical experiments of Bassi and Rebay [3] for the compressible Navier-Stokes equations.

In the following we will discuss the stability and error estimates for the LDG method for convection-diffusion equations. We present details only for the one-dimensional case and will mention briefly the generalization to multi-dimensions in Section 4.4.

4.1 LDG Scheme Formulation

We consider the one-dimensional convection-diffusion equation

$$u_t + f(u)_x = (a(u)u_x)_x \tag{4.2}$$

with $a(u) \geq 0$. We rewrite this equation as the system

$$u_t + f(u)_x = (b(u)q)_x, \qquad q - B(u)_x = 0, \tag{4.3}$$

where

$$b(u) = \sqrt{a(u)}, \qquad B(u) = \int^u b(u)du. \tag{4.4}$$

The finite-element space is still given by (3.8). The semi-discrete LDG scheme is defined as follows. Find $u_h, q_h \in V_h^k$ such that, for all test functions $v_h, p_h \in V_h^k$ and all $1 \leq i \leq N$, we have

$$\begin{aligned} &\int_{I_i} (u_h)_t (v_h)dx - \int_{I_i} (f(u_h) - b(u_h)q_h)(v_h)_x dx \\ &\qquad + (\widehat{f} - \widehat{bq})_{i+\frac{1}{2}} (v_h)^-_{i+\frac{1}{2}} - (\widehat{f} - \widehat{bq})_{i-\frac{1}{2}} (v_h)^+_{i-\frac{1}{2}} = 0, \\ &\int_{I_i} q_h p_h dx + \int_{I_i} B(u_h)(p_h)_x dx - \widehat{B}_{i+\frac{1}{2}} (p_h)^-_{i+\frac{1}{2}} + \widehat{B}_{i-\frac{1}{2}} (p_h)^+_{i-\frac{1}{2}} = 0. \end{aligned} \tag{4.5}$$

Here, all the "hat" terms are the numerical fluxes, namely single-valued functions defined at the cell interfaces which typically depend on the discontinuous numerical solution from both sides of the interface. We already know from Section 3 that the convection flux $\widehat{f}$ should be chosen as a monotone flux. However, the upwinding principle is no longer a valid guiding principle for the design of the diffusion fluxes $\widehat{b}$, $\widehat{q}$ and $\widehat{B}$. In [16], sufficient conditions for the choices of these diffusion fluxes

to guarantee the stability of the scheme (4.5) are given. Here, we will discuss a particularly attractive choice, called "alternating fluxes", defined as

$$\widehat{b} = \frac{B(u_h^+) - B(u_h^-)}{u_h^+ - u_h^-}, \qquad \widehat{q} = q_h^+, \qquad \widehat{B} = B(u_h^-). \tag{4.6}$$

The important point is that $\widehat{q}$ and $\widehat{B}$ should be chosen from different directions. Thus, the choice

$$\widehat{b} = \frac{B(u_h^+) - B(u_h^-)}{u_h^+ - u_h^-}, \qquad \widehat{q} = q_h^-, \qquad \widehat{B} = B(u_h^+)$$

is also fine.

Notice that, from the second equation in the scheme (4.5), we can solve q_h explicitly and locally (in cell I_i) in terms of u_h, by inverting the small mass matrix inside the cell I_i. This is why the method is referred to as the "local" discontinuous Galerkin method.

4.2 Stability Analysis

Similar to the case for hyperbolic conservation laws, we have the following "cell entropy inequality" for the LDG method (4.5).

Proposition 4.1. *The solution u_h, q_h to the semi-discrete* LDG *scheme* (4.5) *satisfies the following "cell entropy inequality"*

$$\frac{1}{2}\frac{d}{dt}\int_{I_i}(u_h)^2\,dx + \int_{I_i}(q_h)^2 dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} \le 0 \tag{4.7}$$

for some consistent entropy flux

$$\widehat{F}_{i+\frac{1}{2}} = \widehat{F}(u_h(x^-_{i+\frac{1}{2}}, t), q_h(x^-_{i+\frac{1}{2}}, t); u_h(x^+_{i+\frac{1}{2}}, t), q_h(x^+_{i+\frac{1}{2}}))$$

satisfying $\widehat{F}(u,u) = F(u) - ub(u)q$ where, as before, $F(u) = \int^u uf'(u)du$.

Proof. We introduce a short-hand notation

$$\begin{aligned} B_i(u_h, q_h; v_h, p_h) = &\int_{I_i}(u_h)_t(v_h)dx - \int_{I_i}(f(u_h) - b(u_h)q_h)(v_h)_x dx \\ &+ (\widehat{f} - \widehat{bq})_{i+\frac{1}{2}}(v_h)^-_{i+\frac{1}{2}} - (\widehat{f} - \widehat{bq})_{i-\frac{1}{2}}(v_h)^+_{i-\frac{1}{2}} \\ &+ \int_{I_i} q_h p_h dx + \int_{I_i} B(u_h)(p_h)_x dx - \widehat{B}_{i+\frac{1}{2}}(p_h)^-_{i+\frac{1}{2}} + \widehat{B}_{i-\frac{1}{2}}(p_h)^+_{i-\frac{1}{2}}. \end{aligned} \tag{4.8}$$

If we take $v_h = u_h$, $p_h = q_h$ in the scheme (4.5), we obtain

$$B_i(u_h, q_h; u_h, q_h) = \int_{I_i} (u_h)_t (u_h) dx \tag{4.9}$$

$$- \int_{I_i} (f(u_h) - b(u_h) q_h)(u_h)_x dx$$

$$+ (\widehat{f} - \widehat{bq})_{i+\frac{1}{2}} (u_h)^-_{i+\frac{1}{2}} - (\widehat{f} - \widehat{bq})_{i-\frac{1}{2}} (u_h)^+_{i-\frac{1}{2}} \tag{4.10}$$

$$+ \int_{I_i} (q_h)^2 dx + \int_{I_i} B(u_h)(q_h)_x dx - \widehat{B}_{i+\frac{1}{2}} (q_h)^-_{i+\frac{1}{2}} + \widehat{B}_{i-\frac{1}{2}} (q_h)^+_{i-\frac{1}{2}}$$

$$= 0.$$

If we denote $\widetilde{F}(u) = \int^u f(u) du$, then (4.9) becomes

$$B_i(u_h, q_h; u_h, q_h) = \frac{1}{2} \frac{d}{dt} \int_{I_i} (u_h)^2 \, dx + \int_{I_i} (q_h)^2 dx$$

$$+ \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}} = 0, \tag{4.11}$$

where

$$\widehat{F} = -\widetilde{F}(u_h^-) + \widehat{f} u_h^- - \widehat{b} q_h^+ u_h^- \tag{4.12}$$

and

$$\Theta = -\widetilde{F}(u_h^-) + \widehat{f} u_h^- + \widetilde{F}(u_h^+) - \widehat{f} u_h^+, \tag{4.13}$$

where we have used the definition of the numerical fluxes (4.6). Notice that we have omitted the subindex $i - \frac{1}{2}$ in the definitions of $\widehat{F}$ and Θ. It is easy to verify that the numerical entropy flux $\widehat{F}$ defined by (4.12) is consistent with the entropy flux $F(u) - ub(u)q$. As Θ in (4.13) is the same as that in (3.16) for the conservation law case, we readily have $\Theta \geq 0$. This finishes the proof of (4.7). □

We again note that the proof does not depend on the accuracy of the scheme, namely it holds for the piecewise polynomial space (3.8) with any degree k. Also, the same proof can be given for multi-dimensional LDG schemes on any triangulation.

As before, the cell entropy inequality trivially implies an L^2-stability of the numerical solution.

Proposition 4.2. *For periodic or compactly supported boundary conditions, the solution u_h, q_h to the semi-discrete* LDG *scheme* (4.5) *satisfies the following L^2-stability*

$$\frac{d}{dt} \int_0^1 (u_h)^2 dx + 2 \int_0^1 (q_h)^2 dx \leq 0, \tag{4.14}$$

or

$$\|u_h(\cdot, t)\| + 2 \int_0^t \|q_h(\cdot, \tau)\| d\tau \leq \|u_h(\cdot, 0)\|. \tag{4.15}$$

□

Notice that both the cell entropy inequality (4.7) and the L^2-stability (4.14) are valid regardless of whether the convection-diffusion equation (4.2) is convection-dominated or diffusion-dominated and regardless of whether the exact solution is smooth or not. The diffusion coefficient $a(u)$ can be degenerate (equal to zero) in any part of the domain. The LDG method is particularly attractive for convection-dominated convection-diffusion equations, when traditional continuous finite-element methods are less stable.

4.3 Error Estimates

Again, if we assume the exact solution of (4.2) is smooth, we can obtain optimal L^2-error estimates. Such error estimates can be obtained for the general nonlinear convection-diffusion equation (4.2), see [53]. However, for simplicity we will give here the proof only for the heat equation:

$$u_t = u_{xx} \tag{4.16}$$

defined on $[0, 1]$ with periodic boundary conditions.

Proposition 4.3. *The solution u_h and q_h to the semi-discrete* DG *scheme* (4.5) *for the* PDE (4.16) *with a smooth solution u satisfies the error estimate*

$$\int_0^1 \left(u(x,t) - u_h(x,t)\right)^2 dx + \int_0^t \int_0^1 \left(u_x(x,\tau) - q_h(x,\tau)\right)^2 dx d\tau \le Ch^{2(k+1)}, \tag{4.17}$$

where C depends on u and its derivatives but is independent of h.

Proof. The DG scheme (4.5), when using the notation in (4.8), can be written as

$$B_i(u_h, q_h; v_h, p_h) = 0, \tag{4.18}$$

for all $v_h, p_h \in V_h$ and for all i. It is easy to verify that the exact solution u and $q = u_x$ of the PDE (4.16) also satisfies

$$B_i(u, q; v_h, p_h) = 0, \tag{4.19}$$

for all $v_h, p_h \in V_h$ and for all i. Subtracting (4.18) from (4.19) and using the linearity of B_i with respect to its first two arguments, we obtain the error equation

$$B_i(u - u_h, q - q_h; v_h, p_h) = 0, \tag{4.20}$$

for all $v_h, p_h \in V_h$ and for all i.

Recall the special projection P defined in (3.37). We also define another special projection Q as follows. For a given smooth function w, the projection Qw is the unique function in V_h which satisfies, for each i,

$$\int_{I_i} (Qw(x) - w(x))v_h(x)dx = 0 \qquad \forall v_h \in P^{k-1}(I_i); \qquad Qw(x^+_{i-\frac{1}{2}}) = w(x_{i-\frac{1}{2}}). \tag{4.21}$$

Similar to P, we also have, by standard approximation theory [7], that

$$\|Qw(x) - w(x)\| \le Ch^{k+1}, \qquad \forall w \in H^{k+1}(\Omega), \tag{4.22}$$

where C is a constant depending on w and its derivatives but independent of h.

We now take

$$v_h = Pu - u_h, \qquad p_h = Qq - q_h \tag{4.23}$$

in the error equation (4.20), and denote

$$e_h = Pu - u_h, \;\; \bar{e}_h = Qq - q_h; \qquad \varepsilon_h = u - Pu, \;\; \bar{\varepsilon}_h = q - Qq, \tag{4.24}$$

to obtain

$$B_i(e_h, \bar{e}_h; e_h, \bar{e}_h) = -B_i(\varepsilon_h, \bar{\varepsilon}_h; e_h, \bar{e}_h). \tag{4.25}$$

For the left-hand side of (4.25), we use the cell entropy inequality (see (4.11)) to obtain

$$B_i(e_h, \bar{e}_h; e_h, \bar{e}_h) = \frac{1}{2}\frac{d}{dt}\int_{I_i} (e_h)^2 dx + \int_{I_i} (\bar{e}_h)^2 dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}}, \tag{4.26}$$

where $\Theta_{i-\frac{1}{2}} \ge 0$ (in fact we can easily verify, from (4.13), that $\Theta_{i-\frac{1}{2}} = 0$ for the special case of the heat equation (4.16)). As to the right-hand side of (4.25), we first write out all the terms

$$\begin{aligned} -B_i(\varepsilon_h, \bar{\varepsilon}_h; e_h, \bar{e}_h) = &- \int_{I_i} (\varepsilon_h)_t e_h dx \\ &- \int_{I_i} \bar{\varepsilon}_h (e_h)_x dx + (\bar{\varepsilon}_h)^+_{i+\frac{1}{2}} (e_h)^-_{i+\frac{1}{2}} - (\bar{\varepsilon}_h)^+_{i-\frac{1}{2}} (e_h)^+_{i-\frac{1}{2}} \\ &- \int_{I_i} \bar{\varepsilon}_h \bar{e}_h dx \\ &- \int_{I_i} \varepsilon_h (\bar{e}_h)_x dx + (\varepsilon_h)^-_{i+\frac{1}{2}} (\bar{e}_h)^-_{i+\frac{1}{2}} - (\varepsilon_h)^-_{i-\frac{1}{2}} (\bar{e}_h)^+_{i-\frac{1}{2}}. \end{aligned}$$

Noticing the properties (3.37) and (4.21) of the projections P and Q, we have

$$\int_{I_i} \bar{\varepsilon}_h (e_h)_x dx = 0, \qquad \int_{I_i} \varepsilon_h (\bar{e}_h)_x dx = 0,$$

because $(e_h)_x$ and $(\bar{e}_h)_x$ are polynomials of degree at most $k-1$, and

$$(\varepsilon_h)^-_{i+\frac{1}{2}} = u_{i+\frac{1}{2}} - (Pu)^-_{i+\frac{1}{2}} = 0, \qquad (\bar{\varepsilon}_h)^+_{i+\frac{1}{2}} = q_{i+\frac{1}{2}} - (Qq)^+_{i+\frac{1}{2}} = 0,$$

for all i. Therefore, the right-hand side of (4.25) becomes

$$\begin{aligned} -B_i(\varepsilon_h, \bar{\varepsilon}_h; e_h, \bar{e}_h) &= -\int_{I_i} (\varepsilon_h)_t e_h dx - \int_{I_i} \bar{\varepsilon}_h \bar{e}_h dx \\ &\le \frac{1}{2}\left(\int_{I_i} ((\varepsilon_h)_t)^2 dx + \int_{I_i} (e_h)^2 dx + \int_{I_i} (\bar{\varepsilon}_h)^2 dx + \int_{I_i} (\bar{e}_h)^2 dx \right). \end{aligned} \tag{4.27}$$

Plugging (4.26) and (4.27) into the equality (4.25), summing up over i, and using the approximation results (3.38) and (4.22), we obtain

$$\frac{d}{dt}\int_0^1 (e_h)^2 dx + \int_0^1 (\bar{e}_h)^2 dx \leq \int_0^1 (e_h)^2 dx + Ch^{2k+2}.$$

A Gronwall's inequality, the fact that the initial error

$$\|u(\cdot,0) - u_h(\cdot,0)\| \leq Ch^{k+1}$$

and the approximation results (3.38) and (4.22) finally give us the error estimate (4.17). □

4.4 Multi-Dimensions

Even though we have only discussed one-dimensional cases in this section, the algorithm and its analysis can be easily generalized to the multi-dimensional equation (4.1). The stability analysis is the same as for the one-dimensional case in Section 4.2. The optimal $O(h^{k+1})$ error estimates can be obtained on tensor product meshes and polynomial spaces, along the same line as that in Section 4.3. For general triangulations and piecewise polynomials of degree k, a sub-optimal error estimate of $O(h^k)$ can be obtained. We will not provide the details here and refer to [16, 53].

Chapter 5

Discontinuous Galerkin Method for PDEs Containing Higher-Order Spatial Derivatives

We now consider the DG method for solving PDEs containing higher-order spatial derivatives. Even though there are other possible DG schemes for such PDEs, e.g. those designed in [6], we will only discuss the local discontinuous Galerkin (LDG) method in this section.

5.1 LDG Scheme for the KdV Equations

We first consider PDEs containing third spatial derivatives. These are usually nonlinear dispersive wave equations, for example the following general KdV-type equations

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} + \sum_{i=1}^{d} \Big(r_i'(u) \sum_{j=1}^{d} g_{ij}(r_i(u)_{x_i})_{x_j} \Big)_{x_i} = 0, \tag{5.1}$$

where $f_i(u)$, $r_i(u)$ and $g_{ij}(q)$ are arbitrary (smooth) nonlinear functions. The one-dimensional KdV equation

$$u_t + (\alpha u + \beta u^2)_x + \sigma u_{xxx} = 0, \tag{5.2}$$

where α, β and σ are constants, is a special case of the general class (5.1).

Stable LDG schemes for solving (5.1) were first designed in [55]. We will concentrate our discussion on the one-dimensional case. For the one-dimensional generalized KdV-type equations

$$u_t + f(u)_x + (r'(u)g(r(u)_x)_x)_x = 0, \tag{5.3}$$

where $f(u)$, $r(u)$ and $g(q)$ are arbitrary (smooth) nonlinear functions, the LDG method is based on rewriting it as the following system,

$$u_t + (f(u) + r'(u)p)_x = 0, \qquad p - g(q)_x = 0, \qquad q - r(u)_x = 0. \tag{5.4}$$

The finite-element space is still given by (3.8). The semi-discrete LDG scheme is defined as follows. Find $u_h, p_h, q_h \in V_h^k$ such that, for all test functions $v_h, w_h, z_h \in V_h^k$ and all $1 \le i \le N$, we have

$$\begin{aligned}
&\int_{I_i} (u_h)_t (v_h) dx - \int_{I_i} (f(u_h) + r'(u_h)p_h)(v_h)_x dx \\
&\quad + (\widehat{f} + \widehat{r'}\widehat{p})_{i+\frac{1}{2}} (v_h)^-_{i+\frac{1}{2}} - (\widehat{f} + \widehat{r'}\widehat{p})_{i-\frac{1}{2}} (v_h)^+_{i-\frac{1}{2}} = 0, \\
&\int_{I_i} p_h w_h dx + \int_{I_i} g(q_h)(w_h)_x dx - \widehat{g}_{i+\frac{1}{2}} (w_h)^-_{i+\frac{1}{2}} + \widehat{g}_{i-\frac{1}{2}} (w_h)^+_{i-\frac{1}{2}} = 0, \\
&\int_{I_i} q_h z_h dx + \int_{I_i} r(u_h)(z_h)_x dx - \widehat{r}_{i+\frac{1}{2}} (z_h)^-_{i+\frac{1}{2}} + \widehat{r}_{i-\frac{1}{2}} (z_h)^+_{i-\frac{1}{2}} = 0.
\end{aligned} \tag{5.5}$$

Here again, all the "hat" terms are the numerical fluxes, namely single-valued functions defined at the cell interfaces which typically depend on the discontinuous numerical solution from both sides of the interface. We already know from Section 3 that the convection flux $\widehat{f}$ should be chosen as a monotone flux. It is important to design the other fluxes suitably in order to guarantee stability of the resulting LDG scheme. In fact, the upwinding principle is still a valid guiding principle here, since the KdV-type equation (5.3) is a dispersive wave equation for which waves are propagating with a direction. For example, the simple linear equation

$$u_t + u_{xxx} = 0,$$

which corresponds to (5.3) with $f(u) = 0$, $r(u) = u$ and $g(q) = q$, admits the following simple wave solution

$$u(x,t) = \sin(x+t),$$

that is, information propagates from right to left. This motivates the following choice of numerical fluxes, discovered in [55]:

$$\widehat{r'} = \frac{r(u_h^+) - r(u_h^-)}{u_h^+ - u_h^-}, \qquad \widehat{p} = p_h^+, \qquad \widehat{g} = \widehat{g}(q_h^-, q_h^+), \qquad \widehat{r} = r(u_h^-). \tag{5.6}$$

Here, $-\widehat{g}(q_h^-, q_h^+)$ is a monotone flux for $-g(q)$, namely $\widehat{g}$ is a non-increasing function in the first argument and a non-decreasing function in the second argument. The important point is again the "alternating fluxes", namely $\widehat{p}$ and $\widehat{r}$ should come from opposite sides. Thus

$$\widehat{r'} = \frac{r(u_h^+) - r(u_h^-)}{u_h^+ - u_h^-}, \qquad \widehat{p} = p_h^-, \qquad \widehat{g} = \widehat{g}(q_h^-, q_h^+), \qquad \widehat{r} = r(u_h^+)$$

would also work.

Notice that, from the third equation in the scheme (5.5), we can solve q_h explicitly and locally (in cell I_i) in terms of u_h, by inverting the small mass matrix inside the cell I_i. Then, from the second equation in the scheme (5.5), we can solve p_h explicitly and locally (in cell I_i) in terms of q_h. Thus only u_h is the global unknown and the auxiliary variables q_h and p_h can be solved in terms of u_h locally. This is why the method is referred to as the "local" discontinuous Galerkin method.

5.1.1 Stability Analysis

Similar to the case for hyperbolic conservation laws and convection-diffusion equations, we have the following "cell entropy inequality" for the LDG method (5.5).

Proposition 5.1. *The solution u_h to the semi-discrete* LDG *scheme* (5.5) *satisfies the following "cell entropy inequality"*

$$\frac{1}{2}\frac{d}{dt}\int_{I_i} (u_h)^2\, dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} \le 0 \tag{5.7}$$

for some consistent entropy flux

$$\widehat{F}_{i+\frac{1}{2}} = \widehat{F}(u_h(x^-_{i+\frac{1}{2}}, t), p_h(x^-_{i+\frac{1}{2}}, t), q_h(x^-_{i+\frac{1}{2}}, t); u_h(x^+_{i+\frac{1}{2}}, t), p_h(x^+_{i+\frac{1}{2}}, t), q_h(x^+_{i+\frac{1}{2}}))$$

satisfying $\widehat{F}(u,u) = F(u) + ur'(u)p - G(q)$ where $F(u) = \int^u uf'(u)du$ and $G(q) = \int^q qg(q)dq$.

Proof. We introduce a short-hand notation

$$\begin{aligned} B_i(u_h,p_h,q_h;v_h,w_h,z_h) &= \int_{I_i}(u_h)_t(v_h)dx - \int_{I_i}(f(u_h)+r'(u_h)p_h)(v_h)_x dx \\ &+ (\widehat{f}+\widehat{r'}\widehat{p})_{i+\frac{1}{2}}(v_h)^-_{i+\frac{1}{2}} - (\widehat{f}+\widehat{r'}\widehat{p})_{i-\frac{1}{2}}(v_h)^+_{i-\frac{1}{2}} \\ &+ \int_{I_i} p_h w_h dx \\ &+ \int_{I_i} g(q_h)(w_h)_x dx - \widehat{g}_{i+\frac{1}{2}}(w_h)^-_{i+\frac{1}{2}} + \widehat{g}_{i-\frac{1}{2}}(w_h)^+_{i-\frac{1}{2}} \\ &+ \int_{I_i} q_h z_h dx \\ &+ \int_{I_i} r(u_h)(z_h)_x dx - \widehat{r}_{i+\frac{1}{2}}(z_h)^-_{i+\frac{1}{2}} + \widehat{r}_{i-\frac{1}{2}}(z_h)^+_{i-\frac{1}{2}}. \end{aligned} \tag{5.8}$$

If we take $v_h = u_h$, $w_h = q_h$ and $z_h = -p_h$ in the scheme (5.5), we obtain

$$\begin{aligned} B_i(u_h,p_h,q_h;u_h,q_h,-p_h) &= \int_{I_i}(u_h)_t(u_h)dx \\ &- \int_{I_i}(f(u_h)+r'(u_h)p_h)(u_h)_x dx \\ &+ (\widehat{f}+\widehat{r'}\widehat{p})_{i+\frac{1}{2}}(u_h)^-_{i+\frac{1}{2}} \\ &- (\widehat{f}+\widehat{r'}\widehat{p})_{i-\frac{1}{2}}(u_h)^+_{i-\frac{1}{2}} \\ &+ \int_{I_i} p_h q_h dx \\ &+ \int_{I_i} g(q_h)(q_h)_x dx - \widehat{g}_{i+\frac{1}{2}}(q_h)^-_{i+\frac{1}{2}} + \widehat{g}_{i-\frac{1}{2}}(q_h)^+_{i-\frac{1}{2}} \\ &- \int_{I_i} q_h p_h dx \\ &- \int_{I_i} r(u_h)(p_h)_x dx + \widehat{r}_{i+\frac{1}{2}}(p_h)^-_{i+\frac{1}{2}} - \widehat{r}_{i-\frac{1}{2}}(p_h)^+_{i-\frac{1}{2}} \\ &= 0. \end{aligned} \tag{5.9}$$

If we denote $\widetilde{F}(u) = \int^u f(u)du$ and $\widetilde{G}(q) = \int^q g(q)dq$, then (5.9) becomes

$$B_i(u_h,p_h,q_h;u_h,q_h,-p_h) = \frac{1}{2}\frac{d}{dt}\int_{I_i}(u_h)^2\,dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}} = 0, \tag{5.10}$$

where

$$\widehat{F} = -\widetilde{F}(u_h^-) + \widehat{f}u_h^- + \widetilde{G}(q_h^-) + \widehat{r'}p_h^+u_h^- - \widehat{g}q_h^-, \tag{5.11}$$

and

$$\Theta = \left(-\widetilde{F}(u_h^-) + \widehat{f}u_h^- + \widetilde{F}(u_h^+) - \widehat{f}u_h^+\right) + \left(\widetilde{G}(q_h^-) - \widehat{g}q_h^- - \widetilde{G}(q_h^+) + \widehat{g}q_h^+\right), \quad (5.12)$$

where we have used the definition of the numerical fluxes (5.6). Notice that we have omitted the subindex $i - \frac{1}{2}$ in the definitions of $\widehat{F}$ and Θ. It is easy to verify that the numerical entropy flux $\widehat{F}$ defined by (5.11) is consistent with the entropy flux $F(u) + ur'(u)p - G(q)$. The terms inside the first parenthesis for Θ in (5.12) are the same as that in (3.16) for the conservation law case; those inside the second parenthesis are the same as those inside the first parenthesis, if we replace q_h by u_h, $-\widetilde{G}$ by $\widetilde{F}$, and $-\widehat{g}$ by $\widehat{f}$ (recall that $-\widehat{g}$ is a monotone flux). We therefore readily have $\Theta \geq 0$. This finishes the proof of (5.7). □

We observe once more that the proof does not depend on the accuracy of the scheme, namely it holds for the piecewise polynomial space (3.8) with any degree k. Also, the same proof can be given for the multi-dimensional LDG scheme solving (5.1) on any triangulation.

As before, the cell entropy inequality trivially implies an L^2-stability of the numerical solution.

Proposition 5.2. *For periodic or compactly supported boundary conditions, the solution u_h to the semi-discrete* LDG *scheme* (5.5) *satisfies the L^2-stability*

$$\frac{d}{dt}\int_0^1 (u_h)^2 dx \leq 0, \quad (5.13)$$

or

$$\|u_h(\cdot, t)\| \leq \|u_h(\cdot, 0)\|. \quad (5.14)$$

□

Again, both the cell entropy inequality (5.7) and the L^2-stability (5.13) are valid regardless of whether the KdV-type equation (5.3) is convection-dominated or dispersion-dominated and regardless of whether the exact solution is smooth or not. The dispersion flux $r'(u)g(r(u)_x)_x$ can be degenerate (equal to zero) in any part of the domain. The LDG method is particularly attractive for convection-dominated convection-dispersion equations, when traditional continuous finite-element methods may be less stable. In [55], this LDG method is used to study the dispersion limit of the Burgers equation, for which the third derivative dispersion term in (5.3) has a small coefficient which tends to zero.

5.1.2 Error Estimates

For error estimates we once again assume the exact solution of (5.3) is smooth. The error estimates can be obtained for a general class of nonlinear convection-dispersion equations which is a subclass of (5.3), see [53]. However, for simplicity we will give here only the proof for the linear equation

$$u_t + u_x + u_{xxx} = 0 \quad (5.15)$$

defined on $[0,1]$ with periodic boundary conditions.

Proposition 5.3. *The solution u_h to the semi-discrete* LDG *scheme* (5.5) *for the* PDE (5.15) *with a smooth solution u satisfies the following error estimate*

$$\|u - u_h\| \le Ch^{k+\frac{1}{2}}, \tag{5.16}$$

where C depends on u and its derivatives but is independent of h.

Proof. The LDG scheme (5.5), when using the notation in (5.8), can be written as

$$B_i(u_h, p_h, q_h; v_h, w_h, z_h) = 0, \tag{5.17}$$

for all $v_h, w_h, z_h \in V_h$ and for all i. It is easy to verify that the exact solution u, $q = u_x$ and $p = u_{xx}$ of the PDE (5.15) also satisfies

$$B_i(u, p, q; v_h, w_h, z_h) = 0, \tag{5.18}$$

for all $v_h, w_h, z_h \in V_h$ and for all i. Subtracting (5.17) from (5.18) and using the linearity of B_i with respect to its first three arguments, we obtain the error equation

$$B_i(u - u_h, p - p_h, q - q_h; v_h, w_h, z_h) = 0, \tag{5.19}$$

for all $v_h, w_h, z_h \in V_h$ and for all i.

Recall the special projection P defined in (3.37). We also denote the standard L^2-projection as R: for a given smooth function w, the projection Rw is the unique function in V_h which satisfies, for each i,

$$\int_{I_i} (Rw(x) - w(x))v_h(x)dx = 0 \qquad \forall v_h \in P^k(I_i). \tag{5.20}$$

Similar to P, we also have, by the standard approximation theory [7], that

$$\|Rw(x) - w(x)\| + \sqrt{h}\|Rw(x) - w(x)\|_\Gamma \le Ch^{k+1} \tag{5.21}$$

for a smooth function w, where C is a constant depending on w and its derivatives but independent of h, and $\|v\|_\Gamma$ is the usual L^2-norm on the cell interfaces of the mesh, which for this one-dimensional case is

$$\|v\|_\Gamma^2 = \sum_i \left((v_{i+\frac{1}{2}}^-)^2 + (v_{i-\frac{1}{2}}^+)^2 \right).$$

We now take

$$v_h = Pu - u_h, \qquad w_h = Rq - q_h, \qquad z_h = p_h - Rp \tag{5.22}$$

in the error equation (5.19), and denote

$$\begin{aligned} &e_h = Pu - u_h, \ \ \bar{e}_h = Rq - q_h, \\ &\bar{\bar{e}}_h = Rp - p_h; \quad \varepsilon_h = u - Pu, \ \ \bar{\varepsilon}_h = q - Rq, \ \ \bar{\bar{\varepsilon}}_h = p - Rp, \end{aligned} \tag{5.23}$$

to obtain

$$B_i(e_h, \bar{\bar{e}}_h, \bar{e}_h, ; e_h, \bar{e}_h, -\bar{\bar{e}}_h) = -B_i(\varepsilon_h, \bar{\bar{\varepsilon}}_h, \bar{\varepsilon}_h; e_h, \bar{e}_h, -\bar{\bar{e}}_h). \tag{5.24}$$

For the left-hand side of (5.24), we use the cell entropy inequality (see (5.10)) to obtain

$$B_i(e_h, \bar{\bar{e}}_h, \bar{e}_h, ; e_h, \bar{e}_h, -\bar{\bar{e}}_h) = \frac{1}{2}\frac{d}{dt}\int_{I_i} (e_h)^2 dx + \widehat{F}_{i+\frac{1}{2}} - \widehat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}} \tag{5.25}$$

where we can easily verify, based on the formula (5.12) and for the PDE (5.15), that

$$\Theta_{i-\frac{1}{2}} = \frac{1}{2}\left((e_h)^+_{i-\frac{1}{2}} - (e_h)^-_{i-\frac{1}{2}}\right)^2 + \frac{1}{2}\left((\bar{e}_h)^+_{i-\frac{1}{2}} - (\bar{e}_h)^-_{i-\frac{1}{2}}\right)^2. \tag{5.26}$$

As to the right-hand side of (5.24), we first write out all the terms

$$\begin{aligned}
&-B_i(\varepsilon_h, \bar{\bar{\varepsilon}}_h, \bar{\varepsilon}_h; e_h, \bar{e}_h, -\bar{\bar{e}}_h) \\
&\quad = -\int_{I_i} (\varepsilon_h)_t e_h dx \\
&\quad + \int_{I_i} (\varepsilon_h + \bar{\bar{\varepsilon}}_h)(e_h)_x dx - (\varepsilon_h^- + \bar{\bar{\varepsilon}}_h^+)_{i+\frac{1}{2}}(e_h)^-_{i+\frac{1}{2}} + (\varepsilon_h^- + \bar{\bar{\varepsilon}}_h^+)_{i-\frac{1}{2}}(e_h)^+_{i-\frac{1}{2}} \\
&\quad - \int_{I_i} \bar{\bar{\varepsilon}}_h \bar{e}_h dx - \int_{I_i} \bar{\varepsilon}_h (\bar{e}_h)_x dx + (\bar{\varepsilon}_h)^+_{i+\frac{1}{2}}(\bar{e}_h)^-_{i+\frac{1}{2}} - (\bar{\varepsilon}_h)^+_{i-\frac{1}{2}}(\bar{e}_h)^+_{i-\frac{1}{2}} \\
&\quad + \int_{I_i} \bar{\varepsilon}_h \bar{\bar{e}}_h dx + \int_{I_i} \varepsilon_h (\bar{\bar{e}}_h)_x dx - (\varepsilon_h)^-_{i+\frac{1}{2}}(\bar{\bar{e}}_h)^-_{i+\frac{1}{2}} + (\varepsilon_h)^-_{i-\frac{1}{2}}(\bar{\bar{e}}_h)^+_{i-\frac{1}{2}}.
\end{aligned}$$

Noticing the properties (3.37) and (5.20) of the projections P and R, we have

$$\int_{I_i} (\varepsilon_h + \bar{\bar{\varepsilon}}_h)(e_h)_x dx = 0, \qquad \int_{I_i} \bar{\bar{\varepsilon}}_h \bar{e}_h dx = 0, \qquad \int_{I_i} \bar{\varepsilon}_h (\bar{e}_h)_x dx = 0,$$

$$\int_{I_i} \bar{\varepsilon}_h \bar{\bar{e}}_h dx = 0, \qquad \int_{I_i} \varepsilon_h (\bar{\bar{e}}_h)_x dx = 0,$$

because $(e_h)_x$, $(\bar{e}_h)_x$ and $(\bar{\bar{e}}_h)_x$ are polynomials of degree at most $k-1$, and $\bar{e}_h$ and $\bar{\bar{e}}_h$ are polynomials of degree at most k. Also,

$$(\varepsilon_h)^-_{i+\frac{1}{2}} = u_{i+\frac{1}{2}} - (Pu)^-_{i+\frac{1}{2}} = 0$$

for all i. Therefore, the right-hand side of (5.24) becomes

$$\begin{aligned}
&-B_i(\varepsilon_h, \bar{\bar{\varepsilon}}_h, \bar{\varepsilon}_h; e_h, \bar{e}_h, -\bar{\bar{e}}_h) \\
&\quad = -\int_{I_i} (\varepsilon_h)_t e_h dx - (\bar{\bar{\varepsilon}}_h)^+_{i+\frac{1}{2}}(e_h)^-_{i+\frac{1}{2}} + (\bar{\bar{\varepsilon}}_h)^+_{i-\frac{1}{2}}(e_h)^+_{i-\frac{1}{2}} \\
&\qquad\qquad + (\bar{\varepsilon}_h)^+_{i+\frac{1}{2}}(\bar{e}_h)^-_{i+\frac{1}{2}} - (\bar{\varepsilon}_h)^+_{i-\frac{1}{2}}(\bar{e}_h)^+_{i-\frac{1}{2}}
\end{aligned}$$

$$
\begin{aligned}
&= -\int_{I_i} (\varepsilon_h)_t e_h dx + \widehat{H}_{i+\frac{1}{2}} - \widehat{H}_{i-\frac{1}{2}} \\
&\quad + (\bar{\bar{\varepsilon}}_h)^+_{i-\frac{1}{2}} \left((e_h)^+_{i-\frac{1}{2}} - (e_h)^-_{i-\frac{1}{2}} \right) - (\bar{\varepsilon}_h)^+_{i-\frac{1}{2}} \left((\bar{e}_h)^+_{i-\frac{1}{2}} - (\bar{e}_h)^-_{i-\frac{1}{2}} \right) \qquad (5.27) \\
&\leq \widehat{H}_{i+\frac{1}{2}} - \widehat{H}_{i-\frac{1}{2}} + \frac{1}{2} \left[\int_{I_i} ((\varepsilon_h)_t)^2 dx + \int_{I_i} (e_h)^2 dx \right. \\
&\qquad\qquad + \left((\bar{\bar{\varepsilon}}_h)^+_{i-\frac{1}{2}} \right)^2 + \left((e_h)^+_{i-\frac{1}{2}} - (e_h)^-_{i-\frac{1}{2}} \right)^2 \\
&\qquad\qquad \left. + \left((\bar{\varepsilon}_h)^+_{i-\frac{1}{2}} \right)^2 + \left((\bar{e}_h)^+_{i-\frac{1}{2}} - (\bar{e}_h)^-_{i-\frac{1}{2}} \right)^2 \right].
\end{aligned}
$$

Plugging (5.25), (5.26) and (5.27) into the equality (5.24), summing up over i, and using the approximation results (3.38) and (5.21), we obtain

$$\frac{d}{dt} \int_0^1 (e_h)^2 dx \leq \int_0^1 (e_h)^2 dx + Ch^{2k+1}.$$

A Gronwall's inequality, the fact that the initial error

$$\|u(\cdot, 0) - u_h(\cdot, 0)\| \leq Ch^{k+1},$$

and the approximation results (3.38) and (5.21) finally give us the error estimate (5.16). □

We note that the error estimate (5.16) is half an order lower than optimal. Technically, this is because we are unable to use the special projections as before to eliminate the interface terms involving $\bar{\varepsilon}_h$ and $\bar{\bar{\varepsilon}}_h$ in (5.27). Numerical experiments in [55] indicate that both the L^2- and L^∞-errors are of the optimal $(k+1)$-th order of accuracy.

5.2 LDG Schemes for Other Higher-Order PDEs

In this subsection we list some of the higher-order PDEs for which stable DG methods have been designed in the literature. We will concentrate on the discussion of LDG schemes.

5.2.1 Bi-harmonic Equations

An LDG scheme for solving the time-dependent convection-bi-harmonic equation

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} + \sum_{i=1}^{d} (a_i(u_{x_i}) u_{x_i x_i})_{x_i x_i} = 0, \qquad (5.28)$$

where $f_i(u)$ and $a_i(q) \geq 0$ are arbitrary functions, was designed in [56]. The numerical fluxes are chosen following the same "alternating fluxes" principle similar to the second-order convection-diffusion equation (4.1), see (4.6). A cell entropy inequality and the L^2-stability of the LDG scheme for the nonlinear equation (5.28) can be proved [56], which do not depend on the smoothness of the solution of (5.28), the order of accuracy of the scheme, or the triangulation.

5.2.2 Fifth-Order Convection-Dispersion Equations

An LDG scheme for solving the following fifth-order convection-dispersion equation

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} + \sum_{i=1}^{d} g_i(u_{x_i x_i})_{x_i x_i x_i} = 0, \tag{5.29}$$

where $f_i(u)$ and $g_i(q)$ are arbitrary functions, was designed in [56]. The numerical fluxes are chosen following the same upwinding and "alternating fluxes" principle similar to the third-order KdV-type equations (5.1), see (5.6). A cell entropy inequality and the L^2-stability of the LDG scheme for the nonlinear equation (5.29) can be proved [56], which again do not depend on the smoothness of the solution of (5.29), the order of accuracy of the scheme, or the triangulation.

Stable LDG schemes for similar equations with sixth or higher derivatives can also be designed along similar lines.

5.2.3 The $K(m, n)$ Equations

LDG methods for solving the $K(m, n)$ equations

$$u_t + (u^m)_x + (u^n)_{xxx} = 0, \tag{5.30}$$

where m and n are positive integers, have been designed in [27]. These $K(m, n)$ equations were introduced by Rosenau and Hyman in [40] to study the so-called *compactons*, namely the compactly supported solitary waves solutions. For the special case of $m = n$ being an odd positive integer, LDG schemes which are stable in the L^{m+1}-norm can be designed (see [27]). For other cases, we can also design LDG schemes based on a linearized stability analysis, which perform well in numerical simulation for the fully nonlinear equation (5.30).

5.2.4 The KdV-Burgers-Type (KdVB) Equations

LDG methods for solving the KdV-Burgers-type (KdVB) equations

$$u_t + f(u)_x - (a(u)u_x)_x + (r'(u)g(r(u)_x)_x)_x = 0, \tag{5.31}$$

where $f(u)$, $a(u) \geq 0$, $r(u)$ and $g(q)$ are arbitrary functions, have been designed in [49]. The design of numerical fluxes follows the same lines as that for the

convection-diffusion equation (4.2) and for the KdV-type equation (5.3). A cell entropy inequality and the L^2-stability of the LDG scheme for the nonlinear equation (5.31) can be proved [49], which again do not depend on the smoothness of the solution of (5.31) and the order of accuracy of the scheme. The LDG scheme is used in [49] to study different regimes when one of the dissipation and the dispersion mechanisms dominates, and when they have comparable influence on the solution. An advantage of the LDG scheme designed in [49] is that it is stable regardless of which mechanism (convection, diffusion, dispersion) actually dominates.

5.2.5 The Fifth-Order KdV-Type Equations

LDG methods for solving the fifth-order KdV-type equations

$$u_t + f(u)_x + (r'(u)g(r(u)_x)_x)_x + (s'(u)h(s(u)_{xx})_{xx})_x = 0, \tag{5.32}$$

where $f(u)$, $r(u)$, $g(q)$, $s(u)$ and $h(p)$ are arbitrary functions, have been designed in [49]. The design of numerical fluxes follows the same lines as that for the KdV-type equation (5.3). A cell entropy inequality and the L^2-stability of the LDG scheme for the nonlinear equation (5.32) can be proved [49], which again do not depend on the smoothness of the solution of (5.32) and the order of accuracy of the scheme. The LDG scheme is used in [49] to simulate the solutions of the Kawahara equation, the generalized Kawahara equation, Ito's fifth-order KdV equation, and a fifth-order KdV-type equations with high nonlinearities, which are all special cases of the equations represented by (5.32).

5.2.6 The Fully Nonlinear $K(n, n, n)$ Equations

LDG methods for solving the fifth-order fully nonlinear $K(n, n, n)$ equations

$$u_t + (u^n)_x + (u^n)_{xxx} + (u^n)_{xxxxx} = 0, \tag{5.33}$$

where n is a positive integer, have been designed in [49]. The design of numerical fluxes follows the same lines as that for the $K(m, n)$ equations (5.30). For odd n, stability in the L^{n+1}-norm of the resulting LDG scheme can be proved for the nonlinear equation (5.33) [49]. This scheme is used to simulate compacton propagation in [49].

5.2.7 The Nonlinear Schrödinger (NLS) Equation

In [50], LDG methods are designed for the generalized nonlinear Schrödinger (NLS) equation

$$i\, u_t + u_{xx} + i\, (g(|u|^2)u)_x + f(|u|^2)u = 0, \tag{5.34}$$

the two-dimensional version

$$i\, u_t + \Delta u + f(|u|^2)u = 0, \tag{5.35}$$

and the coupled nonlinear Schrödinger equation

$$\begin{cases} i\,u_t + i\,\alpha u_x + u_{xx} + \beta\,u + \kappa v + f(|u|^2, |v|^2)u = 0 \\ i\,v_t - i\,\alpha v_x + v_{xx} - \beta\,u + \kappa v + g(|u|^2, |v|^2)v = 0, \end{cases} \tag{5.36}$$

where $f(q)$ and $g(q)$ are arbitrary functions and α, β and κ are constants. With suitable choices of the numerical fluxes, the resulting LDG schemes are proved to satisfy a cell entropy inequality and L^2-stability [50]. The LDG scheme is used in [50] to simulate the soliton propagation and interaction, and the appearance of singularities. The easiness of *h-p* adaptivity of the LDG scheme and rigorous stability for the fully nonlinear case make it an ideal choice for the simulation of Schrödinger equations, for which the solutions often have quite localized structures.

5.2.8 The Kadomtsev-Petviashvili (KP) Equations

The two-dimensional Kadomtsev-Petviashvili (KP) equations

$$(u_t + 6uu_x + u_{xxx})_x + 3\sigma^2 u_{yy} = 0, \tag{5.37}$$

where $\sigma^2 = \pm 1$, are generalizations of the one-dimensional KdV equations and are important models for water waves. Because of the x-derivative for the u_t term, the equation (5.37) is well posed only in a function space with a global constraint, hence it is very difficult to design an efficient LDG scheme which relies on local operations. In [51], an LDG scheme for (5.37) is designed by carefully choosing locally supported bases which satisfy the global constraint needed by the solution of (5.37). The LDG scheme satisfies a cell entropy inequality and is L^2-stable for the fully nonlinear equation (5.37). Numerical simulations are performed in [51] for both the KP-I equations ($\sigma^2 = -1$ in (5.37)) and the KP-II equations ($\sigma^2 = 1$ in (5.37)). Line solitons and lump-type pulse solutions have been simulated.

5.2.9 The Zakharov-Kuznetsov (ZK) Equation

The two-dimensional Zakharov-Kuznetsov (ZK) equation

$$u_t + (3u^2)_x + u_{xxx} + u_{xyy} = 0 \tag{5.38}$$

is another generalization of the one-dimensional KdV equations. An LDG scheme is designed for (5.38) in [51] which is proved to satisfy a cell entropy inequality and to be L^2-stable. An L^2-error estimate is given in [53]. Various nonlinear waves have been simulated by this scheme in [51].

5.2.10 The Kuramoto-Sivashinsky-type Equations

In [52], an LDG method is developed to solve the Kuramoto-Sivashinsky-type equations

$$u_t + f(u)_x - (a(u)u_x)_x + (r'(u)g(r(u)_x)_x)_x + (s(u_x)u_{xx})_{xx} = 0, \tag{5.39}$$

where $f(u)$, $a(u)$, $r(u)$, $g(q)$ and $s(p) \geq 0$ are arbitrary functions. The Kuramoto-Sivashinsky equation

$$u_t + uu_x + \alpha u_{xx} + \beta u_{xxxx} = 0, \tag{5.40}$$

where α and $\beta \geq 0$ are constants, which is a special case of (5.39), is a canonical evolution equation which has attracted considerable attention over the last decades. When the coefficients α and β are both positive, its linear terms describe a balance between long-wave instability and short-wave stability, with the nonlinear term providing a mechanism for energy transfer between wave modes. The LDG method developed in [52] can be proved to satisfy a cell entropy inequality and is therefore L^2-stable, for the general nonlinear equation (5.39). The LDG scheme is used in [52] to simulate chaotic solutions of (5.40).

5.2.11 The Ito-Type Coupled KdV Equations

Also in [52], an LDG method is developed to solve the Ito-type coupled KdV equations

$$\begin{aligned} u_t + \alpha uu_x + \beta vv_x + \gamma u_{xxx} &= 0, \\ v_t + \beta(uv)_x &= 0, \end{aligned} \tag{5.41}$$

where α, β and γ are constants. An L^2-stability is proved for the LDG method. Simulation for the solution of (5.41) in which the result for u behaves like dispersive wave solution and the result for v behaves like shock wave solution is performed in [52] using the LDG scheme.

5.2.12 The Camassa-Holm (CH) Equation

An LDG method for solving the Camassa-Holm (CH) equation

$$u_t - u_{xxt} + 2\kappa u_x + 3uu_x = 2u_x u_{xx} + uu_{xxx}, \tag{5.42}$$

where κ is a constant, is designed in [54]. Because of the u_{xxt} term, the design of an LDG method is non-standard. By a careful choice of the numerical fluxes, the authors obtain an LDG scheme which can be proved to satisfy a cell entropy inequality and to be L^2-stable [54]. A sub-optimal $O(h^k)$ error estimate is also obtained in [54].

5.2.13 The Cahn-Hilliard Equation

LDG methods have been designed for solving the Cahn-Hilliard equation

$$u_t = \nabla \cdot \Big(b(u) \nabla \big(-\gamma \Delta u + \Psi'(u) \big) \Big), \tag{5.43}$$

and the Cahn-Hilliard system

$$\begin{cases} \boldsymbol{u}_t &= \nabla \cdot (\boldsymbol{B}(\boldsymbol{u}) \nabla \boldsymbol{\omega}), \\ \boldsymbol{\omega} &= -\gamma \Delta \boldsymbol{u} + D\Psi(\boldsymbol{u}), \end{cases} \tag{5.44}$$

in [47], where $\{D\Psi(\boldsymbol{u})\}_l = \frac{\partial \Psi(\boldsymbol{u})}{\partial u_l}$ and γ is a positive constant. Here $b(u)$ is the nonnegative diffusion mobility and $\Psi(u)$ is the homogeneous free energy density for the scalar case (5.43). For the system case (5.44), $\boldsymbol{B}(\boldsymbol{u})$ is the symmetric positive semi-definite mobility matrix and $\Psi(\boldsymbol{u})$ is the homogeneous free energy density. The proof of the energy stability for the LDG scheme is given for the general nonlinear solutions. Many simulation results are given in [47].

In [48], a class of LDG methods are designed for the more general Allen-Cahn/Cahn-Hilliard (AC/CH) system in $\Omega \in \mathbb{R}^d$ $(d \leq 3)$

$$\begin{cases} u_t &= \nabla \cdot [b(u,v) \nabla (\Psi_u(u,v) - \gamma \Delta u)], \\ \rho v_t &= -b(u,v)[\Psi_v(u,v) - \gamma \Delta v]. \end{cases} \tag{5.45}$$

Energy stability of the LDG schemes is again proved. Simulation results are provided.

Bibliography

[1] D. Arnold, F. Brezzi, B. Cockburn and L. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* **39** (2002), 1749–1779.

[2] H. Atkins and C.-W. Shu, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations. *AIAA Journal* **36** (1998), 775–782.

[3] F. Bassi and S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics* **131** (1997), 267–279.

[4] C.E. Baumann and J.T. Oden, A discontinuous *hp* finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering* **175** (1999), 311–341.

[5] R. Biswas, K.D. Devine and J. Flaherty, Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics* **14** (1994), 255–283.

[6] Y. Cheng and C.-W. Shu, A discontinuous Galerkin finite element method for time-dependent partial differential equations with higher order derivatives. *Mathematics of Computation* **77** (2008), 699–730.

[7] P. Ciarlet, *The Finite Element Method for Elliptic Problems*. North Holland, 1975.

[8] B. Cockburn, Discontinuous Galerkin methods for convection-dominated problems. In: *High-Order Methods for Computational Physics*, T.J. Barth and H. Deconinck, editors, Lecture Notes in Computational Science and Engineering, volume 9, Springer, 1999, 69–224.

[9] B. Cockburn, B. Dong and J. Guzmán, Optimal convergence of the original DG method for the transport-reaction equation on special meshes. *SIAM Journal on Numerical Analysis* **46** (2008), 1250–1265.

[10] B. Cockburn, S. Hou and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation* **54** (1990), 545–581.

[11] B. Cockburn, G. Karniadakis and C.-W. Shu, The development of discontinuous Galerkin methods. In: *Discontinuous Galerkin Methods: Theory, Computation and Applications*, B. Cockburn, G. Karniadakis and C.-W. Shu, editors, Lecture Notes in Computational Science and Engineering, volume 11, Springer, 2000, Part I: Overview, 3–50.

[12] B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *Journal of Computational Physics* **84** (1989), 90–113.

[13] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation* **52** (1989), 411–435.

[14] B. Cockburn and C.-W. Shu, The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis* (M^2AN) **25** (1991), 337–361.

[15] B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics* **141** (1998), 199–224.

[16] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis* **35** (1998), 2440–2463.

[17] B. Cockburn and C.-W. Shu, Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing* **16** (2001), 173–261.

[18] B. Cockburn and C.-W. Shu, Foreword for the special issue on discontinuous Galerkin method. *Journal of Scientific Computing* **22–23** (2005), 1–3.

[19] C. Dawson, Foreword for the special issue on discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering* **195** (2006), 3183.

[20] S. Gottlieb and C.-W. Shu, Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation* **67** (1998), 73–85.

[21] S. Gottlieb, C.-W. Shu and E. Tadmor, Strong stability preserving high order time discretization methods. *SIAM Reviews* **43** (2001), 89–112.

[22] A. Harten, High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* **49** (1983), 357–393.

[23] G.-S. Jiang and C.-W. Shu, On cell entropy inequality for discontinuous Galerkin methods. *Mathematics of Computation* **62** (1994), 531–538.

[24] C. Johnson and J. Pitkäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation* **46** (1986), 1–26.

[25] P. Lesaint and P.A. Raviart, On a finite element method for solving the neutron transport equation. In: *Mathematical aspects of finite elements in partial differential equations*, C. de Boor, ed., Academic Press, 1974, 89–145.

[26] R.J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, Basel, 1990.

[27] D. Levy, C.-W. Shu and J. Yan, Local discontinuous Galerkin methods for nonlinear dispersive equations. *Journal of Computational Physics* **196** (2004), 751–772.

[28] P.-L. Lions and P.E. Souganidis, Convergence of MUSCL and filtered schemes for scalar conservation law and Hamilton-Jacobi equations. *Numerische Mathematik* **69** (1995), 441–470.

[29] J.T. Oden, I. Babuvska and C.E. Baumann, A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics* **146** (1998), 491–519.

[30] S. Osher, Convergence of generalized MUSCL schemes. *SIAM Journal on Numerical Analysis* **22** (1985), 947–961.

[31] S. Osher and S. Chakravarthy, High resolution schemes and the entropy condition. *SIAM Journal on Numerical Analysis* **21** (1984), 955–984.

[32] S. Osher and E. Tadmor, On the convergence of the difference approximations to scalar conservation laws. *Mathematics of Computation* **50** (1988), 19–51.

[33] T. Peterson, A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis* **28** (1991), 133–140.

[34] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one dimensional case. *Journal of Computational Physics* **193** (2003), 115–135.

[35] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing* **26** (2005), 907–929.

[36] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case. *Computers & Fluids* **34** (2005), 642–663.

[37] W.H. Reed and T.R. Hill, Triangular mesh methods for the neutron transport equation. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[38] J.-F. Remacle, J. Flaherty and M. Shephard, An adaptive discontinuous Galerkin technique with an orthogonal basis applied to Rayleigh-Taylor flow instabilities. *SIAM Review* **45** (2003), 53–72.

[39] G.R. Richter, An optimal-order error estimate for the discontinuous Galerkin method. *Mathematics of Computation* **50** (1988), 75–88.

[40] P. Rosenau and J.M. Hyman, Compactons: solitons with finite wavelength. *Physical Review Letters* **70** (1993), 564–567.

[41] C.-W. Shu, TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation* **49** (1987), 105–121.

[42] C.-W. Shu, Total-Variation-Diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing* **9** (1988), 1073–1084.

[43] C.-W. Shu, A survey of strong stability preserving high order time discretizations. In: *Collected Lectures on the Preservation of Stability under Discretization*, D. Estep and S. Tavener, editors, SIAM, 2002, 51–65.

[44] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* **77** (1988), 439–471.

[45] B. van Leer and S. Nomura, Discontinuous Galerkin for diffusion. 17th AIAA Computational Fluid Dynamics Conference (June 6–9, 2005), AIAA paper 2005–5108.

[46] Y. Xia, Y. Xu and C.-W. Shu, Efficient time discretization for local discontinuous Galerkin methods. *Discrete and Continuous Dynamical Systems – Series B* **8** (2007), 677–693.

[47] Y. Xia, Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for the Cahn-Hilliard type equations. *Journal of Computational Physics* **227** (2007), 472–491.

[48] Y. Xia, Y. Xu and C.-W. Shu, Application of the local discontinuous Galerkin method for the Allen-Cahn/Cahn-Hilliard system. *Communications in Computational Physics* **5** (2009), 821–835.

[49] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for three classes of nonlinear wave equations. *Journal of Computational Mathematics* **22** (2004), 250–274.

[50] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for nonlinear Schrödinger equations. *Journal of Computational Physics* **205** (2005), 72–97.

[51] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for two classes of two dimensional nonlinear wave equations. *Physica D* **208** (2005), 21–58.

[52] Y. Xu and C.-W. Shu, Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations. *Computer Methods in Applied Mechanics and Engineering* **195** (2006), 3430–3447.

[53] Y. Xu and C.-W. Shu, Error estimates of the semi-discrete local discontinuous Galerkin method for nonlinear convection-diffusion and KdV equations. *Computer Methods in Applied Mechanics and Engineering* **196** (2007), 3805–3822.

[54] Y. Xu and C.-W. Shu, A local discontinuous Galerkin method for the Camassa-Holm equation. *SIAM Journal on Numerical Analysis* **46** (2008), 1998–2021.

[55] J. Yan and C.-W. Shu, A local discontinuous Galerkin method for KdV type equations. *SIAM Journal on Numerical Analysis* **40** (2002), 769–791.

[56] J. Yan and C.-W. Shu, Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *Journal of Scientific Computing* **17** (2002), 27–47.

[57] M. Zhang and C.-W. Shu, An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences* (M^3AS) **13** (2003), 395–413.

[58] Q. Zhang and C.-W. Shu, Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin methods for scalar conservation laws. *SIAM Journal on Numerical Analysis* **42** (2004), 641–666.

[59] Q. Zhang and C.-W. Shu, Error estimates to smooth solutions of Runge-Kutta discontinuous Galerkin method for symmetrizable systems of conservation laws. *SIAM Journal on Numerical Analysis* **44** (2006), 1703–1720.

[60] J. Zhu, J.-X. Qiu, C.-W. Shu and M. Dumbser, Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. *Journal of Computational Physics* **227** (2008), 4330–4353.

Chapter 2

An Introduction to the Discontinuous Galerkin Method for Convection-Dominated Problems

Bernardo Cockburn

School of Mathematics, University of Minnesota,
Minneapolis, Minnesota 55455, USA
E-mail: cockburn@math.umn.edu

ABSTRACT

In these notes, we study the Runge Kutta Discontinuous Galerkin method for numericaly solving nonlinear hyperbolic systems and its extension for convection-dominated problems, the so-called Local Discontinuous Galerkin method. Examples of problems to which these methods can be applied are the Euler equations of gas dynamics, the shallow water equations, the equations of magneto-hydrodynamics, the compressible Navier-Stokes equations with high Reynolds numbers, and the equations of the hydrodynamic model for semiconductor device simulation. The main features that make the methods under consideration attractive are their formal high-order accuracy, their nonlinear stability, their high parallelizability, their ability to handle complicated geometries, and their ability to capture the discontinuities or strong gradients of the exact solution without producing spurious oscillations. The purpose of these notes is to provide a short introduction to the devising and analysis of these discontinuous Galerkin methods.

Acknowledgements. The author is grateful to Alfio Quarteroni for the invitation to give a series of lectures at the CIME, June 23–28, 1997, the material of which is contained in these notes. He also thanks F. Bassi and F. Rebay, and I. Lomtev and G.E. Karniadakis for kindly providing pictures from their papers [2] and [3], and [46] and [65], respectively.

Contents

1 Preface

There are several numerical methods using a DG formulation to discretize the equations in time, space, or both. In this monograph, we consider numerical methods that use DG discretizations *in space* and combine it with an *explicit* Runge-Kutta time-marching algorithm. We thus consider the so-called Runge-Kutta discontinuous Galerkin (RKDG) introduced and developed by Cockburn and Shu [17,15,14,13,19] for *nonlinear* hyperbolic systems and the so-called local discontinuous Galerkin (LDG) for *nonlinear* convection-diffusion systems. The LDG methods are an extension of the RKDG methods to convection-diffusion problems proposed first by Bassi and Rebay [3] in the context of the compressible Navier-Stokes and recently extended to general convection-diffusion problems by Cockburn and Shu [18].

Several properties are responsible for the increasing popularity of the above mentioned methods. The use of a DG discretization *in space* gives the methods the high-order accuracy, the flexibility in handling complicated geometries, and the easy to treat boundary conditions typical of the finite element methods. Moreover, the use of *discontinuous* elements produces a block-diagonal *mass* matrix whose blocks can be easily inverted by hand. This why after discretizing in time with a high-order accurate, *explicit* Runge-Kutta method, the resulting algorithm is highly parallelizable. Finally, these methods incorporate in a very natural way the techniques of 'slope limiting' developed by van Leer [62,63] that effectively damp out the spurious oscillations that tend to be produced around the discontinuities or strong gradients of the approximate solution.

In these notes, we sudy these DG methods by following their historical development. Thus, we first study the RKDG method and then the LDG method. To study the RKDG method, we start by considering their definition for the scalar equation in one-space dimension. Then, we consider the scalar equation in several space dimensions and finally, we consider the case of multidimensional systems. The last chapter is devoted to the LDG methods.

To study the RKDG method, we take the point of view that they are formally high-order accurate 'perturbations' of the so-called 'monotone' schemes which are very stable and formally first-order accurate. Indeed, the RKDG methods were devised by trying to see if formally high-order accurate methods could be obtained that retained the remarkable stability of the monotone schemes. Of course, this approach is not new: It has been the basic idea in the devising of the so-called 'high-resolution' schemes for finite-difference and finite-volume methods for nonlinear conservation laws. Thus, the RKDG method incorporates this very successful idea into the framework of DG methods which have all the advantages of finite element methods.

2 A historical overview

2.1 The original Discontinuous Galerkin method

The original discontinuous Galerkin (DG) finite element method was introduced by Reed and Hill [54] for solving the neutron transport equation

$$\sigma\, u + div(\overline{a}\, u) = f,$$

where σ is a real number and $\overline{a}$ a constant vector. Because of the linear nature of the equation, the approximate solution given by the method of Reed and Hill can be computed element by element when the elements are suitably ordered according to the characteristic direction.

LeSaint and Raviart [41] made the first analysis of this method and proved a rate of convergence of $(\Delta x)^k$ for general triangulations and of $(\Delta x)^{k+1}$ for Cartesian grids. Later, Johnson and Pitkaränta [37] proved a rate of convergence of $(\Delta x)^{k+1/2}$ for general triangulations and Peterson [53] confirmed this rate to be optimal. Richter [55] obtained the optimal rate of convergence of $(\Delta x)^{k+1}$ for some structured two-dimensional non-Cartesian grids.

2.2 Nonlinear hyperbolic systems: The RKDG method

The success of this method for linear equations, prompted several authors to try to extend the method to nonlinear hyperbolic conservation laws

$$u_t + \sum_{i=1}^{d} (f_i(u))_{x_i} = 0,$$

equipped with suitable initial or initial–boundary conditions. However, the introduction of the nonlinearity prevents the element-by-element computation of the solution. The scheme defines a nonlinear system of equations that must be solved all at once and this renders it computationally very inefficient for hyperbolic problems.

• **The one-dimensional scalar conservation law.**

To avoid this difficulty, Chavent and Salzano [8] contructed an explicit version of the DG method in the one-dimensional scalar conservation law. To do that, they discretized in space by using the DG method with piecewise linear elements and then discretized in time by using the simple Euler forward method. Although the resulting scheme is explicit, the classical von Neumann analysis shows that it is unconditionally unstable when the ratio $\frac{\Delta t}{\Delta x}$ is held constant; it is stable if $\frac{\Delta t}{\Delta x}$ is of order $\sqrt{\Delta x}$, which is a very restrictive condition for hyperbolic problems.

To improve the stability of the scheme, Chavent and Cockburn [7] modified the scheme by introducing a suitably defined 'slope limiter' following the ideas introduced by vanLeer in [62]. They thus obtained a scheme that

was proven to be total variation diminishing in the means (TVDM) and total variation bounded (TVB) under a fixed CFL number, $f' \frac{\Delta t}{\Delta x}$, that can be chosen to be less than or equal to $1/2$. Convergence of a subsequence is thus guaranteed, and the numerical results given in [7] indicate convergence to the correct entropy solutions. On the other hand, the scheme is only first order accurate in time and the 'slope limiter' has to balance the spurious oscillations in smooth regions caused by linear instability, hence adversely affecting the quality of the approximation in these regions.

These difficulties were overcome by Cockburn and Shu in [17], where the first Runge Kutta Discontinuous Galerkin (RKDG) method was introduced. This method was contructed by (i) retaining the piecewise linear DG method for the space discretization, (ii) using a special explicit TVD second order Runge-Kutta type discretization introduced by Shu and Osher in a finite difference framework [57], [58], and (iii) modifying the 'slope limiter' to maintain the formal accuracy of the scheme at extrema. The resulting explicit scheme was then proven linearly stable for CFL numbers less than $1/3$, formally uniformly second order accurate in space and time including at extrema, and TVBM. Numerical results in [17] indicate good convergence behavior: Second order in smooth regions including at extrema, sharp shock transitions (usually in one or two elements) without oscillations, and convergence to entropy solutions even for non convex fluxes.

In [15], Cockburn and Shu extended this approach to construct (formally) high-order accurate RKDG methods for the scalar conservation law. To device RKDG methods of order $k+1$, they used (i) the DG method with polynomials of degree k for the space discretization, (ii) a TVD $(k+1)$-th order accurate explicit time discretization, and (iii) a generalized 'slope limiter.' The generalized 'slope limiter' was carefully devised with the purpose of enforcing the TVDM property without destroying the accuracy of the scheme. The numerical results in [15], for $k = 1, 2$, indicate $(k+1)$-th order order in smooth regions away from discontinuities as well as sharp shock transitions with no oscillations; convergence to the entropy solutions was observed in all the tests. These RKDG schemes were extended to one-dimensional systems in [14].

• **The multidimensional case.**

The extension of the RKDG method to the multidimensional case was done in [13] for the scalar conservation law. In the multidimensional case, the complicated geometry the spatial domain might have in practical applications can be easily handled by the DG space discretization. The TVD time discretizations remain the same, of course. Only the construction of the generalized 'slope limiter' represents a serious challenge. This is so, not only because of the more complicated form of the elements but also because of inherent accuracy barries imposed by the stability properties.

Indeed, since the main purpose of the 'slope limiter' is to enforce the nonlinear stability of the scheme, it is essential to realize that in the multidimensional case, the constraints imposed by the stability of a scheme on

its accuracy are even greater than in the one dimensional case. Although in the one dimensional case it is possible to devise high-order accurate schemes with the TVD property, this is not true in several space dimensions since Goodman and LeVeque [28] proved that any TVD scheme is at most first order accurate. Thus, any generalized 'slope limiter' that enforces the TVD property, or the TVDM property for that matter, would unavoidably reduce the accuracy of the scheme to first-order accuracy. This is why in [13], Cockburn, Hou and Shu devised a generalized 'slope limiter' that enforced a **local** maximum principles only since they are not incompatible with high-order accuracy. No other class of schemes has a proven maximum principle for genearal nonlinearities $\mathbf{f}$, and arbitrary triangulations.

The extension of the RKDG methods to general multidimensional systems was started by Cockburn and Shu in [16] and has been recently completed in [19]. Bey and Oden [5] and more recently Bassi and Rebay [2] have studied applications of the method to the Euler equations of gas dynamics.

• **The main advantages of the RKDG method.**

The resulting RKDG schemes have several important advantages. First, like finite element methods such as the SUPG-method of Hughes and Brook [29,34,30–33] (which has been analyzed by Johnson *et al* in [38–40]), the RKDG methods are better suited than finite difference methods to handle complicated geometries. Moreover, the particular finite elements of the DG space discretization allow an extremely simple treatment of the boundary conditions; no special numerical treatment of them is required in order to achieve uniform high order accuracy, as is the case for the finite difference schemes.

Second, the method can easily handle adaptivity strategies since the refining or unrefining of the grid can be done without taking into account the continuity restrictions typical of conforming finite element methods. Also, the degree of the approximating polynomial can be easily changed from one element to the other. Adaptivity is of particular importance in hyperbolic problems given the complexity of the structure of the discontinuities. In the one dimensional case the Riemann problem can be solved in closed form and discontinuity curves in the (x,t) plane are simple straight lines passing through the origin. However, in two dimensions their solutions display a very rich structure; see the works of Wagner [64], Lindquist [43], [42], Zhang and Zheng [68], and Zhang and Cheng [67]. Thus, methods which allow triangulations that can be easily adapted to resolve this structure, have an important advantage.

Third, the method is highly parallelizable. Since the elements are discontinuous, the mass matrix is block diagonal and since the order of the blocks is equal to the number of degrees of freedom inside the corresponding elements, the blocks can be inverted by hand once and for all. Thus, at each Runge-Kutta inner step, to update the degrees of freedom inside a given element, only the degrees of freedom of the elements sharing a face are involved; communication between processors is thus kept to a minimum. Extensive studies

of adaptivity and parallelizability issues of the RKDG method were started by Biswas, Devine, and Flaherty [6] and then continued by deCougny *et al.* [20], Devine *et al.* [22,21] and by Özturan *et al.* [52].

2.3 Convection-diffusion systems: The LDG method

The first extensions of the RKDG method to nonlinear, convection-diffusion systems of the form

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{F}(\mathbf{u}, D\,\mathbf{u}) = 0, \text{ in } (0,T) \times \Omega,$$

were proposed by Chen *et al.* [10], [9] in the framework of hydrodynamic models for semiconductor device simulation. In these extensions, approximations of second and third-order derivatives of the discontinuous approximate solution were obtained by using simple projections into suitable finite elements spaces. This projection requires the inversion of global mass matrices, which in [10] and [9] are 'lumped' in order to maintain the high parallelizability of the method. Since in [10] and [9] polynomials of degree one are used, the 'mass lumping' is justified; however, if polynomials of higher degree were used, the 'mass lumping' needed to enforce the full parallelizability of the method could cause a degradation of the formal order of accuracy.

Fortunately, this is not an issue with the methods proposed by Bassi and Rebay [3] (see also Bassi *et al* [2]) for the compressible Navier-Stokes equations. In these methods, the original idea of the RKDG method is applied to *both* u *and* $D\,u$ which are now considered as *independent* unknowns. Like the RKDG methods, the resulting methods are highly parallelizable methods of high-order accuracy which are very efficient for time-dependent, convection-dominated flows. The LDG methods considered by Cockburn and Shu [18] are a generalization of these methods.

The basic idea to construct the LDG methods is to *suitably rewrite* the original system as a larger, degenerate, first-order system and then discretize it by the RKDG method. By a careful choice of this rewriting, nonlinear stability can be achieved even without slope limiters, just as the RKDG method in the purely hyperbolic case; see Jiang and Shu [36].

The LDG methods [18] are very different from the so-called Discontinuous Galerkin (DG) method for parabolic problems introduced by Jamet [35] and studied by Eriksson, Johnson, and Thomée [27], Eriksson and Johnson [23–26], and more recently by Makridakis and Babuška [50]. In the DG method, the approximate solution is discontinuous only in time, not in space; in fact, the space discretization is the standard Galerkin discretization with *continuous* finite elements. This is in strong contrast with the space discretizations of the LDG methods which use *discontinuous* finite elements. To emphasize this difference, those methods are called **Local** Discontinuous Galerkin methods. The large amount of degrees of freedom and the restrictive conditions of the size of the time step for explicit time-discretizations, render the LDG methods inefficient for diffusion-dominated problems; in this situation, the use

of methods with continuous-in-space approximate solutions is recommended. However, as for the successful RKDG methods for purely hyperbolic problems, the extremely local domain of dependency of the LDG methods allows a very efficient parallelization that by far compensates for the extra amount of degrees of freedom in the case of convection-dominated flows.

Karniadakis *et al.* have implemented and tested these methods for the compressible Navier Stokes equations in two and three space dimensions with impressive results; see [44], [45], [46], [47], and [65].

2.4 The content of these notes

In these notes, we study the RKDG and LDG methods. Our exposition will be based on the papers by Cockburn and Shu [17], [15], [14], [13], and [19] in which the RKDG method was developed and on the paper by Cockburn and Shu [18] which is devoted to the LDG methods. Numerical results from the papers by Bassi and Rebay [2], on the Euler equations of gas dynamics, and [3], on the compressible Navier-Stokes equations, are also included.

The emphasis in these notes is on *how the above mentioned schemes were devised.* As a consequence, the sections that follow reflect that development. Thus, section 2, in which the RKDG schemes for the one-dimensional scalar conservation law are constructed, constitutes the core of the notes because it contains all the important ideas for the devicing of the RKDG methods; section 3 contains the extension to multidimensional systems; and section 4, the extension to convection-diffusion problems.

We would like to emphasize that the guiding principle in the devicing of the RKDG methods for scalar conservation laws is to consider them as *perturbations of the so-called monotone schemes.* As it is well-known, monotone schemes for scalar conservation laws are stable and converge to the entropy solution but are only first-order accurate. Following a widespread approach in the field of numerical schemes for nonlinear conservation laws, the RKDG are constructed in such a way that they are high-order accurate schemes that 'become' a monotone scheme when a piecewise-constant approximation is used. Thus, to obtain high-order accurate RKDG schemes, we 'perturb' the piecewise-constant approximation and allow it to be piecewise a polynomial of arbitrary degree. Then, the conditions under which the stability properties of the monotone schemes are still valid are sought and enforced by means of the generalized 'slope limiter.' The fact that it is possible to do so without destroying the accuracy of the RKDG method is the crucial point that makes this method both robust and accurate.

The issues of parallelization and adaptivity developed by Biswas, Devine, and Flaherty [6], deCougny *et al.* [20], Devine *et al.* [22,21] and by Özturan *et al.* [52] are certainly very important. Another issue of importance is how to render the method computationaly more efficient, like the quadrature rule-free versions of the RKDG method recently studied by Atkins and Shu [1].

However, these topics fall beyond the scope of these notes whose main intention is to provide a simple introduction to the topic of discontinuous Galerkin methods for convection-dominated problems.

3 The scalar conservation law in one space dimension

3.1 Introduction

In this section, we introduce and study the RKDG method for the following simple model problem:

$$u_t + f(u)_x = 0, \qquad \text{in } (0,1) \times (0,T), \tag{3.1}$$
$$u(x,0) = u_0(x), \qquad \forall\, x \in (0,1), \tag{3.2}$$

and periodic boundary conditions. This section has material drawn from [17] and [15].

3.2 The discontinuous Galerkin-space discretization

3.3 The weak formulation

To discretize in space, we proceed as follows. For each partition of the interval $(0,1)$, $\{ x_{j+1/2} \}_{j=0}^{N}$, we set $I_j = (x_{j-1/2}, x_{j+1/2})$, $\Delta_j = x_{j+1/2} - x_{j-1/2}$ for $j = 1, \ldots, N$, and denote the quantity $\max_{1 \le j \le N} \Delta_j$ by Δx .

We seek an approximation u_h to u such that for each time $t \in [0,T]$, $u_h(t)$ belongs to the finite dimensional space

$$V_h = V_h^k \equiv \{ v \in L^1(0,1) : v|_{I_j} \in P^k(I_j),\ j = 1, \ldots, N \}, \tag{3.3}$$

where $P^k(I)$ denotes the space of polynomials in I of degree at most k. In order to determine the approximate solution u_h, we use a weak formulation that we obtain as follows. First, we multiply the equations (3.1) and (3.2) by arbitrary, smooth functions v and integrate over I_j, and get, after a simple formal integration by parts,

$$\int_{I_j} \partial_t u(x,t)\, v(x)\, dx - \int_{I_j} f(u(x,t))\, \partial_x v(x)\, dx \tag{3.4}$$
$$+ f(u(x_{j+1/2},t))\, v(x_{j+1/2}^-) - f(u(x_{j-1/2},t))\, v(x_{j-1/2}^+) = 0,$$

$$\int_{I_j} u(x,0)\, v(x)\, dx = \int_{I_j} u_0(x)\, v(x)\, dx. \tag{3.5}$$

Next, we replace the smooth functions v by test functions v_h belonging to the finite element space V_h, and the exact solution u by the approximate solution

u_h. Since the function u_h is discontinuous at the points $x_{j+1/2}$, we must also replace the nonlinear 'flux' $f(u(x_{j+1/2},t))$ by a *numerical* 'flux' that depends on the two values of u_h at the point $(x_{j+1/2},t)$, that is, by the function

$$h(u)_{j+1/2}(t) = h(u(x^-_{j+1/2},t), u(x^+_{j+1/2},t)), \tag{3.6}$$

that will be suitably chosen later. Note that *we always use the same numerical flux regardless of the form of the finite element space.* Thus, the approximate solution given by the DG-space discretization is defined as the solution of the following weak formulation:

$$\forall\ j = 1,\ldots,N, \qquad \forall\ v_h \in P^k(I_j):$$

$$\begin{aligned}&\int_{I_j} \partial_t u_h(x,t)\, v_h(x)\, dx - \int_{I_j} f(u_h(x,t))\, \partial_x v_h(x)\, dx \\ &+h(u_h)_{j+1/2}(t)\, v_h(x^-_{j+1/2}) - h(u_h)_{j-1/2}(t)\, v_h(x^+_{j-1/2}) = 0,\end{aligned} \tag{3.7}$$

$$\int_{I_j} u_h(x,0)\, v_h(x)\, dx = \int_{I_j} u_0(x)\, v_h(x)\, dx. \tag{3.8}$$

3.4 Incorporating the monotone numerical fluxes

To complete the definition of the approximate solution u_h, it only remains to choose the numerical flux h. To do that, we invoke our main point of view, namely, that *we want to construct schemes that are perturbations of the so-called monotone schemes* because monotone schemes, although only first-order accurate, are very stable and converge to the entropy solution. More precisely, we want that in the case $k = 0$, that is, when the approximate solution u_h is a piecewise-constant function, our DG-space discretization gives rise to a monotone scheme.

Since in this case, for $x \in I_j$ we can write

$$u_h(x,t) = u^0_j,$$

we can rewrite our weak formulation (3.7), (3.8) as follows:

$$\forall\ j = 1, \ldots, N:$$

$$\partial_t u_j^0(t) + \{h(u_j^0(t), u_{j+1}^0(t)) - h(u_{j-1}^0(t), u_j^0(t))\}/\Delta_j = 0,$$
$$u_j^0(0) = \frac{1}{\Delta_j} \int_{I_j} u_0(x)\, dx,$$

and it is well-known that this defines a monotone scheme if $h(a, b)$ is a Lipschitz, consistent, monotone flux, that is, if it is,

(i) locally Lipschitz and consistent with the flux $f(u)$, i.e., $h(u, u) = f(u)$,
(ii) a nondecreasing function of its first argument, and
(iii) a nonincreasing function of its second argument.

The best-known examples of numerical fluxes satisfying the above properties are the following:

(i) The Godunov flux:

$$h^G(a, b) = \begin{cases} \min_{a \le u \le b} f(u)\ , & \text{if } a \le b, \\ \max_{a \ge u \ge b} f(u)\ , & \text{if } a > b; \end{cases}$$

(ii) The Engquist-Osher flux:

$$h^{EO}(a, b) = \int_0^b \min(f'(s), 0)\ ds + \int_0^a \max(f'(s), 0)\ ds + f(0);$$

(iii) The Lax-Friedrichs flux:

$$h^{LF}(a, b) = \frac{1}{2}\ [f(a) + f(b) - C\,(b - a)],$$
$$C = \max_{\inf u^0(x) \le s \le \sup u^0(x)} |f'(s)|;$$

(iv) The local Lax–Friedrichs flux:

$$h^{LLF}(a, b) = \frac{1}{2}\ [f(a) + f(b) - C(b - a)],$$
$$C = \max_{\min(a,b) \le s \le \max(a,b)} |f'(s)|;$$

(v) The Roe flux with 'entropy fix':

$$h^R(a, b) = \begin{cases} f(a), & \text{if } f'(u) \ge 0 \quad \text{for} \quad u \in [\min(a, b),\ \max(a, b)], \\ f(b), & \text{if } f'(u) \le 0 \quad \text{for} \quad u \in [\min(a, b), \max(a, b)], \\ h^{LLF}(a, b), & \text{otherwise.} \end{cases}$$

For the flux h, we can use the Godunov flux h^G since it is well-known that this is the numerical flux that produces the smallest amount of artificial viscosity. The local Lax-Friedrichs flux produces more artificial viscosity than the Godunov flux, but their performances are remarkably similar. Of course, if f is too complicated, we can always use the Lax-Friedrichs flux. However, numerical experience suggests that as the degree k of the approximate solution increases, the choice of the numerical flux does not have a significant impact on the quality of the approximations.

3.5 Diagonalizing the mass matrix

If we choose the Legendre polynomials P_ℓ as local basis functions, we can exploit their L^2-orthogonality, namely,

$$\int_{-1}^{1} P_\ell(s)\, P_{\ell'}(s)\, ds = \left(\frac{2}{2\ell+1}\right) \delta_{\ell\,\ell'},$$

and obtain a *diagonal* mass matrix. Indeed, if for $x \in I_j$, we express our approximate solution u_h as follows:

$$u_h(x,t) = \sum_{\ell=0}^{k} u_j^\ell\, \varphi_\ell(x),$$

where

$$\varphi_\ell(x) = P_\ell(2\,(x - x_j)/\Delta_j),$$

the weak formulation (3.7), (3.8) takes the following simple form:

$$\begin{aligned}
&\forall\ j = 1, \ldots, N \text{ and } \ell = 0, \ldots, k:\\
&\left(\frac{1}{2\ell+1}\right)\partial_t\, u_j^\ell(t) - \frac{1}{\Delta_j}\int_{I_j} f(u_h(x,t))\, \partial_x \varphi_\ell(x)\, dx\\
&+\frac{1}{\Delta_j}\Big\{\, h(u_h(x_{j+1/2}))(t) - (-1)^\ell\, h(u_h(x_{j-1/2}))(t)\,\Big\} = 0,\\
&u_j^\ell(0) = \frac{2\ell+1}{\Delta_j}\int_{I_j} u_0(x)\, \varphi_\ell(x)\, dx,
\end{aligned}$$

where we have use the following properties of the Legendre polynomials:

$$P_\ell(1) = 1, \qquad P_\ell(-1) = (-1)^\ell.$$

This shows that after discretizing in space the problem (3.1), (3.2) by the DG method, we obtain a system of ODEs for the degrees of freedom that we can rewrite as follows:

$$\frac{d}{dt} u_h = L_h(u_h), \qquad \text{in } (0,T), \tag{3.9}$$

$$u_h(t=0) = u_{0h}. \tag{3.10}$$

The element $L_h(u_h)$ of V_h is, of course, the approximation to $-f(u)_x$ provided by the DG-space discretization.

Note that if we choose a different local basis, the local mass matrix could be a full matrix but it will always be a matrix of order $(k+1)$. By inverting it by means of a symbolic manipulator, we can always write the equations for the degrees of freedom of u_h as an ODE system of the form above.

3.6 Convergence analysis of the linear case

In the linear case $f(u) = cu$, the $L^\infty(0,T;L^2(0,1))$-accuracy of the method (3.7), (3.8) can be established by using the $L^\infty(0,T;L^2(0,1))$-stability of the method and the approximation properties of the finite element space V_h.

Note that in this case, all the fluxes displayed in the examples above coincide and are equal to

$$h(a,b) = c\frac{a+b}{2} - \frac{|c|}{2}(b-a). \tag{3.11}$$

The following results are thus for this numerical flux.

We state the L^2-stability result in terms of the jumps of u_h across $x_{j+1/2}$ which we denote by

$$[\,u_h\,]_{j+1/2} \equiv u_h(x^+_{j+1/2}) - u_h(x^-_{j+1/2}).$$

Proposition 3.1 *(L^2-stability) We have,*

$$\tfrac{1}{2}\|\,u_h(T)\,\|^2_{L^2(0,1)} + \Theta_T(\,u_h\,) \le \tfrac{1}{2}\|\,u_0\,\|^2_{L^2(0,1)},$$

where

$$\Theta_T(\,u_h\,) = \tfrac{|c|}{2}\int_0^T \sum_{1\le j\le N} [\,u_h(t)\,]^2_{j+1/2}\,dt.$$

Note how the jumps of u_h are controled by the L^2-norm of the initial condition. This control reflects the subtle built-in dissipation mechanism of the DG-methods and is what allows the DG-methods to be more accurate than the standard Galerkin methods. Indeed, the standard Galerkin method has an order of accuracy equal to k whereas the DG-methods have an order of accuray equal to $k + 1/2$ for the same smoothness of the initial condition.

Theorem 3.1 *Suppose that the initial condition u_0 belongs to $H^{k+1}(0,1)$. Let e be the approximation error $u - u_h$. Then we have,*

$$\| e(T) \|_{L^2(0,1)} \leq C \, | u_0 |_{H^{k+1}(0,1)} (\Delta x)^{k+1/2},$$

where C depends solely on k, $|c|$, and T.

It is also possible to prove the following result if we assume that the initial condition is more regular. Indeed, we have the following result.

Theorem 3.2 *Suppose that the initial condition u_0 belongs to $H^{k+2}(0,1)$. Let e be the approximation error $u - u_h$. Then we have,*

$$\| e(T) \|_{L^2(0,1)} \leq C \, | u_0 |_{H^{k+2}(0,1)} (\Delta x)^{k+1},$$

where C depends solely on k, $|c|$, and T.

The Theorem 3.1 is a simplified version of a more general result proven in 1986 by Johnson and Pitkäranta [37] and the Theorem 3.2 is a simplified version of a more general result proven in 1974 by LeSaint and Raviart [41]. To provide a simple introduction to the techniques used in these more general results, we give *new* proofs of these theorems in an appendix to this section.

The above theorems show that the DG-space discretization results in a $(k+1)$th-order accurate scheme, at least in the linear case. This gives a strong indication that the same order of accuracy should hold in the nonlinear case when the exact solution is smooth enough, of course.

Now that we know that the DG-space discretization produces a high-order accurate scheme for smooth exact solutions, we consider the question of how does it behave when the flux is a nonlinear function.

3.7 Convergence analysis in the nonlinear case

To study the convergence properties of the DG-method, we first study the convergence properties of the solution w of the following problem:

$$w_t + f(w)_x = (\nu(w)\, w_x)_x, \qquad \text{in } (0,1) \times (0,T), \tag{3.12}$$

$$w(x,0) = u_0(x), \qquad \forall\, x \in (0,1), \tag{3.13}$$

and periodic boundary conditions. We then mimic the procedure to study the convergence of the DG-method for the piecewise-constant case. The general DG-method will be considered later after having introduced the Runge-Kutta time-discretization.

The continuous case as a model. In order to compare u and w, it is *enough* to have (i) an entropy inequality and (ii) uniform boundedness of $\| w_x \|_{L^1(0,1)}$. Next, we show how to obtain these properties in a formal way.

We start with the entropy inequality. To obtain such an inequality, the basic idea is to multiply the equation (3.12) by $U'(w-c)$, where $U(\cdot)$ denotes the absolute value function and c denotes an arbitrary real number. Since

$$
\begin{aligned}
&U'(w-c)\, w_t = U(w-c)_t, \\
&U'(w-c)\, f(w)_x = \big(U'(w-c)\,(f(w)-f(c))\big) \equiv F(w,c)_x, \\
&U'(w-c)\,(\nu(w)\, w_x)_x = \left(\int_c^w U'(\rho-c)\,\nu(\rho)\, d\rho \right)_{xx} - U''(w-c)\,\nu(w)\,(w_x)^2 \\
&\qquad\qquad \equiv \Phi(w,c)_{xx} - U''(w-c)\,\nu(w)\,(w_x)^2,
\end{aligned}
$$

we obtain

$$
U(w-c)_t + F(w,c)_x - \Phi(w,c)_{xx} \le 0, \qquad \text{in } (0,1)\times(0,T),
$$

which is nothing but the entropy inequality we wanted.

To obtain the uniform boundedness of $\| w_x \|_{L^1(0,1)}$, the idea is to multiply the equation (3.12) by $-(U'(w_x))_x$ and integrate on x from 0 to 1. Since

$$
\begin{aligned}
&\int_0^1 -(U'(w_x))_x\, w_t = \int_0^1 U'(w_x)\,(w_x)_t = \frac{d}{dt}\| w_x \|_{L^1(0,1)}, \\
&\int_0^1 -(U'(w_x))_x\, f(w)_x = -\int_0^1 U''(w_x)\, w_{xx}\, f'(w)\, w_x = 0, \\
&\int_0^1 -(U'(w_x))_x\,(\nu(w)\, w_x)_x = -\int_0^1 U''(w_x)\, w_{xx}\,(\nu'(w)\,(w_x)^2 + \nu(w)\, w_{xx}) \\
&\qquad\qquad = -\int_0^1 U''(w_x)\,\nu(w)\,(w_{xx})^2 \le 0,
\end{aligned}
$$

we immediately get that

$$
\frac{d}{dt}\| w_x \|_{L^1(0,1)} \le 0,
$$

and so,

$$
\| w_x \|_{L^1(0,1)} \le \| (u_0)_x \|_{L^1(0,1)}, \qquad \forall\, t \in (0,T).
$$

When the function u_0 has discontinuities, the same result holds with the total variation of u_0 ,$| u_0 |_{TV(0,1)}$, replacing the quantity $\| (u_0)_x \|_{L^1(0,1)}$; these two quantities coincide when $u_0 \in W^{1,1}(0,1)$.

With the two above ingredients, the following error estimate, obtained in 1976 by Kuznetsov, can be proved:

Theorem 3.3 *We have*

$$\| u(T) - w(T) \|_{L^1(0,1)} \leq | u_0 |_{TV(0,1)} \sqrt{8\,T\,\nu},$$

where $\nu = \sup_{s \in [\inf u_0, \sup u_0]} \nu(s)$.

The piecewise-constant case. Let consider the simple case of the DG-method that uses a piecewise-constant approximate solution:

$$\forall\ j = 1, \ldots, N:$$

$$\partial_t u_j + \{h(u_j, u_{j+1}) - h(u_{j-1}, u_j)\}/\Delta_j = 0,$$
$$u_j(0) = \frac{1}{\Delta_j} \int_{I_j} u_0(x)\, dx,$$

where we have dropped the superindex '0.' We pick the numerical flux h to be the Engquist-Osher flux.

According to the model provided by the continuous case, we must obtain (i) an entropy inequality and (ii) the uniform boundedness of the total variation of u_h.

To obtain the entropy inequality, we multiply our equation by $U'(u_j - c)$:

$$\partial_t U(u_j - c) + U'(u_j - c)\{h(u_j, u_{j+1}) - h(u_{j-1}, u_j)\}/\Delta_j = 0.$$

The second term in the above equation needs to be carefully treated. First, we rewrite the Engquist-Osher flux in the following form:

$$h^{EO}(a, b) = f^+(a) + f^-(b),$$

and, accordingly, rewrite the second term of the equality above as follows:

$$ST_j = U'(u_j - c)\{f^+(u_j) - f^+(u_{j-1})\} + U'(u_j - c)\{f^-(u_{j+1}) - f^-(u_j)\}.$$

Using the simple identity

$$U'(a - c)(g(a) - g(b)) = G(a, c) - G(b, c) + \int_a^b (g(b) - g(\rho))\, U''(\rho - x)\, d\rho$$

where $G(a,c) = \int_c^a U'(\rho - c)\, g(\rho)\, d\rho$, we get

$$\begin{aligned} ST_j &= F^+(u_j, c) - F^+(u_{j-1}, c) + \int_{u_j}^{u_{j-1}} \left(f^+(u_{j-1}) - f^+(\rho)\right) U''(\rho - x)\, d\rho \\ &\quad + F^-(u_{j+1}, c) - F^-(u_j, c) - \int_{u_j}^{u_{j+1}} \left(f^-(u_{j+1}) - f^-(\rho)\right) U''(\rho - x)\, d\rho \\ &= F(u_j, u_{j+1}; c) - F(u_{j-1}, u_j; c) + \Theta_{diss,j} \end{aligned}$$

where

$$\begin{aligned} F(a, b; c) &= F^+(a, c) + F^-(b, c), \\ \Theta_{diss,j} &= + \int_{u_j}^{u_{j-1}} \left(f^+(u_{j-1}) - f^+(\rho)\right) U''(\rho - x)\, d\rho \\ &\quad - \int_{u_j}^{u_{j+1}} \left(f^-(u_{j+1}) - f^-(\rho)\right) U''(\rho - x)\, d\rho. \end{aligned}$$

We thus get

$$\partial_t U(u_j - c) + \left\{F(u_j, u_{j+1}; c) - F(u_{j-1}, u_j; c)\right\}/\Delta_j + \Theta_{diss,j}/\Delta_j = 0.$$

Since, f^+ and $-f^-$ are nondecreasing functions, we easily see that

$$\Theta_{diss,j} \geq 0,$$

and we obtain our entropy inequality:

$$\partial_t U(u_j - c) + \left\{F(u_j, u_{j+1}; c) - F(u_{j-1}, u_j; c)\right\}/\Delta_j \leq 0.$$

Next, we obtain the uniform boundedness on the total variation. To do that, we follow our model and multiply our equation by a discrete version of $-(U'(w_x))_x$, namely,

$$v_j^0 = -\frac{1}{\Delta_j}\left\{U'\left(\frac{u_{j+1} - u_j}{\Delta_{j+1/2}}\right) - U'\left(\frac{u_j - u_{j-1}}{\Delta_{j-1/2}}\right)\right\},$$

where $\Delta_{j+1/2} = (\Delta_j + \Delta_{j+1})/2$, multiply it by Δ_j and sum over j from 1 to N. We easily obtain

$$\frac{d}{dt}|\, u_h\, |_{TV(0,1)} + \sum_{1 \leq j \leq N} v_j^0 \left\{h(u_j, u_{j+1}) - h(u_{j-1}, u_j)\right\} = 0,$$

where

$$| u_h |_{TV(0,1)} \equiv \sum_{1 \leq j \leq N} | u_{j+1} - u_j |.$$

According to our continuous model, the second term in the above equality should be positive. Let us see that this is indeed the case:

$$\begin{aligned} v_j^0 \left\{ h(u_j, u_{j+1}) - h(u_{j-1}, u_j) \right\} &= v_j^0 \left\{ f^+(u_j) - f^+(u_{j-1}) \right\} \\ &\quad + v_j^0 \left\{ f^-(u_{j+1}) - f^-(u_j) \right\} \geq 0, \end{aligned}$$

by the definition of v_j^0, f^+, and f^-. This implies that

$$| u_h(t) |_{TV(0,1)} \leq | u_h(0) |_{TV(0,1)} \leq | u_0 |_{TV(0,1)}.$$

With the two above ingredients, the following error estimate, obtained in 1976 by Kuznetsov, can be proved:

Theorem 3.4 *We have*

$$\| u(T) - u_h(T) \|_{L^1(0,1)} \leq \| u_0 - u_h(0) \|_{L^1(0,1)} + C | u_0 |_{TV(0,1)} \sqrt{T \, \Delta x}.$$

3.8 The TVD-Runge-Kutta time discretization

To discretize our ODE system in time, we use the TVD Runge Kutta time discretization introduced in [60]; see also [57] and [58].

3.9 The discretization

Thus, if $\{t^n\}_{n=0}^N$ is a partition of $[0, T]$ and $\Delta t^n = t^{n+1} - t^n, n = 0, ..., N-1$, our time-marching algorithm reads as follows:

- Set $u_h^0 = u_{0h}$;
- For $n = 0, ..., N-1$ compute u_h^{n+1} from u_h^n as follows:
  1. set $u_h^{(0)} = u_h^n$;
  2. for $i = 1, ..., k+1$ compute the intermediate functions:

$$u_h^{(i)} = \left\{ \sum_{l=0}^{i-1} \alpha_{il} u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}) \right\};$$

  3. set $u_h^{n+1} = u_h^{(k+1)}$.

Note that this method is very easy to code since *only a single subroutine defining $L_h(u_h)$ is needed.* Some Runge-Kutta time discretization parameters are displayed on the table below.

Table 1

Parameters of some practical Runge-Kutta time discretizations			
order	α_{il}	β_{il}	$\max\{\beta_{il}/\alpha_{il}\}$
2	1 $\frac{1}{2}\ \frac{1}{2}$	1 $0\ \frac{1}{2}$	1
3	1 $\frac{3}{4}\ \frac{1}{4}$ $\frac{1}{3}\ 0\ \frac{2}{3}$	1 $0\ \frac{1}{4}$ $0\ 0\ \frac{2}{3}$	1

3.10 The stability property

Note that all the values of the parameters α_{il} displayed in the table below are nonnegative; this is not an accident. Indeed, this is a condition on the parameters α_{il} that ensures the stability property

$$|\, u_h^{n+1} \,| \leq |\, u_h^n \,|,$$

provided that the 'local' stability property

$$|\, w \,| \leq |\, v \,|, \tag{3.14}$$

where w is obtained from v by the following 'Euler forward' step,

$$w = v + \delta\ L_h(v), \tag{3.15}$$

holds for values of $|\, \delta \,|$ smaller than a given number δ_0.

For example, the second-order Runke-Kutta method displayed in the table above can be rewritten as follows:

$$\begin{aligned} u_h^{(1)} &= u_h^n + \Delta t\, L_h(u_h^n), \\ w_h &= u_h^{(1)} + \Delta t\, L_h(u_h^{(1)}), \\ u_h^{n+1} &= \frac{1}{2}(u_h^n + w_h). \end{aligned}$$

Now, assuming that the stability property (3.14), (3.15) is satisfied for

$$\delta_0 = |\, \Delta t \, \max\{\beta_{il}/\alpha_{il}\} \,| = \Delta t,$$

we have

$$|\, u_h^{(1)} \,| \le |\, u_h^n \,|, \qquad |\, w_h \,| \le |\, u_h^{(1)} \,|,$$

and so,

$$|\, u_h^{n+1} \,| \le \frac{1}{2}(|\, u_h^n \,| + |\, w_h \,|) \le |\, u_h^n \,|.$$

Note that we can obtain this result because the coefficients α_{il} are positive! Runge-Kutta methods of this type of order up to order 5 can be found in [58].

The above example shows how to prove the following more general result.

Theorem 3.5 *Assume that the stability property for the single 'Euler forward' step (3.14), (3.15) is satisfied for*

$$\delta_0 = \max_{0 \le n \le N} |\, \Delta t^n \, \max\{\beta_{il}/\alpha_{il}\} \,|.$$

Assume also that all the coeficients α_{il} are nonnegative and satisfy the following condition:

$$\sum_{l=0}^{i-1} \alpha_{il} = 1, \qquad i = 1, \ldots, k+1.$$

Then

$$|\, u_h^n \,| \le |\, u_h^0 \,|, \qquad \forall\, n \ge 0.$$

This stability property of the TVD-Runge-Kutta methods is crucial since it allows us to obtain the stability of the method from the stability of a single 'Euler forward' step.

Proof of Theorem 3.5. We start by rewriting our time discretization as follows:

- Set $u_h^0 = u_{0h}$;
- For $n = 0, ..., N-1$ compute u_h^{n+1} from u_h^n as follows:
  1. set $u_h^{(0)} = u_h^n$;
  2. for $i = 1, ..., k+1$ compute the intermediate functions:

$$u_h^{(i)} = \sum_{l=0}^{i-1} \alpha_{il} \, w_h^{(il)},$$

where

$$w_h^{(il)} = u_h^{(l)} + \frac{\beta_{il}}{\alpha_{il}} \, \Delta t^n \, L_h(u_h^{(l)});$$

  3. set $u_h^{n+1} = u_h^{(k+1)}$.

We then have

$$
\begin{aligned}
|u_h^{(i)}| &\leq \sum_{l=0}^{i-1} \alpha_{il} |w_h^{(il)}|, \quad \text{since } \alpha_{il} \geq 0, \\
&\leq \sum_{l=0}^{i-1} \alpha_{il} |u_h^{(l)}|, \quad \text{by the stability property (3.14), (3.15),} \\
&\leq \max_{0 \leq l \leq i-1} |u_h^{(l)}|, \quad \text{since } \sum_{l=0}^{i-1} \alpha_{il} = 1.
\end{aligned}
$$

It is clear now that that Theorem 3.5 follows from the above inequality by a simple induction argument. □

3.11 Remarks about the stability in the linear case

For the linear case $f(u) = cu$, Chavent and Cockburn [7] proved that for the case $k = 1$, i.e., for piecewise-linear approximate solutions, the single 'Euler forward' step is *unconditionally* $L^\infty(0,T;L^2(0,1))$-unstable for any fixed ratio $\Delta t/\Delta x$. On the other hand, in [17] it was shown that if a Runge-Kutta method of second order is used, the scheme is $L^\infty(0,T;L^2(0,1))$-stable provided that

$$
c\frac{\Delta t}{\Delta x} \leq \frac{1}{3}.
$$

This means that we cannot deduce the stability of the complete Runge-Kutta method from the stability of the single 'Euler forward' step. As a consequence, we cannot apply Theorem 3.5 and we must consider the complete method at once.

Our numerical experiments show that when polynomial of degree k are **used, a Runge-Kutta of order $(k+1)$ must be used. In this case, the $L^\infty(0,T;L^2(0,1))$-stability condition is the following:**

$$
c\frac{\Delta t}{\Delta x} \leq \frac{1}{2k+1}.
$$

There is no rigorous proof of this fact yet.

At a first glance, this stability condition, also called the Courant-Friedrichs-Levy (CFL) condition, seems to compare unfavorably with that of the well-known finite difference schemes. However, we must remember that in the DG-methods there are $(k+1)$ degrees of freedom in each element of size Δx whereas for finite difference schemes there is a single degree of freedom of

each cell of size Δx. Also, if a finite difference scheme is of order $(k+1)$ its so-called stencil must be of at least $(2k+1)$ points, whereas the DG-scheme has a stencil of $(k+1)$ elements only.

3.12 Convergence analysis in the nonlinear case

Now, we explore what is the impact of the explicit Runge-Kutta time-discretization on the convergence properties of the methods under consideration. We start by considering the piecewise-constant case.

The piecewise-constant case. Let us begin by considering the simplest case, namely,

$$\forall\ j = 1, \ldots, N:$$

$$\begin{aligned} &(u_j^{n+1} - u_j^n)/\Delta t + \{h(u_j^n, u_{j+1}^n) - h(u_{j-1}^n, u_j^n)\}/\Delta_j = 0, \\ &u_j(0) = \frac{1}{\Delta_j}\int_{I_j} u_0(x)\,dx, \end{aligned}$$

where we pick the numerical flux h to be the Engquist-Osher flux.

According to the model provided by the continuous case, we must obtain (i) an entropy inequality and (ii) the uniform boundedness of the total variation of u_h.

To obtain the entropy inequality, we proceed as in the semidiscrete case and obtain the following result; see [12] for details.

Theorem 3.6 *We have*

$$\begin{aligned} \{U(u_j^{n+1} - c) - U(u_j^n - c)\}/\Delta t + \{F(u_j^n, u_{j+1}^n; c) - F(u_{j-1}^n, u_j^n; c)\}/\Delta_j \\ + \Theta_{diss,j}^n/\Delta t = 0, \end{aligned}$$

where

$$\begin{aligned} \Theta_{diss,j}^n = &\int_{u_j^{n+1}}^{u_j^n} (p_j(u_j^n) - p_j(\rho))\,U''(\rho - x)\,d\rho \\ &+\frac{\Delta t}{\Delta_j}\int_{u_j^{n+1}}^{u_{j-1}^n} (f^+(u_{j-1}^n) - f^+(\rho))\,U''(\rho - x)\,d\rho \\ &-\frac{\Delta t}{\Delta_j}\int_{u_j^{n+1}}^{u_{j+1}^n} (f^-(u_{j+1}^n) - f^-(\rho))\,U''(\rho - x)\,d\rho, \end{aligned}$$

and

$$p_j(w) = w - \frac{\Delta t}{\Delta_j}(f^+(w) - f^-(w)).$$

Moreover, if the following CFL condition is satisfied

$$\max_{1\le j\le N} \frac{\Delta t}{\Delta_j} \mid f' \mid \le 1,$$

then $\Theta^n_{diss,j} \ge 0$, *and the following entropy inequality holds:*

$$\{U(u_j^{n+1} - c) - U(u_j^n - c)\}/\Delta t + \{F(u_j^n, u_{j+1}; c) - F(u_{j-1}, u_j; c)\}/\Delta_j \le 0.$$

Note that $\Theta^n_{diss,j} \ge 0$ because f^+, $-f^-$, are nondecreasing and because p_j is also nondecreasing under the above CFL condition.

Next, we obtain the uniform boundedness on the total variation. Proceeding as before, we easily obtain the following result.

Theorem 3.7 *We have*

$$\mid u_h^{n+1} \mid_{TV(0,1)} - \mid u_h^n \mid_{TV(0,1)} + \Theta^n_{TV} = 0,$$

where

$$\begin{aligned}\Theta^n_{TV} = &\sum_{1\le j\le N} \left(U'^n_{j+1/2} - U'^{n+1}_{j+1/2} \right) (p_{j+1/2}(u^n_{j+1}) - p_{j+1/2}(u^n_j)) \\ &+ \sum_{1\le j\le N} \frac{\Delta t}{\Delta_j} \left(U'^n_{j-1/2} - U'^{n+1}_{j+1/2} \right) (f^+(u^n_j) - f^+(u^n_{j-1})) \\ &- \sum_{1\le j\le N} \frac{\Delta t}{\Delta_j} \left(U'^n_{j+1/2} - U'^{n+1}_{j-1/2} \right) (f^-(u^n_{j+1}) - f^-(u^n_j))\end{aligned}$$

where

$$U'^m_{i+1/2} = U'\left(\frac{u^m_{i+1} - u^m_i}{\Delta_{i+1/2}} \right),$$

and

$$p_{j+1/2}(w) = s - \frac{\Delta t}{\Delta_{j+1}} f^+(w) + \frac{\Delta t}{\Delta_j} f^-(w).$$

Moreover, if the following CFL condition is satisfied

$$\max_{1\le j\le N} \frac{\Delta t}{\Delta_j} \mid f' \mid \le 1,$$

then $\Theta^n_{TV} \ge 0$, *and we have*

$$\mid u_h^n \mid_{TV(0,1)} \le \mid u_0 \mid_{TV(0,1)}.$$

With the two above ingredients, the following error estimate, obtained in 1976 by Kuznetsov, can be proved:

Theorem 3.8 *We have*

$$\| u(T) - u_h(T) \|_{L^1(0,1)} \leq \| u_0 - u_h(0) \|_{L^1(0,1)} + C \, | u_0 |_{TV(0,1)} \sqrt{T \, \Delta x}.$$

The general case. The study of the general case is much more difficult than the study of the monotone schemes. In these notes, we restrict ourselves to the study of the stability of the RKDG schemes. Hence, we restrict ourselves to the task of studying under what conditions the total variation of the *local means* is uniformly bounded.

If we denote by $\overline{u}_j$ the mean of u_h on the interval I_j, by setting $v_h = 1$ in the equation (3.7), we obtain,

$$\forall \; j = 1, \ldots, N :$$

$$(\overline{u}_j)_t + \left\{ h(u^-_{j+1/2}, u^+_{j+1/2}) - h(u^-_{j-1/2}, u^+_{j-1/2}) \right\} / \Delta_j = 0,$$

where $u^-_{j+1/2}$ denotes the limit from the left and $u^+_{j+1/2}$ the limit from the right. We pick the numerical flux h to be the Engquist-Osher flux.

This shows that if we set w_h equal to the Euler forward step $u_h + \delta \, L_h(u_h)$, we obtain

$$\forall \; j = 1, \ldots, N :$$

$$(\overline{w}_j - \overline{u}_j \,) / \delta + \left\{ h(u^-_{j+1/2}, u^+_{j+1/2}) - h(u^-_{j-1/2}, u^+_{j-1/2}) \right\} / \Delta_j = 0.$$

Proceeding exactly as in the piecewise-constant case, we obtain the following result for the total variation of the avergages,

$$| \, \overline{u}_h \, |_{TV(0,1)} \equiv \sum_{1 \leq j \leq N} | \, \overline{u}_{j+1} - \overline{u}_j \, |.$$

Theorem 3.9 *We have*

$$| \, \overline{w}_h \, |_{TV(0,1)} - | \, \overline{u}_h \, |_{TV(0,1)} + \Theta_{TVM} = 0,$$

where

$$\Theta_{TVM} = \sum_{1\le j\le N} \left(U'_{j+1/2} - U'_{j+1/2} \right) (p_{j+1/2}(u_h|_{I_{j+1}}) - p_{j+1/2}(u_h|_{I_j})$$
$$+ \sum_{1\le j\le N} \frac{\delta}{\Delta_j} \left(U'_{j-1/2} - U'_{j+1/2} \right) (f^+(u^-_{j+1/2}) - f^+(u^-_{j-1/2}))$$
$$- \sum_{1\le j\le N} \frac{\delta}{\Delta_j} \left(U'_{j+1/2} - U'_{j-1/2} \right) (f^-(u^+_{j+1/2}) - f^-(u^+_{j-1/2}))$$

where

$$U'_{i+1/2} = U'\left(\frac{u_{i+1} - u_i}{\Delta_{i+1/2}} \right),$$

and

$$p_{j+1/2}(u_h|_{I_m}) = \overline{u}_m - \frac{\delta}{\Delta_{j+1}} f^+(u^-_{m+1/2}) + \frac{\delta}{\Delta_j} f^-(u^+_{m-1/2}).$$

From the above result, we see that the total variation of the means of the Euler forward step is nonincreasing if the following three conditions are satisfied:

$$sgn(\,\overline{u}_{j+1} - \overline{u}_j\,) = sgn(\,p_{j+1/2}(u_h|_{I_{j+1}}) - p_{j+1/2}(u_h|_{I_j})\,), \tag{3.16}$$
$$sgn(\,\overline{u}_j - \overline{u}_{j-1}\,) = sgn(\,u^{n,-}_{j+1/2} - u^{n,-}_{j-1/2}\,), \tag{3.17}$$
$$sgn(\,\overline{u}_{j+1} - \overline{u}_j\,) = sgn(\,u^{n,+}_{j+1/2} - u^{n,+}_{j-1/2}\,). \tag{3.18}$$

Note that if the properties (3.16) and (3.17) are satisfied, then the property (3.18) can always be satisfied for a small enough values of $|\,\delta\,|$.

Of course, the numerical method under consideration does not provide an approximate solution automatically satisfying the above conditions. It is thus necessary to *enforce* them by means of a suitably defined generalized slope limiter,' $\Lambda\Pi_h$.

3.13 The generalized slope limiter

High-order accuracy versus the TVDM property: Heuristics The ideal generalized slope limiter $\Lambda\Pi_h$

- Maintains the conservation of *mass* element by element,
- Satifies the properties (3.16), (3.17), and (3.18),
- Does not degrade the accuracy of the method.

The first requirement simply states that the slope limiting must not change the total mass contained in each interval, that is, if $u_h = \Lambda\Pi_h(v_h)$,

$$\overline{u}_j = \overline{v}_j, \qquad j = 1, \ldots, N.$$

This is, of course a very sensible requirement because after all we are dealing with consevation laws. It is also a requirement very easy to satisfy.

The second requirement, states that if $u_h = \Lambda\Pi_h(v_h)$ and $w_h = u_h + \delta\, L_h(u_h)$ then

$$|\,\overline{w}_h\,|_{TV(0,1)} \leq |\,\overline{u}_h\,|_{TV(0,1)},$$

for small enough values of $|\,\delta\,|$.

The third requirement deserves a more delicate discussion. Note that if u_h is a very good approximation of a smooth solution u in a neigborhood of the point x_0, it behaves (asymptotically as Δx goes to zero) as a straight line if $u_x(x_0) \neq 0$. If x_0 is an isolated extrema of u, then it behaves like a parabola provided $u_{xx}(x_0) \neq 0$. Now, if u_h is a straightline, it trivially satisfies conditions (3.16) and (3.17). However, if u_h is a parabola, conditions (3.16) and (3.17) are not always satisfied. This shows that it is impossible to construct the above ideal generalized 'solpe limiter,' or, in other words, that in order to enforce the TVDM property, we must loose high-order accuracy at the local extrema. This is a very well-known phenomenon for TVD finite difference schemes!

Fortunatelly, it is still possible to construct generalized slope limiters that do preserve high-order accuracy even at local extrema. The resulting scheme will then not be TVDM but total variation bounded in the means (TVBM) as we will show.

In what follows we first consider generalized slope limiters that render the RKDG schemes TVDM. Then we suitably modify them in order to obtain TVBM schemes.

Constructing TVDM generalized slope limiters Next, we look for simple, sufficient conditions on the function u_h that imply the conditions (3.16), (3.17), and (3.18). These conditions will be stated in terms of the *minmod* function m defined as follows:

$$m\,(a_1, \ldots, a_\nu) = \begin{cases} s\,\min_{1\leq n\leq \nu} |\,a_n\,|, & \text{if } s = sign(a_1) = \cdots = sign(a_\nu), \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 3.10 *Suppose the the following CFL condition is satisfied:*

$$|\,\delta\,|\,(\frac{|\,f^+\,|_{Lip}}{\Delta_{j+1}} + \frac{|\,f^-\,|_{Lip}}{\Delta_j}) \leq 1/2, \qquad j = 1, \ldots, N. \tag{3.19}$$

Then, conditions (3.16), (3.17), and (3.18) are satisfied if, for all $j = 1, \ldots, N$, *we have that*

$$u^-_{j+1/2} - \overline{u}_j = m\,(u^-_{j+1/2} - \overline{u}_j,\ \overline{u}_j - \overline{u}_{j-1},\ \overline{u}_{j+1} - \overline{u}_j) \tag{3.20}$$

$$\overline{u}_j - u^+_{j-1/2} = m\,(\overline{u}_j - u^+_{j-1/2},\ \overline{u}_j - \overline{u}_{j-1},\ \overline{u}_{j+1} - \overline{u}_j). \tag{3.21}$$

Proof. Let us start by showing that the property (3.17) is satisfied. We have:

$$\begin{aligned} u^-_{j+1/2} - u^-_{j-1/2} &= (u^-_{j+1/2} - \overline{u}_j) + (\overline{u}_j - \overline{u}_{j-1}) + (\overline{u}_{j-1} - u^-_{j-1/2}) \\ &= \Theta\,(\overline{u}_j - \overline{u}_{j-1}), \end{aligned}$$

where

$$\Theta = 1 + \frac{u^-_{j+1/2} - \overline{u}_j}{\overline{u}_j - \overline{u}_{j-1}} - \frac{u^-_{j-1/2} - \overline{u}_{j-1}}{\overline{u}_j - \overline{u}_{j-1}} \in [0, 2],$$

by conditions (3.20) and (3.21). This implies that the property (3.17) is satisfied. Properties (3.18) and (3.16) are proven in a similar way. This completes the proof. □

Examples of TVDM generalized slope limiters

a. The MUSCL limiter. In the case of piecewise linear approximate solutions, that is,

$$v_h|_{I_j} = \overline{v}_j + (x - x_j)\, v_{x,j}, \qquad j = 1, \ldots, N,$$

the following generalized slope limiter does satisfy the conditions (3.20) and (3.21):

$$u_h|_{I_j} = \overline{v}_j + (x - x_j)\, m\,(v_{x,j}, \frac{\overline{v}_{j+1} - \overline{v}_j}{\Delta_j}, \frac{\overline{v}_j - \overline{v}_{j-1}}{\Delta_j}).$$

This is the well-known slope limiter of the MUSCL schemes of vanLeer [62,63].

b. The less restrictive limiter $\Lambda\Pi^1_h$**.** The following less restrictive slope limiter also satisfies the conditions (3.20) and (3.21):

$$u_h|_{I_j} = \overline{v}_j + (x - x_j)\, m\,(v_{x,j}, \frac{\overline{v}_{j+1} - \overline{v}_j}{\Delta_j/2}, \frac{\overline{v}_j - \overline{v}_{j-1}}{\Delta_j/2}).$$

Moreover, it can be rewritten as follows:

$$u^-_{j+1/2} = \overline{v}_j + m\,(\,v^-_{j+1/2} - \overline{v}_j,\ \overline{v}_j - \overline{v}_{j-1},\ \overline{v}_{j+1} - \overline{v}_j) \tag{3.22}$$

$$u^+_{j-1/2} = \overline{v}_j - m\,(\,\overline{v}_j - v^+_{j-1/2},\ \overline{v}_j - \overline{v}_{j-1},\ \overline{v}_{j+1} - \overline{v}_j). \tag{3.23}$$

We denote this limiter by $\Lambda\Pi^1_h$.

Note that we have that

$$\|\,\overline{v}_h - \Lambda\Pi^1_h(v_h)\,\|_{L^1(0,1)} \leq \frac{\Delta x}{2}\,|\,\overline{v}_h\,|_{TV(0,1)}.$$

See Theorem 3.13 below.

c. The limiter $\Lambda\Pi^k_h$. In the case in which the approximate solution is piecewise a polynomial of degree k, that is, when

$$v_h(x,t) = \sum_{\ell=0}^{k} v^\ell_j\,\varphi_\ell(x),$$

where

$$\varphi_\ell(x) = P_\ell(2\,(x - x_j)/\Delta_j),$$

and P_ℓ are the Legendre polynomials, we can define a generalized slope limiter in a very simple way. To do that, we need the define what could be called the P^1-part of v_h:

$$v^1_h(x,t) = \sum_{\ell=0}^{1} v^\ell_j\,\varphi_\ell(x),$$

We define $u_h = \Lambda\Pi_h(v_h)$ as follows:

- For $j = 1, ..., N$ compute $u_h|_{I_j}$ as follows:
  1. Compute $u^-_{j+1/2}$ and $u^+_{j-1/2}$ by using (3.22) and (3.23),
  2. If $u^-_{j+1/2} = v^-_{j+1/2}$ and $u^+_{j-1/2} = v^+_{j-1/2}$ set $u_h|_{I_j} = v_h|_{I_j}$,
  3. If not, take $u_h|_{I_j}$ equal to $\Lambda\Pi^1_h(v^1_h)$.

d. The limiter $\Lambda\Pi^k_{h,\alpha}$. When instead of (3.22) and (3.23), we use

$$u^-_{j+1/2} = \overline{v}_j + m\,(\,v^-_{j+1/2} - \overline{v}_j,\ \overline{v}_j - \overline{v}_{j-1},\ \overline{v}_{j+1} - \overline{v}_j, C\,(\Delta x)^\alpha) \tag{3.24}$$

$$u^+_{j-1/2} = \overline{v}_j - m\,(\,\overline{v}_j - v^+_{j-1/2},\ \overline{v}_j - \overline{v}_{j-1},\ \overline{v}_{j+1} - \overline{v}_j, C\,(\Delta x)^\alpha), \tag{3.25}$$

for some fixed constant C and $\alpha \in (0,1)$, we obtain a generalized slope limiter we denote by $\Lambda\Pi^k_{h,\alpha}$.

This generalized slope limiter is never used in practice, but we consider it here because it is used for theoretical purposes; see Theorem 3.13 below.

The complete RKDG method Now that we have our generalized slope limiters, we can display the complete RKDG method. It is contained in the following algorith:

- Set $u_h^0 = \Lambda\Pi_h \, P_{V_h}(u_0)$;
- For $n = 0, ..., N-1$ compute u_h^{n+1} as follows:
  1. set $u_h^{(0)} = u_h^n$;
  2. for $i = 1, ..., k+1$ compute the intermediate functions:

$$u_h^{(i)} = \Lambda\Pi_h \left\{ \sum_{l=0}^{i-1} \alpha_{il} \, u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}) \right\};$$

  3. set $u_h^{n+1} = u_h^{(k+1)}$.

This algorithm describes the complete RKDG method. Note how the generalized slope limiter has to be applied at each intermediate computation of the Runge-Kutta method. This way of appying the generalized slope limiter in the time-marching algorithm ensures that the scheme is TVDM, as we next show.

The TVDM property of the RKDG method To do that, we start by noting that if we set

$$u_h = \Lambda\Pi_h(v_h), \qquad w_h = u_h + \delta \, L_h(u_h),$$

then we have that

$$|\overline{u}_h|_{TV(0,1)} \leq |\overline{v}_h|_{TV(0,1)}, \tag{3.26}$$

$$|\overline{w}_h|_{TV(0,1)} \leq |\overline{u}_h|_{TV(0,1)}, \qquad \forall |\delta| \leq \delta_0, \tag{3.27}$$

where

$$\delta_0^{-1} = 2 \max_j \left(\frac{|f^+|_{Lip}}{\Delta_{j+1}} + \frac{|f^-|_{Lip}}{\Delta_j} \right) \qquad j = 1, \ldots, N,$$

by Theorem 3.10. By using the above two properties of the generalized slope limiter,' it is possible to show that the RKDG method is TVDM.

Theorem 3.11 *Assume that the generalized slope limiter $\Lambda\Pi_h$ satisfies the properties (3.26) and (3.27). Assume also that all the coeficients α_{il} are non-negative and satisfy the following condition:*

$$\sum_{l=0}^{i-1} \alpha_{il} = 1, \qquad i = 1, \ldots, k+1.$$

Then

$$|\,\overline{u}_h^n\,|_{TV(0,1)} \leq |\,u_0\,|_{TV(0,1)}, \qquad \forall\, n \geq 0.$$

Proof of Theorem 3.11. The proof of this result is very similar to the proof of Theorem 3.5. Thus, we start by rewriting our time discretization as follows:

- Set $u_h^0 = u_{0h}$;
- For $n = 0, ..., N-1$ compute u_h^{n+1} from u_h^n as follows:
  1. set $u_h^{(0)} = u_h^n$;
  2. for $i = 1, ..., k+1$ compute the intermediate functions:

$$u_h^{(i)} = \Lambda \Pi_h \left\{ \sum_{l=0}^{i-1} \alpha_{il}\, w_h^{(il)} \right\},$$

where

$$w_h^{(il)} = u_h^{(l)} + \frac{\beta_{il}}{\alpha_{il}}\, \Delta t^n\, L_h(u_h^{(l)});$$

  3. set $u_h^{n+1} = u_h^{(k+1)}$.

Then have,

$$\begin{aligned}
|\,\overline{u}_h^{(i)}\,|_{TV(0,1)} &\leq |\sum_{l=0}^{i-1} \alpha_{il}\, \overline{w}_h^{(il)}\,|_{TV(0,1)}, \quad \text{by (3.26)},\\
&\leq \sum_{l=0}^{i-1} \alpha_{il}\, |\,\overline{w}_h^{(il)}\,|_{TV(0,1)}, \quad \text{since } \alpha_{il} \geq 0,\\
&\leq |\sum_{l=0}^{i-1} \alpha_{il}\, \overline{u}_h^{(l)}\,|_{TV(0,1)}, \quad \text{by (3.27)},\\
&\leq \max_{0 \leq l \leq i-1} |\,\overline{u}_h^{(l)}\,|_{TV(0,1)}, \quad \text{since } \sum_{l=0}^{i-1} \alpha_{il} = 1.
\end{aligned}$$

It is clear now that that the inequality

$$|\,\overline{u}_h^n\,|_{TV(0,1)} \leq |\,\overline{u}_h^0\,|_{TV(0,1)}, \qquad \forall\, n \geq 0.$$

follows from the above inequality by a simple induction argument. To obtain the result of the theorem, it is enough to note that we have

$$|\,\overline{u}_h^0\,|_{TV(0,1)} \leq |\,u_0\,|_{TV(0,1)},$$

by the definition of the initial condition u_h^0. This completes the proof. □

TVBM generalized slope limiters As was pointed out before, it is possible to modify the generalized slope limiters displayed in the examples above in such a way that the degradation of the accuracy at local extrema is avoided. To achieve this, we follow Shu [59] and modify the definition of the generalized slope limiters by simply replacing the *minmod* function m by the TVB corrected *minmod* function $\bar{m}$ defined as follows:

$$\bar{m}(a_1, ..., a_m) = \begin{cases} a_1, & if \ |a_1| \leq M(\Delta x)^2, \\ m(a_1, ..., a_m), & otherwise, \end{cases} \tag{3.28}$$

where M is a given constant. We call the generalized slope limiters thus constructed, TVBM slope limiters.

The constant M is, of course, an upper bound of the absolute value of the second-order derivative of the solution at local extrema. In the case of the nonlinear conservation laws under consideration, it is easy to see that, if the initial data is piecewise C^2, we can take

$$M = \sup\{ \mid (u_0)_{xx}(y) \mid, y : (u_0)_x(y) = 0\}.$$

See [15] for other choices of M.

Thus, if the constant M is is taken as above, there is no degeneracy of accuracy at the extrema and the resulting RKDG scheme retains its optimal accuracy. Moreover, we have the following stability result.

Theorem 3.12 *Assume that the generalized slope limiter $\Lambda\Pi_h$ is a TVBM slope limiter. Assume also that all the coeficients α_{il} are nonnegative and satisfy the following condition:*

$$\sum_{l=0}^{i-1} \alpha_{il} = 1, \qquad i = 1, \ldots, k+1.$$

Then

$$| \overline{u}_h^n |_{TV(0,1)} \leq | \overline{u}_0 |_{TV(0,1)} + C\, M, \qquad \forall\, n \geq 0,$$

where C depends on k only.

Convergence in the nonlinear case By using the stability above stability results, we can use the Ascoli-Arzelá theorem to prove the following convergence result.

Theorem 3.13 *Assume that the generalized slope limiter $\Lambda\Pi_h$ is a TVDM or a TVBM slope limiter. Assume also that all the coeficients α_{il} are nonnegative and satisfy the following condition:*

$$\sum_{l=0}^{i-1} \alpha_{il} = 1, \qquad i = 1, \ldots, k+1.$$

Then there is a subsequence $\{\overline{u}_{h'}\}_{h'>0}$ *of the sequence* $\{\overline{u}_h\}_{h>0}$ *generate by the RKDG scheme that converges in* $L^\infty(0,T;L^1(0,1))$ *to a weak solution of the problem (3.1), (3.2).*

Moreover, if the TVBM version of the slope limiter $\Lambda\Pi^k_{h,\alpha}$ *is used, the weak solution is the entropy solution and the whole sequence converges.*

Finally, if the generalized slope limiter $\Lambda\Pi_h$ *is such that*

$$\| \overline{v}_h - \Lambda\Pi_h(v_h) \|_{L^1(0,1)} \leq C\, \Delta x \,|\, \overline{v}_h \,|_{TV(0,1)},$$

then the above results hold not only to the sequence of the means $\{\overline{u}_h\}_{h>0}$ *but to the sequence of the functions* $\{u_h\}_{h>0}$.

3.14 Computational results

In this subsection, we display the performance of the RKDG schemes in a simple but typical test problem. We use piecewise linear ($k = 1$) and piecewise quadratic ($k = 2$) elements; the $\Lambda\Pi^k_h$ generalized slope limter is used. Our purpose is to show that (i) when the constant M is properly chosen, the RKDG method using polynomials of degree k is is order $k+1$ in the uniform norm away from the discontinuities, that (ii) it is computationally more efficient to use high-degree polynomial approximations, and that (iii) shocks are captured in a few elements without production of spurious oscillations

We solve the Burger's equation with a periodic boundary condition:

$$u_t + (\frac{u^2}{2})_x = 0,$$
$$u(x,0) = u_0(x) = \frac{1}{4} + \frac{1}{2}\ \sin(\pi(2\,x-1)).$$

The exact solution is smooth at $T = .05$ and has a well developed shock at $T = 0.4$. Notice that there is a sonic point. In Tables 1,2, and 3, the history of convergence of the RKDG method using piecewise linear elements is dsplayed and in Tables 4,5, and 6, the history of convergence of the RKDG method using piecewise quadratic elements. It can be seen that when the TVDM generalized slope limiter is used, i.e., when we take $M = 0$, there is degradation of the accuracy of the scheme, whereas when the TVBM generalized slope limiter is used with a properly chosen constant M, i.e., when $M = 20 \geq 2\,\pi^2$, the scheme is uniformly high order in regions of smoothness that include critical and sonic points.

Next, we compare the efficiency of the RKDG schemes for $k = 1$ and $k = 2$ for the case $M = 20$ and $T = 0.05$. We define the inverse of the efficiency of the method as the product of the error times the number of operations. Since the RKDG method that uses quadratic elements has 0.3/0.2 times more time steps, 3/2 times more inner iterations per time step, and 3/2 time more unknowns in space, its number of operations is 27/8 times bigger than the one of the RKDH method using linear elements. Hence, the ratio of the efficiency

of the RKDG method with quadratic elements to that of the RKDG method with linear elements is

$$r = \frac{8}{27} \frac{error(RKDG(k=1)}{error(RKDG(k=2)}.$$

The results are displayed in Table 7. We can see that the efficiency of the RKDG scheme with quadratic polynomials is several times that of the RKDG scheme with linear polynomials even for very small values of Δx. We can also see that the ratio r of efficiencies is proportional to $(\Delta x)^{-1}$, which is expected for smooth solutions. This indicates that it is indeed more efficient to work with RKDG methods using polynomials of higher degree.

That this is also true when the solution displays discontinuities can be seen figures 3.22, and 3.23. In the figure 3.22, it can be seen that the shock is captured in essentially two elements. A zoom of these figures is shown in figure 3.23, where the approximation right in front of the shock is shown. It is clear that the approximation using quadratic elements is superior to the approximation using linear elements.

3.15 Concluding remarks

In this subsection, which is the core of these notes, we have devised the general RKDG method for nonlinear scalar conservation laws with periodic boundary conditions.

We have seen that the RKDG are constructed in three steps. First, the Discontinuous Galerkin method is used to discretize in space the conservation law. Then, an explicit TVB-Runge-Kutta time discretizationis used to discretize the resulting ODE system. Finally, a generalized slope limiter is introduced that enforces nonlinear stability without degrading the accuracy of the scheme.

We have seen that the numerical results show that the RKDG methods using polynomials of degree $k, k = 1, 2$ are uniformly $(k+1)$-th order accurate away from discontinuities and that the use of high degree polynomials render the RKDG method more efficient, even close to discontinuities.

All these results can be extended to the initial boundary value problem, see [15]. In what follows, we extend the RKDG methods to multidimensional systems.

Table 1
P^1, $M = 0$, CFL$= 0.3$, $T = 0.05$.

	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	1286.23	-	3491.79	-
1/20	334.93	1.85	1129.21	1.63
1/40	85.32	1.97	449.29	1.33
1/80	21.64	1.98	137.30	1.71
1/160	5.49	1.98	45.10	1.61
1/320	1.37	2.00	14.79	1.61
1/640	0.34	2.01	4.85	1.60
1/1280	0.08	2.02	1.60	1.61

Table 2
P^1, $M = 20$, CFL$= 0.3$, $T = 0.05$.

	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	1073.58	-	2406.38	-
1/20	277.38	1.95	628.12	1.94
1/40	71.92	1.95	161.65	1.96
1/80	18.77	1.94	42.30	1.93
1/160	4.79	1.97	10.71	1.98
1/320	1.21	1.99	2.82	1.93
1/640	0.30	2.00	0.78	1.86
1/1280	0.08	2.00	0.21	1.90

Table 3
Errors in smooth region $\Omega = \{x : |x - shock| \geq 0.1\}$.
P^1, $M = 20$, CFL= 0.3, $T = 0.4$.

	$L^1(\Omega) - error$		$L^\infty(\Omega) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	1477.16	-	17027.32	-
1/20	155.67	3.25	1088.55	3.97
1/40	38.35	2.02	247.35	2.14
1/80	9.70	1.98	65.30	1.92
1/160	2.44	1.99	17.35	1.91
1/320	0.61	1.99	4.48	1.95
1/640	0.15	2.00	1.14	1.98
1/1280	0.04	2.00	0.29	1.99

Table 4
P^2, $M = 0$, CFL= 0.2, $T = 0.05$.

	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	2066.13	-	16910.05	-
1/20	251.79	3.03	3014.64	2.49
1/40	42.52	2.57	1032.53	1.55
1/80	7.56	2.49	336.62	1.61

Table 5
P^2, $M = 20$, CFL$= 0.2$, $T = 0.05$.

	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	37.31	-	101.44	-
1/20	4.58	3.02	13.50	2.91
1/40	0.55	3.05	1.52	3.15
1/80	0.07	3.08	0.19	3.01

Table 6
Errors in smooth region $\Omega = \{x : |x - shock| \geq 0.1\}$.
P^2, $M = 20$, CFL$= 0.2$, $T = 0.4$.

	$L^1(\Omega) - error$		$L^\infty(\Omega) - error$	
Δx	$10^5 \cdot error$	order	$10^5 \cdot error$	order
1/10	786.36	-	16413.79	-
1/20	5.52	7.16	86.01	7.58
1/40	0.36	3.94	15.49	2.47
1/80	0.06	2.48	0.54	4.84

Table 7
Comparison of the efficiencies of RKDG schemes for $k = 2$ and $k = 1$
$M = 20$, $T = 0.05$.

	L^1-norm		L^∞-norm	
Δx	$eff.ratio$	order	$eff.ratio$	order
1/10	8.52	-	7.03	-
1/20	17.94	-1.07	46.53	-2.73
1/40	38.74	-1.11	106.35	-1.19
1/80	79.45	-1.04	222.63	-1.07

3.16 Appendix: Proof of the L^2-error estimates in the linear case

Proof of the L^2-stability In this subsection, we prove the the stability result of Proposition 3.1. To do that, we first show how to obtain the corresponding stability result for the exact solution and then mimic the argument to obtain Proposition 3.1.

The continuous case as a model. We start by rewriting the equations (3.4) in *compact form.* If in the equations (3.4) we replace $v(x)$ by $v(x,t)$, sum on j from 1 to N, and integrate in time from 0 to T, we obtain

$$\mathbb{B}(u,v) = 0, \qquad \forall\ v:\ v(t) \text{ is smooth} \quad \forall\ t \in (0,T), \tag{3.29}$$

where

$$\mathbb{B}(u,v) = \int_0^T \int_0^1 \{\, \partial_t u(x,t)\, v(x,t) - c\, u(x,t)\, \partial_x\, v(x,t) \,\}\, dx\, dt. \tag{3.30}$$

Taking $v = u$, we easily see that we see that

$$\mathbb{B}(u,u) = \frac{1}{2} \| u(T) \|^2_{L^2(0,1)} - \frac{1}{2} \| u_0 \|^2_{L^2(0,1)},$$

and since

$$\mathbb{B}(u,u) = 0,$$

by (3.29), we immediately obtain the following L^2-stability result:

$$\frac{1}{2} \| u(T) \|^2_{L^2(0,1)} = \frac{1}{2} \| u_0 \|^2_{L^2(0,1)}.$$

This is the argument we have to mimic in order to prove Proposition 3.1.

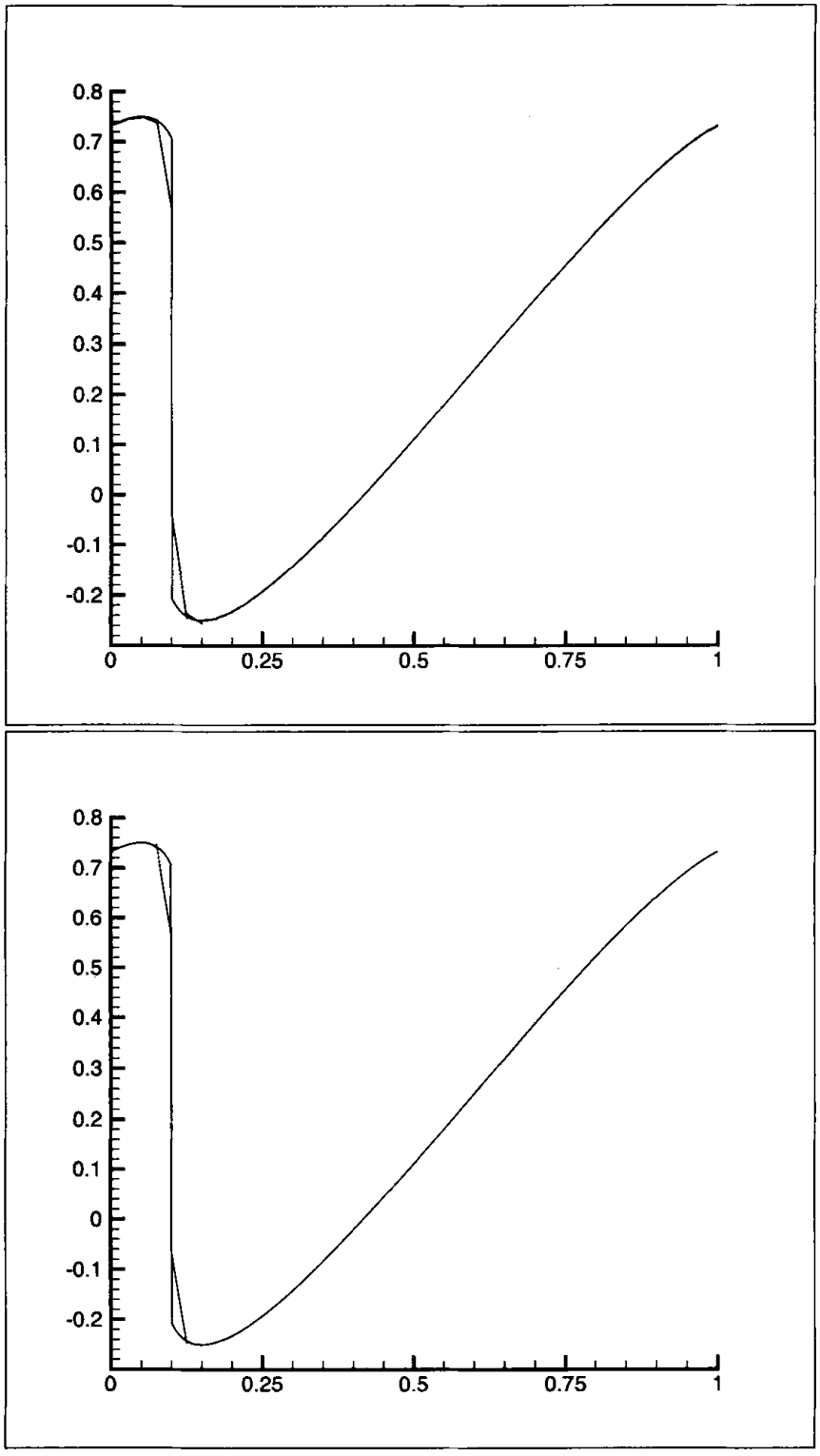


Fig. 3.22: Comparison of the exact and the approximate solution obtained with $M = 20$, $\Delta x = 1/40$ at $T = .4$: Piecewise linear elements (top) and piecewise quadratic elements (bottom)

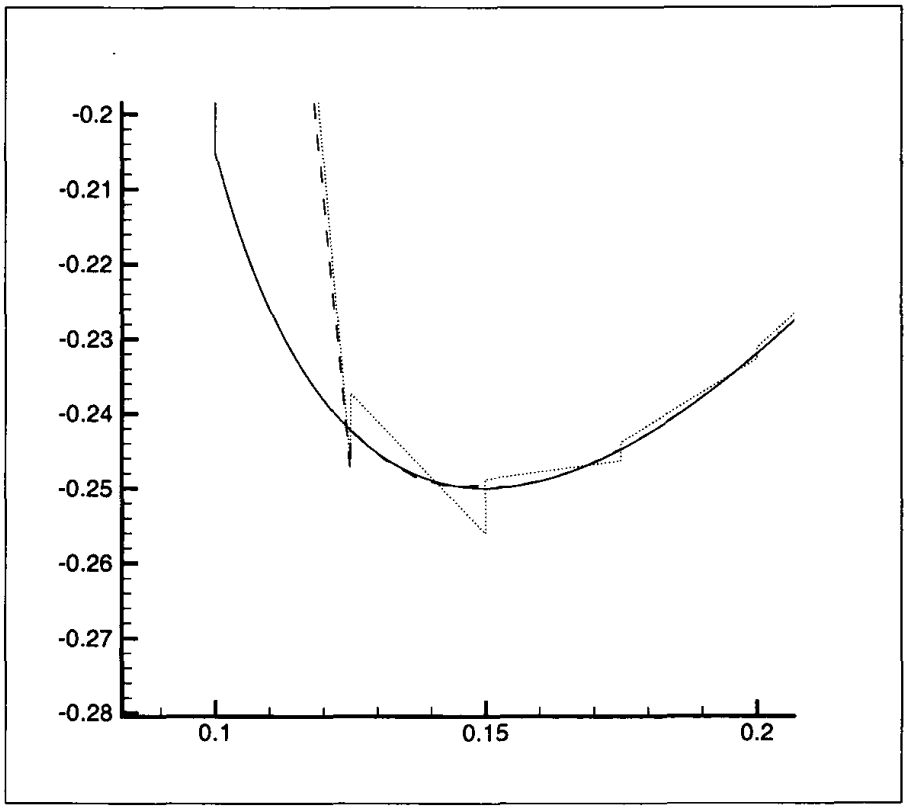


Fig. 3.23: Detail of previous figure. Behavior of the approximate solutions four elements in front of the shock: Exact solution (solid line), piecewise linear solution (dotted line), and piecewise quadratic solution (dashed line).

The discrete case. Thus, we start by finding the discrete version of the form $\mathbb{B}(\cdot,\cdot)$. If we replace $v(x)$ by $v_h(x,t)$ in the equation (3.7), sum on j from 1 to N, and integrate in time from 0 to T, we obtain

$$\mathbb{B}_h(u_h, v_h) = 0, \qquad \forall\, v_h :\ v_h(t) \in V_h^k \quad \forall\, t \in (0,T). \tag{3.31}$$

where

$$\begin{aligned}\mathbb{B}_h(u_h, v_h) \ &= \int_0^T \int_0^1 \partial_t u_h(x,t)\, v_h(x,t)\, dx\, dt \\ &\quad - \int_0^T \sum_{1\le j\le N} \int_{I_j} c\, u_h(x,t)\, \partial_x v_h(x,t)\, dx\, dt \\ &\quad - \int_0^T \sum_{1\le j\le N} h(u_h)_{j+1/2}(t)\, [\, v_h(t)\,]_{j+1/2}\, dt.\end{aligned} \tag{3.32}$$

Following the model provided by the continuous case, we next obtain an expression for $\mathbb{B}_h(w_h, w_h)$. It is contained in the following result which will proved later.

Lemma 3.1 *We have*

$$\mathbb{B}_h(w_h, w_h) = \frac{1}{2} \| w_h(T) \|^2_{L^2(0,1)} + \Theta_T(w_h) - \frac{1}{2} \| w_h(0) \|^2_{L^2(0,1)},$$

where

$$\Theta_T(w_h) = \tfrac{|c|}{2} \int_0^T \textstyle\sum_{1\le j\le N} [\, w_h(t)\,]^2_{j+1/2}\, dt.$$

Taking $w_h = u_h$ in the above result and noting that by (3.31),

$$\mathbb{B}_h(u_h, u_h) = 0,$$

we get the equality

$$\tfrac{1}{2} \| u_h(T) \|^2_{L^2(0,1)} + \Theta_T(u_h) = \tfrac{1}{2} \| u_h(0) \|^2_{L^2(0,1)},$$

from which Proposition 3.1 easily follows, since

$$\frac{1}{2} \| u_h(T) \|^2_{L^2(0,1)} \le \frac{1}{2} \| u_0 \|^2_{L^2(0,1)},$$

by (3.8). It only remains to prove Lemma 3.1.

Proof of Lemma 3.1. After setting $u_h = v_h = w_h$ in the definition of $\mathbb{B}_h$, (3.32), we get

$$\mathbb{B}_h(w_h, w_h) = \frac{1}{2}\| w_h(T) \|^2_{L^2(0,1)} + \int_0^T \Theta_{diss}(t)\, dt - \frac{1}{2}\| w_h(0) \|^2_{L^2(0,1)},$$

where

$$\Theta_{diss}(t) = -\sum_{1\le j\le N} \left\{ h(w_h)_{j+1/2}(t)\, [\, w_h(t)\,]_{j+1/2} + \int_{I_j} c\, w_h(x,t)\, \partial_x\, w_h(x,t)\, dx \right\}$$

We only have to show that $\int_0^T \Theta_{diss}(t)\, dt = \Theta_T(w_h)$. To do that, we proceed as follows. Dropping the dependence on the variable t and setting

$$\overline{w}_h(x_{j+1/2}) = \frac{1}{2}(\, w_h(x^-_{j+1/2}) + w_h(x^+_{j+1/2})\,),$$

we have, by the definition of the flux h, (3.11),

$$-\sum_{1\le j\le N} \int_{I_j} h(w_h)_{j+1/2}\, [\, w_h\,]_{j+1/2} = -\sum_{1\le j\le N} \{\, c\,\overline{w}_h\, [\, w_h\,] - \frac{|c|}{2}\, [\, w_h\,]^2 \,\}_{j+1/2},$$

and

$$\begin{aligned} -\sum_{1\le j\le N} \int_{I_j} c\, w_h(x)\, \partial_x\, w_h(x)\, dx &= \frac{c}{2} \sum_{1\le j\le N} [\, w_h^2\,]_{j+1/2} \\ &= c \sum_{1\le j\le N} \{\overline{w}_h\, [\, w_h\,]\}_{j+1/2} \end{aligned}$$

Hence

$$\Theta_{diss}(t) = \frac{|c|}{2} \sum_{1\le j\le N} [u_h(t)]^2_{j+1/2},$$

and the result follows. This completes the proof of Lemma 3.1. □

This completes the proof of Proposition 3.1.

Proof of the Theorem 3.1 In this subsection, we prove the error estimate of Theorem 3.1 which holds for the linear case $f(u) = cu$. To do that, we first show how to estimate the error between the solutions $w_\nu = (u_\nu, q_\nu)^t$, $\nu = 1, 2$, of

$$\begin{aligned} &\partial_t\, u_\nu + \partial_x\, f(u_\nu) = 0 \quad \text{in } (0,T)\times(0,1), \\ &u_\nu(t=0) = u_{0,\nu}, \quad \text{on } (0,1). \end{aligned}$$

Then, we mimic the argument in order to prove Theorem 3.1.

The continuous case as a model. By the definition of the form $\mathbb{B}(\cdot,\cdot)$, (3.30), we have, for $\nu = 1, 2$,

$$\mathbb{B}(w_\nu, v) = 0, \qquad \forall\ v:\ v(t) \text{ is smooth} \quad \forall\ t \in (0,T).$$

Since the form $\mathbb{B}(\cdot,\cdot)$ is bilinear, from the above equation we obtain the so-called *error equation*:

$$\mathbb{B}(e, v) = 0, \qquad \forall\ v:\ v(t) \text{ is smooth} \quad \forall\ t \in (0,T). \tag{3.33}$$

where $e = w_1 - w_2$. Now, since

$$\mathbb{B}(e,e) = \frac{1}{2}\| e(T) \|^2_{L^2(0,1)} - \frac{1}{2}\| e(0) \|^2_{L^2(0,1)},$$

and

$$\mathbb{B}(e,e) = 0,$$

by the error equation (3.33), we immediately obtain the error estimate we sought:

$$\frac{1}{2}\| e(T) \|^2_{L^2(0,1)} = \frac{1}{2}\| u_{0,1} - u_{0,2} \|^2_{L^2(0,1)}.$$

To prove Theorem 3.1, we only need to obtain a discrete version of this argument.

The discrete case. Since,

$$\begin{aligned} \mathbb{B}_h(u_h, v_h) &= 0, \qquad \forall\ v_h:\ v(t) \in V_h \quad \forall\ t \in (0,T), \\ \mathbb{B}_h(u, v_h) &= 0, \qquad \forall\ v_h:\ v_h(t) \in V_h \quad \forall\ t \in (0,T), \end{aligned}$$

by (3.7) and by equations (3.4), respectively, we easily obtain our *error equation*:

$$\mathbb{B}_h(e, v_h) = 0, \qquad \forall\ v_h:\ v_h(t) \in V_h \quad \forall\ t \in (0,T), \tag{3.34}$$

where $e = w - w_h$.

Now, according to the continuous case argument, we should consider next the quantity $\mathbb{B}_h(e,e)$; however, since $e(t)$ is not in the finite element space V_h, it is more convenient to consider $\mathbb{B}_h(\mathbb{P}_h(e), \mathbb{P}_h(e))$, where $\mathbb{P}_h(e(t))$ is the L^2-projection of the error $e(t)$ into the finite element space V_h^k.

The L^2-projection of the function $p \in \mathrm{L}^2(0,1)$ into V_h, $\mathbb{P}_h(p)$, is defined as the only element of the finite element space V_h such that

$$\int_0^1 \big(\mathbb{P}_h(p)(x) - p(x)\big)\, v_h(x)\, dx = 0, \qquad \forall\ v_h \in V_h. \tag{3.35}$$

Note that in fact $u_h(t=0) = \mathbb{P}_h(u_0)$, by (3.8).

Thus, by Lemma 3.1, we have

$$\mathbb{B}_h(\mathbb{P}_h(e),\mathbb{P}_h(e)) = \frac{1}{2}\| \mathbb{P}_h(e(T)) \|^2_{L^2(0,1)} + \Theta_T(\mathbb{P}_h(e)) - \frac{1}{2}\| \mathbb{P}_h(e(0)) \|^2_{L^2(0,1)},$$

and since

$$\mathbb{P}_h(e(0)) = \mathbb{P}_h(u_0 - u_h(0)) = \mathbb{P}_h(u_0) - u_h(0) = 0,$$

and

$$\mathbb{B}_h(\mathbb{P}_h(e),\mathbb{P}_h(e)) = \mathbb{B}_h(\mathbb{P}_h(e) - e, \mathbb{P}_h(e)) = \mathbb{B}_h(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)),$$

by the *error equation* (3.34), we get

$$\frac{1}{2}\| \mathbb{P}_h(e(T)) \|^2_{L^2(0,1)} + \Theta_T(\mathbb{P}_h(e)) = \mathbb{B}_h(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)). \tag{3.36}$$

It only remains to estimate the right-hand side

$$\mathbb{B}(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)),$$

which, according to our continuous model, should be small.

Estimating the right-hand side. To show that this is so, we must suitably treat the term $\mathbb{B}(\mathbb{P}_h(w) - w, \mathbb{P}_h(e))$. We start with the following remarkable result.

Lemma 3.2 *We have*

$$\mathbb{B}_h(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)) = -\int_0^T \sum_{1\le j\le N} h(\mathbb{P}_h(u) - u)_{j+1/2}(t)\, [\, \mathbb{P}_h(e)(t)\,]_{j+1/2}\, dt.$$

Proof Setting $p = \mathbb{P}_h(u) - u$ and $v_h = \mathbb{P}_h(e)$ and recalling the definition of $\mathbb{B}_h(\cdot,\cdot)$, (3.32), we have

$$\begin{aligned}
\mathbb{B}_h(p, v_h) &= \int_0^T \int_0^1 \partial_t p(x,t)\, v_h(x,t)\, dx\, dt \\
&\quad - \int_0^T \sum_{1\le j\le N} \int_{I_j} c\, p(x,t)\, \partial_x v_h(x,t)\, dx\, dt \\
&\quad - \int_0^T \sum_{1\le j\le N} h(p)_{j+1/2}(t)\, [\, v_h(t)\,]_{j+1/2}\, dt \\
&= -\int_0^T \sum_{1\le j\le N} h(p)_{j+1/2}(t)\, [\, v_h(t)\,]_{j+1/2}\, dt,
\end{aligned}$$

by the definition of the L^2-projection (3.35). This completes the proof. □

Now, we can see that a simple application of Young's inequality and a standard approximation result should give us the estimate we were looking for. The approximation result we need is the following.

Lemma 3.3 *If* $w \in H^{k+1}(I_j \cup I_{j+1})$, *then*

$$| h(\mathbb{P}_h(w) - w)(x_{j+1/2}) | \leq c_k \, (\Delta x)^{k+1/2} \, \frac{|c|}{2} \, | w |_{H^{k+1}(I_j \cup I_{j+1})},$$

where the constant c_k *depends solely on* k.

Proof. Dropping the argument $x_{j+1/2}$ we have, by the definition (3.11) of the flux h,

$$\begin{aligned} | h(\mathbb{P}(w) - w) | &= \frac{c}{2}(\mathbb{P}_h(w)^+ + \mathbb{P}_h(w)^-) - \frac{|c|}{2}(\mathbb{P}_h(w)^+ - \mathbb{P}_h(w)^-) - cw \\ &= \frac{c - |c|}{2}(\mathbb{P}_h(w)^+ - w) + \frac{c + |c|}{2}(\mathbb{P}_h(w)^- - w) \\ &\leq |c| \, \max\{ | \mathbb{P}_h(w)^+ - w |, | \mathbb{P}_h(w)^- - w | \} \end{aligned}$$

and the result follows from the properties of $\mathbb{P}_h$ after a simple application of the Bramble-Hilbert lemma; see [11]. This completes the proof. □

An immediate consequence of this result is the estimate we wanted.

Lemma 3.4 *We have*

$$\mathbb{B}_h(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)) \leq c_k^2 \, (\Delta x)^{2k+1} \, \frac{|c|}{2} \, T \, | u_0 |^2_{H^{k+1}(0,1)} + \frac{1}{2} \, \Theta_T(\mathbb{P}_h(e)),$$

where the constant c_k *depends solely on* k.

Proof. After using Young's inequality in the right-hand side of Lemma 3.2, we get

$$\begin{aligned} \mathbb{B}_h(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)) \leq & \int_0^T \sum_{1 \leq j \leq N} \frac{1}{|c|} \, | h(\mathbb{P}_h(u) - u)_{j+1/2}(t) \, |^2 \\ & + \int_0^T \sum_{1 \leq j \leq N} \frac{|c|}{4} [\, \mathbb{P}_h(e)(t) \,]^2_{j+1/2} \, dt. \end{aligned}$$

By Lemma 3.3 and the definition of the form Θ_T, we get

$$\begin{aligned} \mathbf{B_h}(\mathbb{P}_h(u) - u, \mathbb{P}_h(e)) &\leq c_k^2 \, (\Delta x)^{2k+1} \, \frac{|c|}{4} \int_0^T \sum_{1 \leq j \leq N} | u |^2_{H^{k+1}(I_j \cup I_{j+1})} + \frac{1}{2} \, \Theta_T(\mathbb{P}_h(e)) \\ &\leq c_k^2 \, (\Delta x)^{2k+1} \, \frac{|c|}{2} \, T \, | u_0 |^2_{H^{k+1}(0,1)} + \frac{1}{2} \, \Theta_T(\mathbb{P}_h(e)). \end{aligned}$$

This completes the proof. □

Conclusion. Finally, inserting in the equation (3.36) the estimate of its right hand side obtained in Lemma 3.4, we get

$$\| \, \mathbb{P}_h(e(T)) \, \|^2_{L^2(0,1)} + \Theta_T(\mathbb{P}_h(e)) \leq c_k \, (\Delta x)^{2k+1} \, | \, c \, | \, T \, | \, u_0 \, |^2_{H^{k+1}(0,1)},$$

Theorem 3.1 now follows from the above estimate and from the following inequality:

$$\begin{aligned} \| e(T) \|_{L^2(0,1)} &\leq \| u(T) - \mathbb{P}_h(u(T)) \|_{L^2(0,1)} + \| \mathbb{P}_h(e(T)) \|_{L^2(0,1)} \\ &\leq c'_k \, (\Delta x)^{k+1} \, | u_0 |_{H^{k+1}(0,1)} + \| \mathbb{P}_h(e(T)) \|_{L^2(0,1)}. \end{aligned}$$

Proof of the Theorem 3.2 To prove Theorem 3.2, we only have to suitably modify the proof of Theorem 3.1. The modification consists in *replacing* the L^2-projection of the error, $\mathbb{P}_h(e)$, by another projection that we denote by $\mathbb{R}_h(e)$.

Given a function $p \in L^\infty(0,1)$ that is continuous on each element I_j, we define $\mathbb{R}_h(p)$ as the only element of the finite element space V_h such that

$$\forall j = 1, \ldots, N : \qquad \mathbb{R}_h(p)(x_{j,\ell}) - p(x_{j,\ell}) = 0, \qquad \ell = 0, \ldots, k, \tag{3.37}$$

where the points $x_{j,\ell}$ are the Gauss-Radau quadrature points of the interval I_j. We take

$$x_{j,k} = x_{j+1/2}, \quad \text{if } c > 0, \quad \text{and} \quad x_{j,0} = x_{j-1/2}, \quad \text{if } c < 0. \tag{3.38}$$

The special nature of the Gauss-Radau quadrature points is captured in the following property:

$$\begin{aligned} &\forall \varphi \in P^\ell(I_j), \quad \ell \leq k, \quad \forall p \in P^{2k-\ell}(I_j) : \\ &\int_{I_j} (\mathbb{R}_h(p)(x) - p(x)) \, \varphi(x) \, dx = 0. \end{aligned} \tag{3.39}$$

Compare this equality with (3.35).

The quantity $\mathbb{B}_h(\mathbb{R}_h(e), \mathbb{R}_h(e))$. To prove our error estimate, we start by considering the quantity $\mathbb{B}_h(\mathbb{R}_h(e), \mathbb{R}_h(e))$. By Lemma 3.1, we have

$$\mathbb{B}_h(\mathbb{R}_h(e), \mathbb{R}_h(e)) = \frac{1}{2} \| \mathbb{R}_h(e(T)) \|^2_{L^2(0,1)} + \Theta_T(\mathbb{R}_h(e)) - \frac{1}{2} \| \mathbb{R}_h(e(0)) \|^2_{L^2(0,1)},$$

and since

$$\mathbb{B}_h(\mathbb{R}_h(e), \mathbb{R}_h(e)) = \mathbb{B}_h(\mathbb{R}_h(e) - e, \mathbb{R}_h(e)) = \mathbb{B}_h(\mathbb{R}_h(u) - u, \mathbb{R}_h(e)),$$

by the *error equation* (3.34), we get

$$\frac{1}{2} \| \mathbb{R}_h(e(T)) \|^2_{L^2(0,1)} + \Theta_T(\mathbb{R}_h(e)) = \frac{1}{2} \| \mathbb{R}_h(e(0)) \|^2_{L^2(0,1)} + \mathbb{B}_h(\mathbb{R}_h(u) - u, \mathbb{R}_h(e)).$$

Next, we estimate the term $\mathbb{B}(\mathbb{R}_h(u) - u, \mathbb{R}_h(e))$.

Estimating $\mathbb{B}(\mathbb{R}_h(u) - u, \mathbb{R}_h(e))$. The following result corresponds to Lemma 3.2.

Lemma 3.5 *We have*

$$\mathbb{B}_h(\mathbb{R}_h(u) - u, v_h) = \int_0^T \int_0^1 (\mathbb{R}_h(\partial_t u)(x,t) - \partial_t u(x,t))\, v_h(x,t)\, dx\, dt - \int_0^T \sum_{1 \le j \le N} \int_{I_j} c\, (\mathbb{R}_h(u)(x,t) - u(x,t))\, \partial_x v_h(x,t)\, dx\, dt.$$

Proof Setting $p = \mathbb{R}_h(u) - u$ and $v_h = \mathbb{R}_h(e)$ and recalling the definition of $\mathbb{B}_h(\cdot,\cdot)$, (3.32), we have

$$\begin{aligned}\mathbb{B}_h(p, v_h) = &\int_0^T \int_0^1 \partial_t p(x,t)\, v_h(x,t)\, dx\, dt \\ &- \int_0^T \sum_{1 \le j \le N} \int_{I_j} c\, p(x,t)\, \partial_x v_h(x,t)\, dx\, dt \\ &- \int_0^T \sum_{1 \le j \le N} h(p)_{j+1/2}(t)\, [\, v_h(t)\,]_{j+1/2}\, dt.\end{aligned}$$

But, from the definition (3.11) of the flux h, we have

$$\begin{aligned}h(\mathbb{R}(u) - u) &= \frac{c}{2}(\mathbb{R}_h(u)^+ + \mathbb{R}_h(u)^-) - \frac{|c|}{2}(\mathbb{R}_h(u)^+ - \mathbb{R}_h(u)^-) - c\,u \\ &= \frac{c - |c|}{2}(\mathbb{R}_h(u)^+ - u) + \frac{c + |c|}{2}(\mathbb{R}_h(u)^- - u) \\ &= 0,\end{aligned}$$

by (3.38) and the result follows. □

Next, we need some approximation results.

Lemma 3.6 *If* $w \in H^{k+2}(I_j)$, *and* $v_h \in P^k(I_j)$, *then*

$$\left| \int_{I_j} (\mathbb{R}_h(w) - w)(x)\, v_h(x)\, dx \right| \le c_k\, (\Delta x)^{k+1}\, |\, w\, |_{H^{k+1}(I_j)}\, \|\, v_h\, \|_{L^2(I_j)},$$

$$\left| \int_{I_j} (\mathbb{R}_h(w) - w)(x)\, \partial_x v_h(x)\, dx \right| \le c_k\, (\Delta x)^{k+1}\, |\, w\, |_{H^{k+2}(I_j)}\, \|\, v_h\, \|_{L^2(I_j)},$$

where the constant c_k *depends solely on* k.

Proof. The first inequality follows from the property (3.39) with $\ell = k$ and from standard approximation results. The second follows in a similar way from the property 3.39 with $\ell = k - 1$ and a standard scaling argument. This completes the proof. □

An immediate consequence of this result is the estimate we wanted.

Lemma 3.7 *We have*

$$\mathbb{B}_h(\mathbb{R}_h(u)-u,\mathbb{R}_h(e)) \leq c_k\,(\Delta x)^{k+1}\,|\,u_0\,|_{H^{k+2}(0,1)}\int_0^T \|\,\mathbb{R}_h(e(t))\,\|_{L^2(0,1)}\,dt,$$

where the constant c_k depends solely on k and $|\,c\,|$.

Conclusion. Finally, inserting in the equation (3.36) the estimate of its right hand side obtained in Lemma 3.7, we get

$$\begin{aligned}\|\,\mathbb{R}_h(e(T))\,\|^2_{L^2(0,1)} + \Theta_T(\mathbb{R}_h(e)) &\leq \|\,\mathbb{R}_h(e(0))\,\|^2_{L^2(0,1)}\\ &\quad + c_k\,(\Delta x)^{k+1}\,|\,u_0\,|_{H^{k+2}(0,1)}\int_0^T \|\,\mathbb{R}_h(e(t))\,\|_{L^2(0,1)}\,dt.\end{aligned}$$

After applying a simple variation of the Gronwall lemma, we obtain

$$\begin{aligned}\|\,\mathbb{R}_h(e(T))\,\|_{L^2(0,1)} &\leq \|\,\mathbb{R}_h(e(0))(x)\,\|_{L^2(0,1)} + c_k\,(\Delta x)^{k+1}\,T\,|\,u_0\,|_{H^{k+2}(0,1)}\\ &\leq c'_k(\Delta x)^{k+1}\,|\,u_0\,|_{H^{k+2}(0,1)}.\end{aligned}$$

Theorem 3.2 now follows from the above estimate and from the following inequality:

$$\begin{aligned}\|\,e(T)\,\|_{L^2(0,1)} &\leq \|\,u(T)-\mathbb{R}_h(u(T))\,\|_{L^2(0,1)} + \|\,\mathbb{R}_h(e(T))\,\|_{L^2(0,1)}\\ &\leq c'_k\,(\Delta x)^{k+1}\,|\,u_0\,|_{H^{k+1}(0,1)} + \|\,\mathbb{R}_h(e(T))\,\|_{L^2(0,1)}.\end{aligned}$$

4 The RKDG method for multidimensional systems

4.1 Introduction

In this section, we extend the RKDG methods to multidimensional systems:

$$u_t + \nabla f(u) = 0, \qquad \text{in } \Omega \times (0,T), \tag{4.1}$$
$$u(x,0) = u_0(x), \qquad \forall\, x \in \Omega, \tag{4.2}$$

and periodic boundary conditions. For simplicity, we assume that Ω is the unit cube.

This section is essentially devoted to the description of the algorithms and their implementation details. The practitioner should be able to find here all the necessary information to completely code the RKDG methods.

This section also contains two sets of numerical results for the Euler equations of gas dynamics in two space dimensions. The first set is devoted to transient computations and domains that have corners; the effect of using triangles or rectangles and the effect of using polynomials of degree one or two are explored. The main conclusions from these computations are that (i) the RKDG method works as well with triangles as it does with rectangles and that (ii) the use of high-order polynomials does not deteriorate the approximation of strong shocks and is advantageous in the approximation of contact discontinuities.

The second set concerns steady state computations with smooth solutions. For these computations, no generalized slope limiter is needed. The effect of (i) the quality of the approximation of curved boundaries and of (ii) the degree of the polynomials on the quality of the approximate solution is explored. The main conclusions from these computations are that (i) a high-order approximation of the curve boundaries introduces a dramatic improvement on the quality of the solution and that (ii) the use of high-degree polynomials is advantageous when smooth solutions are shought.

This section contains material from the papers [14], [13], and [19]. It also contains numerical results from the paper by Bassi and Rebay [2] in two dimensions and from the paper by Warburton, Lomtev, Kirby and Karniadakis [65] in three dimensions.

4.2 The general RKDG method

The RKDG method for multidimensional systems has the same structure it has for one-dimensional scalar conservation laws, that is,

- Set $u_h^0 = \Lambda\Pi_h\ P_{V_h}(u_0)$;

- For $n = 0, ..., N-1$ compute u_h^{n+1} as follows:
  1. set $u_h^{(0)} = u_h^n$;
  2. for $i = 1, ..., k+1$ compute the intermediate functions:

$$u_h^{(i)} = \Lambda\Pi_h \left\{ \sum_{l=0}^{i-1} \alpha_{il} u_h^{(l)} + \beta_{il} \Delta t^n L_h(u_h^{(l)}) \right\};$$

  3. set $u_h^{n+1} = u_h^{(k+1)}$.

In what follows, we describe the operator L_h that results form the DG-space discretization, and the generalized slope limiter $\Lambda\Pi_h$.

The Discontinuous Galerkin space discretization To show how to discretize in space by the DG method, it is enough to consider the case in which u is a scalar quantity since to deal with the general case in which u, we apply the same procedure component by component.

Once a triangulation $\mathbb{T}_h$ of Ω has been obtained, we determine $L_h(\cdot)$ as follows. First, we multiply (4.1) by v_h in the finite elemen space V_h, integrate over the element K of the triangulation $\mathbb{T}_h$ and replace the exact solution u by its approximation $u_h \in V_h$:

$$\frac{d}{dt} \int_K u_h(t,x)\, v_h(x)\, dx + \int_K div\, f(u_h(t,x))\, v_h(x)\, dx = 0, \ \forall v_h \in V_h.$$

Integrating by parts formally we obtain

$$\begin{aligned} &\tfrac{d}{dt} \textstyle\int_K u_h(t,x)\, v_h(x)\, dx + \sum_{e \in \partial K} \int_e f(u_h(t,x)) \cdot n_{e,K}\ v_h(x)\, d\Gamma \\ &\quad - \textstyle\int_K f(u_h(t,x)) \cdot \ \text{grad}\ v_h(x)\, dx = 0, \quad \forall v_h \in V_h, \end{aligned}$$

where $n_{e,K}$ is the outward unit normal to the edge e. Notice that $f(u_h(t,x)) \cdot n_{e,K}$ does not have a precise meaning, for u_h is discontinuous at $x \in e \in \partial K$. Thus, as in the one dimensional case, we replace $f(u_h(t,x)) \cdot n_{e,K}$ by the function $h_{e,K}(u_h(t, x^{int(K)}), u_h(t, x^{ext(K)}))$. The function $h_{e,K}(\cdot,\cdot)$ is any consistent two–point monotone Lipschitz flux, consistent with $f(u) \cdot n_{e,K}$.

In this way we obtain

$$\begin{aligned} &\tfrac{d}{dt} \textstyle\int_K u_h(t,x) v_h(x)\, dx + \sum_{e \in \partial K} \int_e h_{e,K}(t,x)\ v_h(x)\, d\,\Gamma \\ &- \textstyle\int_K f(u_h(t,x)) \cdot \ \text{grad}\ v_h(x)\, dx = 0, \qquad \forall\ v_h \in V_h. \end{aligned}$$

Finally, we replace the integrals by quadrature rules that we shall choose as follows:

$$\int_e h_{e,K}(t,x)\, v_h(x)\, d\Gamma \approx \sum_{l=1}^{L} \omega_l\, h_{e,K}(t,x_{el})\, v(x_{el})|e|, \tag{4.3}$$

$$\int_K f(u_h(t,x)) \cdot \text{ grad } v_h(x)\, dx \approx$$

$$\sum_{j=1}^{M} \omega_j\, f(u_h(t,x_{Kj})) \cdot \text{ grad } v_h(x_{Kj})|K|. \tag{4.4}$$

Thus, we finally obtain the weak formulation:

$$\frac{d}{dt}\int_k u_h(t,x)v_h(x)dx + \sum_{e\in\partial K} \sum_{l=1}^{L} \omega_l\, h_{e,K}(t,x_{el})\, v(x_{el})|e|$$
$$-\sum_{j=1}^{M} \omega_j\, f(u_h(t,x_{Kj})) \cdot \text{grad} v_h(x_{Kj})|K| = 0, \quad \forall v_h \in V_h, \quad \forall K \in \mathbb{T}_h.$$

These equations can be rewritten in ODE form as $\frac{d}{dt}u_h = L_h(u_h, \gamma_h)$. This defines the operator $L_h(u_h)$, which is a discrete approximation of $-div\, f(u)$. The following result gives an indication of the quality of this approximation.

Proposition 4.1 *Let $f(u) \in W^{k+2,\infty}(\Omega)$, and set $\gamma = trace(u)$. Let the quadrature rule over the edges be exact for polynomials of degree $(2k+1)$, and let the one over the element be exact for polynomials of degree $(2k)$. Assume that the family of triangulations $\mathbb{F} = \{\mathbb{T}_h\}_{h>0}$ is regular, i.e., that there is a constant σ such that:*

$$\frac{h_K}{\rho_K} \geq \sigma, \quad \forall K \in \mathbb{T}_h, \quad \forall \mathbb{T}_h \in \mathbb{F}, \tag{4.5}$$

where h_K is the diameter of K, and ρ_K is the diameter of the biggest ball included in K. Then, if $V(K) \supset P^k(K), \quad \forall\, K \in \mathbb{T}_h$:

$$\|L_h(u,\gamma) + div\, f(u)\|_{L^\infty(\Omega)} \leq C\, h^{k+1} |f(u)|_{W^{k+2,\infty}(\Omega)}.$$

For a proof, see [13].

The form of the generalized slope limiter $\Lambda\Pi_h$. The construction of generalized slope limiters $\Lambda\Pi_h$ for several space dimensions is not a trivial matter and will not be discussed in these notes; we refer the interested reader to the paper by Cockburn, Hou, and Shu [13].

In these notes, we restrict ourselves to displaying very simple, practical, and effective generalized slope limiters $\Lambda\Pi_h$ which are closely related to the generalized slope limiters $\Lambda\Pi_h^k$ of the previous section.

To compute $\Lambda\Pi_h u_h$, we rely on the *assumption* that spurious oscillations are present in u_h only if they are present in its P^1 part u_h^1, which is its L^2-projection into the space of piecewise linear functions V_h^1. Thus, if they are not present in u_h^1, i.e., if

$$u_h^1 = \Lambda\Pi_h u_h^1,$$

then we assume that they are not present in u_h and hence do not do any limiting:

$$\Lambda\Pi_h u_h = u_h .$$

On the other hand, if spurious oscillations are present in the P^1 part of the solution u_h^1, i.e., if

$$u_h^1 \neq \Lambda\Pi_h u_h^1,$$

then we chop off the higher order part of the numerical solution, and limit the remaining P^1 part:

$$\Lambda\Pi_h u_h = \Lambda\Pi_h u_h^1.$$

In this way, in order to define $\Lambda\Pi_h$ for arbitrary space V_h, we only need to actually define it for piecewise linear functions V_h^1. The exact way to do that, both for the triangular elements and for the rectangular elements, will be discussed in the next section.

4.3 Algorithm and implementation details

In this section we give the algorithm and implementation details, including numerical fluxes, quadrature rules, degrees of freedom, fluxes, and limiters of the RKDG method for both piecewise-linear and piecewise-quadratic approximations in both triangular and rectangular elements.

Fluxes The numerical flux we use is the simple Lax-Friedrichs flux:

$$h_{e,K}(a,b) = \frac{1}{2}\left[\mathbf{f}(a)\cdot n_{e,K} + \mathbf{f}(b)\cdot n_{e,K} - \alpha_{e,K}\,(b-a)\right].$$

The numerical viscosity constant $\alpha_{e,K}$ should be an estimate of the biggest eigenvalue of the Jacobian $\frac{\partial}{\partial u}\mathbf{f}(u_h(x,t))\cdot n_{e,K}$ for (x,t) in a neighborhood of the edge e.

For the triangular elements, we use the local Lax-Friedrichs recipe:

- Take $\alpha_{e,K}$ to be the larger one of the largest eigenvalue (in absolute value) of $\frac{\partial}{\partial u}\mathbf{f}(\bar{u}_K)\cdot n_{e,K}$ and that of $\frac{\partial}{\partial u}\mathbf{f}(\bar{u}_{K'})\cdot n_{e,K}$, where $\bar{u}_K$ and $\bar{u}_{K'}$ are the means of the numerical solution in the elements K and K' sharing the edge e.

For the rectangular elements, we use the local Lax-Friedrichs recipe :

- Take $\alpha_{e,K}$ to be the largest of the largest eigenvalue (in absolute value) of $\frac{\partial}{\partial u}\mathbf{f}(\bar{u}_{K''})\cdot n_{e,K}$, where $\bar{u}_{K''}$ is the mean of the numerical solution in the element K'', which runs over all elements on the same line (horizontally or vertically, depending on the direction of $n_{e,K}$) with K and K' sharing the edge e.

Quadrature rules According to the analysis done in [13], the quadrature rules for the edges of the elements, (4.3), must be exact for polynomials of degree $2k+1$, and the quadrature rules for the interior of the elements, (4.4), must be exact for polynomials of degree $2k$, if P^k methods are used. Here we discuss the quadrature points used for P^1 and P^2 in the triangular and rectangular element cases.

The rectangular elements For the edge integral, we use the following two point Gaussian rule

$$\int_{-1}^{1} g(x)dx \approx g\left(-\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}\right), \tag{4.1}$$

for the P^1 case, and the following three point Gaussian rule

$$\int_{-1}^{1} g(x)dx \approx \frac{5}{9}\left[g\left(-\frac{3}{5}\right) + g\left(\frac{3}{5}\right)\right] + \frac{8}{9}g(0), \tag{4.2}$$

for the P^2 case, suitably scaled to the relevant intervals.

For the interior of the elements, we could use a tensor product of (4.1), with four quadrature points, for the P^1 case. But to save cost, we "recycle" the values of the fluxes at the element boundaries, and only add one new quadrature point in the middle of the element. Thus, to approximate the integral $\int_{-1}^{1}\int_{-1}^{1} g(x,y)dxdy$, we use the following quadrature rule:

$$\begin{aligned}\approx \frac{1}{4}\Bigg[&g\left(-1,\frac{1}{\sqrt{3}}\right) + g\left(-1,-\frac{1}{\sqrt{3}}\right) + g\left(-\frac{1}{\sqrt{3}},-1\right) + g\left(\frac{1}{\sqrt{3}},-1\right)\\ &+g\left(1,-\frac{1}{\sqrt{3}}\right) + g\left(1,\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}},1\right) + g\left(-\frac{1}{\sqrt{3}},1\right)\Bigg] + 2\,g(0,0).\end{aligned}$$

For the P^2 case, we use a tensor product of (4.2), with 9 quadrature points.

The triangular elements For the edge integral, we use the same two point or three point Gaussian quadratures as in the rectangular case, (4.1) and (4.2), for the P^1 and P^2 cases, respectively.

For the interior integrals (4.4), we use the three mid-point rule

$$\int_K g(x,y)dxdy \approx \frac{|K|}{3}\sum_{i=1}^{3} g(m_i),$$

where m_i are the mid-points of the edges, for the P^1 case. For the P^2 case, we use a seven-point quadrature rule which is exact for polynomials of degree 5 over triangles.

Basis and degrees of freedom We emphasize that the choice of basis and degrees of freedom does not affect the algorithm, as it is completely determined by the choice of function space $V(h)$, the numerical fluxes, the quadrature rules, the slope limiting, and the time discretization. However, a suitable choice of basis and degrees of freedom may simplify the implementation and calculation.

The rectangular elements For the P^1 case, we use the following expression for the approximate solution $u_h(x,y,t)$ inside the rectangular element $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$:

$$u_h(x,y,t) = \bar{u}(t) + u_x(t)\varphi_i(x) + u_y(t)\psi_j(y) \tag{4.3}$$

where

$$\varphi_i(x) = \frac{x - x_i}{\Delta x_i/2}, \qquad \psi_j(y) = \frac{y - y_j}{\Delta y_j/2}, \tag{4.4}$$

and

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}.$$

The degrees of freedoms, to be evolved in time, are then

$$\bar{u}(t), \;\; u_x(t), \;\; u_y(t).$$

Here we have omitted the subscripts ij these degrees of freedom should have, to indicate that they belong to the element ij which is $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$.

Notice that the basis functions

$$1, \;\; \varphi_i(x), \;\; \psi_j(y),$$

are orthogonal, hence the local mass matrix is diagonal:

$$M = \Delta x_i \Delta y_j \, diag\left(1, \frac{1}{3}, \frac{1}{3}\right).$$

For the P^2 case, the expression for the approximate solution $u_h(x,y,t)$ inside the rectangular element $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ is:

$$\begin{aligned} u_h(x,y,t) = \bar{u}(t) &+ u_x(t)\varphi_i(x) + u_y(t)\psi_j(y) + u_{xy}(t)\varphi_i(x)\psi_j(y) \\ &+ u_{xx}(t)\left(\varphi_i^2(x) - \frac{1}{3}\right) + u_{yy}(t)\left(\psi_j^2(y) - \frac{1}{3}\right), \end{aligned} \tag{4.5}$$

where $\varphi_i(x)$ and $\psi_j(y)$ are defined by (4.4). The degrees of freedoms, to be evolved in time, are

$$\bar{u}(t), \;\; u_x(t), \;\; u_y(t), \;\; u_{xy}(t), \;\; u_{xx}(t), \;\; u_{yy}(t).$$

Again the basis functions

$$1,\ \varphi_i(x),\ \psi_j(y),\ \varphi_i(x)\psi_j(y),\ \varphi_i^2(x)-\frac{1}{3},\ \psi_j^2(y)-\frac{1}{3},$$

are orthogonal, hence the local mass matrix is diagonal:

$$M = \Delta x_i \Delta y_j\, diag\left(1, \frac{1}{3}, \frac{1}{3}, \frac{1}{9}, \frac{4}{45}, \frac{4}{45}\right).$$

The triangular elements For the P^1 case, we use the following expression for the approximate solution $u_h(x,y,t)$ inside the triangle K:

$$u_h(x,y,t) = \sum_{i=1}^{3} u_i(t)\varphi_i(x,y)$$

where the degrees of freedom $u_i(t)$ are values of the numerical solution at the midpoints of edges, and the basis function $\varphi_i(x,y)$ is the linear function which takes the value 1 at the mid-point m_i of the i-th edge, and the value 0 at the mid-points of the two other edges. The mass matrix is diagonal

$$M = |K| diag\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

For the P^2 case, we use the following expression for the approximate solution $u_h(x,y,t)$ inside the triangle K:

$$u_h(x,y,t) = \sum_{i=1}^{6} u_i(t)\xi_i(x,y)$$

where the degrees of freedom, $u_i(t)$, are values of the numerical solution at the three midpoints of edges and the three vertices. The basis function $\xi_i(x,y)$, is the quadratic function which takes the value 1 at the point i of the six points mentioned above (the three midpoints of edges and the three vertices), and the value 0 at the remaining five points. The mass matrix this time is not diagonal.

Limiting We construct slope limiting operators $\Lambda\Pi_h$ on piecewise linear functions u_h in such a way that the following properties are satisfied:

1. Accuracy: if u_h is linear then $\Lambda\Pi_h u_h = u_h$.
2. Conservation of mass: for every element K of the triangulation $\mathbb{T}_h$, we have:
$$\int_K \Lambda\Pi_h u_h = \int_K u_h.$$
3. Slope limiting: on each element K of $\mathbb{T}_h$, the gradient of $\Lambda\Pi_h u_h$ is not bigger than that of u_h.

The actual form of the slope limiting operators is closely related to that of the slope limiting operators studied in [15] and [13].

The rectangular elements The limiting is performed on u_x and u_y in (4.3), using the differences of the means. For a scalar equation, u_x would be limited (replaced) by

$$\bar{m}\left(u_x, \bar{u}_{i+1,j} - \bar{u}_{ij}, \bar{u}_{ij} - \bar{u}_{i-1,j}\right) \tag{4.6}$$

where the function $\bar{m}$ is the TVB corrected *minmod* function defined in the previous section.

The TVB correction is needed to avoid unnecessary limiting near smooth extrema, where the quantity u_x or u_y is on the order of $O(\Delta x^2)$ or $O(\Delta y^2)$. For an estimate of the TVB constant M in terms of the second derivatives of the function, see [15]. Usually, the numerical results are not sensitive to the choice of M in a large range. In all the calculations in this paper we take M to be 50.

Similarly, u_y is limited (replaced) by

$$\bar{m}(u_y, \bar{u}_{i,j+1} - \bar{u}_{ij}, \bar{u}_{ij} - \bar{u}_{i,j-1}).$$

with a change of Δx to Δy in (4.6).

For systems, we perform the limiting in the local characteristic variables. To limit the vector u_x in the element ij, we proceed as follows:

- Find the matrix R and its inverse R^{-1}, which diagonalize the Jacobian evaluated at the mean in the element ij in the x-direction:

$$R^{-1}\,\frac{\partial f_1(\bar{u}_{ij})}{\partial u}R = \Lambda\,,$$

  where Λ is a diagonal matrix containing the eigenvalues of the Jacobian. Notice that the columns of R are the right eigenvectors of $\frac{\partial f_1(\bar{u}_{ij})}{\partial u}$ and the rows of R^{-1} are the left eigenvectors.
- Transform all quantities needed for limiting, i.e., the three vectors u_{xij}, $\tilde{u}_{i+1,j} - \bar{u}_{ij}$ and $\bar{u}_{ij} - \bar{u}_{i-1,j}$, to the characteristic fields. This is achieved by left multiplying these three vectors by R^{-1}.
- Apply the scalar limiter (4.6) to each of the components of the transformed vectors.
- The result is transformed back to the original space by left multiplying R on the left.

The triangular elements To construct the slope limiting operators for triangular elements, we proceed as follows. We start by making a simple observation. Consider the triangles in Figure 4.1, where m_1 is the mid-point of the edge on the boundary of K_0 and b_i denotes the barycenter of the triangle K_i for $i = 0, 1, 2, 3$.

Since we have that

$$m_1 - b_0 = \alpha_1\left(b_1 - b_0\right) + \alpha_2\left(b_2 - b_0\right),$$

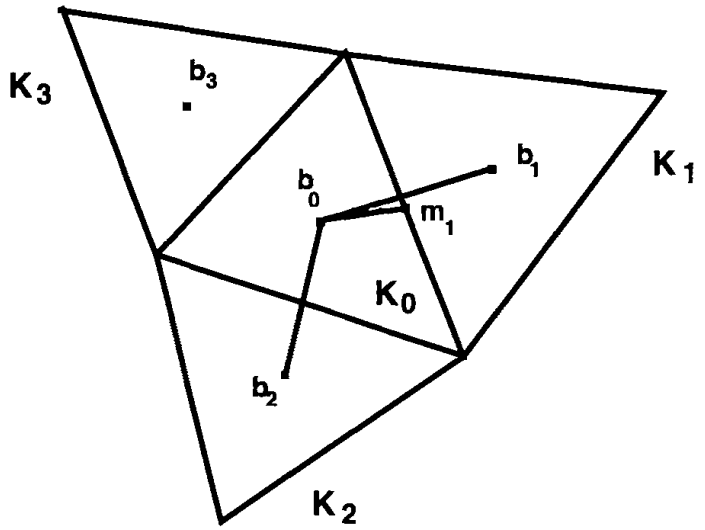


Fig. 4.1: Illustration of limiting.

for some nonnegative coefficients α_1, α_2 which depend only on m_1 and the geometry, we can write, for any linear function u_h,

$$u_h(m_1) - u_h(b_0) = \alpha_1 \left(u_h(b_1) - u_h(b_0)\right) + \alpha_2 \left(u_h(b_2) - u_h(b_0)\right),$$

and since

$$\bar{u}_{K_i} = \frac{1}{|K_i|} \int_{K_i} u_h = u_h(b_i), \qquad i = 0, 1, 2, 3,$$

we have that

$$\tilde{u}_h(m_1, K_0) \equiv u_h(m_1) - \bar{u}_{K_0} = \alpha_1 \left(\bar{u}_{K_1} - \bar{u}_{K_0}\right) + \alpha_2 \left(\bar{u}_{K_2} - \bar{u}_{K_0}\right) \equiv \Delta\bar{u}(m_1, K_0)$$

Now, we are ready to describe the slope limiting. Let us consider a piecewise linear function u_h, and let $m_i, i = 1, 2, 3$ be the three mid-points of the edges of the triangle K_0. We then can write, for $(x, y) \in K_0$,

$$u_h(x, y) = \sum_{i=1}^{3} u_h(m_i)\varphi_i(x, y) = \bar{u}_{K_0} + \sum_{i=1}^{3} \tilde{u}_h(m_i, K_0)\varphi_i(x, y).$$

To compute $\Lambda\Pi_h u_h$, we first compute the quantities

$$\Delta_i = \bar{m}(\tilde{u}_h(m_i, K_0), \nu\, \Delta\bar{u}(m_i, K_0)),$$

where $\bar{m}$ is the TVB modified *minmod* function and $\nu > 1$. We take $\nu = 1.5$ in our numerical runs. Then, if $\sum_{i=1}^{3} \Delta_i = 0$, we simply set

$$\Lambda\Pi_h u_h(x, y) = \bar{u}_{K_0} + \sum_{i=1}^{3} \Delta_i\, \varphi_i(x, y).$$

If $\sum_{i=1}^{3} \Delta_i \neq 0$, we compute

$$pos = \sum_{i=1}^{3} \max(0, \Delta_i), \qquad neg = \sum_{i=1}^{3} \max(0, -\Delta_i),$$

and set

$$\theta^+ = \min\left(1, \frac{neg}{pos}\right), \qquad \theta^- = \min\left(1, \frac{pos}{neg}\right).$$

Then, we define

$$\Lambda\Pi_h u_h(x, y) = \bar{u}_{K_0} + \sum_{i=1}^{3} \hat{\Delta}_i \, \varphi_i(x, y),$$

where

$$\hat{\Delta}_i = \theta^+ \max(0, \Delta_i) - \theta^- \max(0, -\Delta_i).$$

It is very easy to see that this slope limiting operator satisfies the three properties listed above.

For systems, we perform the limiting in the local characteristic variables. To limit Δ_i, we proceed as in the rectangular case, the only difference being that we work with the following Jacobian

$$\frac{\partial}{\partial u} f(\bar{u}_{K_0}) \cdot \frac{m_i - b_0}{|m_i - b_0|}.$$

4.4 Computational results: Transient, nonsmooth solutions

In this section we present several numerical results obtained with the P^1 and P^2 (second and third order accurate) RKDG methods with either rectangles or triangles in the triangulation. These are standard test problems for Euler equations of compressible gas dynamics.

The double-Mach reflection problem Double Mach reflection of a strong shock. This problem was studied extensively in Woodward and Colella [66] and later by many others. We use exactly the same setup as in [66], namely a Mach 10 shock initially makes a 60° angle with a reflecting wall. The undisturbed air ahead of the shock has a density of 1.4 and a pressure of 1.

For the rectangle based triangulation, we use a rectangular computational domain $[0, 4] \times [0, 1]$, as in [66]. The reflecting wall lies at the bottom of the computational domain for $\frac{1}{6} \leq x \leq 4$. Initially a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = \frac{1}{6}$, to mimic an angled wedge. Reflective boundary condition is used for the rest. At the top boundary of our computational domain, the flow values are set to describe the exact motion of the Mach

10 shock. Inflow/outflow boundary conditions are used for the left and right boundaries. As in [66], only the results in $[0,3] \times [0,1]$ are displayed.

For the triangle based triangulation, we have the freedom to treat irregular domains and thus use a true wedged computational domain. Reflective boundary conditions are then used for all the bottom boundary, including the sloped portion. Other boundary conditions are the same as in the rectangle case.

Uniform rectangles are used in the rectangle based triangulations. Four different meshes are used: 240×60 rectangles ($\Delta x = \Delta y = \frac{1}{60}$); 480×120 rectangles ($\Delta x = \Delta y = \frac{1}{120}$); 960×240 rectangles ($\Delta x = \Delta y = \frac{1}{240}$); and 1920×480 rectangles ($\Delta x = \Delta y = \frac{1}{480}$). The density is plotted in Figure 4.2 for the P^1 case and in 4.3 for the P^2 case.

To better appreciate the difference between the P^1 and P^2 results in these pictures, we show a "blowed up" portion around the double Mach region in Figure 4.4 and show one-dimensional cuts along the line $y = 0.4$ in Figures 4.5 and 4.6. In Figure 4.4, w can see that P^2 with $\Delta x = \Delta y = \frac{1}{240}$ has qualitatively the same resolution as P^1 with $\Delta x = \Delta y = \frac{1}{480}$, for the fine details of the complicated structure in this region. P^2 with $\Delta x = \Delta y = \frac{1}{480}$ gives a much better resolution for these structures than P^1 with the same number of rectangles.

Moreover, from Figure 4.5, we clearly see that the difference between the results obtained by using P^1 and P^2, on the same mesh, increases dramatically as the mesh size decreases. This indicates that the use of polynomials of high degree might be beneficial for capturing the above mentioned structures. From Figure 4.6, we see that the results obtained with P^1 are qualitatively similar to those obtained with P^2 in a coarser mesh; the similarity increases as the meshsize decreases. The conclusion here is that, if one is interested in the above mentioned fine structures, then one can use the third order scheme P^2 with only half of the mesh points in each direction as in P^1. This translates into a reduction of a factor of 8 in space-time grid points for 2D time dependent problems, and will more than off-set the increase of cost per mesh point and the smaller CFL number by using the higher order P^2 method. This saving will be even more significant for 3D.

The optimal strategy, of course, is to use adaptivity and concentrate triangles around the interesting region, and/or change the order of the scheme in different regions.

The forward-facing step problem Flow past a forward facing step. This problem was again studied extensively in Woodward and Colella [66] and later by many others. The set up of the problem is the following: A right going Mach 3 uniform flow enters a wind tunnel of 1 unit wide and 3 units long. The step is 0.2 units high and is located 0.6 units from the left-hand end of the tunnel. The problem is initialized by a uniform, right-going Mach 3 flow. Reflective boundary conditions are applied along the walls of the tunnel

and in-flow and out-flow boundary conditions are applied at the entrance (left-hand end) and the exit (right-hand end), respectively.

The corner of the step is a singularity, which we study carefully in our numerical experiments. Unlike in [66] and many other papers, we do not modify our scheme near the corner in any way. It is well known that this leads to an errorneous entropy layer at the downstream bottom wall, as well as a spurious Mach stem at the bottom wall. However, these artifacts decrease when the mesh is refined. In Figure 4.7, second order P^1 results using rectangle triangulations are shown, for a grid refinement study using $\Delta x = \Delta y = \frac{1}{40}$, $\Delta x = \Delta y = \frac{1}{80}$, $\Delta x = \Delta y = \frac{1}{160}$, and $\Delta x = \Delta y = \frac{1}{320}$ as mesh sizes. We can clearly see the improved resolution (especially at the upper slip line from the triple point) and decreased artifacts caused by the corner, with increased mesh points. In Figure 4.8, third order P^2 results using the same meshes are shown.

In order to verify that the erroneous entropy layer at the downstream bottom wall and the spurious Mach stem at the bottom wall are both artifacts caused by the corner singularity, we use our triangle code to locally refine near the corner progressively; we use the meshes displayed in Figure 4.9. In Figure 4.10, we plot the density obtained by the P^1 triangle code, with triangles (roughly the resolution of $\Delta x = \Delta y = \frac{1}{40}$, except around the corner). In Figure 4.11, we plot the entropy around the corner for the same runs. We can see that, with more triangles concentrated near the corner, the artifacts gradually decrease. Results with P^2 codes in Figures 4.12 and 4.13 show a similar trend.

4.5 Computational results: Steady state, smooth solutions

In this section, we present some of the numerical results of Bassi and Rebay [2] in two dimensions and Warburton, Lomtev, Kirby and Karniadakis [65] in three dimensions.

The purpose of the numerical results of Bassi and Rebay [2] we are presenting is to assess (i) the effect of the quality of the approximation of curved boundaries and of (ii) the effect of the degree of the polynomials on the quality of the approximate solution. The test problem we consider here is the two-dimensional steady-state, subsonic flow around a disk at Mach number $M_\infty = 0.38$. Since the solution is smooth and can be computed analytically, the quality of the approximation can be easily assessed.

In the figures 4.14, 4.15, 4.16, and 4.17, details of the meshes around the disk are shown together with the approximate solution given by the RKDG method using piecewise linear elements. These meshes approximate the circle with a polygonal. It can be seen that the approximate solution are of very low quality even for the most refined grid. This is an effect caused by the kinks of the polygonal approximating the circle.

This statement can be easily verified by taking a look to the figures 4.18, 4.19, 4.20, and 4.21. In these pictures the approximate solutions with piece-

wise linear, quadratic, and cubic elements are shown; the meshes have been modified to render *exactly* the circle. It is clear that the improvement in the quality of the approximation is enormous. Thus, a high-quality approximation of the boundaries has a dramatic improvement on the quality of the approximations.

Also, it can be seen that the higher the degree of the polynomials, the better the quality of the approximations, in particular from figures 4.18 and 4.19. In [2], Bassi and Rebay show that the RKDG method using polynomilas of degree k are $(k+1)$-th order accurate for $k = 1, 2, 3$. As a consequence, a RKDG method using polynomials of a higher degree is more efficient than a RKDG method using polynomials of lower degree.

In [65], Warburton, Lomtev, Kirby and Karniadakis present the same test problem in a three dimensional setting. In Figure 4.22, we can see the three-dimensional mesh and the density isosurfaces. We can also see how, while the mesh is being kept fixed and the degree of the polynomials k is increased from 1 to 9, the maximum error on the entropy goes exponentialy to zero. (In the picture, a so-called 'mode' is equal to $k+1$).

4.6 Concluding remarks

In this section, we have extended the RKDG methods to multidimensional systems. We have described in full detail the algorithms and displayed numerical results showing the performance of the methods for the Euler equations of gas dynamics.

The flexibility of the RKDG method to handle nontrivial geometries and to work with different elements has been displayed. Moreover, it has been shown that the use of polynomials of high degree not only does not degrade the resolution of strong shocks, but enhances the resolution of the contact discontinuities and renders the scheme more efficient on smooth regions.

Next, we extend the RKDG methods to convection-dominated problems.

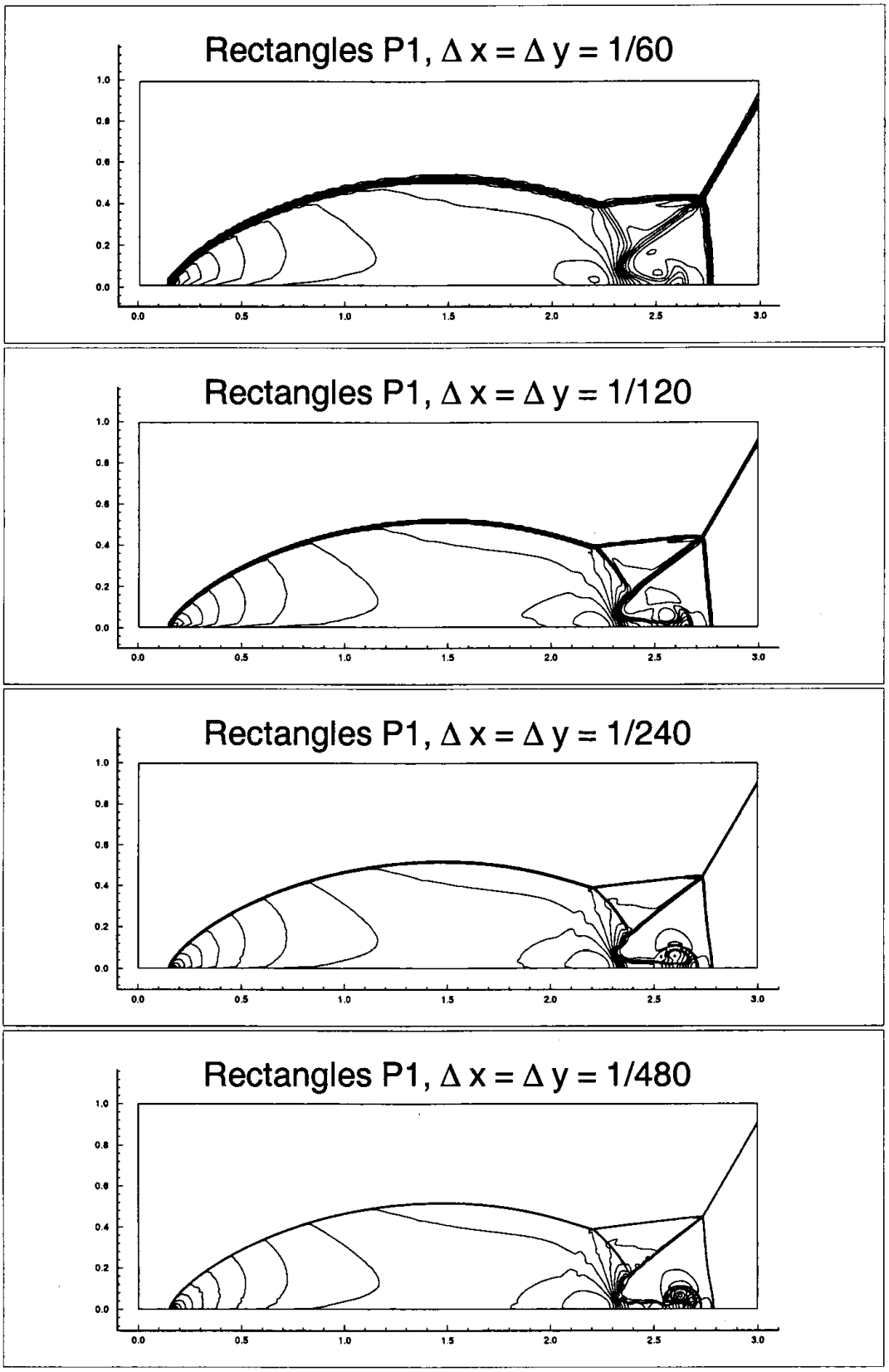


Fig. 4.2: Double Mach reflection problem. Second order P^1 results. Density ρ. 30 equally spaced contour lines from $\rho = 1.3965$ to $\rho = 22.682$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{60}, \frac{1}{120}, \frac{1}{240}$, and $\frac{1}{480}$.

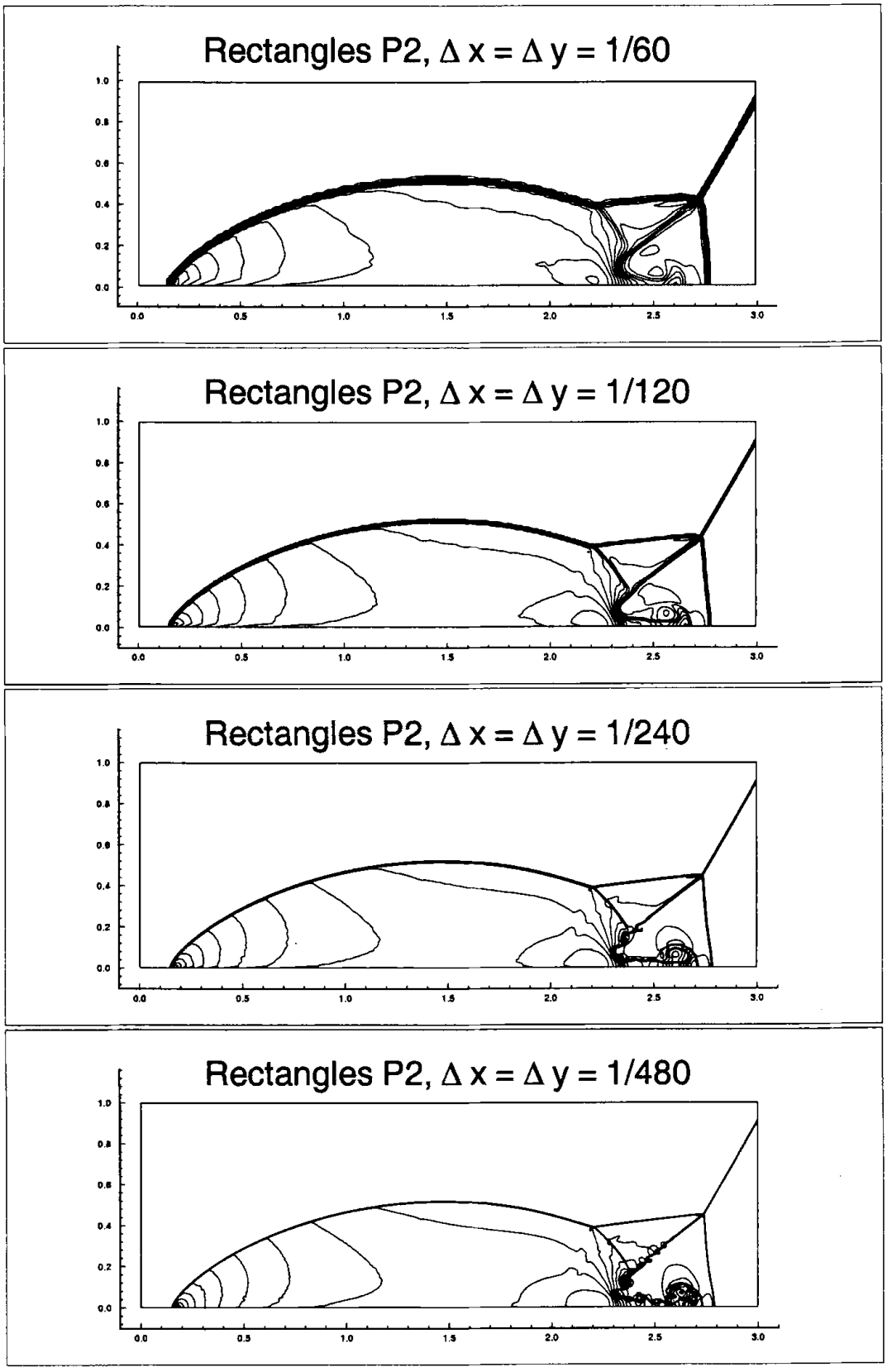


Fig. 4.3: Double Mach reflection problem. Third order P^2 results. Density ρ. 30 equally spaced contour lines from $\rho = 1.3965$ to $\rho = 22.682$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{60}, \frac{1}{120}, \frac{1}{240}$, and $\frac{1}{480}$.

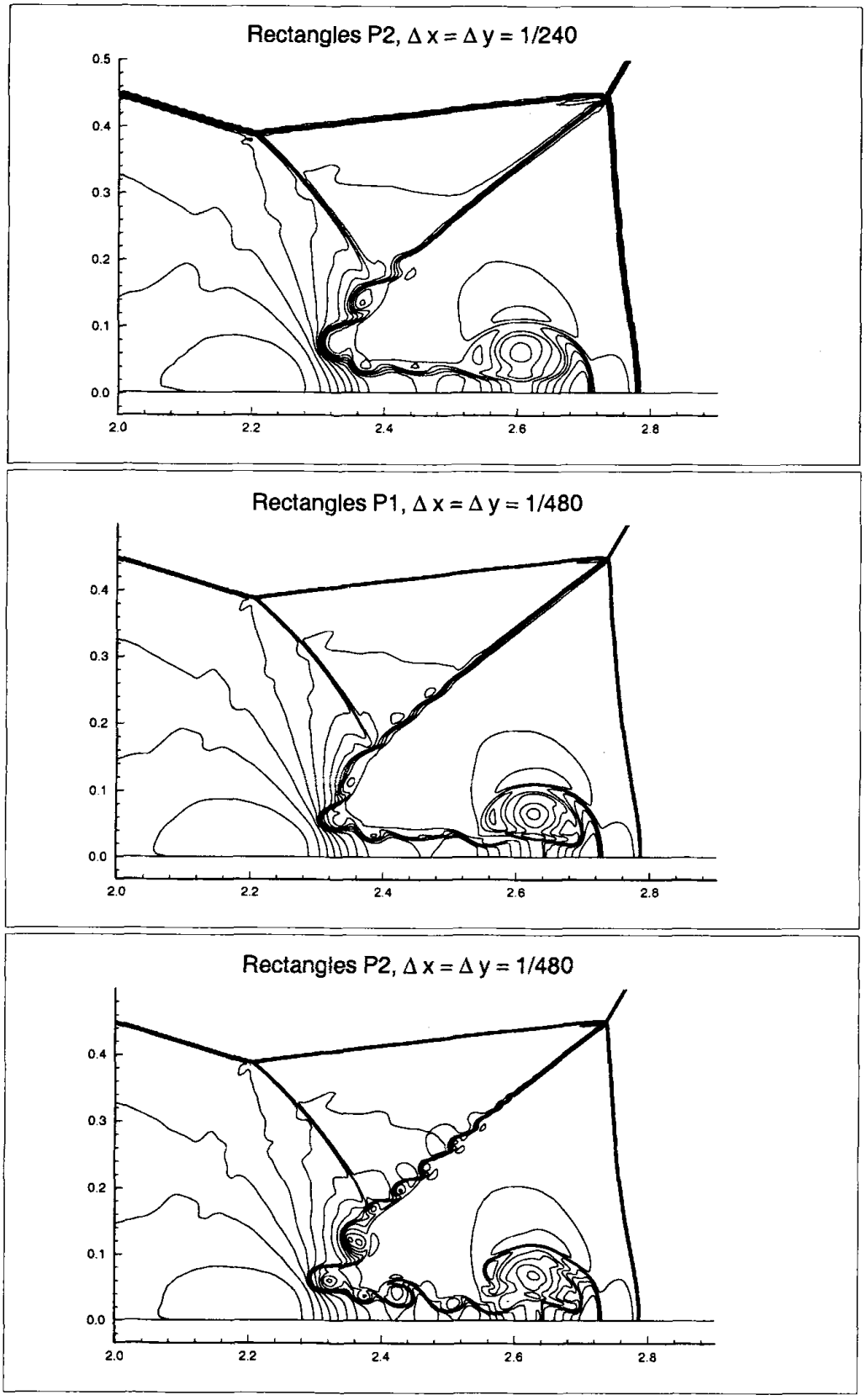


Fig. 4.4: Double Mach reflection problem. Blowed-up region around the double Mach stems. Density ρ. Third order P^2 with $\Delta x = \Delta y = \frac{1}{240}$ (top); second order P^1 with $\Delta x = \Delta y = \frac{1}{480}$ (middle); and third order P^2 with $\Delta x = \Delta y = \frac{1}{480}$ (bottom).

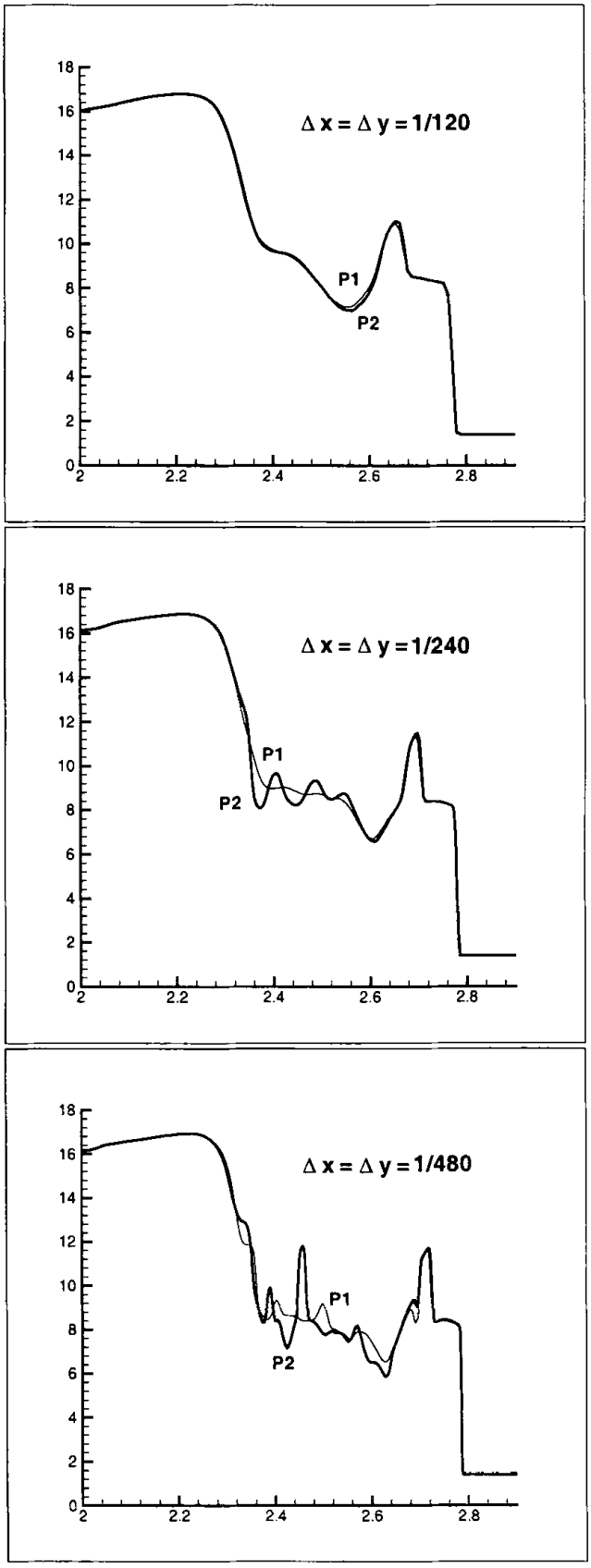


Fig. 4.5: Double Mach reflection problem. Cut $y = 0.4$ of the blowed-up region. Density ρ. Comparison of second order P^1 with third order P^2 on the same mesh

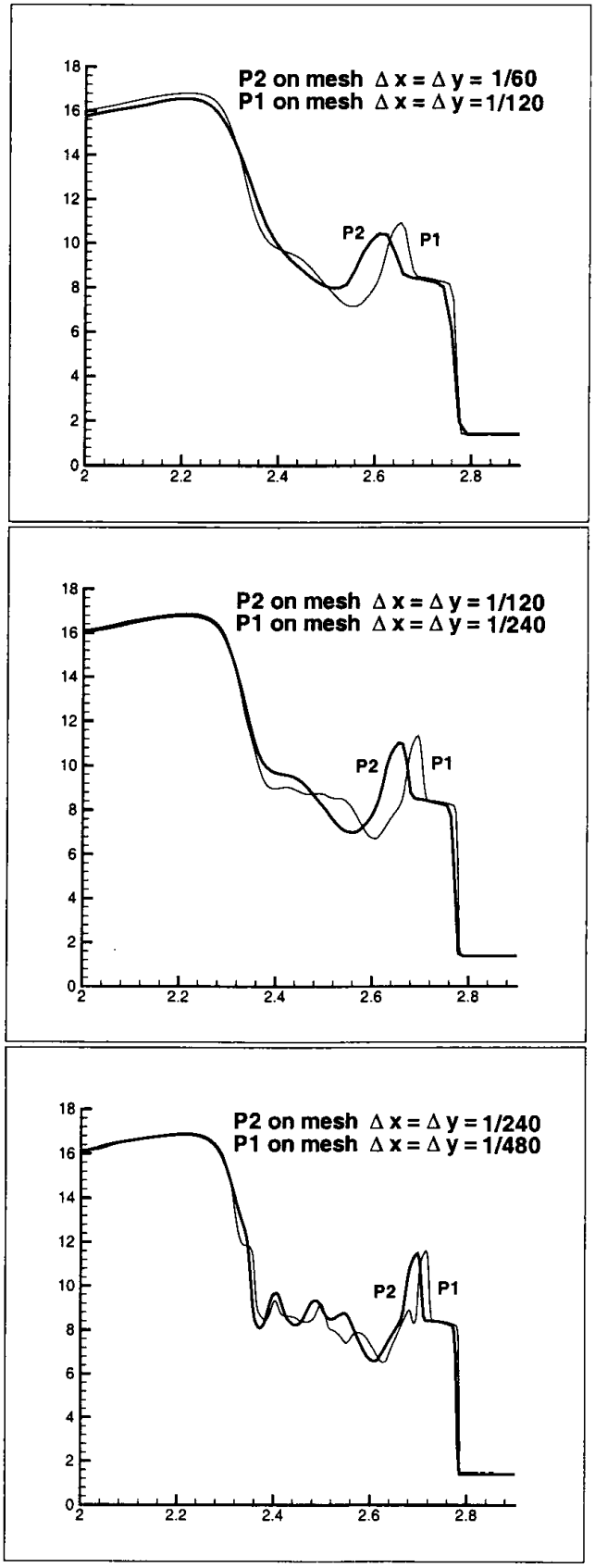


Fig. 4.6: Double Mach reflection problem. Cut $y = 0.4$ of the blowed-up region. Density ρ. Comparison of second order P^1 with third order P^2 on a coarser mesh

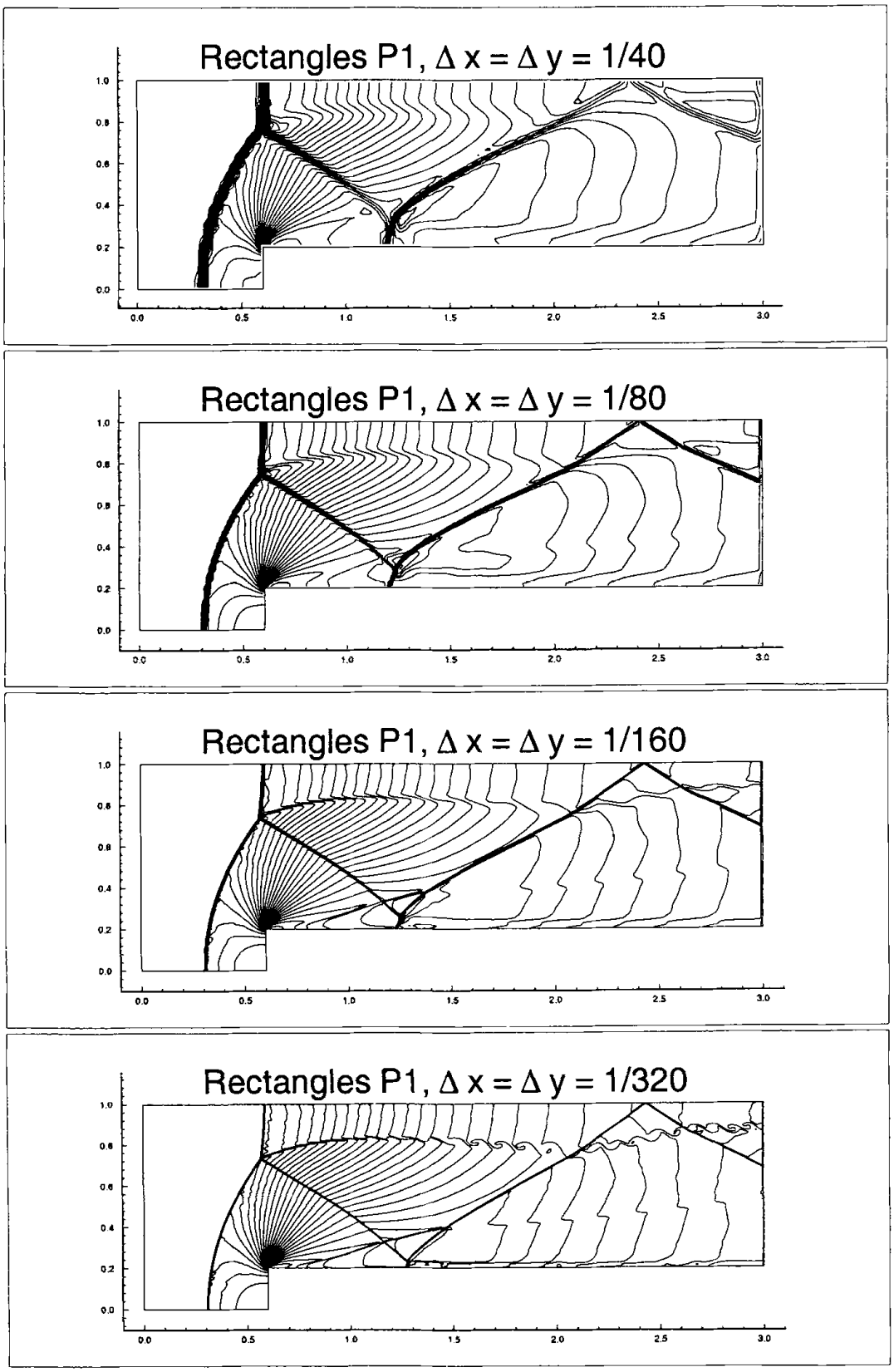


Fig. 4.7: Forward facing step problem. Second order P^1 results. Density ρ. 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$, and $\frac{1}{320}$.

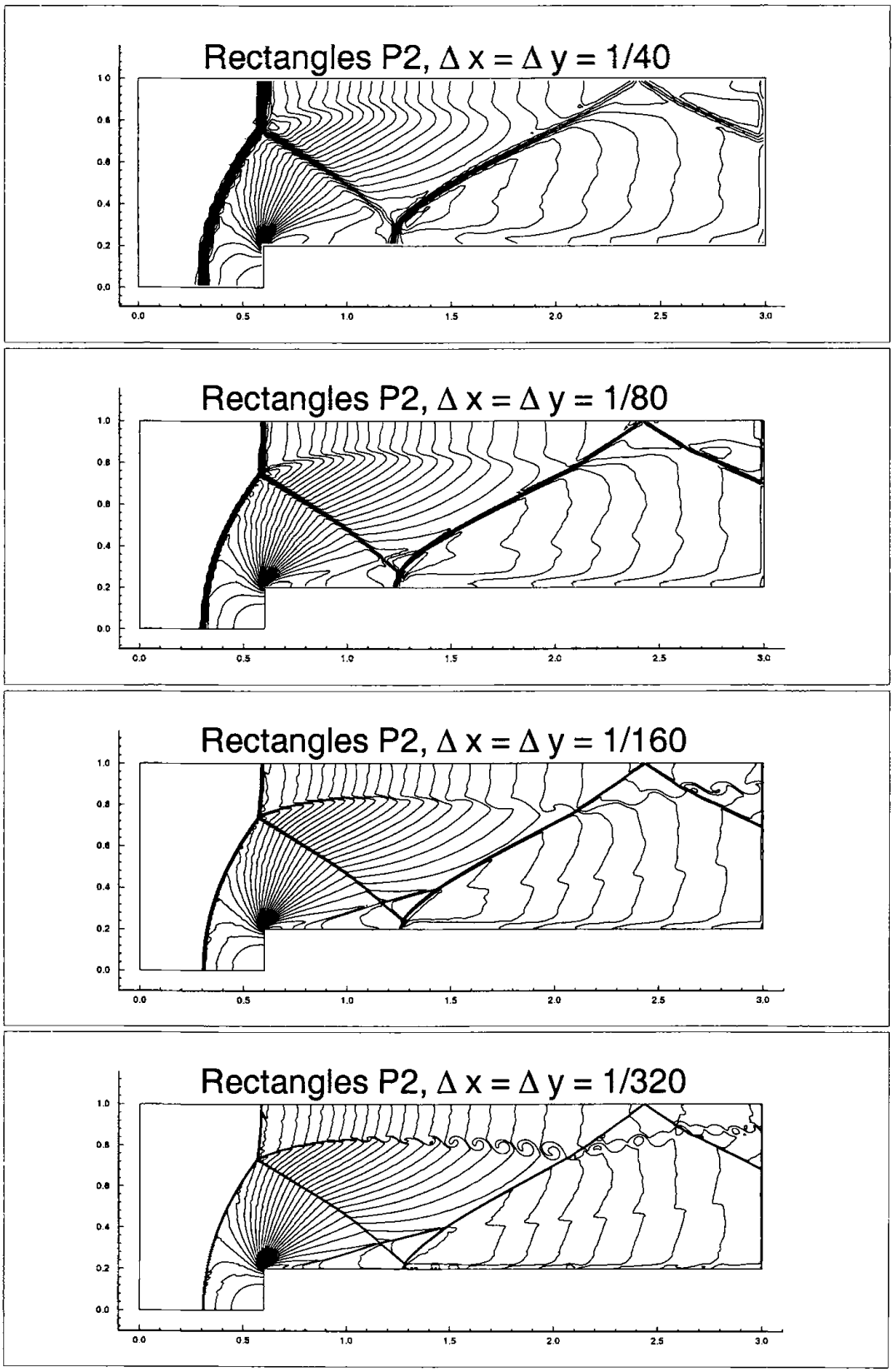


Fig. 4.8: Forward facing step problem. Third order P^2 results. Density ρ. 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$, and $\frac{1}{320}$.

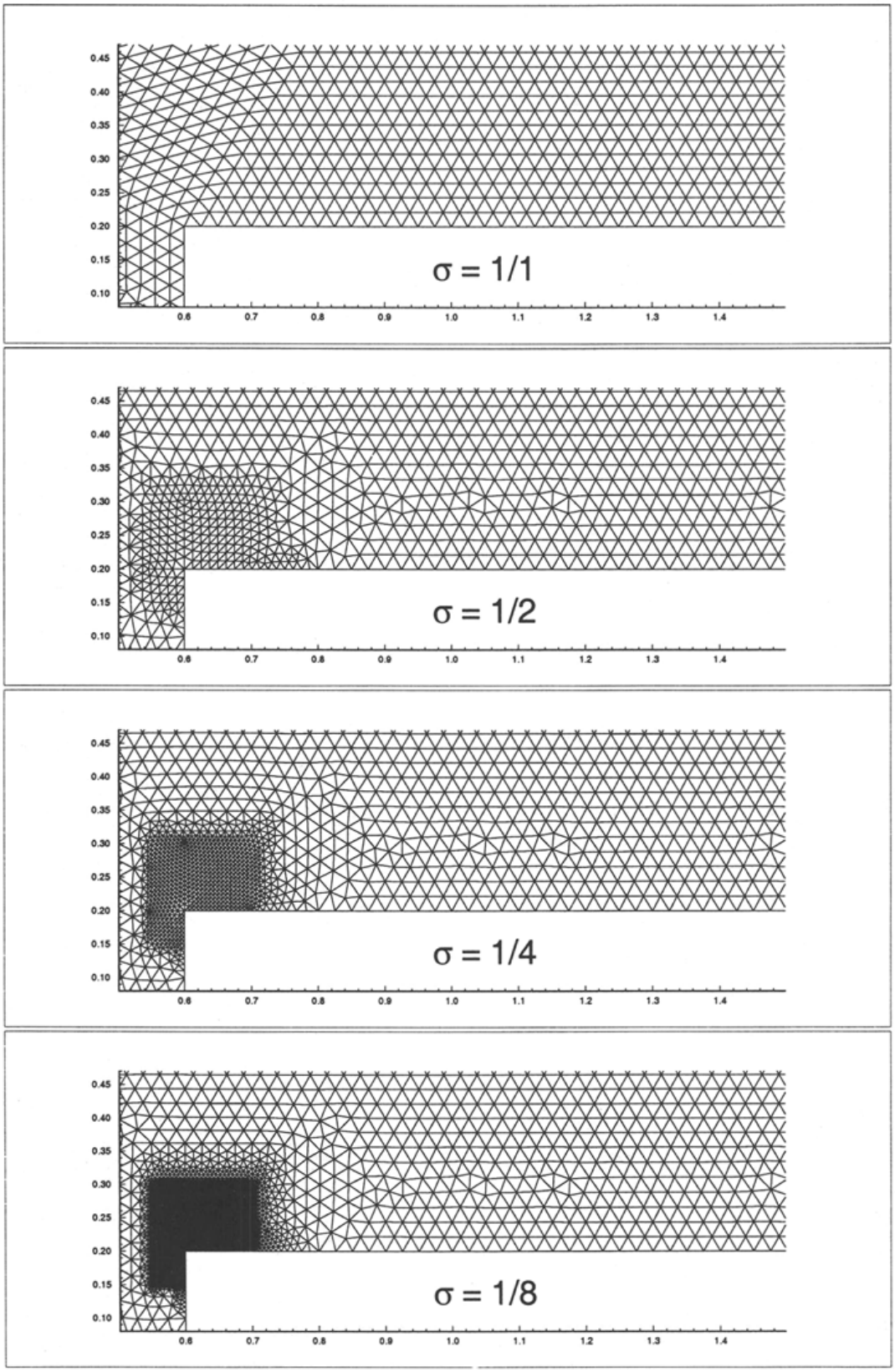


Fig. 4.9: Forward facing step problem. Detail of the triangulations associated with the different values of σ. The parameter σ is the ratio between the typical size of the triangles near the corner and that elsewhere.

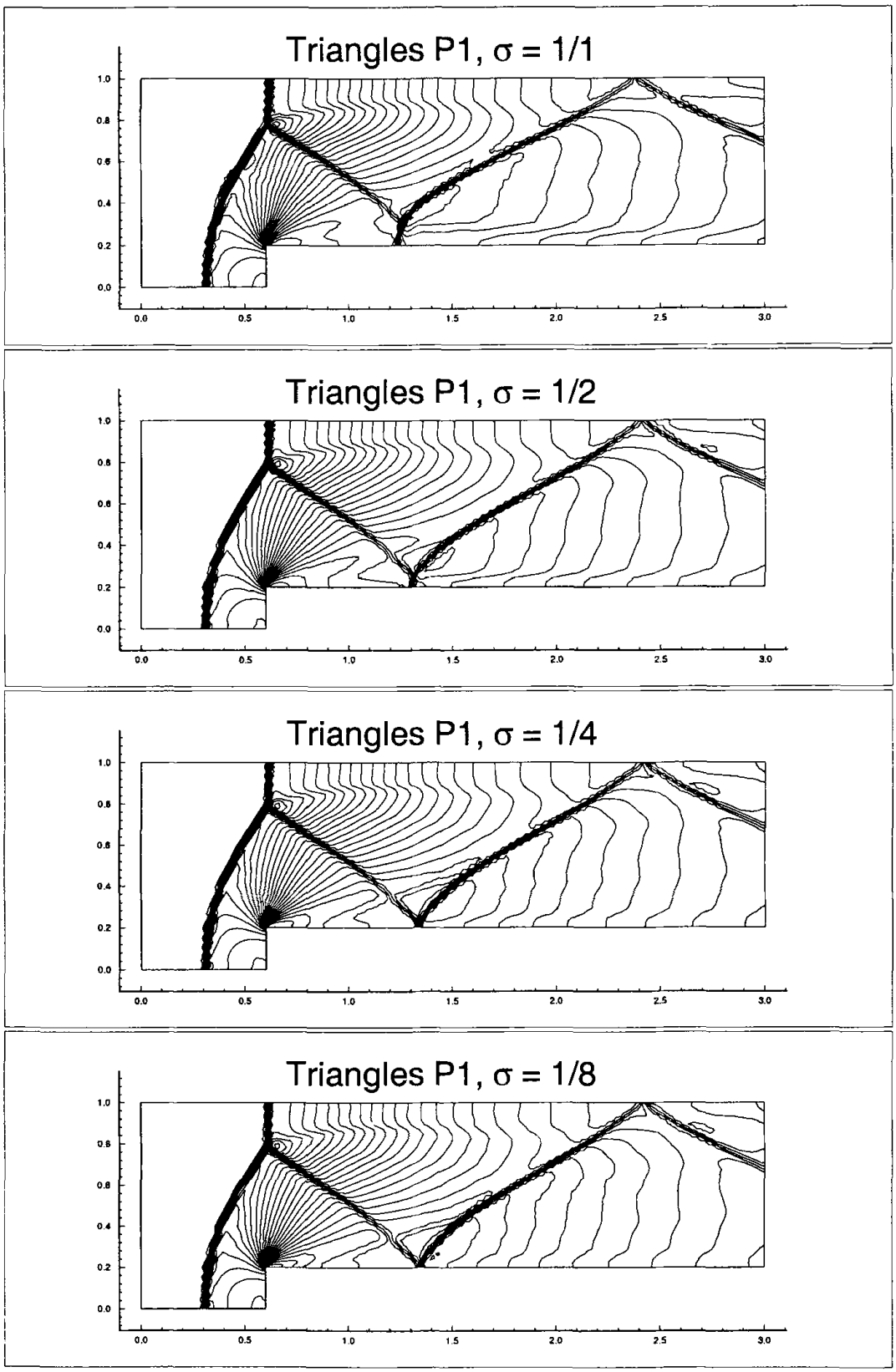


Fig. 4.10: Forward facing step problem. Second order P^1 results. Density ρ. 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Triangle code. Progressive refinement near the corner

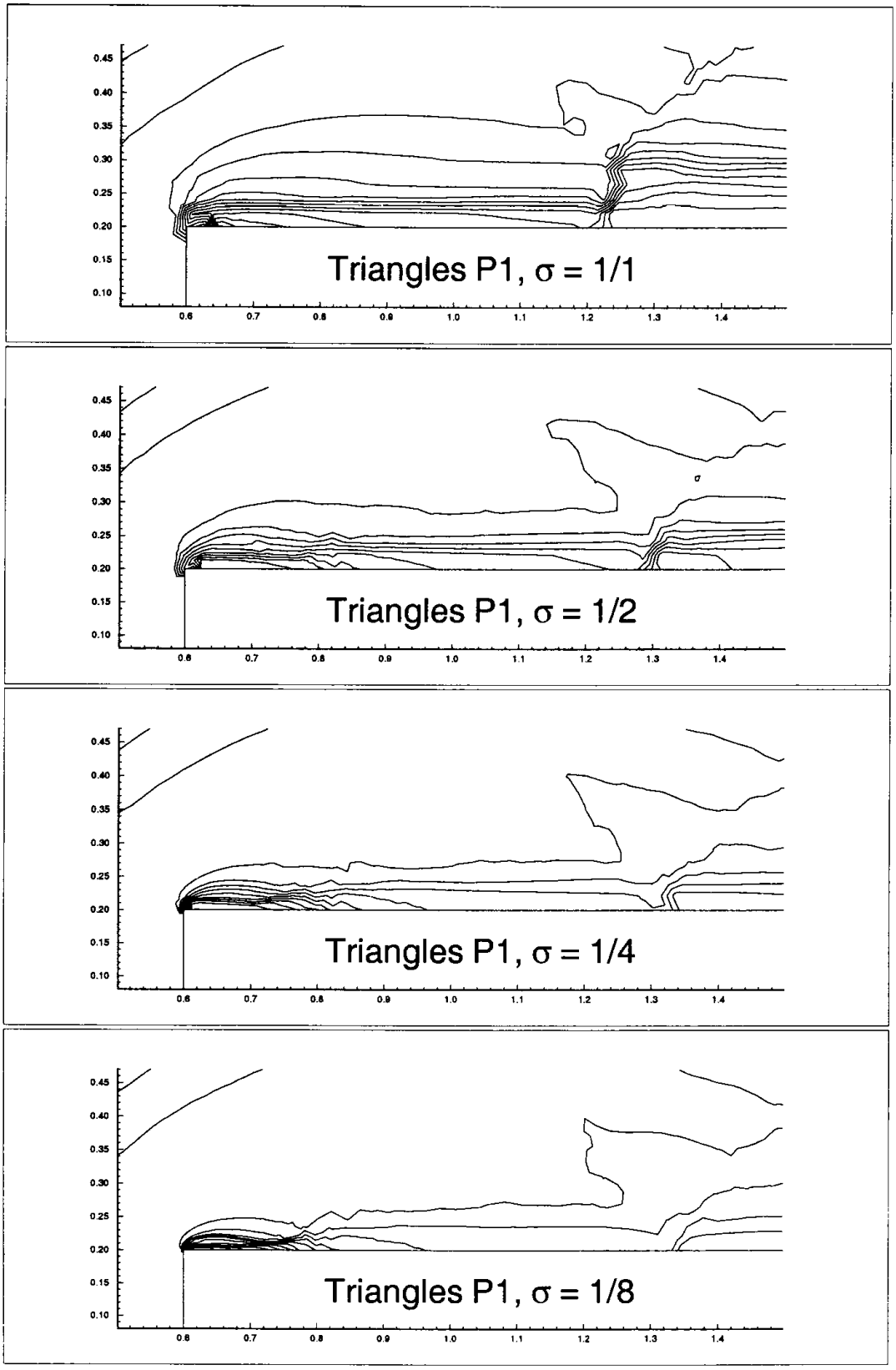


Fig. 4.11: Forward facing step problem. Second order P^1 results. Entropy level curves around the corner. Triangle code. Progressive refinement near the corner

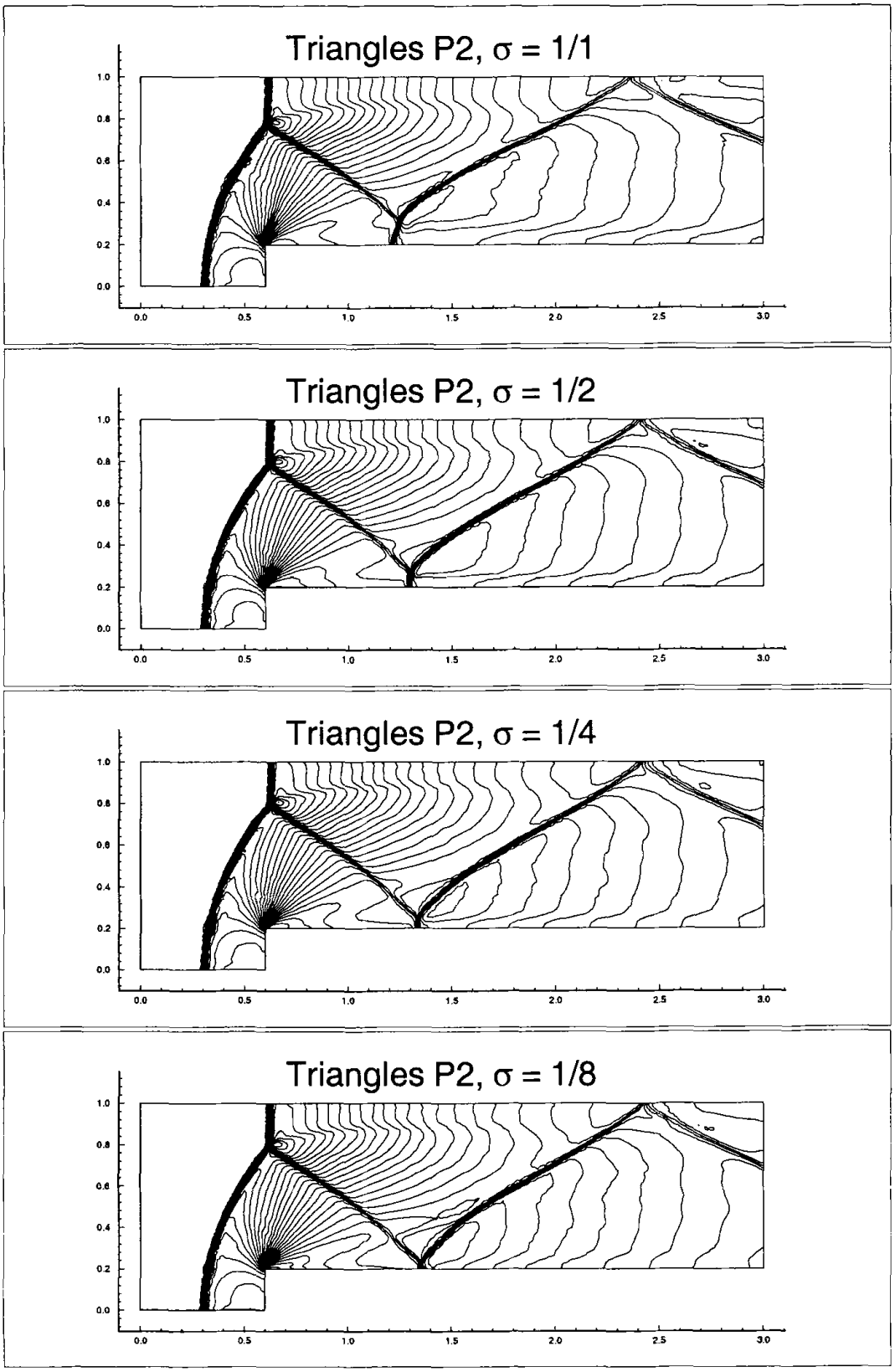


Fig. 4.12: Forward facing step problem. Third order P^2 results. Density ρ. 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Triangle code. Progressive refinement near the corner

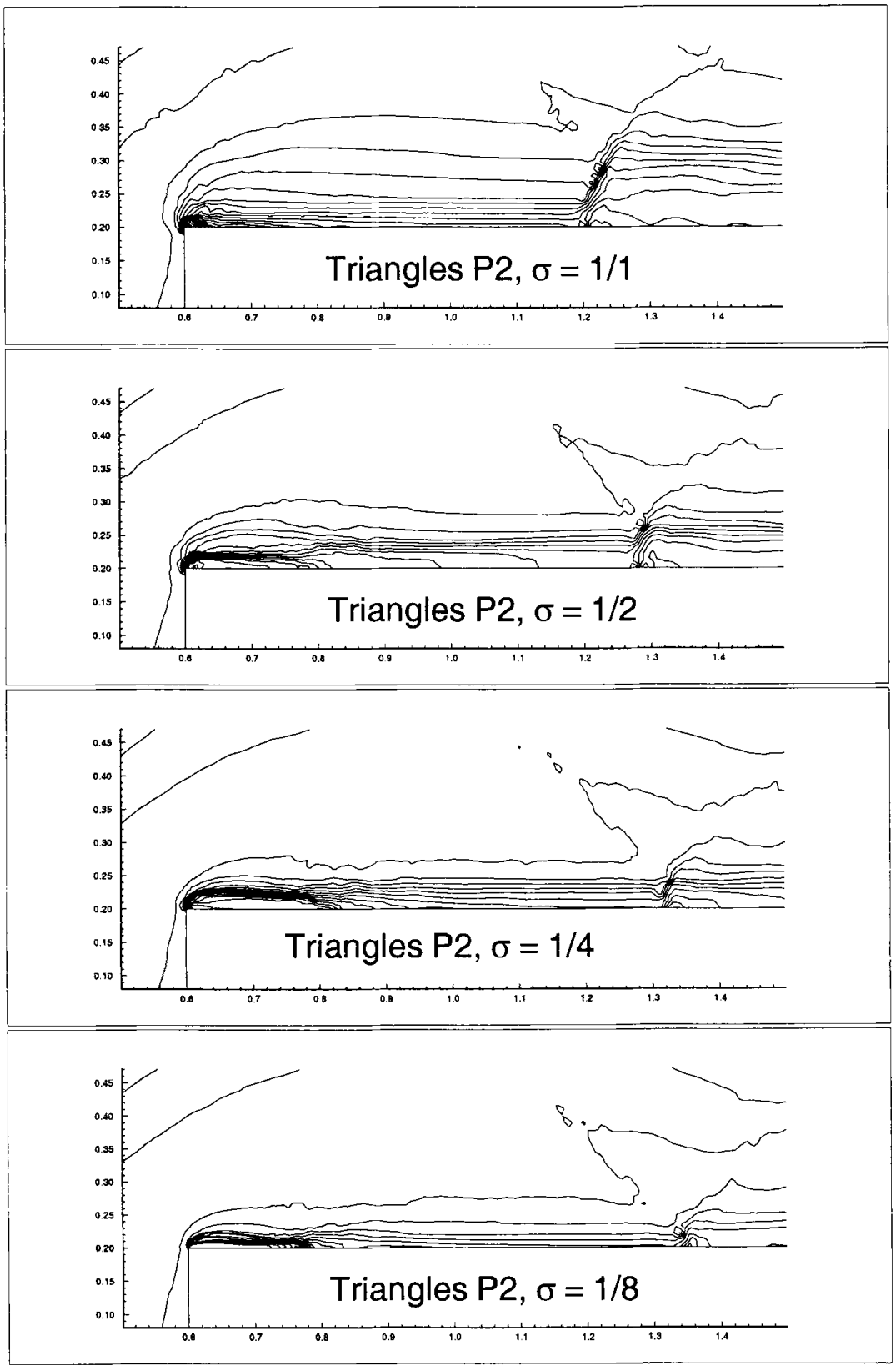


Fig. 4.13: Forward facing step problem. Third order P^1 results. Entropy level curves around the corner. Triangle code. Progressive refinement near the corner

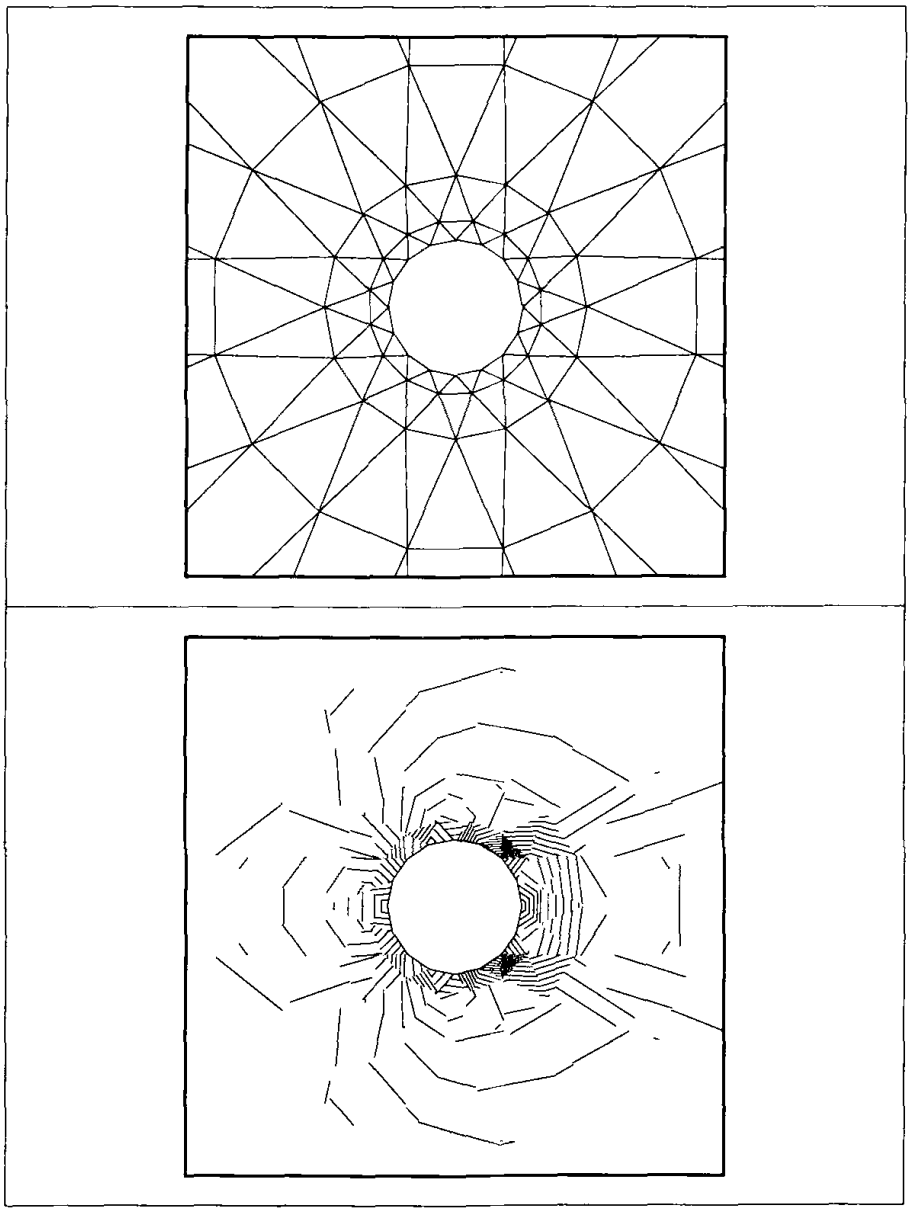


Fig. 4.14: Grid "16 × 8" with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

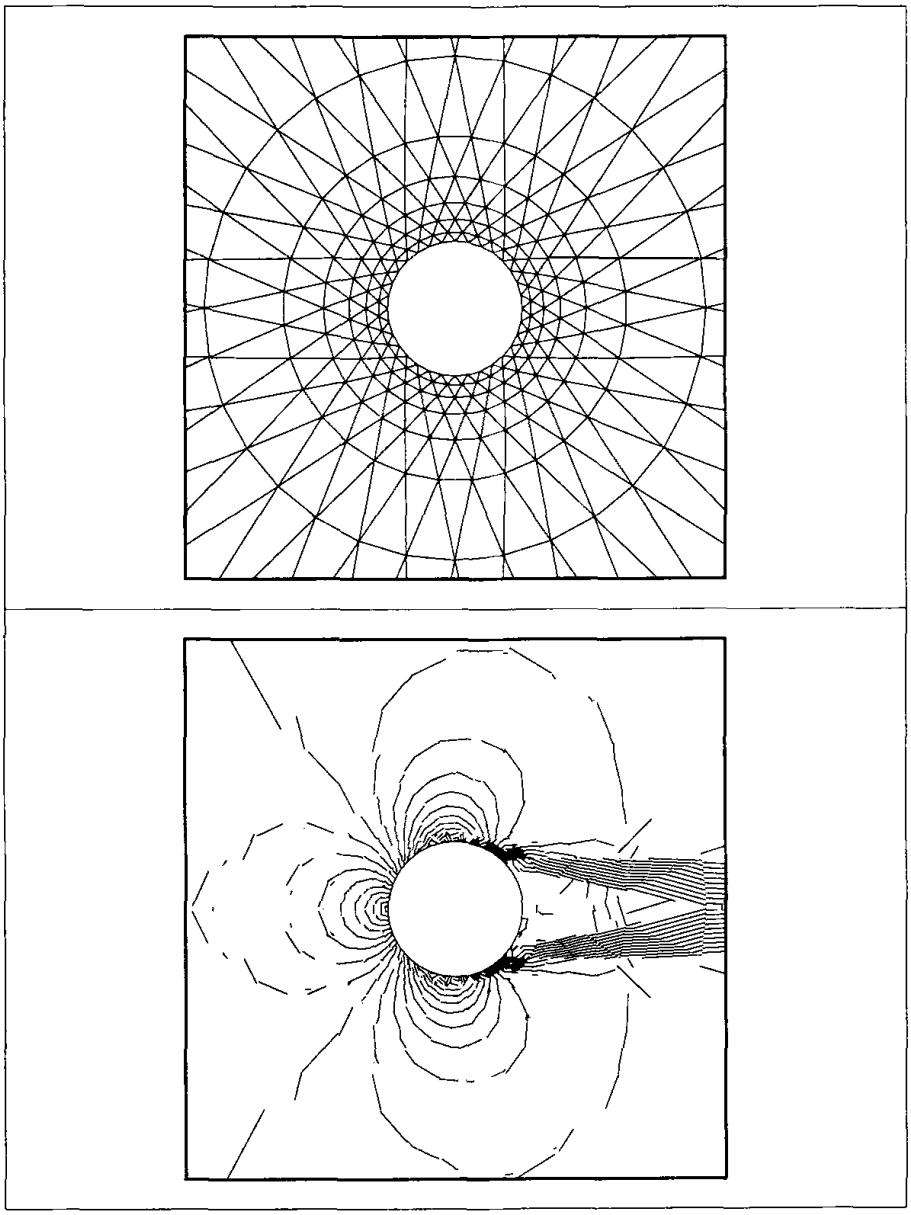


Fig. 4.15: Grid "32 × 8" with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

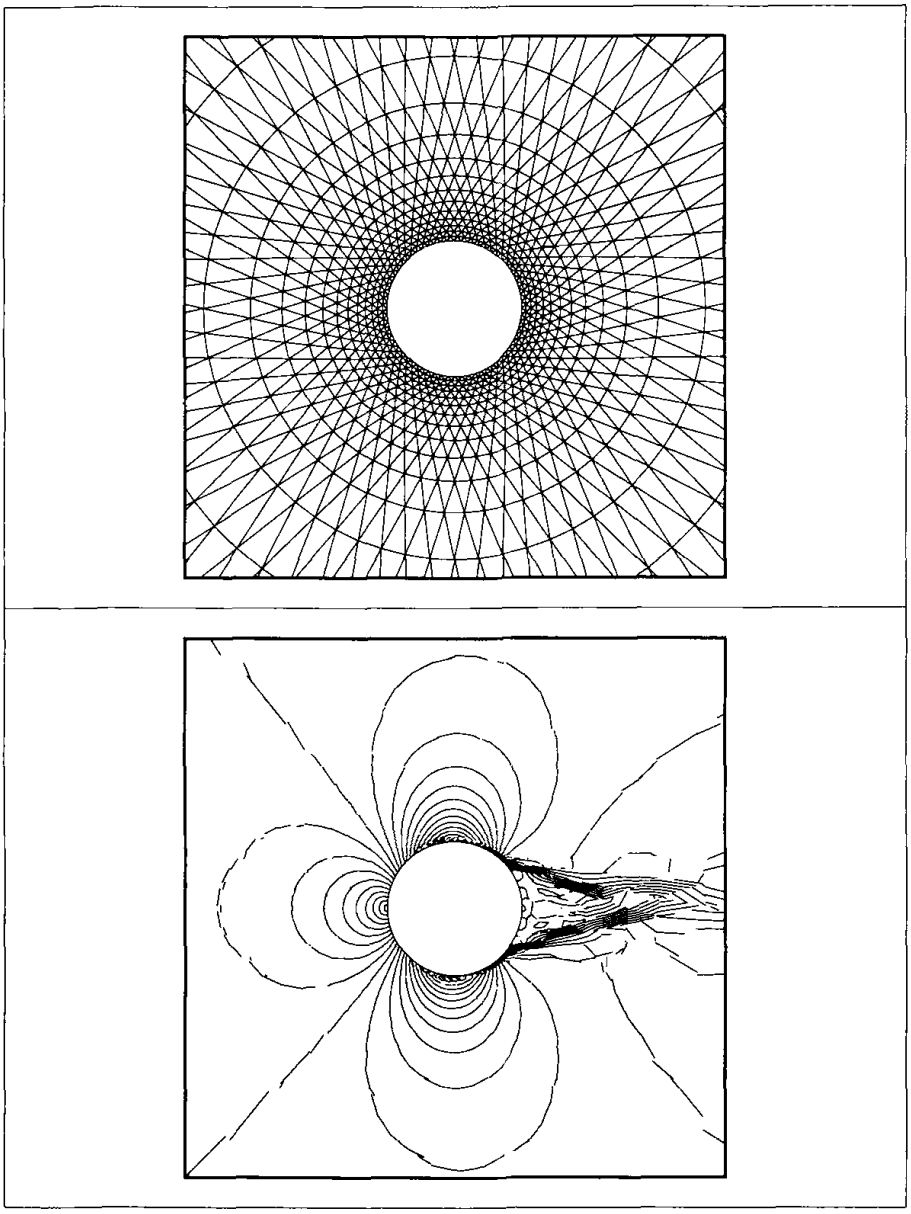


Fig. 4.16: Grid "64 × 16" with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

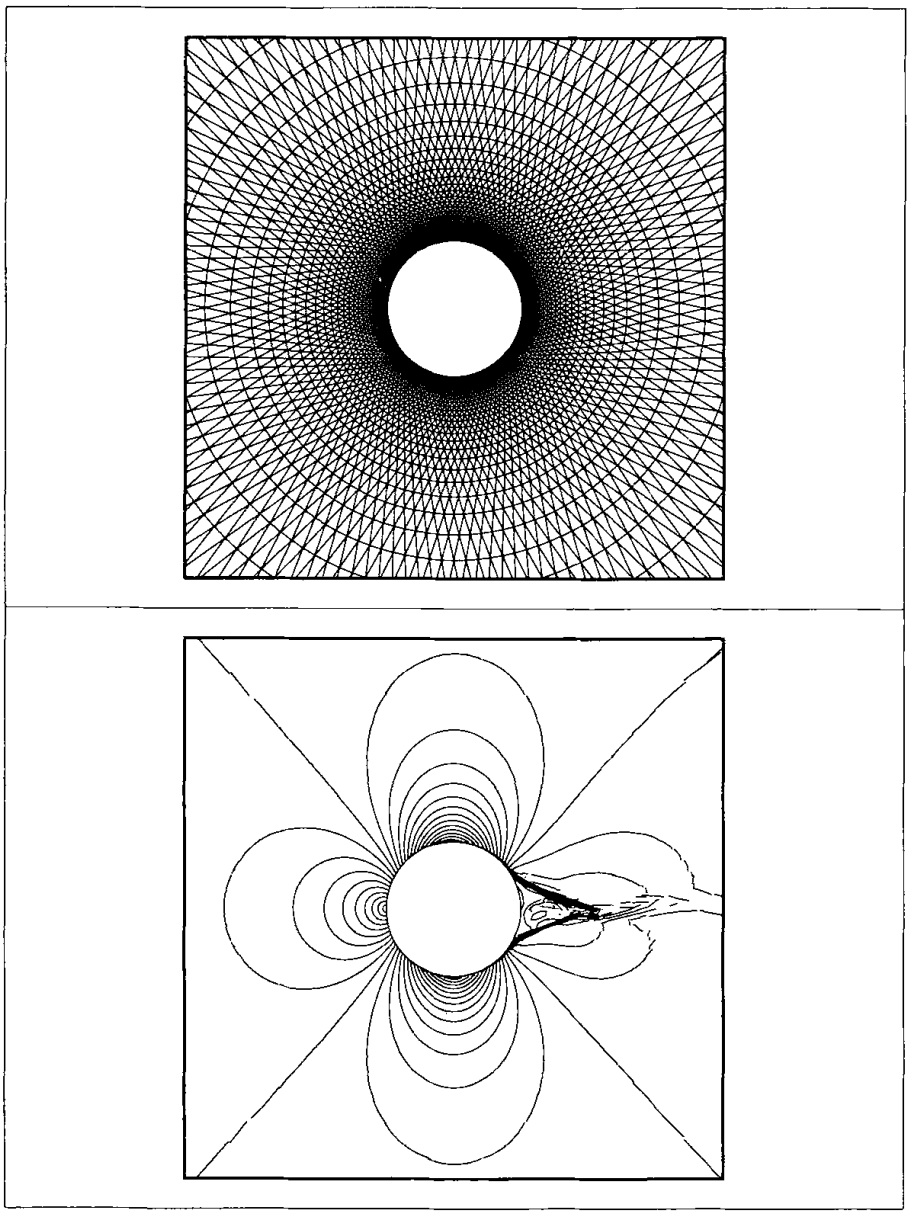


Fig. 4.17: Grid "128×32" a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

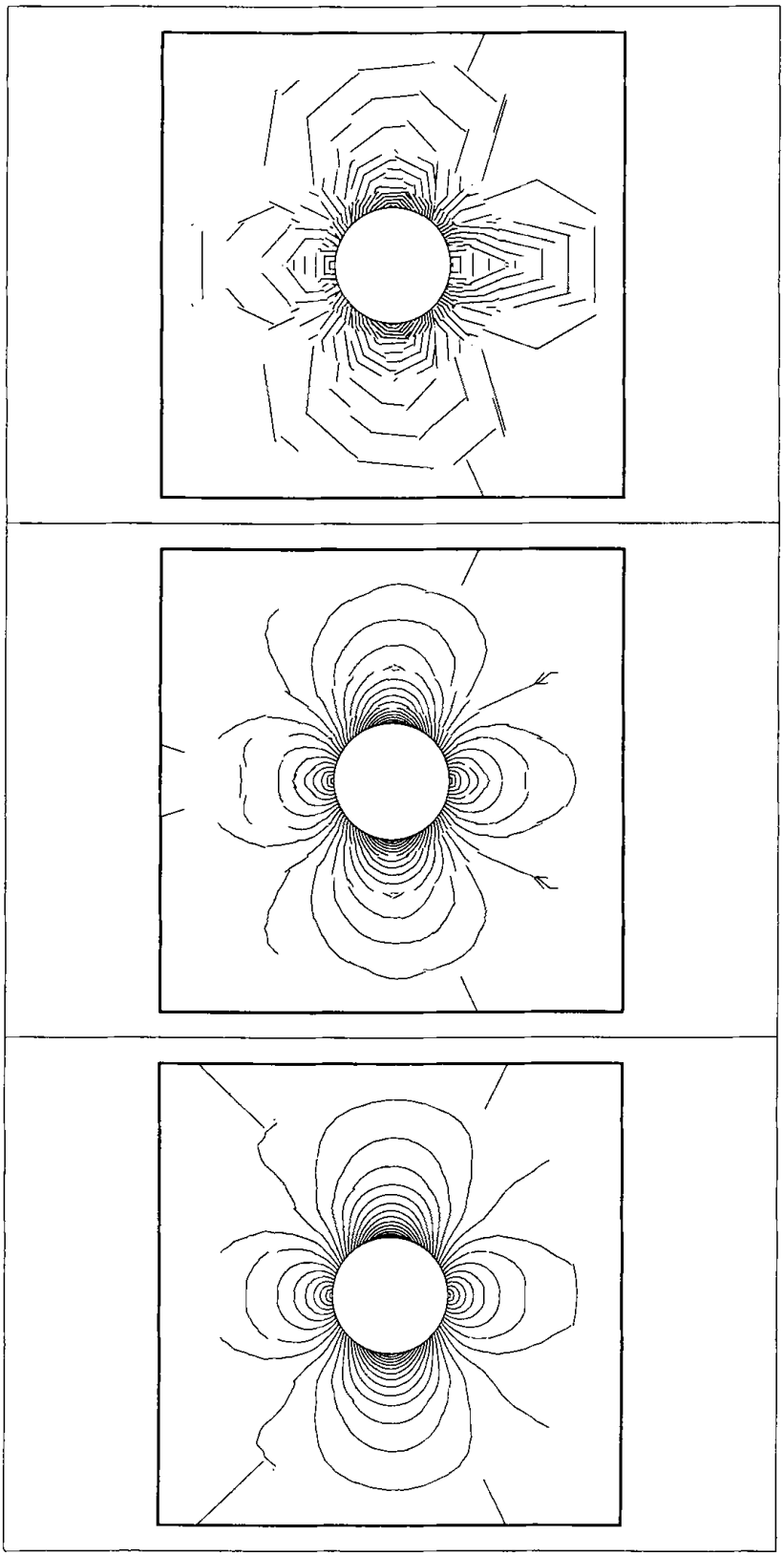


Fig. 4.18: Grid "16×4" with exact rendering of the circle and the corresponding P^1 (top), P^2(middle), and P^3 (bottom) approximations (Mach isolines).

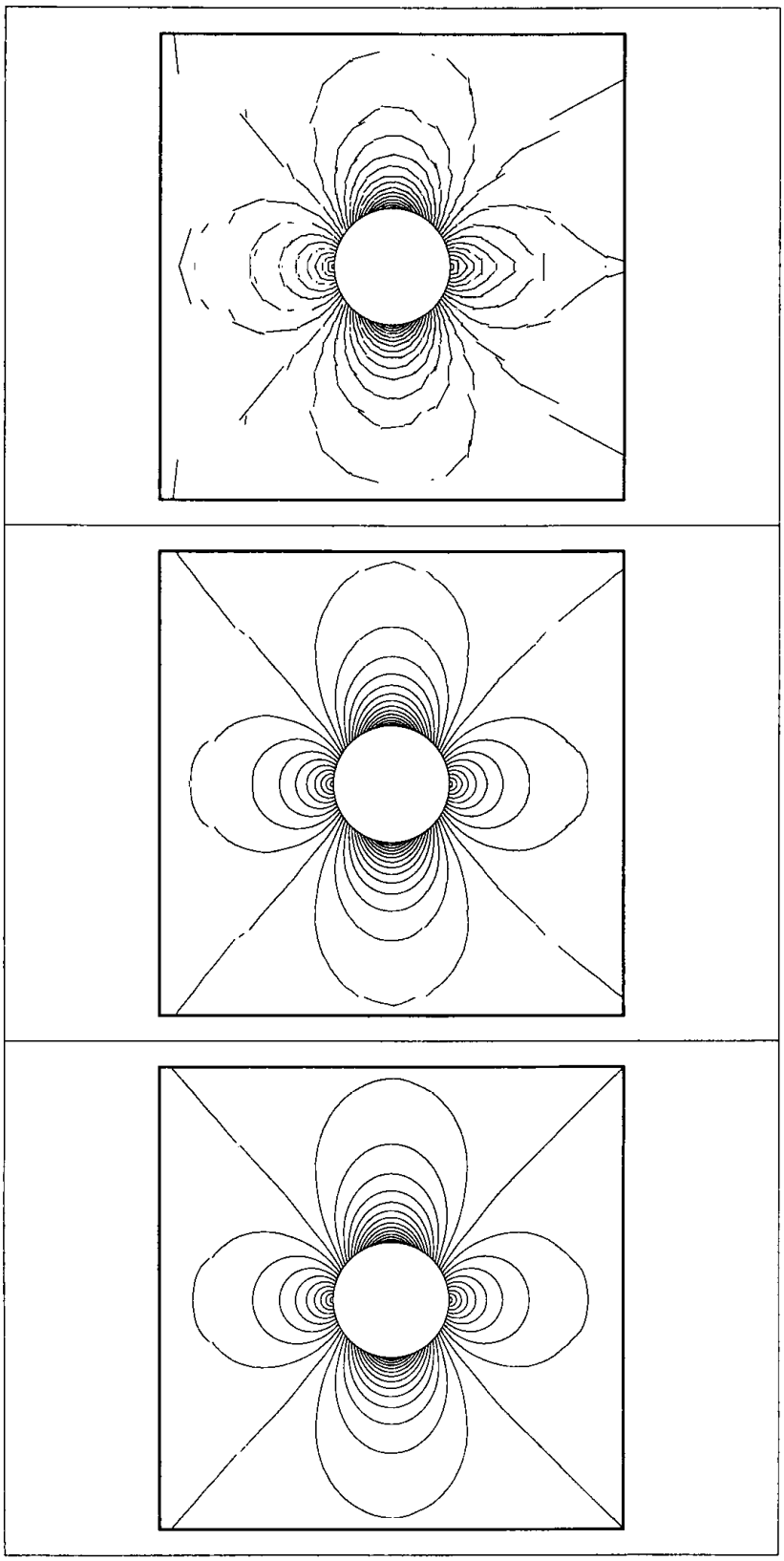


Fig. 4.19: Grid "32×8" with exact rendering of the circle and the corresponding P^1 (top), P^2(middle), and P^3 (bottom) approximations (Mach isolines).

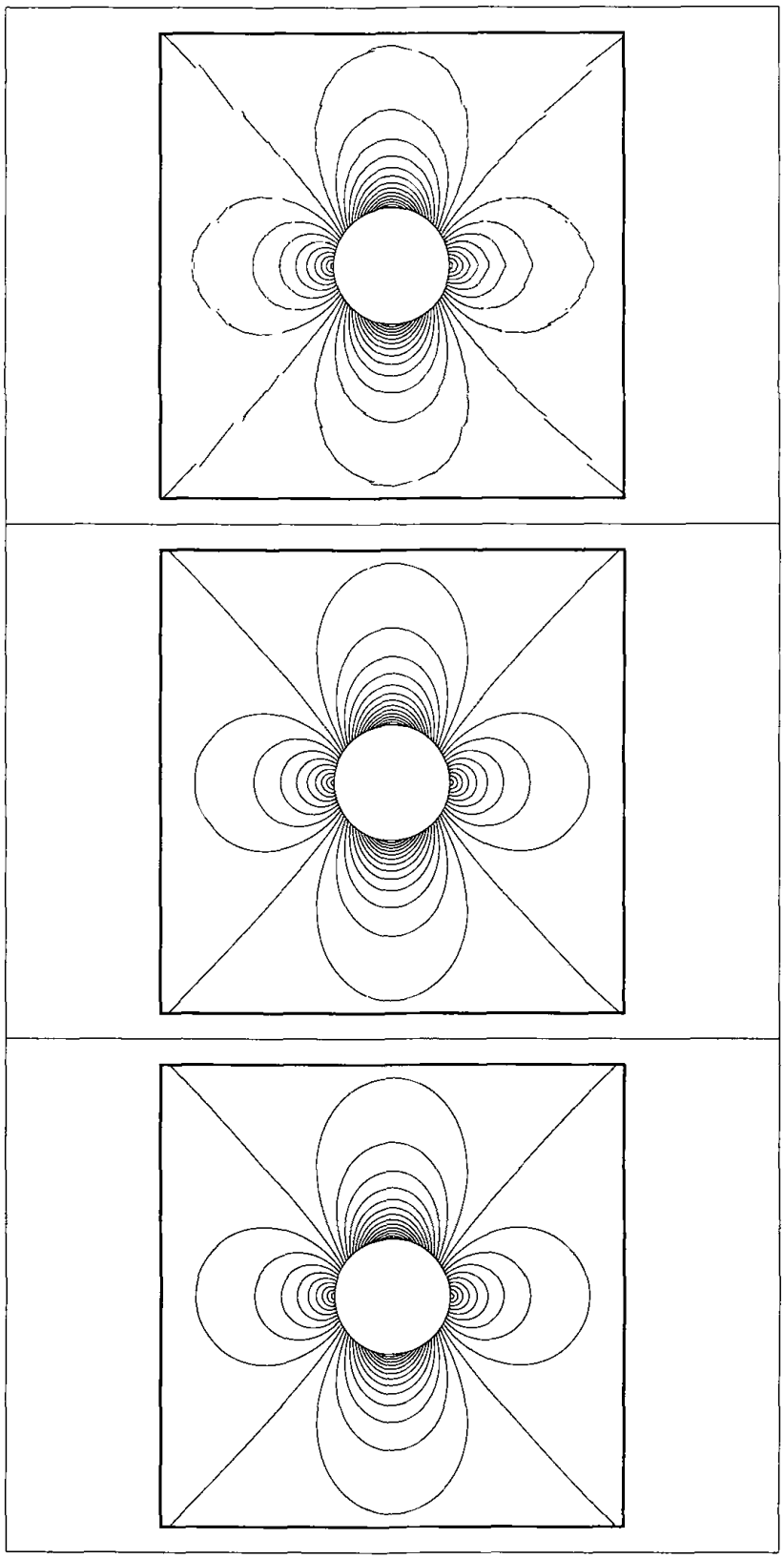


Fig. 4.20: Grid "64 × 16" with exact rendering of the circle and the corresponding P^1 (top), P^2(middle), and P^3 (bottom) approximations (Mach isolines).

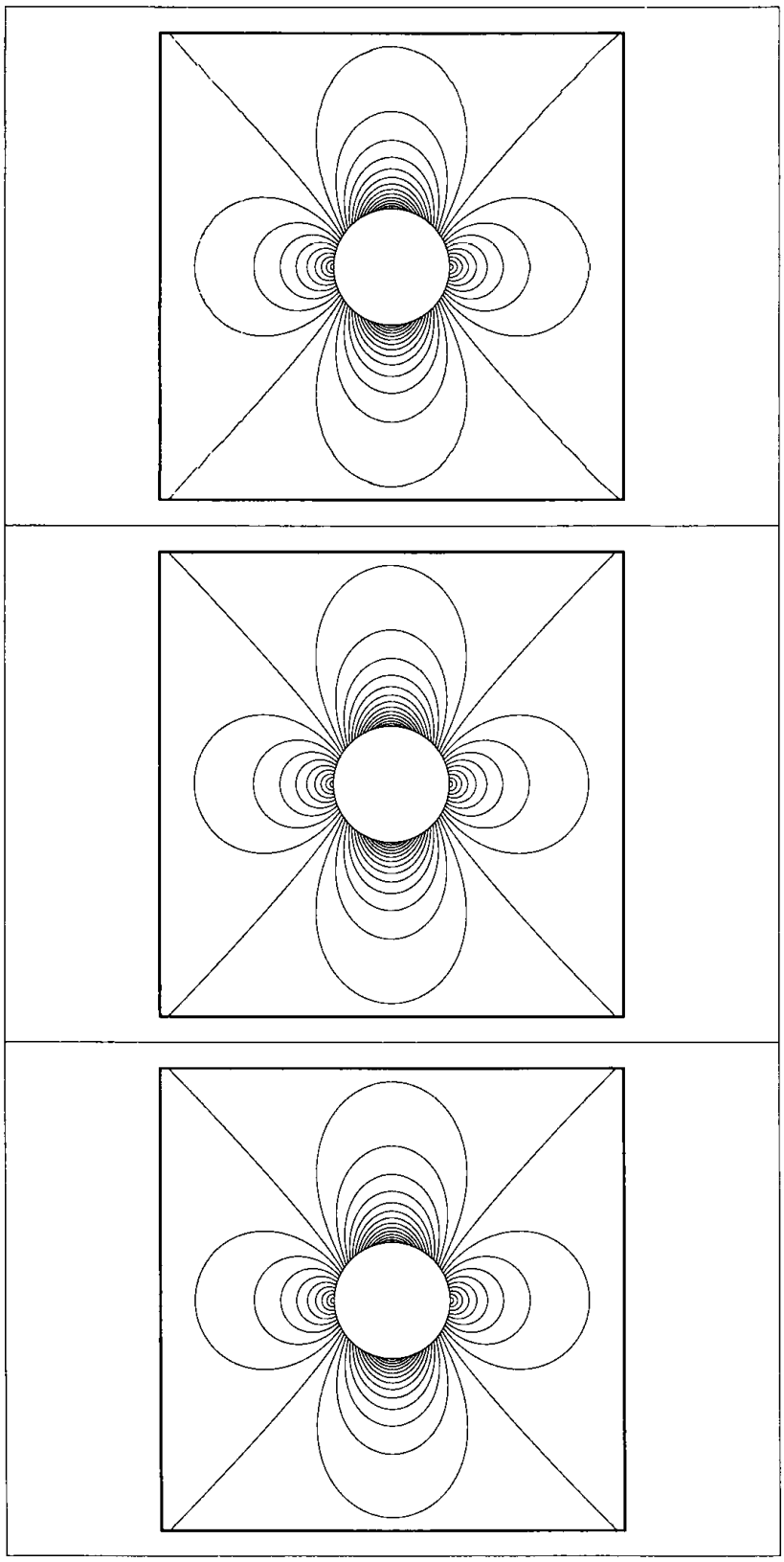


Fig. 4.21: Grid "128 × 32" with exact rendering of the circle and the corresponding P^1 (top), P^2(middle), and P^3 (bottom) approximations (Mach isolines).

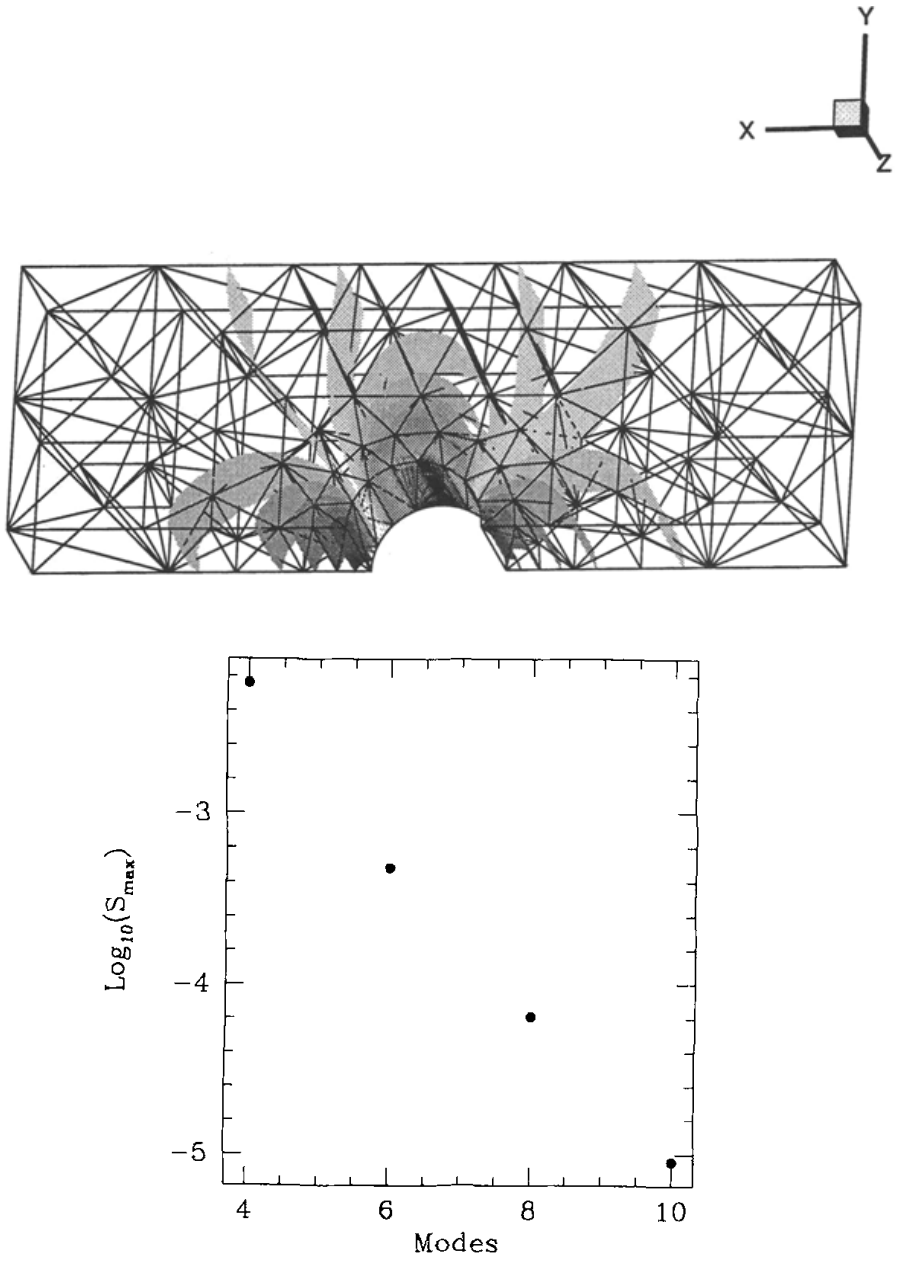


Fig. 4.22: Three-dimensional flow over a semicircular bump. Mesh and density isosurfaces (top) and history of convergence with p-refinement of the maximum entropy generated (bottom). The degree of the polynomial plus one is plotted on the 'modes' axis.

5 Convection-diffusion problems: The LDG method

5.1 Introduction

In this chapter, which follows the work by Cockburn and Shu [18], we restrict ourselves to the semidiscrete LDG methods for convection-diffusion problems with periodic boundary conditions. Our aim is to clearly display the most distinctive features of the LDG methods in a setting as simple as possible; the extension of the method to the fully discrete case is straightforward. In §2, we introduce the LDG methods for the simple one-dimensional case $d = 1$ in which

$$\mathbf{F}(u, Du) = f(u) - a(u)\,\partial_x u,$$

u is a scalar and $a(u) \geq 0$ and show some preliminary numerical results displaying the performance of the method. In this simple setting, the main ideas of how to device the method and how to analyze it can be clearly displayed in a simple way. Thus, the L^2-stability of the method is proven in the general nonlinear case and the rate of convergence of $(\Delta x)^k$ in the $\mathrm{L}^\infty(0,T;\mathrm{L}^2)$-norm for polynomials of degree $k \geq 0$ in the linear case is obtained; this estimate is sharp. In §3, we extend these results to the case in which u is a scalar and

$$\mathbf{F}_i(u, Du) = f_i(u) - \sum_{1 \leq j \leq d} a_{ij}(u)\,\partial_{x_j} u,$$

where a_{ij} defines a positive semidefinite matrix. Again, the L^2-stability of the method is proven for the general nonlinear case and the rate of convergence of $(\Delta x)^k$ in the $\mathrm{L}^\infty(0,T;\mathrm{L}^2)$-norm for polynomials of degree $k \geq 0$ and arbitrary triangulations is proven in the linear case. In this case, the multidimensionality of the problem and the arbitrariness of the grids increase the technicality of the analysis of the method which, nevertheless, uses the same ideas of the one-dimensional case. In §4, the extension of the LDG method to multidimensional systems is briefly described some numerical results for the compressible Navier-Stokes equations from the paper by Bassi and Rebay [3] and from the paper by Lomtev and Karniadakis [46] are presented.

5.2 The LDG methods for the one-dimensional case

In this section, we present and analyze the LDG methods for the following simple model problem:

$$\partial_t u + \partial_x \left(f(u) - a(u)\,\partial_x u\right) = 0 \quad \text{in } (0,T) \times (0,1), \tag{5.1}$$

$$u(t=0) = u_0, \quad \text{on } (0,1), \tag{5.2}$$

with periodic boundary conditions.

General formulation and main properties To define the LDG method, we introduce the new variable $q = \sqrt{a(u)}\,\partial_x u$ and rewrite the problem (5.1), (5.2) as follows:

$$\partial_t u + \partial_x (f(u) - \sqrt{a(u)}\, q) = 0 \quad \text{in } (0,T)\times(0,1), \tag{5.3}$$
$$q - \partial_x g(u) = 0 \quad \text{in } (0,T)\times(0,1), \tag{5.4}$$
$$u(t=0) = u_0, \quad \text{on } (0,1), \tag{5.5}$$

where $g(u) = \int^u \sqrt{a(s)}\, ds$. The LDG method for (5.1), (5.2) is now obtained by simply discretizing the above system with the Discontinuous Galerkin method.

To do that, we follow [15] and [14]. We define the flux $\mathbf{h} = (\, h_u, h_q\,)^t$ as follows:

$$\mathbf{h}(u,q) = (\, f(u) - \sqrt{a(u)}\, q\,,\ -g(u)\,)^t. \tag{5.6}$$

For each partition of the interval $(0,1)$, $\{\, x_{j+1/2}\,\}_{j=0}^N$, we set $I_j = (x_{j-1/2}, x_{j+1/2})$, and $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ for $j = 1,\ldots,N$; we denote the quantity $\max_{1\le j\le N} \Delta x_j$ by Δx . We seek an approximation $\mathbf{w}_h = (u_h, q_h)^t$ to $\mathbf{w} = (u,q)^t$ such that for each time $t \in [0,T]$, both $u_h(t)$ and $q_h(t)$ belong to the finite dimensional space

$$V_h = V_h^k = \{v \in L^1(0,1) : v|_{I_j} \in P^k(I_j),\ j = 1,\ldots,N\}, \tag{5.7}$$

where $P^k(I)$ denotes the space of polynomials in I of degree at most k. In order to determine the approximate solution (u_h, q_h), we first note that by multiplying (5.3), (5.4), and (5.5) by arbitrary, smooth functions v_u, v_q, and v_i, respectively, and integrating over I_j, we get, after a simple formal integration by parts in (5.3) and (5.4),

$$\int_{I_j} \partial_t u(x,t)\, v_u(x)\, dx - \int_{I_j} h_u(\mathbf{w}(x,t))\, \partial_x v_u(x)\, dx$$
$$+ h_u(\mathbf{w}(x_{j+1/2},t))\, v_u(x_{j+1/2}^-) - h_u(\mathbf{w}(x_{j-1/2},t))\, v_u(x_{j-1/2}^+) = 0, \tag{5.8}$$
$$\int_{I_j} q(x,t)\, v_q(x)\, dx - \int_{I_j} h_q(\mathbf{w}(x,t))\, \partial_x v_q(x)\, dx$$
$$+ h_q(\mathbf{w}(x_{j+1/2},t))\, v_q(x_{j+1/2}^-) - h_q(\mathbf{w}(x_{j-1/2},t))\, v_q(x_{j-1/2}^+) = 0, \tag{5.9}$$
$$\int_{I_j} u(x,0)\, v_i(x)\, dx = \int_{I_j} u_0(x)\, v_i(x)\, dx. \tag{5.10}$$

Next, we replace the smooth functions v_u, v_q, and v_i by test functions $v_{h,u}$, $v_{h,q}$, and $v_{h,i}$, respectively, in the finite element space V_h and the exact solution $\mathbf{w} = (u, q)^t$ by the approximate solution $\mathbf{w}_h = (u_h, q_h)^t$. Since this function is discontinuous in each of its components, we must also replace the nonlinear flux $\mathbf{h}(\mathbf{w}(x_{j+1/2}, t))$ by a numerical flux $\hat{\mathbf{h}}(\mathbf{w})_{j+1/2}(t) = (\hat{h}_u(\mathbf{w}_h)_{j+1/2}(t), \hat{h}_q(\mathbf{w}_h)_{j+1/2}(t))$ that will be suitably chosen later. Thus, the approximate solution given by the LDG method is defined as the solution of the following weak formulation:

$$\begin{aligned}
&\forall\, v_{h,u} \in P^k(I_j):\\
&\int_{I_j} \partial_t\, u_h(x,t)\, v_{h,u}(x)\, dx - \int_{I_j} h_u(\mathbf{w}_h(x,t))\, \partial_x\, v_{h,u}(x)\, dx\\
&+\hat{h}_u(\mathbf{w}_h)_{j+1/2}(t)\, v_{h,u}(x^-_{j+1/2}) - \hat{h}_u(\mathbf{w}_h)_{j-1/2}(t)\, v_{h,u}(x^+_{j-1/2}) = 0, \qquad (5.11)\\
&\forall\, v_{h,q} \in P^k(I_j):\\
&\int_{I_j} q_h(x,t)\, v_{h,q}(x)\, dx - \int_{I_j} h_q(\mathbf{w}_h(x,t))\, \partial_x\, v_{h,q}(x)\, dx\\
&+\hat{h}_q(\mathbf{w}_h)_{j+1/2}(t)\, v_{h,q}(x^-_{j+1/2}) - \hat{h}_q(\mathbf{w}_h)_{j-1/2}(t)\, v_{h,q}(x^+_{j-1/2}) = 0, \qquad (5.12)\\
&\forall\, v_{h,i} \in P^k(I_j):\\
&\int_{I_j} u_h(x,0)\, v_{h,i}(x)\, dx = \int_{I_j} u_0(x)\, v_{h,i}(x)\, dx. \qquad (5.13)
\end{aligned}$$

It only remains to choose the numerical flux $\hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}(t)$. We use the notation:

$$[p] = p^+ - p^-, \quad \text{and} \quad \overline{p} = \frac{1}{2}(p^+ + p^-), \text{and } p^{\pm}_{j+1/2} = p(x^{\pm}_{j+1/2}).$$

To be consistent with the type of numerical fluxes used in the RKDG methods, we consider numerical fluxes of the form

$$\hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}(t) \equiv \hat{\mathbf{h}}(\mathbf{w}_h(x^-_{j+1/2}, t), \mathbf{w}_h(x^+_{j+1/2}, t)),$$

that (i) are locally Lipschitz and consistent with the flux $\mathbf{h}$, (ii) allow for a local resolution of q_h in terms of u_h, (iii) reduce to an E-flux (see Osher [51]) when $a(\cdot) \equiv 0$, and that (iv) enforce the L^2-stability of the method.

To reflect the convection-diffusion nature of the problem under consideration, we write our numerical flux as the sum of a convective flux and a diffusive flux:

$$\hat{\mathbf{h}}(\mathbf{w}^-, \mathbf{w}^+) = \hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+) + \hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+). \tag{5.14}$$

The convective flux is given by

$$\hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+) = (\hat{f}(u^-, u^+), 0)^t, \tag{5.15}$$

where $\hat{f}(u^-, u^+)$ is any locally Lipschitz E-flux consistent with the nonlinearity f, and the diffusive flux is given by

$$\hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+) = \Big(-\frac{[\,g(u)\,]}{[\,u\,]}\,\overline{q},\ -\overline{g(u)}\ \Big)^t - \mathbb{C}_{diff}\,[\,\mathbf{w}\,], \tag{5.16}$$

where

$$\mathbb{C}_{diff} = \begin{pmatrix} 0 & c_{12} \\ -c_{12} & 0 \end{pmatrix}, \tag{5.17}$$

$$c_{12} = c_{12}(\mathbf{w}^-, \mathbf{w}^+) \quad \text{is locally Lipschitz,} \tag{5.18}$$

$$c_{12} \equiv 0 \quad \text{when } a(\cdot) \equiv 0. \tag{5.19}$$

We claim that this flux satisfies the properties (i) to (iv).

Let us prove our claim. That the flux $\hat{\mathbf{h}}$ is consistent with the flux $\mathbf{h}$ easily follows from their definitions. That $\hat{\mathbf{h}}$ is locally Lipschitz follows from the fact that $\hat{f}(\cdot,\cdot)$ is locally Lipschitz and from (5.17); we assume that $f(\cdot)$ and $a(\cdot)$ are locally Lipschitz functions, of course. Property (i) is hence satisfied.

That the approximate solution q_h can be resolved element by element in terms of u_h by using (5.12) follows from the fact that, by (5.16), the flux $\hat{h}_q = -\overline{g(u)} - c_{12}\,[\,u\,]$ is independent of q_h. Property (ii) is hence satisfied.

Property (iii) is also satisfied by (5.19) and by the construction of the convective flux.

To see that the property (iv) is satisfied, let us first rewrite the flux $\hat{\mathbf{h}}$ in the following way:

$$\hat{\mathbf{h}}(\mathbf{w}^-, \mathbf{w}^+) = \Big(\frac{[\,\varphi(u)\,]}{[\,u\,]} - \frac{[\,g(u)\,]}{[\,u\,]}\,\overline{q},\ -\overline{g(u)}\ \Big)^t - \mathbb{C}\,[\,\mathbf{w}\,],$$

where

$$\mathbb{C} = \begin{pmatrix} c_{11} & c_{12} \\ -c_{12} & 0 \end{pmatrix}, \quad c_{11} = \frac{1}{[u]} \left(\frac{[\varphi(u)]}{[u]} - \hat{f}(u^-, u^+) \right). \tag{5.20}$$

with $\varphi(u)$ defined by $\varphi(u) = \int^u f(s)\, ds$. Since $\hat{f}(\cdot, \cdot)$ is an E-flux,

$$c_{11} = \frac{1}{[u]^2} \int_{u^-}^{u^+} \left(f(s) - \hat{f}(u^-, u^+) \right) ds \geq 0,$$

and so, by (5.17), the matrix $\mathbb{C}$ is semipositive definite. The property (iv) follows from this fact and from the following result.

Theorem 5.1 *We have,*

$$\tfrac{1}{2} \int_0^1 u_h^2(x,T)\, dx + \int_0^T \int_0^1 q_h^2(x,t)\, dx\, dt + \Theta_{T,\mathbb{C}}([\mathbf{w}_h]) \leq \tfrac{1}{2} \int_0^1 u_0^2(x)\, dx,$$

where

$$\Theta_{T,\mathbb{C}}([\mathbf{w}_h]) = \int_0^T \sum_{1 \leq j \leq N} \left\{ [\mathbf{w}_h(t)]^t \mathbb{C}\, [\mathbf{w}_h(t)] \right\}_{j+1/2} dt.$$

For a proof, see [18]. Thus, this shows that the flux $\hat{\mathbf{h}}$ under consideration does satisfy the properties (i) to (iv)- as claimed.

Now, we turn to the question of the quality of the approximate solution defined by the LDG method. In the linear case $f' \equiv c$ and $a(\cdot) \equiv a$, from the above stability result and from the the approximation properties of the finite element space V_h, we can prove the following error estimate. We denote the $\mathrm{L}^2(0,1)$-norm of the ℓ-th derivative of u by $|\, u \,|_\ell$.

Theorem 5.2 *Let* $\mathbf{e}$ *be the approximation error* $\mathbf{w} - \mathbf{w}_h$. *Then we have,*

$$\left\{ \int_0^1 |\, e_u(x,T) \,|^2\, dx + \int_0^T \int_0^1 |\, e_q(x,t) \,|^2\, dx\, dt + \Theta_{T,\mathbb{C}}([\mathbf{e}]) \right\}^{1/2} \leq C\, (\Delta x)^k,$$

where $C = C(k, |\, u \,|_{k+1}, |\, u \,|_{k+2})$. *In the purely hyperbolic case* $a = 0$, *the constant* C *is of order* $(\Delta x)^{1/2}$. *In the purely parabolic case* $c = 0$, *the constant* C *is of order* Δx *for even values of* k *for uniform grids and for* $\mathbb{C}$ *identically zero.*

For a proof, see [18]. The above error estimate gives a suboptimal order of convergence, but it is sharp for the LDG methods. Indeed, Bassi *et al* [4]

report an order of convergence of order $k+1$ for even values of k and of order k for odd values of k for a steady state, purely elliptic problem for uniform grids and for $\mathbb{C}$ identically zero. The numerical results for a purely parabolic problem that will be displayed later lead to the same conclusions; see Table 5 in the section §2.b.

The error estimate is also sharp in that the optimal order of convergence of $k+1/2$ is recovered in the purely hyperbolic case, as expected. This improvement of the order of convergence is a reflection of the *semipositive definiteness* of the matrix $\mathbb{C}$, which enhances the stability properties of the LDG method. Indeed, since in the purely hyperbolic case

$$\Theta_{T,\mathbb{C}}([\mathbf{w}_h]) = \int_0^T \sum_{1\le j\le N} \left\{ [u_h(t)]^t \, c_{11} \, [u_h(t)] \right\}_{j+1/2} dt,$$

the method enforces a control of the jumps of the variable u_h, as shown in Proposition lemenergy. This additional control is reflected in the improvement of the order of accuracy from k in the general case to $k+1/2$ in the purely hyperbolic case.

However, this can only happen in the purely hyperbolic case for the LDG methods. Indeed, since $c_{11}=0$ for $c=0$, the control of the jumps of u_h is not enforced in the purely parabolic case. As indicated by the numerical experiments of Bassi *et al.* [4] and those of section §2.b below, this can result in the effective degradation of the order of convergence. To remedy this situation, the control of the jumps of u_h in the purely parabolic case can be easily enforced by letting c_{11} be strictly positive if $|c|+|a|>0$. Unfortunately, this is not enough to guarantee an improvement of the accuracy: an additional control on the jumps of q_h is required! This can be easily achieved by allowing the matrix $\mathbb{C}$ to be *symmetric and positive definite* when $a>0$. In this case, the order of convergence of $k+1/2$ can be easily obtained for the general convection-diffusion case. However, this would force the matrix entry c_{22} to be nonzero and the property (ii) of local resolvability of q_h in terms of u_h would not be satisfied anymore. As a consequence, the high parallelizability of the LDG would be lost.

The above result shows how strongly the order of convergence of the LDG methods depend on the choice of the matrix $\mathbb{C}$. In fact, the numerical results of section §2.b in uniform grids indicate that with yet another choice of the matrix $\mathbb{C}$, see (5.21), the LDG method converges with the optimal order of $k+1$ in the general case. The analysis of this phenomenon constitutes the subject of ongoing work.

5.3 Numerical results in the one-dimensional case

In this section we present some numerical results for the schemes discussed in this paper. We will only provide results for the following one dimensional,

linear convection diffusion equation

$$\partial_t u + c\,\partial_x u - a\,\partial_x^2 u = 0 \quad \text{in } (0,T)\times(0,2\pi),$$
$$u(t=0,x) = \sin(x), \quad \text{on } (0,2\pi),$$

where c and $a \geq 0$ are both constants; periodic boundary conditions are used. The exact solution is $u(t,x) = e^{-at}\sin(x-ct)$. We compute the solution up to $T = 2$, and use the LDG method with $\mathbb{C}$ defined by

$$\mathbb{C} = \begin{pmatrix} \frac{|c|}{2} & -\frac{\sqrt{a}}{2} \\ \frac{\sqrt{a}}{2} & 0 \end{pmatrix}. \tag{5.21}$$

We notice that, for this choice of fluxes, the approximation to the convective term cu_x is the standard upwinding, and that the approximation to the diffusion term $a\,\partial_x^2 u$ is the standard three point central difference, for the P^0 case. On the other hand, if one uses a central flux corresponding to $c_{12} = -c_{21} = 0$, one gets a spread-out five point central difference approximation to the diffusion term $a\,\partial_x^2 u$.

The LDG methods based on P^k, with $k = 1,2,3,4$ are tested. Elements with equal size are used. Time discretization is by the third-order accurate TVD Runge-Kutta method [58], with a sufficiently small time step so that error in time is negligible comparing with spatial errors. We list the L_∞ errors and numerical orders of accuracy, for u_h, as well as for its derivatives suitably scaled $\Delta x^m \partial_x^m u_h$ for $1 \leq m \leq k$, at the center of of each element. This gives the complete description of the error for u_h over the whole domain, as u_h in each element is a polynomial of degree k. We also list the L_∞ errors and numerical orders of accuracy for q_h at the element center.

In all the convection-diffusion runs with $a > 0$, accuracy of at least $(k+1)$-th order is obtained, for both u_h and q_h, when P^k elements are used. See Tables 1 to 3. The P^4 case for the purely convection equation $a = 0$ seems to be not in the asymptotic regime yet with $N = 40$ elements (further refinement with $N = 80$ suffers from round-off effects due to our choice of non-orthogonal basis functions), Table 4. However, the absolute values of the errors are comparable with the convection dominated case in Table 3.

Table 1. The heat equation $a = 1$, $c = 0$. L_∞ errors and numerical order of accuracy, measured at the center of each element, for $\Delta x^m \partial_x^m u_h$ for $0 \leq m \leq k$, and for q_h.

k	variable	$N = 10$	$N = 20$		$N = 40$	
		error	error	order	error	order
1	u	4.55E-4	5.79E-5	2.97	7.27E-6	2.99
	$\Delta x\, \partial_x u$	9.01E-3	2.22E-3	2.02	5.56E-4	2.00
	q	4.17E-5	2.48E-6	4.07	1.53E-7	4.02
2	u	1.43E-4	1.76E-5	3.02	2.19E-6	3.01
	$\Delta x\, \partial_x u$	7.87E-4	1.03E-4	2.93	1.31E-5	2.98
	$(\Delta x)^2\, \partial_x^2 u$	1.64E-3	2.09E-4	2.98	2.62E-5	2.99
	q	1.42E-4	1.76E-5	3.01	2.19E-6	3.01
3	u	1.54E-5	9.66E-7	4.00	6.11E-8	3.98
	$\Delta x\, \partial_x u$	3.77E-5	2.36E-6	3.99	1.47E-7	4.00
	$(\Delta x)^2\, \partial_x^2 u$	1.90E-4	1.17E-5	4.02	7.34E-7	3.99
	$(\Delta x)^3\, \partial_x^3 u$	2.51E-4	1.56E-5	4.00	9.80E-7	4.00
	q	1.48E-5	9.66E-7	3.93	6.11E-8	3.98
4	u	2.02E-7	5.51E-9	5.20	1.63E-10	5.07
	$\Delta x\, \partial_x u$	1.65E-6	5.14E-8	5.00	1.61E-9	5.00
	$(\Delta x)^2\, \partial_x^2 u$	6.34E-6	2.04E-7	4.96	6.40E-9	4.99
	$(\Delta x)^3\, \partial_x^3 u$	2.92E-5	9.47E-7	4.95	2.99E-8	4.99
	$(\Delta x)^4\, \partial_x^4 u$	3.03E-5	9.55E-7	4.98	2.99E-8	5.00
	q	2.10E-7	5.51E-9	5.25	1.63E-10	5.07

Table 2. The convection diffusion equation $a = 1$, $c = 1$. L_∞ errors and numerical order of accuracy, measured at the center of each element, for $\Delta x^m \partial_x^m u_h$ for $0 \leq m \leq k$, and for q_h.

k	variable	$N = 10$	$N = 20$		$N = 40$	
		error	error	order	error	order
1	u	6.47E-4	1.25E-4	2.37	1.59E-5	2.97
	$\Delta x\, \partial_x u$	9.61E-3	2.24E-3	2.10	5.56E-4	2.01
	q	2.96E-3	1.20E-4	4.63	1.47E-5	3.02
2	u	1.42E-4	1.76E-5	3.02	2.18E-6	3.01
	$\Delta x\, \partial_x u$	7.93E-4	1.04E-4	2.93	1.31E-5	2.99
	$(\Delta x)^2\, \partial_x^2 u$	1.61E-3	2.09E-4	2.94	2.62E-5	3.00
	q	1.26E-4	1.63E-5	2.94	2.12E-6	2.95
3	u	1.53E-5	9.75E-7	3.98	6.12E-8	3.99
	$\Delta x\, \partial_x u$	3.84E-5	2.34E-6	4.04	1.47E-7	3.99
	$(\Delta x)^2\, \partial_x^2 u$	1.89E-4	1.18E-5	4.00	7.36E-7	4.00
	$(\Delta x)^3\, \partial_x^3 u$	2.52E-4	1.56E-5	4.01	9.81E-7	3.99
	q	1.57E-5	9.93E-7	3.98	6.17E-8	4.01
4	u	2.04E-7	5.50E-9	5.22	1.64E-10	5.07
	$\Delta x\, \partial_x u$	1.68E-6	5.19E-8	5.01	1.61E-9	5.01
	$(\Delta x)^2\, \partial_x^2 u$	6.36E-6	2.05E-7	4.96	6.42E-8	5.00
	$(\Delta x)^3\, \partial_x^3 u$	2.99E-5	9.57E-7	4.97	2.99E-8	5.00
	$(\Delta x)^4\, \partial_x^4 u$	2.94E-5	9.55E-7	4.95	3.00E-8	4.99
	q	1.96E-7	5.35E-9	5.19	1.61E-10	5.06

Table 3. The convection dominated convection diffusion equation $a = 0.01$, $c = 1$. L_∞ errors and numerical order of accuracy, measured at the center of each element, for $\Delta x^m \partial_x^m u_h$ for $0 \leq m \leq k$, and for q_h.

k	variable	$N = 10$	$N = 20$		$N = 40$	
		error	error	order	error	order
1	u	7.14E-3	9.30E-4	2.94	1.17E-4	2.98
	$\Delta x\, \partial_x u$	6.04E-2	1.58E-2	1.93	4.02E-3	1.98
	q	8.68E-4	1.09E-4	3.00	1.31E-5	3.05
2	u	9.59E-4	1.25E-4	2.94	1.58E-5	2.99
	$\Delta x\, \partial_x u$	5.88E-3	7.55E-4	2.96	9.47E-5	3.00
	$(\Delta x)^2\, \partial_x^2 u$	1.20E-2	1.50E-3	3.00	1.90E-4	2.98
	q	8.99E-5	1.11E-5	3.01	1.10E-6	3.34
3	u	1.11E-4	7.07E-6	3.97	4.43E-7	4.00
	$\Delta x\, \partial_x u$	2.52E-4	1.71E-5	3.88	1.07E-6	4.00
	$(\Delta x)^2\, \partial_x^2 u$	1.37E-3	8.54E-5	4.00	5.33E-6	4.00
	$(\Delta x)^3\, \partial_x^3 u$	1.75E-3	1.13E-4	3.95	7.11E-6	3.99
	q	1.18E-5	7.28E-7	4.02	4.75E-8	3.94
4	u	1.85E-6	4.02E-8	5.53	1.19E-9	5.08
	$\Delta x\, \partial_x u$	1.29E-5	3.76E-7	5.10	1.16E-8	5.01
	$(\Delta x)^2\, \partial_x^2 u$	5.19E-5	1.48E-6	5.13	4.65E-8	4.99
	$(\Delta x)^3\, \partial_x^3 u$	2.21E-4	6.93E-6	4.99	2.17E-7	5.00
	$(\Delta x)^4\, \partial_x^4 u$	2.25E-4	6.89E-6	5.03	2.17E-7	4.99
	q	3.58E-7	3.06E-9	6.87	5.05E-11	5.92

Table 4. The convection equation $a = 0$, $c = 1$. L_∞ errors and numerical order of accuracy, measured at the center of each element, for $\Delta x^m \partial_x^m u_h$ for $0 \leq m \leq k$.

k	variable	$N = 10$	$N = 20$		$N = 40$	
		error	error	order	error	order
1	u	7.24E-3	9.46E-4	2.94	1.20E-4	2.98
	$\Delta x\, \partial_x u$	6.09E-2	1.60E-2	1.92	4.09E-3	1.97
2	u	9.96E-4	1.28E-4	2.96	1.61E-5	2.99
	$\Delta x\, \partial_x u$	6.00E-3	7.71E-4	2.96	9.67E-5	3.00
	$(\Delta x)^2\, \partial_x^2 u$	1.23E-2	1.54E-3	3.00	1.94E-4	2.99
3	u	1.26E-4	7.50E-6	4.07	4.54E-7	4.05
	$\Delta x\, \partial_x u$	1.63E-4	2.00E-5	3.03	1.07E-6	4.21
	$(\Delta x)^2\, \partial_x^2 u$	1.52E-3	9.03E-5	4.07	5.45E-6	4.05
	$(\Delta x)^3\, \partial_x^3 u$	1.35E-3	1.24E-4	3.45	7.19E-6	4.10
4	u	3.55E-6	8.59E-8	5.37	3.28E-10	8.03
	$\Delta x\, \partial_x u$	1.89E-5	1.27E-7	7.22	1.54E-8	3.05
	$(\Delta x)^2\, \partial_x^2 u$	8.49E-5	2.28E-6	5.22	2.33E-8	6.61
	$(\Delta x)^3\, \partial_x^3 u$	2.36E-4	5.77E-6	5.36	2.34E-7	4.62
	$(\Delta x)^4\, \partial_x^4 u$	2.80E-4	8.93E-6	4.97	1.70E-7	5.72

Finally, to show that the order of accuracy could really degenerate to k for P^k, as was already observed in [4], we rerun the heat equation case $a = 1, c = 0$ with the central flux

$$\mathbb{C} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{5.22}$$

This time we can see that the global order of accuracy in L_∞ is only k when P^k is used with an odd value of k.

Table 5. The heat equation $a = 1$, $c = 0$. L_∞ errors and numerical order of accuracy, measured at the center of each element, for $\Delta x^m \partial_x^m u_h$ for $0 \leq m \leq k$, and for q_h, using the central flux.

k	variable	$N = 10$	$N = 20$		$N = 40$	
		error	error	order	error	order
1	u	3.59E-3	8.92E-4	2.01	2.25E-4	1.98
	$\Delta x\, \partial_x u$	2.10E-2	1.06E-2	0.98	5.31E-3	1.00
	q	2.39E-3	6.19E-4	1.95	1.56E-4	1.99
2	u	6.91E-5	4.12E-6	4.07	2.57E-7	4.00
	$\Delta x\, \partial_x u$	7.66E-4	1.03E-4	2.90	1.30E-5	2.98
	$(\Delta x)^2\, \partial_x^2 u$	2.98E-4	1.68E-5	4.15	1.03E-6	4.02
	q	6.52E-5	4.11E-6	3.99	2.57E-7	4.00
3	u	1.62E-5	1.01E-6	4.00	6.41E-8	3.98
	$\Delta x\, \partial_x u$	1.06E-4	1.32E-5	3.01	1.64E-6	3.00
	$(\Delta x)^2\, \partial_x^2 u$	1.99E-4	1.22E-5	4.03	7.70E-7	3.99
	$(\Delta x)^3\, \partial_x^3 u$	6.81E-4	8.68E-5	2.97	1.09E-5	2.99
	q	1.54E-5	1.01E-6	3.93	6.41E-8	3.98
4	u	8.25E-8	1.31E-9	5.97	2.11E-11	5.96
	$\Delta x\, \partial_x u$	1.62E-6	5.12E-8	4.98	1.60E-9	5.00
	$(\Delta x)^2\, \partial_x^2 u$	1.61E-6	2.41E-8	6.06	3.78E-10	6.00
	$(\Delta x)^3\, \partial_x^3 u$	2.90E-5	9.46E-7	4.94	2.99E-8	4.99
	$(\Delta x)^4\, \partial_x^4 u$	5.23E-6	7.59E-8	6.11	1.18E-9	6.01
	q	7.85E-8	1.31E-9	5.90	2.11E-11	5.96

5.4 The LDG methods for the multi-dimensional case

In this section, we consider the LDG methods for the following convection-diffusion model problem

$$\partial_t u + \sum_{1\le i\le d} \partial_{x_i} (f_i(u) - \sum_{1\le j\le d} a_{ij}(u)\, \partial_{x_j} u) = 0 \quad \text{in } (0,T)\times(0,1)^d, \tag{5.23}$$

$$u(t=0) = u_0, \quad \text{on } (0,1)^d, \tag{5.24}$$

with periodic boundary conditions. Essentially, the one-dimensional case and the multidimensional case can be studied in exactly the same way. However, there are two important differences that deserve explicit discussion. The first is the treatment of the matrix of entries $a_{ij}(u)$, which is assumed to be *symmetric, semipositive definite* and the introduction of the variables q_ℓ, and the second is the treatment of arbitrary meshes. 4 To define the LDG method, we first notice that, since the matrix $a_{ij}(u)$ is assumed to be symmetric and semipositive definite, there exists a symmetric matrix $b_{ij}(u)$ such that

$$a_{ij}(u) = \textstyle\sum_{1\le \ell\le d} b_{i\ell}(u)\, b_{\ell j}(u). \tag{5.25}$$

Then we define the new scalar variables $q_\ell = \sum_{1\le j\le d} b_{\ell j}(u)\, \partial_{x_j} u$ and rewrite the problem (5.23), (5.24) as follows:

$$\partial_t u + \sum_{1\le i\le d} \partial_{x_i} (f_i(u) - \sum_{1\le \ell\le d} b_{i\ell}(u)\, q_\ell) = 0 \quad \text{in } (0,T)\times(0,1)^d, \tag{5.26}$$

$$q_\ell - \sum_{1\le j\le d} \partial_{x_j} g_{\ell j}(u) = 0, \quad \ell = 1,\dots d, \quad \text{in } (0,T)\times(0,1)^d, \tag{5.27}$$

$$u(t=0) = u_0, \quad \text{on } (0,1)^d, \tag{5.28}$$

where $g_{\ell j}(u) = \int^u b_{\ell j}(s)\, ds$. The LDG method is now obtained by discretizing the above equations by the Discontinuous Galerkin method.

We follow what was done in §2. So, we set $\mathbf{w} = (u, \mathbf{q})^t = (u, q_1, \cdots, q_d)^t$ and, for each $i = 1, \cdots, d$, introduce the flux

$$\mathbf{h}_i(\mathbf{w}) = (\, f_i(u) - \textstyle\sum_{1\le \ell\le d} b_{i\ell}(u)\, q_\ell, -g_{1i}(u), \cdots, -g_{di}(u)\,)^t. \tag{5.29}$$

We consider triangulations of $(0,1)^d$, $\mathbb{T}_{\Delta x} = \{\, K \,\}$, made of non-overlapping polyhedra. We require that for any two elements K and K', $\overline{K} \cap \overline{K}'$ is either

a face e of both K and K' with nonzero $(d-1)$-Lebesgue measure $|\,e\,|$, or has Hausdorff dimension less than $d-1$. We denote by $\mathbb{E}_{\Delta x}$ the set of all faces e of the border of K for all $K \in \mathbb{T}_{\Delta x}$. The diameter of K is denoted by Δx_K and the maximum Δx_K, for $K \in \mathbb{T}_{\Delta x}$ is denoted by Δx. We require, for the sake of simplicity, that the triangulations $\mathbb{T}_{\Delta x}$ be regular, that is, there is a constant independent of Δx such that

$$\frac{\Delta x_K}{\rho_K} \leq \sigma \quad \forall\, K \in \mathbb{T}_{\Delta x},$$

where ρ_K denotes the diameter of the maximum ball included in K.

We seek an approximation $\mathbf{w}_h = (u_h, \mathbf{q}_h)^t = (u_h, q_{h1}, \cdots, q_{hd})^t$ to $\mathbf{w}$ such that for each time $t \in [0,T]$, each of the components of $\mathbf{w}_h$ belong to the finite element space

$$V_h = V_h^k = \{\, v \in L^1((0,1)^d) :\ v|_K \in P^k(K)\ \forall\, K \in \mathbb{T}_{\Delta x}\}, \qquad (5.30)$$

where $P^k(K)$ denotes the space of polynomials of total degree at most k. In order to determine the approximate solution $\mathbf{w}_h$, we proceed exactly as in the one-dimensional case. This time, however, the integrals are made on each element K of the triangulation $\mathbb{T}_{\Delta x}$. We obtain the following weak formulation on each element K of the triangulation $\mathbb{T}_{\Delta x}$:

$$\begin{aligned}
&\textstyle\int_K \partial_t\, u_h(x,t)\, v_{h,u}(x)\, dx - \sum_{1\leq i\leq d} \int_K h_{i\,u}(\mathbf{w}_h(x,t))\, \partial_{x_i}\, v_{h,u}(x)\, dx \\
&+ \textstyle\int_{\partial K} \hat{h}_u(\mathbf{w}_h, \mathbf{n}_{\partial K})(x,t)\, v_{h,u}(x)\, d\,\Gamma(x) = 0, \qquad \forall\, v_{h,u} \in P^k(K), \quad (5.31)
\end{aligned}$$

$$\text{For } \ell = 1, \cdots, d:$$

$$\begin{aligned}
&\textstyle\int_K q_{h\ell}(x,t)\, v_{h,q_\ell}(x)\, dx - \sum_{1\leq j\leq d} \int_K h_{j\,q_\ell}(\mathbf{w}_h(x,t))\, \partial_{x_j}\, v_{h,q_\ell}(x)\, dx \\
&+ \textstyle\int_{\partial K} \hat{h}_{q_\ell}(\mathbf{w}_h, \mathbf{n}_{\partial K})(x,t)\, v_{h,q_\ell}(x)\, d\,\Gamma(x) = 0, \quad \forall\, v_{h,q_\ell} \in P^k(K), \quad (5.32) \\
&\textstyle\int_K u_h(x,0)\, v_{h,i}(x)\, dx = \int_K u_0(x)\, v_{h,i}(x)\, dx, \qquad \forall\, v_{h,i} \in P^k(K), \quad (5.33)
\end{aligned}$$

where $\mathbf{n}_{\partial K}$ denotes the outward unit normal to the element K at $x \in \partial K$. It **remains to choose the numerical flux $(\hat{h}_u, \hat{h}_{q_1}, \cdots, \hat{h}_{q_d})^t \equiv \hat{\mathbf{h}} \equiv \hat{\mathbf{h}}(\mathbf{w}_h, \mathbf{n}_{\partial K})(x,t)$.**

As in the one-dimensional case, we require that the fluxes $\hat{\mathbf{h}}$ be of the form

$$\hat{\mathbf{h}}(\mathbf{w}_h, \mathbf{n}_{\partial K})(x) \equiv \hat{\mathbf{h}}(\mathbf{w}_h(x^{int_K}, t), \mathbf{w}_h(x^{ext_K}, t); \mathbf{n}_{\partial K}),$$

where $\mathbf{w}_h(x^{int_K})$ is the limit at x taken from the interior of K and $\mathbf{w}_h(x^{ext_K})$ the limit at x from the exterior of K, and consider fluxes that (i) are locally Lipschitz, conservative, that is,

$$\hat{\mathbf{h}}(\mathbf{w}_h(x^{int_K}), \mathbf{w}_h(x^{ext_K}); \mathbf{n}_{\partial K}) + \hat{\mathbf{h}}(\mathbf{w}_h(x^{ext_K}), \mathbf{w}_h(x^{int_K}); -\mathbf{n}_{\partial K}) = 0,$$

and consistent with the flux

$$\sum_{1 \le i \le d} \mathbf{h}_i \, n_{\partial K, i},$$

(ii) allow for a local resolution of each component of $\mathbf{q}_h$ in terms of u_h *only*, (iii) reduce to an E-flux when $a(\cdot) \equiv 0$, and that (iv) enforce the L^2-stability of the method.

Again, we write our numerical flux as the sum of a convective flux and a diffusive flux:

$$\hat{\mathbf{h}} = \hat{\mathbf{h}}_{conv} + \hat{\mathbf{h}}_{diff},$$

where the convective flux is given by

$$\hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+; \mathbf{n}) = \big(\hat{f}(u^-, u^+; \mathbf{n}), 0\big)^t,$$

where $\hat{f}(u^-, u^+; \mathbf{n})$ is any locally Lipschitz E-flux which is conservative and consistent with the nonlinearity

$$\sum_{1 \le i \le d} f_i(u) \, n_i,$$

and the diffusive flux $\hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+; \mathbf{n})$ is given by

$$\Big(- \sum_{1 \le i, \ell \le d} \frac{[\, g_{i\ell}(u) \,]}{[\, u \,]} \, \overline{q_\ell} \, n_i, \; - \sum_{1 \le i \le d} \overline{g_{i1}(u)} \, n_i, \cdots, - \sum_{1 \le i \le d} \overline{g_{id}(u)} \, n_i \Big)^t - \mathbb{C}_{diff} \, [\, \mathbf{w} \,],$$

where

$$\mathbb{C}_{diff} = \begin{pmatrix} 0 & c_{12} & c_{13} & \cdots & c_{1d} \\ -c_{12} & 0 & 0 & \cdots & 0 \\ -c_{13} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_{1d} & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$c_{1j} = c_{1j}(\mathbf{w}^-, \mathbf{w}^+) \quad \text{is locally Lipschitz for } j = 1, \cdots, d,$$

$$c_{1j} \equiv 0 \quad \text{when } a(\cdot) \equiv 0 \quad \text{for } j = 1, \cdots, d.$$

We claim that this flux satisfies the properties (i) to (iv).

To prove that properties (i) to (iii) are satisfied is now a simple exercise. To see that the property (iv) is satisfied, we first rewrite the flux $\hat{\mathbf{h}}$ in the following way:

$$\Big(-\sum_{1\le i,\ell\le d} \frac{[\,g_{i\ell}(u)\,]}{[\,u\,]}\,\overline{q_\ell}\,n_i,\ -\sum_{1\le i\le d} \overline{g_{i1}(u)}\,n_i, \cdots,\ -\sum_{1\le i\le d} \overline{g_{id}(u)}\,n_i \Big)^t - \mathbb{C}\,[\,\mathbf{w}\,],$$

where

$$\mathbb{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1d} \\ -c_{12} & 0 & 0 & \cdots & 0 \\ -c_{13} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_{1d} & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$c_{11} = \frac{1}{[\,u\,]}\Big(\textstyle\sum_{1\le i\le d} \dfrac{[\,\varphi_i(u)\,]}{[\,u\,]}\,n_i - \hat{f}(u^-, u^+; \mathbf{n}) \Big),$$

where $\varphi_i(u) = \int^u f_i(s)\,ds$. Since $\hat{f}(\cdot,\cdot;\mathbf{n})$ is an E-flux,

$$c_{11} = \frac{1}{[\,u\,]^2} \int_{u^-}^{u^+} \big(\textstyle\sum_{1\le i\le d} f_i(s)\,n_i - \hat{f}(u^-, u^+; \mathbf{n}) \big)\,ds \ge 0,$$

and so the matrix $\mathbb{C}$ is semipositive definite. The property (iv) follows from this fact and from the following result.

Theorem 5.3 *We have,*

$$\frac{1}{2}\int_{(0,1)^d} u_h^2(x,T)\,dx + \int_0^T \int_{(0,1)^d} |\,\mathbf{q}_h(x,t)\,|^2\,dx\,dt + \Theta_{T,\mathbf{C}}([\mathbf{w}_h]) \le \frac{1}{2}\int_{(0,1)^d} u_0^2(x)\,dx,$$

where

$$\Theta_{T,\mathbb{C}}([\mathbf{w}_h]) = \int_0^T \sum_{e \in \mathbb{E}_{\Delta x}} \int_e [\mathbf{w}_h(x,t)]^t \mathbb{C}\, [\mathbf{w}_h(x,t)]\, d\Gamma(x)\, dt.$$

We can also prove the following error estimate. We denote the integral over $(0,1)^d$ of the sum of the squares of all the derivatives of order $(k+1)$ of u by $|\, u \,|^2_{k+1}$.

Theorem 5.4 *Let* $\mathbf{e}$ *be the approximation error* $\mathbf{w} - \mathbf{w}_h$. *Then we have, for arbitrary, regular grids,*

$$\left\{ \int_{(0,1)^d} |\, e_u(x,T) \,|^2\, dx + \int_0^T \int_{(0,1)^d} |\, \mathbf{e}_q(x,t) \,|^2\, dx\, dt + \Theta_{T,\mathbb{C}}([\mathbf{e}]) \right\}^{1/2} \leq C\, (\Delta x)^k,$$

where $C = C(k, |\, u \,|_{k+1}, |\, u \,|_{k+2})$. *In the purely hyperbolic case* $a_{ij} = 0$, *the constant* C *is of order* $(\Delta x)^{1/2}$. *In the purely parabolic case* $c = 0$, *the constant* C *is of order* Δx *for even values of* k *and of order* 1 *otherwise for Cartesian products of uniform grids and for* $\mathbb{C}$ *identically zero provided that the local spaces* Q^k *are used instead of the spaces* P^k, *where* Q^k *is the space of tensor products of one dimensional polynomials of degree* k.

5.5 Extension to multidimensional systems

In this chapter, we have considered the so-called LDG methods for convection-diffusion problems. For scalar problems in multidimensions, we have shown that they are L^2-stable and that in the linear case, they are of order k if polynomials of order k are used. We have also shown that this estimate is sharp and have displayed the strong dependence of the order of convergence of the LDG methods on the choice of the numerical fluxes.

The main advantage of these methods is their extremely high parallelizability and their high-order accuracy which render them suitable for computations of convection-dominated flows. Indeed, although the LDG method have a large amount of degrees of freedom per element, and hence more computations per element are necessary, its extremely local domain of dependency allows a very efficient parallelization that by far compensates for the extra amount of local computations.

The LDG methods for multidimensional systems, like for example the compressible Navier-Stokes equations and the equations of the hydrodynamic model for semiconductor device simulation, can be easily defined by simply applying the procedure described for the multidimensional scalar case to each

component of $\mathbf{u}$. In practice, especially for viscous terms which are not symmetric but still semipositive definite, such as for the compressible Navier-Stokes equations, we can use $\mathbf{q} = (\partial_{x_1} u, ..., \partial_{x_d} u)$ as the auxilary variables. Although with this choice, the L^2-stability result will not be available theoretically, this would not cause any problem in practical implementations.

5.6 Some numerical results

Next, we present some numerical results from the papers by Bassi and Rebay [3] and Lomtev and Karniadakis [46].

• **Smooth, steady state solutions**. We start by displaying the convergence of the method for a p-refinement done by Lomtev and Karniadakis [46]. In Figure 5.23, we can see how the maximum errors in density, momentum, and energy decrease exponentially to zero as the degree k of the approximating polynomials increases while the grid is kept fixed; details about the exact solution can be found in [46].

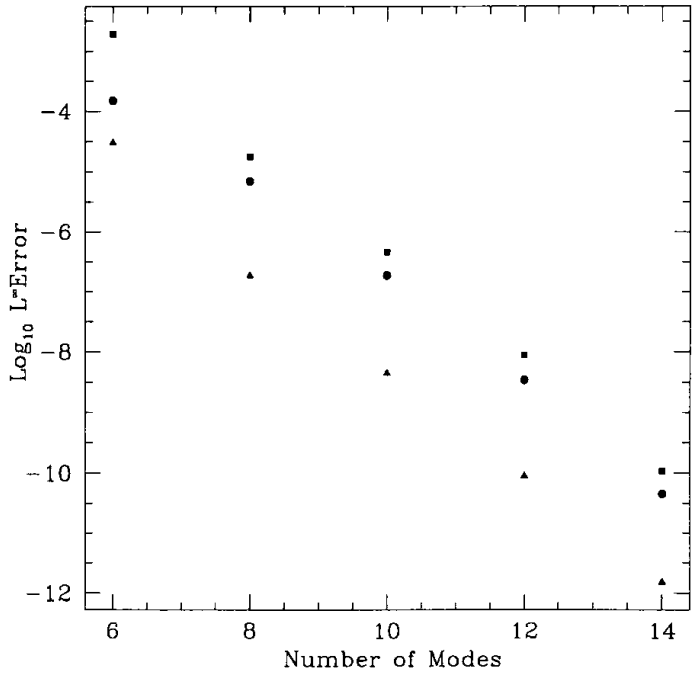


Fig. 5.23: Maximum errors of the density (triangles), momemtum (circles) and energy (squares) as a function of the degree of the approximating polynomial plus one (called "number of modes" in the picture).

Now, let us consider the laminar, transonic flow around the NACA0012 airfoil at an angle of attack of ten degrees, freestream Mach number $M =$

0.8, and Reynolds number (based on the freestream velocity and the airfoil chord) equal to 73; the wall temperature is set equal to the freestream total temperature. Bassy and Rebay [3] have computed the solution of this problem with polynomials of degree $1, 2$, and 3 and Lomtev and Karniadakis [46] have tried the same test problem with polynomials of degree $2, 4$, and 6 in a mesh of 592 elements which is about four times less elements than the mesh used by Bassi and Rebay [3]. In Figure 5.25, taken from [46], we display the pressure and drag coefficient distributions computed by Bassi and Rebay [3] with polynomials on degree 3 and the ones computed by Lomtev and Karniadakis [46] computed with polynomials of degree 6. We can see good agreement of both computations. In Figure 5.24, taken from [46], we see the mesh and the Mach isolines obtained with polynomials of degree two and four; note the improvement of the solution.

Next, we show a result from the paper by Bassi and Rebay [3]. We consider the laminar, subsonic flow around the NACA0012 airfoil at an angle of attack of zero degrees, freestream Mach number $M = 0.5$, and Reynolds number equal to 5000. In figure 5.26, we can see the Mach isolines corresponding to linear, quadratic, and cubic elements. In the figures 5.27, 5.28, and 5.29 details of the results with cubic elements are shown. Note how the boundary layer is captured withing a few layers of elements and how its separation at the trailing edge of the airfoil has been clearly resolved. Bassi and Rebay [3] report that these results are comparable to common structured and unstructures finite volume methods on much finer grids- a result consistent with the computational results we have displayed in these notes.

Finally, we present a not-yet-published result kindly provided by Lomtev and Karniadakis about the simulation of an expansion pipe flow. The smaller cylinder has a diameter of 1 and the larger cylinder has a diameter of 2. In Figure 5.30, we display the velocity profile and some streamlines for a Reynolds number equal to 50 and Mach number 0.2. The computation was made with polynomials of degree 5 and a mesh of 600 tetrahedra; of course the tetrahedra have curved faces to accomodate the exact boundaries. In Figure 5.31, we display a comparison between computational and experimental results. As a function of the Reynolds number, two quantities are plotted. The first is the distance between the step and the center of the vertex (lower brach) and the second is the distance from the step to the separation point (upper branch). The computational results are obtained by the method under consideration with polynomials of degree 5 for the compressible Navier Stokes equations, and by a standard Galerkin formulation in terms of velocity-pressure (NEKTAR), by Sherwin and Karniadakis [56], or in terms of velocity-vorticity (IVVA), by Trujillo [61], for the *incompressible* Navier Stokes equations; results produced by the code called PRISM are also included, see Newmann [48]. The experimental data was taken from Macagno and Tung [49]. The agreement between computations and experiments is remarkable.

• **Unsteady solutions.** To end this chapter, we present the computation of an unsteady solution by Lomtev and Karniadakis [46]. The test problem

is the classical problem of a flow around a cylinder in two space dimensions. The Reynolds number is 10,000 and the Mach number 0.2.

In Figure 5.32, the streamlines are shown for a computation made on a grid of 680 triangles (with curved sides fitting the cylinder) and polynomials whose degree could vary from element to element; the maximum degree was 5. In Figure 5.33, details of the mesh and the density around the cylinder are shown. Note how the method is able to capture the shear layer instability observed experimentally. For more details, see [46].

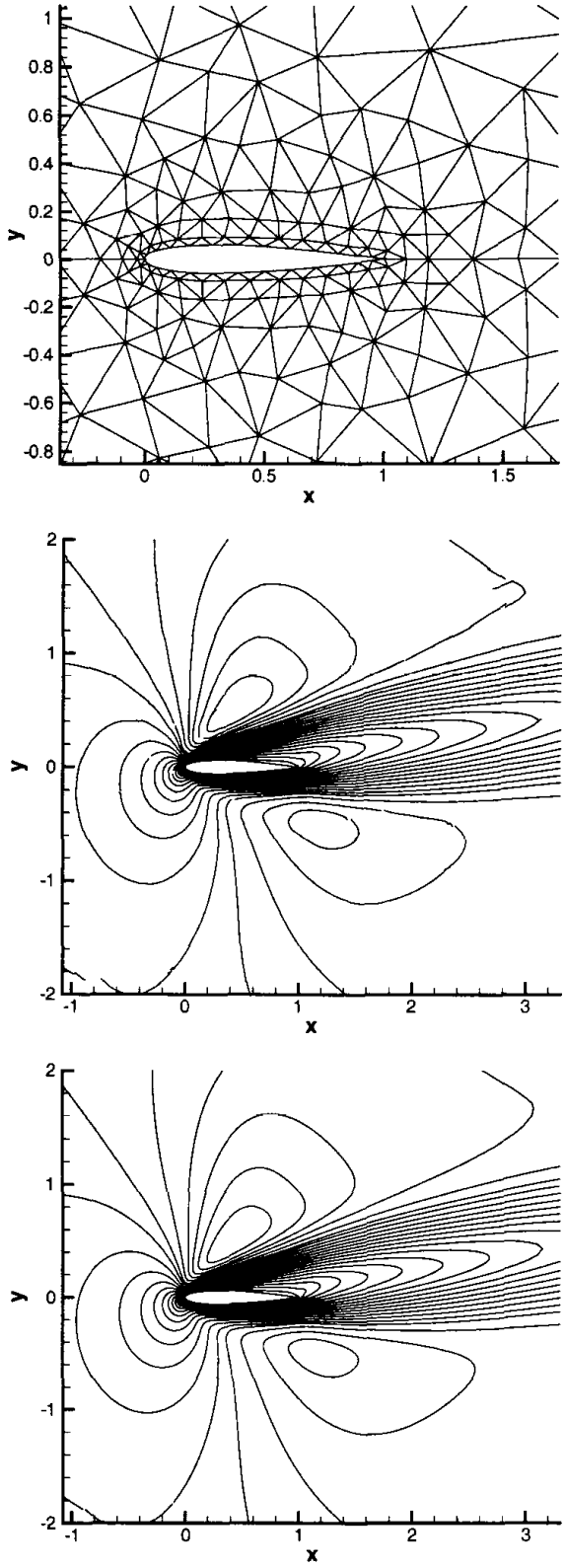


Fig. 5.24: Mesh (top) and Mach isolines around the NACA0012 airfoil, ($Re = 73, M = 0.8$, angle of attack of ten degrees) for quadratic (middle) and quartic (bottom) elements.

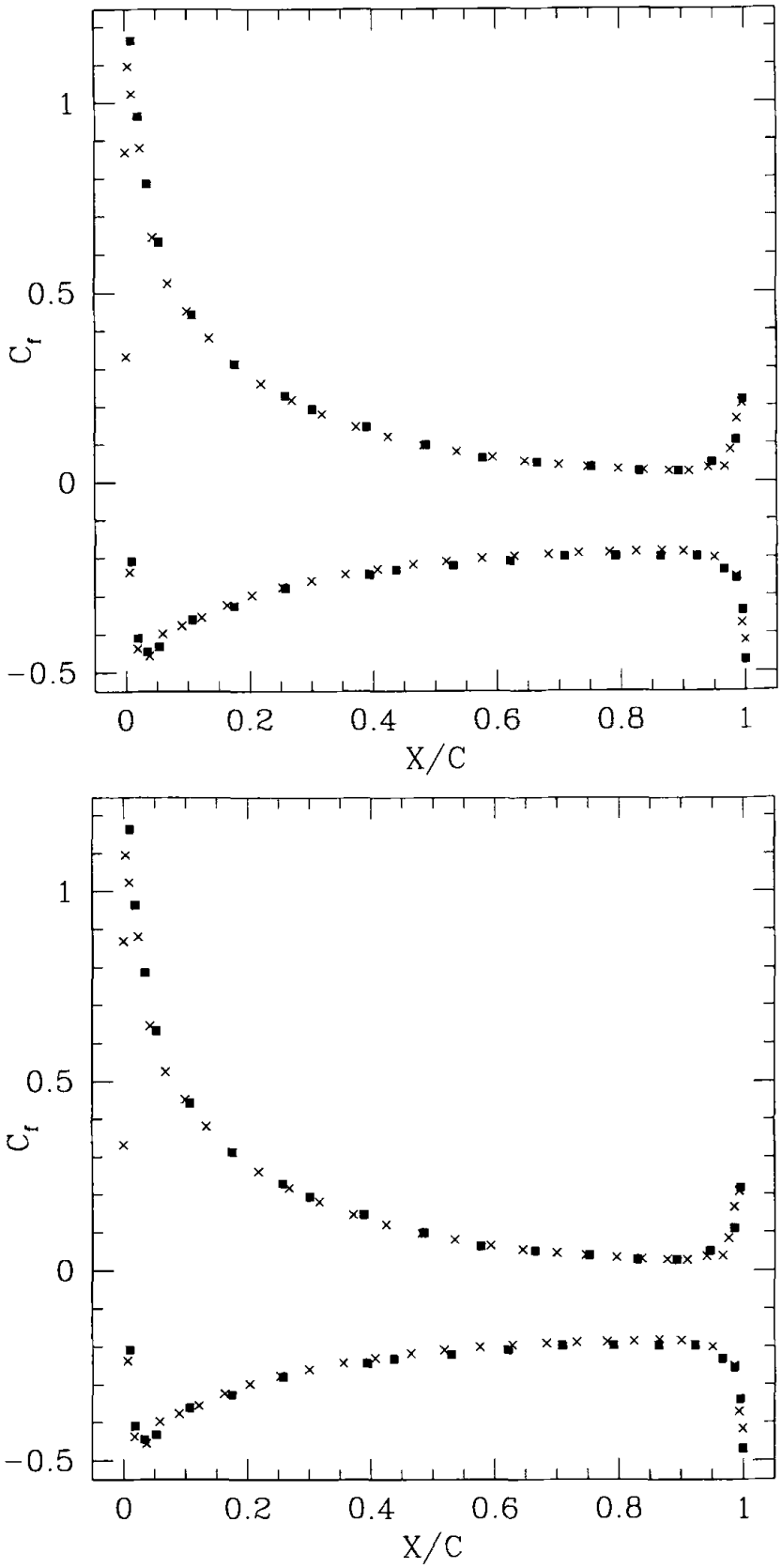


Fig. 5.25: Pressure (top) and drag(bottom) coefficient distributions. The squares were obtained by Bassi and Rebay [3] with cubics and the crosses by Lomtev and Karniadakis [46] with polynomials of degree 6.

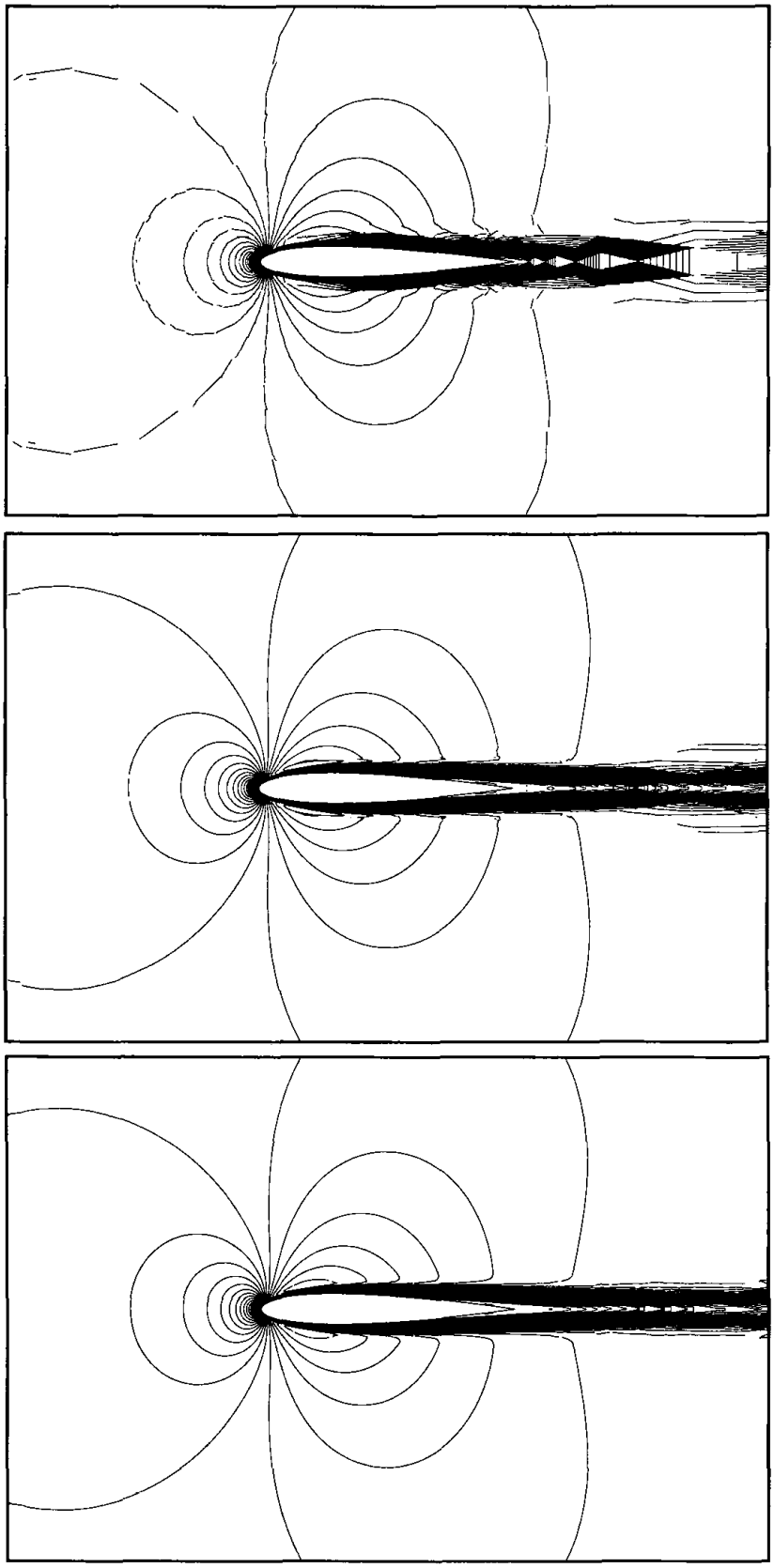


Fig. 5.26: Mach isolines around the NACA0012 airfoil, ($Re = 5000, M = 0.5$, zero angle of attack) for the linear (top), quadratic (middle), and cubic (bottom) elements.

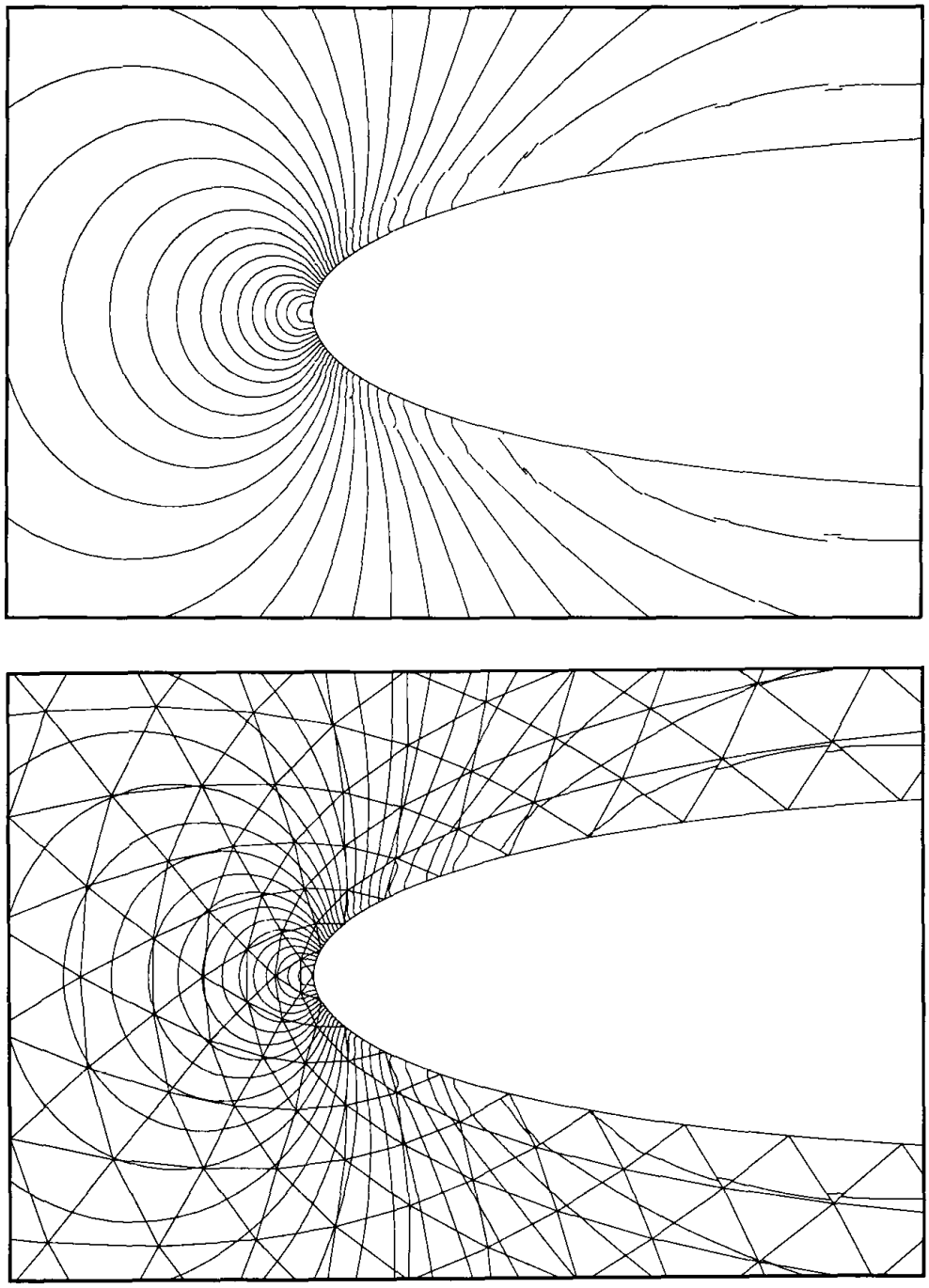


Fig. 5.27: Pressure isolines around the NACA0012 airfoil, ($Re = 5000, M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

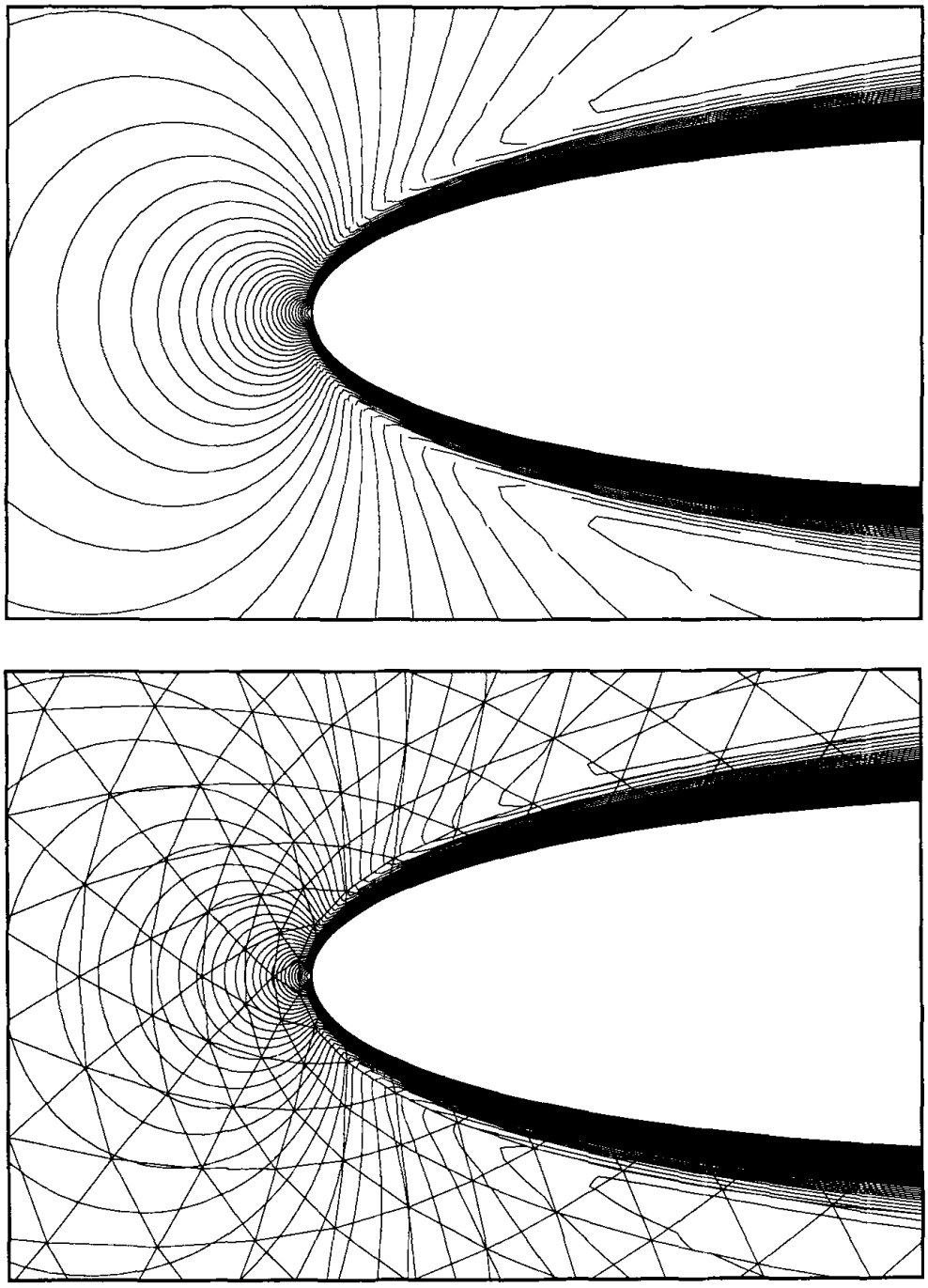


Fig. 5.28: Mach isolines around the leading edge of the NACA0012 airfoil, ($Re = 5000, M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

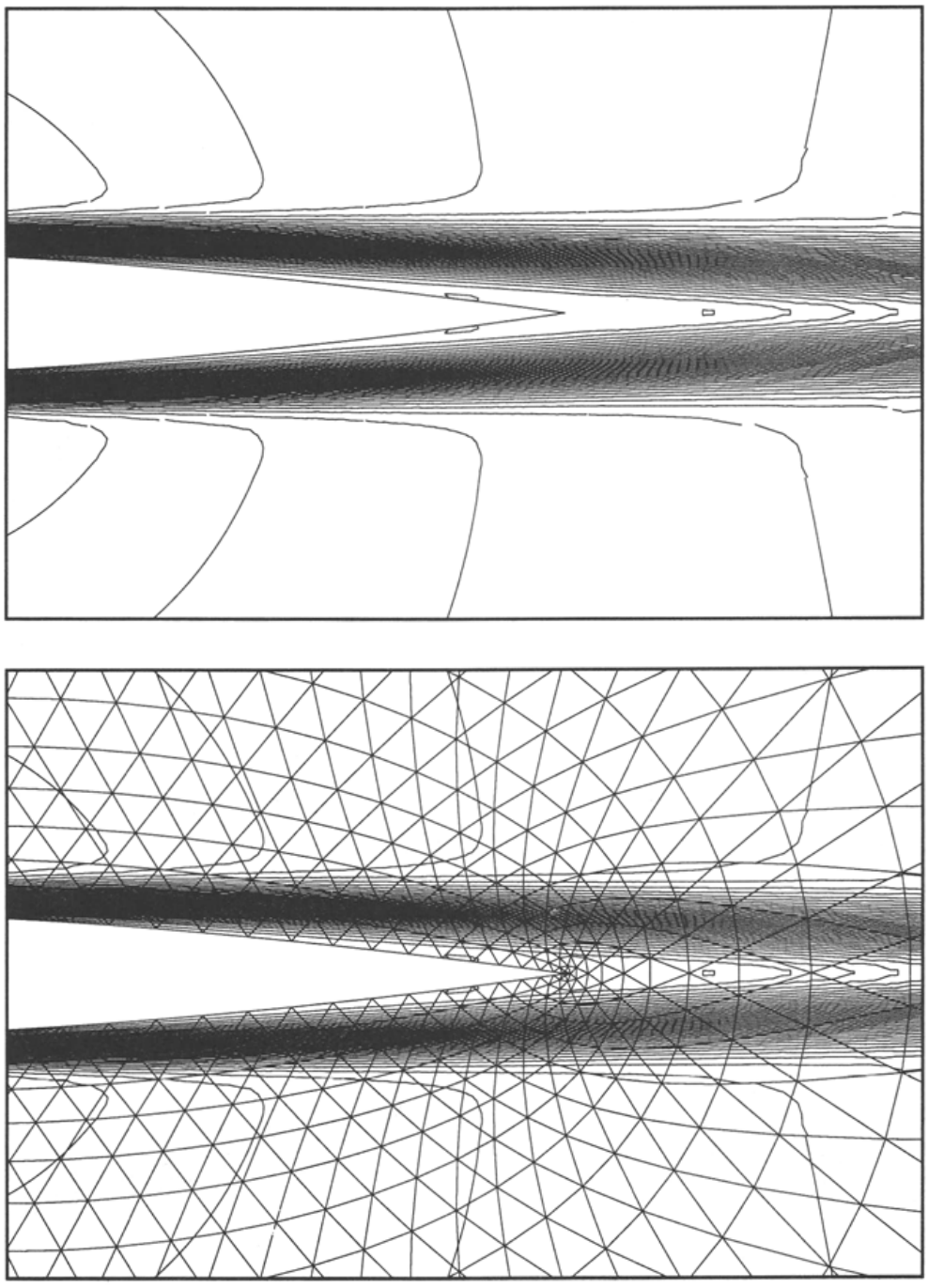


Fig. 5.29: Mach isolines around the trailing edge of the NACA0012 airfoil, ($Re = 5000, M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

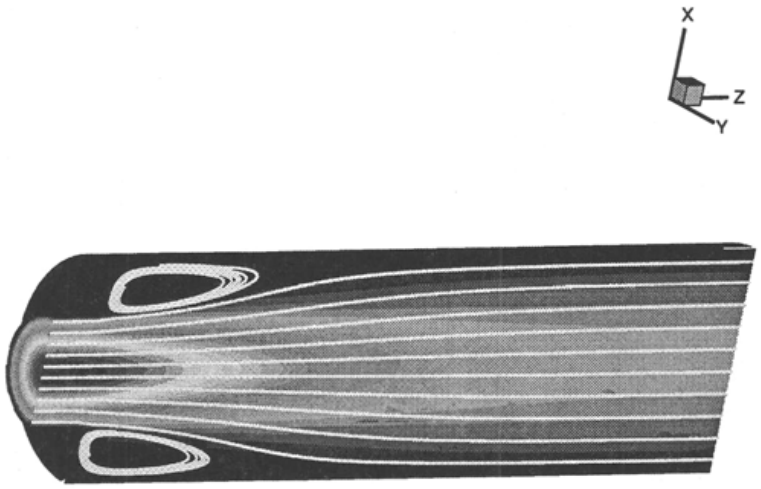


Fig. 5.30: Expansion pipe flow at Reynolds number 50 and Mach number 0.2. Velocity profile and streamlines computed with a mesh of 600 elements and polynomials of degree 5.

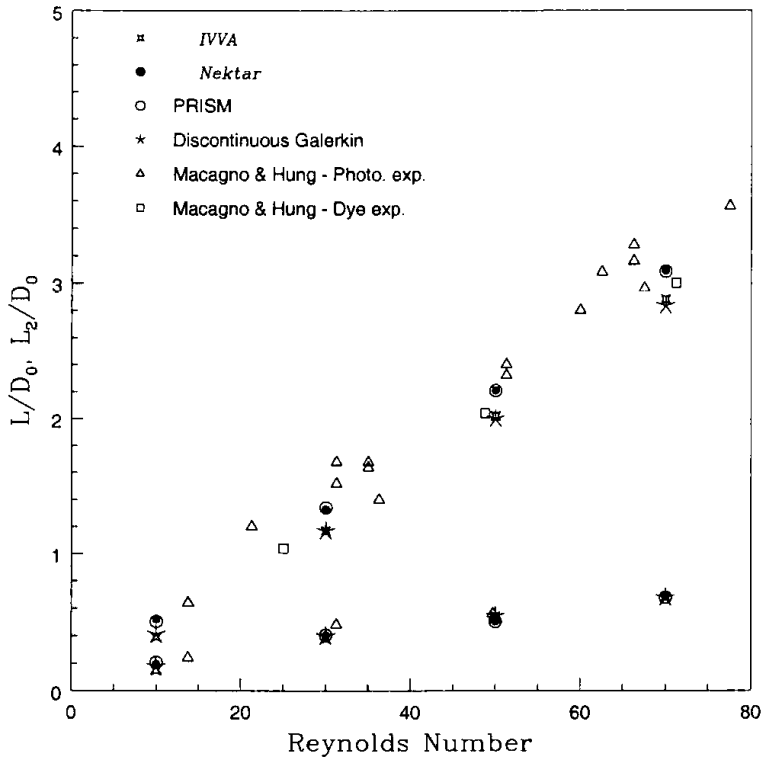


Fig. 5.31: Expansion pipe flow: Comparison between computational and experimental results.

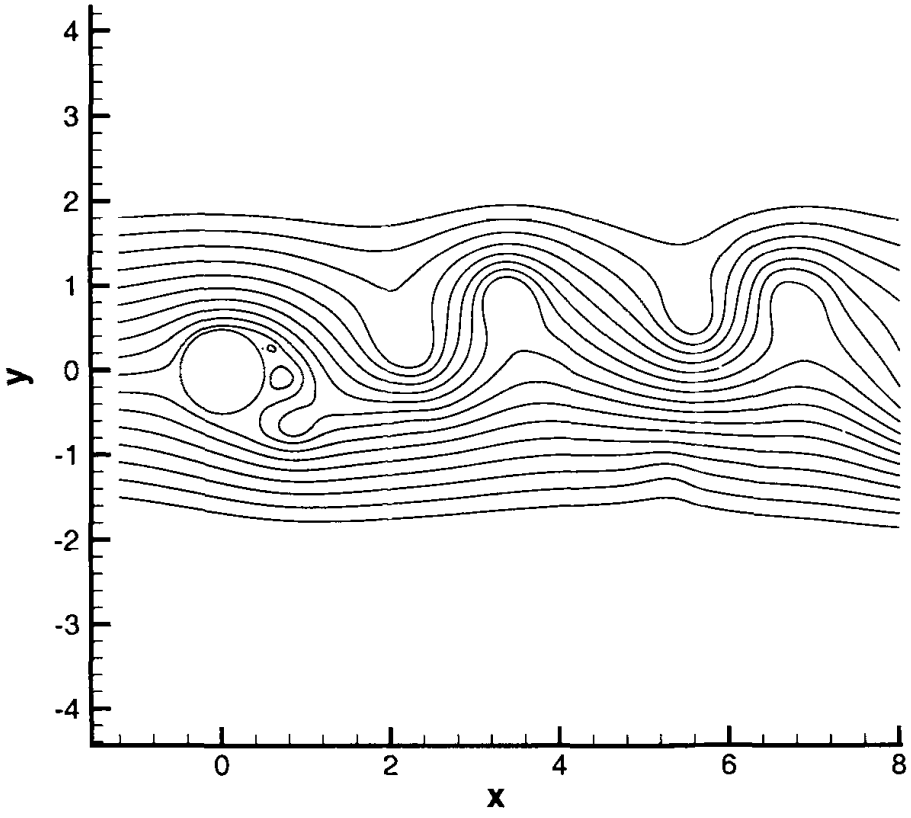


Fig. 5.32: Flow around a cylinder with Reynolds number 10,000 and Mach number 0.2. Streamlines. A mesh of 680 elements was used with polynomials that could change degree from element to element; the maximum degree was 5.

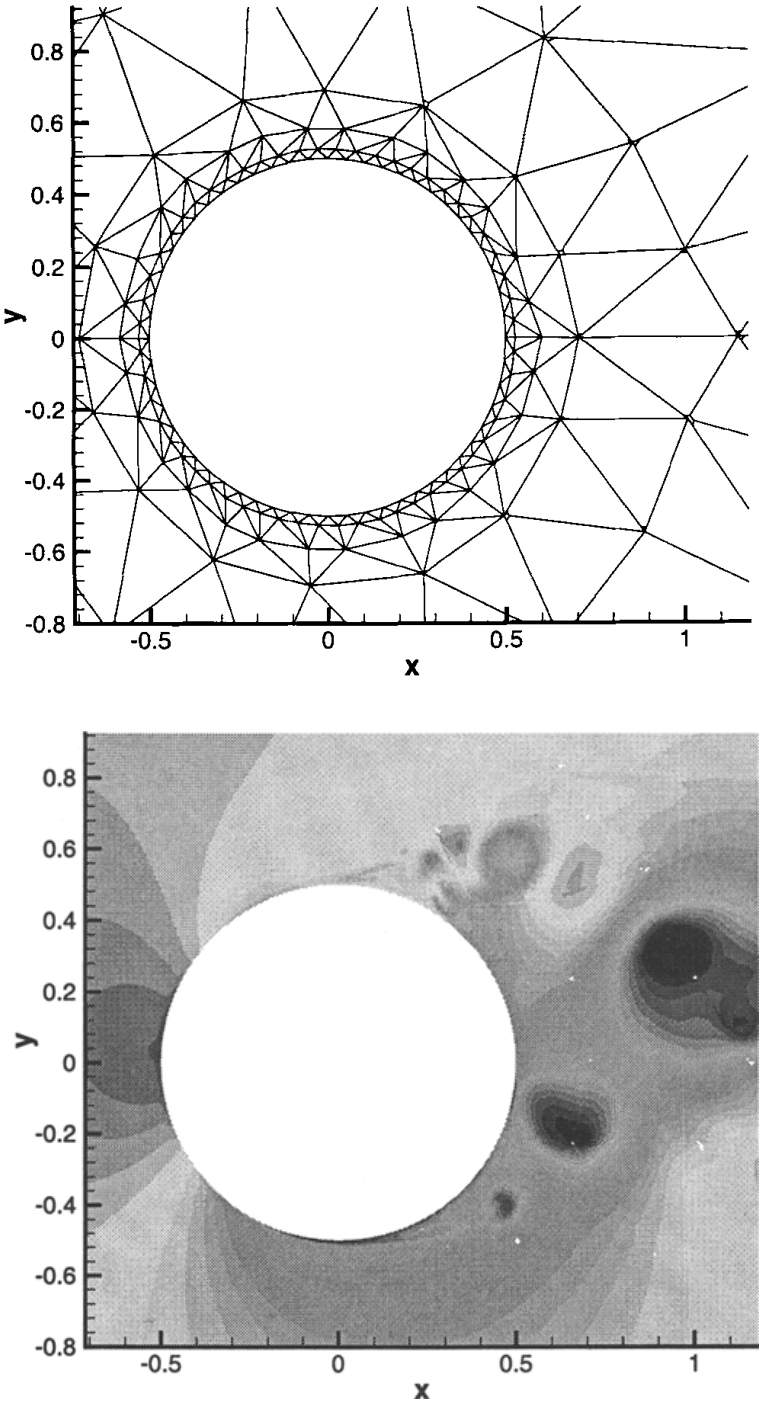


Fig. 5.33: Flow around a cylinder with Reynolds number 10,000 and Mach number 0.2. Detail of the mesh (top) and density (bottom) around the cylinder.

References

1. H.L. Atkins and C.-W. Shu. Quadrature-free implementation of discontinuous Galerkin methods for hyperbolic equations. *ICASE Report 96-51*, 1996. submitted to AIAA J.
2. F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d Euler equations. *J. Comput. Phys.* to appear.
3. F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys*, 131:267–279, 1997.
4. F. Bassi, S. Rebay, M. Savini, G. Mariotti, and S. Pedinotti. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. *Proceedings of the Second European Conference ASME on Turbomachinery Fluid Dynamics and Thermodynamics*, 1995.
5. K.S. Bey and J.T. Oden. A Runge-Kutta discontinuous Galerkin finite element method for high speed flows. *info AIAA* 10^{th} *Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24-27*, 1991.
6. R. Biswas, K.D. Devine, and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.
7. G. Chavent and B. Cockburn. The local projection p^0 p^1-discontinuous-Galerkin finite element method for scalar conservation laws. M^2AN, 23:565–592, 1989.
8. G. Chavent and G. Salzano. A finite element method for the 1d water flooding problem with gravity. *J. Comput. Phys*, 45:307–344, 1982.
9. Z. Chen, B. Cockburn, C. Gardner, and J. Jerome. Quantum hydrodynamic simulation of hysteresis in the resonant tunneling diode. *J. Comput. Phys*, 117:274–280, 1995.
10. Z. Chen, B. Cockburn, J. Jerome, and C.-W. Shu. Mixed-RKDG finite element method for the drift-diffusion semiconductor device equations. *VLSI Design*, 3:145–158, 1995.
11. P. Ciarlet. *The finite element method for elliptic problems.* North Holland, 1975.
12. B. Cockburn and P.-A. Gremaud. A priori error estimates for numerical methods for scalar conservation laws. part i: The general approach. *Math. Comp.*, 65:533–573, 1996.
13. B. Cockburn, S. Hou, and C.W. Shu. Tvb Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws iv: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.
14. B. Cockburn, S.Y. Lin, and C.W. Shu. Tvb Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws iii: One dimensional systems. *J. Comput. Phys*, 84:90–113, 1989.
15. B. Cockburn and C.W. Shu. Tvb Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws ii: General framework. *Math. Comp.*, 52:411–435, 1989.
16. B. Cockburn and C.W. Shu. The p^1-Rkdg method for two-dimensional Euler equations of gas dynamics. *ICASE Report No.91-32*, 1991.
17. B. Cockburn and C.W. Shu. The Runge-Kutta local projection p^1-discontinuous Galerkin method for scalar conservation laws. M^2AN, 25:337–361, 1991.

18. B. Cockburn and C.W. Shu. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM J. Numer. Anal.*. to appear.
19. B. Cockburn and C.W. Shu. The Runge-Kutta discontinuous Galerkin finite element method for conservation laws v: Multidimensional systems. *J. Comput. Phys.*. to appear.
20. H.L. deCougny, K.D. Devine, J.E. Flaherty, R.M. Loy, C. Ozturan, and M.S. Shephard. High-order accurate discontinuous finite element solution of the 2d Euler equations. *Applied Numerical Mathematics*, 16:157–182, 1994.
21. K.D. Devine, J.E. Flaherty, R.M. Loy, and S.R. Wheat. Parallel partitioning strategies for the adaptive solution of conservation laws. *Rensselaer Polytechnic Institute Report No. 94-1*, 1994.
22. K.D. Devine, J.E. Flaherty, S.R. Wheat, and A.B. Maccabe. A massively parallel adaptive finite element method with dynamic load balancing. *SAND 93-0936C*, 1993.
23. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems i: A linear model problem. *SIAM J. Numer. Anal.*, 28:43–77, 1991.
24. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems ii: Optimal error estimates in $l_\infty l_2$ and $l_\infty l_\infty$. *SIAM J. Numer. Anal.*, 32:706–740, 1995.
25. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems iv: A nonlinear model problem. *SIAM J. Numer. Anal.*, 32:1729–1749, 1995.
26. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems v: Long time integration. *SIAM J. Numer. Anal.*, 32:1750–1762, 1995.
27. K. Eriksson, C. Johnson, and V. Thomée. Time discretization of parabolic problems by the discontinuous Galerkin method. *RAIRO, Anal. Numér.*, 19:611–643, 1985.
28. J. Goodman and R. LeVeque. On the accuracy of stable schemes for 2d scalar conservation laws. *Math. Comp.*, 45:15–21, 1985.
29. T. Hughes and A. Brook. Streamline upwind-Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Comp. Meth. in App. Mech. and Eng.*, 32:199–259, 1982.
30. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, i. *Comp. Meth. in App. Mech. and Eng.*, 54:223–234, 1986.
31. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, ii. *Comp. Meth. in App. Mech. and Eng.*, 54:341–355, 1986.
32. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, iii. *Comp. Meth. in App. Mech. and Eng.*, 58:305–328, 1986.
33. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, iv. *Comp. Meth. in App. Mech. and Eng.*, 58:329–336, 1986.
34. T. Hughes and M. Mallet. A high-precision finite element method for shock-tube calculations. *Finite Element in Fluids*, 6:339–, 1985.

35. P. Jamet. Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain. *SIAM J. Numer. Anal.*, 15:912–928, 1978.
36. G. Jiang and C.-W. Shu. On cell entropy inequality for discontinuous Galerkin methods. *Math. Comp.*, 62:531–538, 1994.
37. C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46:1–26, 1986.
38. C. Johnson and J. Saranen. Streamline diffusion methods for problems in fluid mechanics. *Math. Comp.*, 47:1–18, 1986.
39. C. Johnson and A. Szepessy. On the convergence of a finite element method for a non-linear hyperbolic conservation law. *Math. Comp.*, 49:427–444, 1987.
40. C. Johnson, A. Szepessy, and P. Hansbo. On the convergence of shock capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Math. Comp.*, 54:107–129, 1990.
41. P. LeSaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations (C. de Boor, Ed.), Academic Press*, pages 89–145, 1974.
42. W. B. Lindquist. Construction of solutions for two-dimensional riemann problems. *Comp. & Maths. with Appls.*, 12:615–630, 1986.
43. W. B. Lindquist. The scalar Riemann problem in two spatial dimensions: piecewise smoothness of solutions and its breakdown. *SIAM J. Numer. Anal.*, 17:1178–1197, 1986.
44. I. Lomtev and G.E. Karniadakis. A discontinuous spectral/hp element Galerkin method for the Navier-Stokes equations on unstructured grids. *Proc. IMACS WC'97*, Berlin, Germany, 1997.
45. I. Lomtev and G.E. Karniadakis Simulations of viscous supersonic flows on unstructured h-p meshes. *AIAA 97-0754, 35th Aerospace Sciences Meeting*, Reno, 1997.
46. I. Lomtev and G.E. Karniadakis A Discontinuous Galerkin Method for the Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, submitted.
47. I. Lomtev, C.B. Quillen and G.E. Karniadakis. Spectral/hp methods for viscous compressible flows on unstructured 2D meshes. *J. Comp. Phys.*, in press.
48. D. Newmann. A Computational Study of Fluid/Structure Interactions: Flow-Induced Vibrations of a Flexible Cable Ph.D., Princeton, 1996.
49. E.O. Macagno and T. Hung. Computational and experimental study of a captive annular eddy. *J.F.M.*, 28:43 –, 1967.
50. X. Makridakis and I. Babusška. On the stability of the discontinuous Galerkin method for the heat equation. *SIAM J. Numer. Anal.*, 34:389–401, 1997.
51. S. Osher. Riemann solvers, the entropy condition and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235, 1984.
52. C. Ozturan, H.L. deCougny, M.S. Shephard, and J.E. Flaherty. Parallel adaptive mesh refinement and redistribution on distributed memory computers. *Comput. Methods in Appl. Mech. and Engrg.*, 119:123–137, 1994.
53. T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28:133–140, 1991.
54. W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Scientific Laboratory Report LA-UR-73-479*, 1973.
55. G.R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Math. Comp.*, 50:75–88, 1988.

56. S.J. Sherwin and G. Karniadakis Tetrahedral hp finite elements: Algorithms and flow simulations. *J. Comput. Phys*, 124:314–45, 1996.
57. C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys*, 77:439–471, 1988.
58. C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes, ii. *J. Comput. Phys*, 83:32–78, 1989.
59. C.W. Shu. TVB uniformly high order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.
60. C.W. Shu. TVD time discretizations. *SIAM J. Sci. Stat. Comput.*, 9:1073–1084, 1988.
61. J.R. Trujillo. Effective High-Order Vorticity-Velocity Formulation. Ph.D., Princeton, 1997.
62. B. van Leer. Towards the ultimate conservation difference scheme, ii. *J. Comput. Phys*, 14:361–376, 1974.
63. B. van Leer. Towards the ultimate conservation difference scheme, v. *J. Comput. Phys*, 32:1–136, 1979.
64. D. Wagner. The Riemann problem in two space dimensions for a single conservation law. *SIAM J. Math. Anal.*, 14:534–559, 1983.
65. T.C. Warburton, I. Lomtev, R.M. Kirby and G.E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations on hybrid grids. *Center for Fluid Mechanics # 97-14*, Division of Applied Mathematics, Brown University, 1997.
66. P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys*, 54:115–173, 1984.
67. T. Zhang and G.Q.. Chen. Some fundamental concepts about systems of two spatial dimensional conservation laws. *Acta Math. Sci. (English Ed.)*, 6:463–474, 1986.
68. T. Zhang and Y.X. Zheng. Two dimensional Riemann problems for a single conservation law. *Trans. Amer. Math. Soc.*, 312:589–619,1989.

J Sci Comput (2018) 77:1621–1659
https://doi.org/10.1007/s10915-018-0750-6

Globally Divergence-Free Discontinuous Galerkin Methods for Ideal Magnetohydrodynamic Equations

Pei Fu[1] · **Fengyan Li**[2] · **Yan Xu**[1]

Received: 3 October 2017 / Revised: 5 April 2018 / Accepted: 29 May 2018 / Published online: 8 June 2018

Abstract Ideal magnetohydrodynamic (MHD) equations are widely used in many areas in physics and engineering, and these equations have a divergence-free constraint on the magnetic field. In this paper, we propose high order globally divergence-free numerical methods to solve the ideal MHD equations. The algorithms are based on discontinuous Galerkin methods in space. The induction equation is discretized separately to approximate the normal components of the magnetic field on elements interfaces, and to extract additional information about the magnetic field when higher order accuracy is desired. This is then followed by an element by element reconstruction to obtain the globally divergence-free magnetic field. In time, strong-stability-preserving Runge–Kutta methods are applied. In consideration of accuracy and stability of the methods, a careful investigation is carried out, both numerically and analytically, to study the choices of the numerical fluxes associated with the electric field at element interfaces and vertices. The resulting methods are local and the approximated magnetic fields are globally divergence-free. Numerical examples are presented to demonstrate the accuracy and robustness of the methods.

Keywords MHD equations · Divergence-free magnetic field · Discontinuous Galerkin methods · $H(\mathrm{div})$-conforming finite element spaces · Fourier analysis

Research of F. Li is supported in part by NSF Grants DMS-1318409 and DMS-1719942. Research of Y. Xu is supported by NSFC Grant Nos. 11722112 and 91630207.

✉ Yan Xu
yxu@ustc.edu.cn

Pei Fu
sxfp2013@mail.ustc.edu.cn

Fengyan Li
lif@rpi.edu

[1] School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, Anhui, China

[2] Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

1 Introduction

In this paper, we will develop globally divergence-free discontinuous Galerkin (DG) methods to numerically simulate ideal magnetohydrodynamic (MHD) equations. MHD equations model ionized plasmas under some simplified assumptions and are widely used for describing many problems in physics and engineering. The ideal MHD equations considered in this work can be written as a system of nonlinear hyperbolic conservation laws, in addition to a divergence-free constraint on the magnetic field. Even though the magnetic field in the exact solution satisfies the divergence-free condition as long as it does initially, insufficient preservation of this property numerically may lead to numerical instability or nonphysical features of approximated solutions [8,9,19,37].

To handle the divergence-free constraint, various strategies have been developed in numerical modeling and mathematical analysis within divergence-cleaning or divergence-free algorithms. In [9], Brackbill and Barnes proposed a simple divergence correction technique based on Hodge decomposition. They projected the computed magnetic field into a divergence-free vector space by solving a Poisson equation and used the divergence-free magnetic field in the next time step. One widely used framework to achieve the preservation of the divergence, in some discrete or continuous sense, is "constrained transport", introduced by Yee [40] in the context of the electromagnetism, and adapted by Evans and Hawley [19] to MHD simulations. This idea was further developed by many researchers within frameworks of finite difference, finite volume, and finite element methods, either upwind (also called Godunov) or central types, and with various accuracy [8,17,20,26,29]. Among the developments, there are exactly divergence-free numerical methods [1,3,26,28,29,35]. Other approaches which attract different practitioners include Powell's source term formulation [30] by adding source terms depending on $\nabla \cdot \mathbf{B}$, and generalized multiplier methods [18] with divergence cleaning technique.

In recent years, Li et al. [25–27,38] developed divergence-free numerical methods for ideal MHD equations based on DG and central DG spatial discretizations. In [25], locally divergence-free DG methods were formulated, and they utilize divergence-free vector spaces inside each mesh element to approximate the magnetic field. In [26,27], exactly divergence-free central DG methods were proposed for ideal MHD equations, and the methods can be of arbitrary accuracy. The discrete space to represent and to compute the magnetic field is a divergence-free subspace of the Brezzi–Douglas–Marini (BDM) finite element space [10], a well-established $H(\mathrm{div})$-conforming finite element space. DG method was first introduced in 1973 by Reed and Hill for linear neutron transport problems [33]. A major breakthrough was made by Cockburn et al. [13–16] to develop DG spatial discretizations for nonlinear hyperbolic conservation laws, coupled with high order Runge–Kutta methods in time. Exact or approximate Riemann solvers are used as numerical fluxes at element interfaces, and total variation bounded (TVB) nonlinear limiters [36] are applied in the presence of strong shocks to achieve non-oscillatory property. With their great flexibility in local approximations and geometry, local conservation, and high parallel efficiency, DG methods since then have been formulated and analyzed to various mathematical models, with broad applications in areas such as electromagnetism, gas dynamics, granular flow, plasma physics etc. One can refer to [22,24,34] for a more systematic description of the methods as well as their implementation and applications.

Our present work follows the development of exactly divergence-free central DG methods for ideal MHD equations in [26,27], and it is related to the exactly divergence-free DG methods for the induction equation using multi-dimensional Riemann solvers [7]. On the one

hand, the methods in [26,27] achieve exactly divergence-free approximations for the magnetic field within a relatively simple formulation due to that the methods involve two copies of numerical solutions from two overlapping meshes, and no numerical fluxes are needed either on element interfaces or at mesh vertices. On the other hand, two copies of numerical solutions double the total number of unknowns and hence increase the computational complexity of the algorithms. In this work, we want to design exactly divergence-free DG methods that are defined on one mesh, which is structured, for ideal MHD equations in two dimensions. Similar as in central DG framework, our new methods will discretize the hydrodynamic variables, such as density, momentum, total energy using standard DG methods, while the equations evolving the magnetic fields, referred to as the induction equation, will be discretized differently by DG-type methods. More specifically, the normal components of the magnetic field along element edges will be updated first by DG methods defined on edges, and this is followed by an element-wise reconstruction to produce an exactly divergence-free magnetic field. For higher order accuracy, additional information will be computed for the magnetic field, and it will be used together with the normal components of the magnetic field to uniquely determine the reconstruction. It turns out that the entire algorithm to discretize the induction equation to obtain the magnetic field approximation can be equivalently reformulated to a form without any reconstruction. The magnetic fields will still be approximated by the exactly divergence-free $H(\mathrm{div})$-conforming BDM finite element functions as in [26,27] (see Sect. 2.2 for comments on the use of general $H(\mathrm{div})$-conforming finite element spaces), and the new challenge comes from the need for numerical fluxes to approximate the electric field on element interfaces and at vertices.

It is known that the choices of numerical fluxes play an important role for the accuracy and stability of DG methods. To finalize our methods, we first identify two necessary conditions (see Theorem 3.1) on the numerical fluxes used in the different parts of the numerical methods, to ensure the reconstructed magnetic field is exactly divergence-free. We then adapt the proposed methods to a constant coefficient linear model, the induction equation with a given constant velocity field, and carry out both a numerical study and a Fourier analysis, to learn about the choices of numerical fluxes for the electric field especially at the mesh vertices, and their roles to the accuracy and numerical stability of the methods. Even though such study is only for a linear model for the magnetic field, the experience we have with it informs us how to choose numerical fluxes (see Sect. 4.3) for the proposed schemes to solve the full ideal MHD equations accurately and robustly. Our final choice of the electric field flux at mesh vertices is one type of multi-dimensional Riemann solver used in [7]. Our numerical tests in Sect. 4.1 imply that multi-dimensional Riemann solvers, when they introduce enough numerical dissipation, can make a good approximation to the electric field flux at vertices. Multi-dimensional Riemann solvers have been used within the WENO finite volume method frameworks in [4–6] to solve ideal MHD equations.

The rest of this paper is organized as follows. In Sect. 2, we describe the ideal MHD equations and introduce notations for meshes and discrete spaces. In Sect. 3, we present the proposed DG methods, and identify the conditions on the numerical fluxes to ensure the overall algorithms to be exactly divergence-free. In order to know what choices of numerical fluxes, especially for the electric field on element interfaces and at vertices, will render accurate and stable algorithms, in Sect. 4 we adapt the proposed methods to the induction equation and carry out numerical and analytical studies. In Sect. 5, nonlinear limiters are discussed, and the entire algorithm is also presented. Numerical examples are presented in Sect. 6 to illustrate the performance of the proposed methods, and this is followed by concluding remarks in Sect. 7.

2 MHD Equations, Notations and Discrete Spaces

2.1 MHD Equations

We consider the ideal MHD equations consisting of a set of nonlinear hyperbolic conservations laws

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot \left[\rho \mathbf{u}\mathbf{u}^{\mathrm{T}} + \left(p + \frac{1}{2}|\mathbf{B}|^2 \right) \mathbf{I} - \mathbf{B}\mathbf{B}^{\mathrm{T}} \right] = 0, \tag{2}$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) = 0, \tag{3}$$

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[\left(\mathcal{E} + p + \frac{1}{2}|\mathbf{B}|^2 \right) \mathbf{u} - \mathbf{B}(\mathbf{u} \cdot \mathbf{B}) \right] = 0, \tag{4}$$

with a divergence-free constraint

$$\nabla \cdot \mathbf{B} = 0. \tag{5}$$

Here ρ is the density, p is the hydrodynamic pressure, $\mathbf{u} = (u_x, u_y, u_z)^{\mathrm{T}}$ is the velocity, and $\mathbf{B} = (B_x, B_y, B_z)^{\mathrm{T}}$ is the magnetic field. The total energy is given by $\mathcal{E} = \frac{1}{2}\rho|\mathbf{u}|^2 + \frac{1}{2}|\mathbf{B}|^2 + \frac{p}{\gamma - 1}$ with γ as the ratio of the specific heats. The superscript T denotes the vector transpose. $\mathbf{I}$ is the identity matrix, $\nabla\cdot$ is the divergence operator, and $\nabla\times$ is the curl operator. In two dimensions, all functions depend on the spatial variables x and y. Hence only B_x and B_y contribute to $\nabla \cdot \mathbf{B}$. The Eqs. (1)–(4) can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}, \mathcal{B}) = 0, \tag{6}$$

$$\frac{\partial \mathcal{B}}{\partial t} + \widehat{\nabla} \times E_z(\mathbf{U}, \mathcal{B}) = 0, \tag{7}$$

where $\mathbf{U} = (\rho, \rho u_x, \rho u_y, \rho u_z, B_z, \mathcal{E})^{\mathrm{T}}$, $\mathcal{B} = (B_x, B_y)^{\mathrm{T}}$, and $\mathbf{F} = (F_1, F_2)$ with

$$\begin{aligned} F_1(\mathbf{U}, \mathcal{B}) = \Bigg(& \rho u_x, \rho u_x^2 + p + \frac{1}{2}|\mathbf{B}|^2 - B_x^2, \rho u_x u_y - B_x B_y, \rho u_x u_z - B_x B_z, \\ & u_x B_z - u_z B_x, u_x \left(\mathcal{E} + p + \frac{1}{2}|\mathbf{B}|^2 \right) - B_x (\mathbf{u} \cdot \mathbf{B}) \Bigg)^{\mathrm{T}}, \end{aligned} \tag{8}$$

$$\begin{aligned} F_2(\mathbf{U}, \mathcal{B}) = \Bigg(& \rho u_y, \rho u_x u_y - B_x B_y, \rho u_y^2 + p + \frac{1}{2}|\mathbf{B}|^2 - B_y^2, \rho u_y u_z - B_y B_z, \\ & u_y B_z - u_z B_y, u_y \left(\mathcal{E} + p + \frac{1}{2}|\mathbf{B}|^2 \right) - B_y (\mathbf{u} \cdot \mathbf{B}) \Bigg)^{\mathrm{T}}. \end{aligned} \tag{9}$$

In addition, $E_z(\mathbf{u}, \mathcal{B}) = u_y B_x - u_x B_y$ which is the z-component of the electric field $\mathbf{E} = -\mathbf{u} \times \mathbf{B}$, and $\widehat{\nabla} \times E_z = (\frac{\partial E_z}{\partial y}, -\frac{\partial E_z}{\partial x})^{\mathrm{T}}$ is the first two components of $\nabla \times (0, 0, E_z)^{\mathrm{T}}$. Without confusion, we will refer to $\mathcal{B}$ as the magnetic field.

2.2 Notations and Discrete Spaces

In this subsection, notations and discrete spaces for numerical schemes are introduced. We assume the computational domain is $\Omega = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \subset \mathbb{R}^d$ with $d = 2$. Let

$\{x_i\}_i$ and $\{y_j\}_j$ be the partitions of $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$, respectively. We define $x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$, $y_{j+\frac{1}{2}} = \frac{1}{2}(y_j + y_{j+1})$ and $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ as an rectangle element with (x_i, y_j) as the center. Let $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, $\Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$, and let $\mathcal{T}_h = \bigcup_{ij} I_{ij}$ be a partition of the domain Ω.

The discrete spaces are defined over the mesh. For variable $\mathbf{U}$, we use the piecewise polynomial vector space

$$\mathcal{V}_h^k = \left\{ \mathbf{v} : \mathbf{v}|_K \in [P^k(K)]^{8-d}, \forall K \in \mathcal{T}_h \right\}, \tag{10}$$

where $P^k(K)$ is the space of polynomials with the total degree at most k in K, and $[P^k(K)]^n$ is its vector version. For the magnetic field $\mathcal{B}$, we approximate it using globally (also called exactly) divergence-free polynomial functions, which are piecewise divergence-free with continuous normal components across element interfaces. This space is defined as

$$\begin{aligned} \mathcal{M}_h^k &= \left\{ \mathbf{v} \in H(\mathrm{div}^0; \Omega) : \mathbf{v}|_K \in \mathcal{W}^k(K), \forall K \in \mathcal{T}_h \right\} \\ &= \Big\{ \mathbf{v} : \mathbf{v}|_K \in \mathcal{W}^k(K), \nabla \cdot \mathbf{v}|_K = 0, \forall K \in \mathcal{T}_h, \\ &\qquad \text{and the normal component of } \mathbf{v} \text{ is continuous across each element interface} \Big\}, \end{aligned} \tag{11}$$

with $\mathcal{W}^k(K)$ defined as

$$\mathcal{W}^k(K) = [P^k(K)]^d \oplus \mathrm{span}\left\{ \widehat{\nabla} \times (x^{k+1} y), \widehat{\nabla} \times (x y^{k+1}) \right\}. \tag{12}$$

$\mathcal{M}_h^k$ is the divergence-free subspace of the $H(\mathrm{div})$-conforming Brezzi–Douglas–Marini (BDM) finite element space

$$\mathrm{BDM}^k = \left\{ \mathbf{v} \in H(\mathrm{div}) : \mathbf{v}\,|_K \in \mathcal{W}^k(K), \forall K \in \mathcal{T}_h \right\}, \tag{13}$$

and it has optimal accuracy to approximate functions in $H(\mathrm{div}^0) = \{\mathbf{v} \in [L^2(\Omega)]^d : \nabla \cdot \mathbf{v} = 0\}$ [10]. As pointed out in [26], divergence-free subspaces of other $H(\mathrm{div})$-conforming finite element spaces, such as Brezzi–Douglas–Fortin–Marini (BDFM) [11] or Raviart–Thomas (RT) [32] finite element spaces can also be used to provide divergence-free approximations for the magnetic field by following the same framework proposed in the present paper. The BDM finite element space is chosen here as it is the smallest among these candidates to achieve the same order of accuracy in the L^2 norm.

3 Proposed Numerical Methods for Ideal MHD Equations

In this section, we will formulate the DG methods with a globally divergence-free magnetic field to solve the MHD equations (6)–(7). For simplicity, we use the forward Euler method as time discretization to present the schemes. For high order accuracy in time, strong-stability-preserving Runge–Kutta methods will be used [21]. Such time integrators can be expressed as convex combinations of the forward Euler method, and hence they preserve the globally divergence-free property of the magnetic field. To describe the proposed methods, we assume the numerical solutions at time $t = t_n$ are available, that is $(\mathbf{U}_h^n, \mathcal{B}_h^n) \in \mathcal{V}_h^k \times \mathcal{M}_h^k$ with $\mathcal{B}_h^n = (B_{x,h}^n, B_{y,h}^n)^{\mathrm{T}}$. We want to compute the numerical solutions at $t_{n+1} = t_n + \Delta t$, denoted as $(\mathbf{U}_h^{n+1}, \mathcal{B}_h^{n+1}) \in \mathcal{V}_h^k \times \mathcal{M}_h^k$ with $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}}$.

3.1 DG Methods to Update $\mathbf{U}_h^{n+1}$

We update the variable $\mathbf{U}_h^{n+1}$ by applying to (6) the standard DG method [16] as the spatial discretization and forward Euler method as the time discretization. That is, we look for $\mathbf{U}_h^{n+1} \in \mathcal{V}_h^k$, such that for any $\mathbf{w} \in \mathcal{V}_h^k$ and any element $I_{ij} \in \mathcal{T}_h$,

$$\int_{I_{ij}} \mathbf{U}_h^{n+1} \cdot \mathbf{w} dxdy = \int_{I_{ij}} \mathbf{U}_h^n \cdot \mathbf{w} dxdy - \Delta t \left(\int_{\partial I_{ij}} \mathbf{H}_{\mathbf{e},\mathbf{I_{ij}}} \left(\mathbf{v}^{int(I_{ij})}, \mathbf{v}^{ext(I_{ij})}; \mathbf{n} \right) \cdot \mathbf{w} ds - \int_{I_{ij}} \mathbf{F}(\mathbf{U}_h^n, \mathcal{B}_h^n) \cdot \nabla \mathbf{w} dxdy \right). \tag{14}$$

Here, $\mathbf{H}_{\mathbf{e},\mathbf{I_{ij}}}(\mathbf{v}^{int(I_{ij})}, \mathbf{v}^{ext(I_{ij})}; \mathbf{n})$ is the numerical flux to approximate $\mathbf{F}(\mathbf{U}, \mathcal{B}) \cdot \mathbf{n}$, and $\mathbf{n} = (n_1, n_2)^{\mathrm{T}}$ is the outward normal vector of an edge e of the element I_{ij}. $\mathbf{v}$ is a symbol which denotes the variables $(\mathbf{U}_h^n, \mathcal{B}_h^n)$, and $\mathbf{v}^{int(I_{ij})}$, $\mathbf{v}^{ext(I_{ij})}$ are the limits of $\mathbf{v}$ from the interior and exterior of an element I_{ij} along its edge e. In our simulation, we take the Lax–Friedrichs numerical flux

$$\mathbf{H}_{\mathbf{e},\mathbf{I_{ij}}}(\mathbf{a}, \mathbf{b}; \mathbf{n}) = \frac{1}{2} \left(\mathbf{F}(\mathbf{a}) \cdot \mathbf{n} + \mathbf{F}(\mathbf{b}) \cdot \mathbf{n} - \alpha(\mathbf{b} - \mathbf{a}) \right), \tag{15}$$

where α is an estimate of the maximal absolute eigenvalue of the Jacobian $\frac{\partial \mathbf{F}(\mathbf{U},\mathcal{B}) \cdot \mathbf{n}}{\partial (\mathbf{U},\mathcal{B})}$ in the neighborhood of the edge e.

3.2 DG Methods for Globally Divergence-Free Magnetic Field $\mathcal{B}$

In this subsection, we present DG methods for the induction equation (7) to generate a globally divergence-free approximation $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}} \in \mathcal{M}_h^k$ for the magnetic field $\mathcal{B}$. It is known that a piecewise divergence-free vector field is globally divergence-free if its normal component is continuous on element interfaces. Therefore, we first approximate the normal component of the magnetic field $\mathcal{B} \cdot \mathbf{n}$ on element interfaces based on the DG methods (see Sect. 3.2.1). Then, an element by element reconstruction is used to reconstruct the globally divergence-free magnetic field (see Sect. 3.2.3). When $k \geq 2$, more information about the magnetic field is obtained by approximating the two-dimensional system (7) using a standard DG method that is *less* accurate (see Sect. 3.2.2). In Sect. 3.2.4, we will present a reformulation of the schemes, equivalent to that in Sects. 3.2.1–3.2.3 to update the magnetic field yet free of reconstruction. Throughout this subsection, E_z in numerical schemes and its related numerical fluxes are from time t_n.

3.2.1 Approximation of $\mathcal{B} \cdot \mathbf{n}$ on the Element Interfaces

To get the continuous normal component $\mathcal{B} \cdot \mathbf{n}$ of the magnetic field, we formulate a DG-type scheme for magnetic field equations on the element interfaces. For the rectangular mesh, $\mathcal{B} \cdot \mathbf{n}$ is $B_{x,h}^{n+1}$ along y-direction edges with $\mathbf{n} = (1, 0)^{\mathrm{T}}$, and it is $B_{y,h}^{n+1}$ along the x-direction edges with $\mathbf{n} = (0, 1)^{\mathrm{T}}$.

To propose the DG method for Eq. (7) on the element interface, we consider the equation

$$\frac{\partial \mathcal{B} \cdot \mathbf{n}}{\partial t} + \widehat{\nabla} \times E_z(\mathbf{U}, \mathcal{B}) \cdot \mathbf{n} = 0. \tag{16}$$

To this end, on a rectangular mesh, we need to consider two one-dimensional equations of the system (16)

$$\frac{\partial B_x}{\partial t} + \frac{\partial E_z}{\partial y} = 0, \tag{17}$$

$$\frac{\partial B_y}{\partial t} - \frac{\partial E_z}{\partial x} = 0. \tag{18}$$

We use the DG method as the spatial discretization and forward Euler method as the time discretization for Eqs. (17) and (18) on element interfaces. The method is as follows: look for $b_{ij}^x(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$, such that for any $\varphi(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$

$$\begin{aligned}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{ij}^x(y)\varphi(y)dy &= \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_x^n(x_{i+\frac{1}{2}}, y)\varphi(y)dy \\ &- \Delta t\left(\widehat{E_z}(x_{i+\frac{1}{2}}, y)\varphi(y)\Big|_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \overline{E_z}(x_{i+\frac{1}{2}}, y)\frac{\partial\varphi(y)}{\partial y}dy\right),\end{aligned} \tag{19}$$

and look for $b_{ij}^y(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$, such that for any $\varphi(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$

$$\begin{aligned}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b_{ij}^y(x)\varphi(x)dx &= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_y^n(x, y_{j+\frac{1}{2}})\varphi(x)dx \\ &- \Delta t\left(\widehat{\overline{-E_z}}(x, y_{j+\frac{1}{2}})\varphi(x)\Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \overline{\overline{-E_z}}(x, y_{j+\frac{1}{2}})\frac{\partial\varphi(x)}{\partial x}dx\right).\end{aligned} \tag{20}$$

Here b_{ij}^x and b_{ij}^y denote the approximations of $B_x(x_{i+\frac{1}{2}}, y)$ for $y \in [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ and $B_y(x, y_{j+\frac{1}{2}})$ for $x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ at time $t = t_{n+1}$, respectively. $\widehat{E_z}$ and $\widehat{\overline{-E_z}}$ are exact or approximate Riemann solvers to approximate the electric field flux E_z at the vertices of a mesh element, while $\overline{E_z}$, $\overline{\overline{-E_z}}$ are exact or approximate Riemann solvers to approximate E_z on the element interfaces, and their choices will be discussed in Theorem 3.1 and specified in Sect. 4.3. $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$ will be used to reconstruct the globally divergence-free magnetic field.

3.2.2 Additional Information for the Magnetic Field $\mathfrak{B}$: $\widetilde{\mathfrak{B}}_h$ in Mesh Elements

When $k \geqslant 2$, $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$ do not provide enough information to reconstruct a two-dimensional function in $\mathcal{M}_h^k$. For more information, a standard DG method with lower accuracy is applied to the two-dimensional system (7). For $k \geqslant 2$, we look for

$\widetilde{\mathcal{B}}_h \in [P^{k-2}(I_{ij})]^2$ such that for any $\Phi \in [P^{k-2}(I_{ij})]^2$ with $\Phi = (\Phi_1, \Phi_2)^{\mathrm{T}}$,

$$\begin{aligned}\int_{I_{ij}} \widetilde{\mathcal{B}}_h \cdot \Phi dxdy &= \int_{I_{ij}} \mathcal{B}_h^n \cdot \Phi dxdy \\ &- \Delta t\Bigg(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j-\frac{1}{2}})dx \\ &+ \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i-\frac{1}{2}}, y)dy \\ &- \int_{I_{ij}} \left(E_z \frac{\partial \Phi_1}{\partial y} - E_z \frac{\partial \Phi_2}{\partial x} \right) dxdy \Bigg).\end{aligned} \tag{21}$$

Here $\widetilde{E_z}$ is the numerical flux for $E_z = (0, E_z)^{\mathrm{T}} \cdot \mathbf{n}$ with $\mathbf{n} = (0, 1)^{\mathrm{T}}$ along an x-direction edge, and $\widetilde{\widetilde{-E_z}}$ is the numerical flux for $-E_z = (-E_z, 0)^{\mathrm{T}} \cdot \mathbf{n}$ with $\mathbf{n} = (1, 0)^{\mathrm{T}}$ along a y-direction edge. Both $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ will be taken as the one-dimensional Lax–Friedrichs flux (15). It will be seen from Theorem 3.1 that the numerical fluxes $\overline{E_z}$ and $\overline{\overline{-E_z}}$ in (19)–(20) need to be related to $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ in order to ensure the globally divergence-free reconstruction, also see Sect. 4.3.

3.2.3 Reconstruct the Globally Divergence-Free Magnetic Field $\mathcal{B}_h^{n+1}$

Once we have $\{b_{ij}^x\}_{ij}$, $\{b_{ij}^y\}_{ij}$ on element interfaces from (19) and (20) as well as $\widetilde{\mathcal{B}}_h$ from (21), we will follow the idea of the BDM projection [10] (also see [26,27]) to carry out an element-by-element reconstruction of a globally divergence-free magnetic field $\mathcal{B}_h^{n+1}$. Given an element I_{ij}, the reconstruction is to obtain $\mathcal{B}_h^{n+1}|_{I_{ij}} \in \mathcal{W}^k(I_{ij})$ on I_{ij}, such that $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}}$ satisfies

R1 $\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left(B_{x,h}^{n+1}(x_{l+\frac{1}{2}}, y) - b_{lj}^x(y) \right) \varphi(y)dy = 0$ on the y-direction edge with $l = i-1, i$ and any $\varphi(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$,

R2 $\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(B_{y,h}^{n+1}(x, y_{l+\frac{1}{2}}) - b_{il}^y(x) \right) \varphi(x)dx = 0$ on the x-direction edge with $l = j-1, j$ and any $\varphi(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$,

R3 $\int_{I_{ij}} \left(\mathcal{B}_h^{n+1}(x, y) - \widetilde{\mathcal{B}}_h(x, y) \right) \Phi(x, y)dxdy = 0$ for any $\Phi(x, y) \in [P^{k-2}(I_{ij})]^2$ when $k \geq 2$.

From the reconstruction, one can see that the normal component of the magnetic field $\mathcal{B}_h^{n+1}$, given by $\{b_{ij}^x\}_{ij}$ or $\{b_{ij}^y\}_{ij}$, is single-valued, and hence it is continuous on element interfaces. When $k \geq 2$, additional information is provided by $\widetilde{\mathcal{B}}_h$ via L^2 projection. In the next theorem, we will show that the reconstruction produces a globally divergence-free approximation for the magnetic field under some conditions for the numerical fluxes in schemes (19)–(21).

Theorem 3.1 *Under the conditions that*

1. *the electric field flux approximations in* (19)–(21) *along the same edge satisfy*

$$\overline{E_z} = -(\widetilde{\widetilde{-E_z}}), \quad \overline{\overline{-E_z}} = -(\widetilde{E_z}), \tag{22}$$

2. *and the electric field flux approximations in* (19)–(20) *at the same vertex is single-valued, satisfying*

$$\widehat{\widehat{-E_z}} = -\widehat{E_z}, \tag{23}$$

then for any $k \geq 0$, *the reconstructed* $\mathcal{B}_h^{n+1}(I_{ij})$ *exists uniquely in* $\mathcal{W}^k(I_{ij})$. *In addition,* $\nabla \cdot \mathcal{B}_h^{n+1}|_{I_{ij}} = 0$.

Proof One can follow the same proof as in [26] to show the unique existence of the reconstructed $\mathcal{B}_h^{n+1}(I_{ij}) \in \mathcal{W}^k(I_{ij})$. We here will only show the divergence-free property of $\mathcal{B}_h^{n+1}$.

For any $\omega \in P^{k-1}(I_{ij})$, from the reconstruction step **R3** and equation (21), we have

$$\begin{aligned}\int_{I_{ij}} \mathcal{B}_h^{n+1}\nabla\omega dxdy &= \int_{I_{ij}} \widetilde{\mathcal{B}}_h\nabla\omega dxdy \\ &= \int_{I_{ij}} \mathcal{B}_h^{n}\nabla\omega dxdy - \Delta t\left(\Theta_{inside} - \int_{I_{ij}}\left(E_z\frac{\partial^2\omega}{\partial x\partial y} - E_z\frac{\partial^2\omega}{\partial y\partial x}\right)dxdy\right) \\ &= \int_{I_{ij}} \mathcal{B}_h^{n}\nabla\omega dxdy - \Delta t\Theta_{inside},\end{aligned} \tag{24}$$

where

$$\begin{aligned}\Theta_{inside} = &\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\widetilde{E_z}\frac{\partial\omega}{\partial x}\right)(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\widetilde{E_z}\frac{\partial\omega}{\partial x}\right)(x, y_{j-\frac{1}{2}})dx \\ &+ \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\left(\widetilde{\widetilde{-E_z}}\frac{\partial\omega}{\partial y}\right)(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\left(\widetilde{\widetilde{-E_z}}\frac{\partial\omega}{\partial y}\right)(x_{i-\frac{1}{2}}, y)dy,\end{aligned}$$

and $\mathcal{B}_h^n \in \mathcal{M}_h^k$ is the globally divergence-free approximation of $\mathcal{B}$ at time t_n.

From the reconstruction steps **R1** and **R2**, we have

$$\begin{aligned}\int_{\partial I_{ij}} \mathcal{B}_h^{n+1}\cdot\mathbf{n}\omega ds = &\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b_{ij}^y(x)\omega(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b_{ij-1}^y(x)\omega(x, y_{j-\frac{1}{2}})dx \\ &+ \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{ij}^x(y)\omega(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{i-1j}^x(y)\omega(x_{i-\frac{1}{2}}, y)dy.\end{aligned} \tag{25}$$

With the schemes (19) and (20), we further get

$$\int_{\partial I_{ij}} \mathcal{B}_h^{n+1}\cdot\mathbf{n}\omega ds = \int_{\partial I_{ij}} \mathcal{B}_h^{n}\cdot\mathbf{n}\omega ds + \Delta t(\Theta_{edge} - \Theta_{vertex}), \tag{26}$$

with

$$\begin{aligned}\Theta_{edge} = &\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\left(\overline{E_z}\frac{\partial\omega}{\partial y}\right)(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\left(\overline{E_z}\frac{\partial\omega}{\partial y}\right)(x_{i-\frac{1}{2}}, y)dy \\ &+ \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\overline{\overline{-E_z}}\frac{\partial\omega}{\partial x}\right)(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\overline{\overline{-E_z}}\frac{\partial\omega}{\partial x}\right)(x, y_{j-\frac{1}{2}})dx,\end{aligned}$$

and

$$\begin{aligned}\Theta_{vertex} = &\left(\widehat{E_z\omega}\right)(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - \left(\widehat{E_z\omega}\right)\left(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}\right)\\ &- \left(\widehat{E_z\omega}\right)(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) + \left(\widehat{E_z\omega}\right)\left(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}\right)\\ &+ \left(\widehat{\widehat{-E_z\omega}}\right)(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - \left(\widehat{\widehat{-E_z\omega}}\right)\left(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}\right)\\ &- \left(\widehat{\widehat{-E_z\omega}}\right)(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + \left(\widehat{\widehat{-E_z\omega}}\right)\left(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}\right).\end{aligned}$$

Under the condition in (23) that the electric field flux approximations at vertices are single-valued, we have $\Theta_{vertex} = 0$. Moreover, under the condition (22), we get $\Theta_{edge} + \Theta_{inside} = 0$. Now we can apply Gauss theorem, utilize the relations in (24) and (26), and get

$$\begin{aligned}\int_{I_{ij}} \nabla\cdot\mathcal{B}_h^{n+1}\omega dxdy &= \int_{\partial I_{ij}} \mathcal{B}_h^{n+1}\cdot\mathbf{n}\omega ds - \int_{I_{ij}} \mathcal{B}_h^{n+1}\nabla\omega dxdy\\ &= \int_{\partial I_{ij}} \mathcal{B}_h^{n}\cdot\mathbf{n}\omega ds - \int_{I_{ij}} \mathcal{B}_h^{n}\nabla\omega dxdy + \Delta t(\Theta_{edge} + \Theta_{inside} - \Theta_{vertex})\\ &= \int_{I_{ij}} \nabla\cdot\mathcal{B}_h^{n}\omega dxdy + \Delta t(\Theta_{edge} + \Theta_{inside} - \Theta_{vertex}) = 0. \end{aligned} \tag{27}$$

Here we have used the fact that $\nabla\cdot\mathcal{B}_h^n = 0$ at time t_n. Finally, note that $\nabla\cdot\mathcal{B}_h^{n+1} \in P^{k-1}(I_{ij})$, by taking $\omega = \nabla\cdot\mathcal{B}_h^{n+1}$ in (27), we conclude $\nabla\cdot\mathcal{B}_h^{n+1} = 0$. □

Remark 3.2 Two conditions (22)–(23) are needed to ensure the exactly divergence-free reconstructions. The one in (23) that requires a single-valued electric field flux approximation at vertices has long been used for many constrained transport methods in various frameworks such as finite difference and finite volume methods, while the condition in (22) is needed only in finite element type of methods including DG methods. Both conditions can be avoided if central DG methods are used, see [26,27].

3.2.4 Equivalent form of Numerical Schemes for $\mathcal{B}_h^{n+1}$: Without Reconstruction

From the reconstruction **R1**–**R3** in Sect. 3.2.3, one can see that the normal components of $\mathcal{B}_h^{n+1}$ along the edges of an element are identical to b_{ij}^x and b_{ij}^y (at most up to a sign difference, or a shift in index i or j), and its L^2 projection onto $[P^{k-2}(I_{ij})]^2$ is identical to $\widetilde{\mathcal{B}}_h$. Therefore the schemes to compute the globally divergence-free $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}} \in \mathcal{M}_h^k$ in Sects. 3.2.1–3.2.3 can be rewritten into an equivalent formulation as follows, *without* any reconstruction: look for $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}}$ such that $\mathcal{B}_h^{n+1}|_{I_{ij}} \in \mathcal{W}^k(I_{ij})$ for any i, j, satisfying

$$\begin{aligned}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_{x,h}^{n+1}(x_{l+\frac{1}{2}}, y)\varphi(y)dy &= \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_x^{n}(x_{l+\frac{1}{2}}, y)\varphi(y)dy\\ &- \Delta t\left(\widehat{E_z}(x_{l+\frac{1}{2}}, y)\varphi(y)\Big|_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \overline{E_z}(x_{l+\frac{1}{2}}, y)\frac{\partial\varphi(y)}{\partial y}dy\right)\end{aligned} \tag{28}$$

for any $\varphi(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$ and with $l = i-1, i$;

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_{y,h}^{n+1}(x, y_{l+\frac{1}{2}})\varphi(x)dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_y^n(x, y_{l+\frac{1}{2}})\varphi(x)dx - \Delta t\left(\widehat{\widehat{-E_z}}(x, y_{l+\frac{1}{2}})\varphi(x)\Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \overline{\overline{-E_z}}(x, y_{l+\frac{1}{2}})\frac{\partial\varphi(x)}{\partial x}dx\right) \tag{29}$$

for any $\varphi(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$ and with $l = j-1, j$; in addition,

$$\begin{aligned}\int_{I_{ij}} \mathcal{B}_h^{n+1} \cdot \Phi dxdy = \int_{I_{ij}} \mathcal{B}_h^n \Phi dxdy \\ - \Delta t\Big(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j-\frac{1}{2}})dx \\ + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i-\frac{1}{2}}, y)dy \\ - \int_{I_{ij}} \left(E_z\frac{\partial\Phi_1}{\partial y} - E_z\frac{\partial\Phi_2}{\partial x}\right)dxdy\Big)\end{aligned} \tag{30}$$

for any $\Phi \in [P^{k-2}(I_{ij})]^2$ with $\Phi = (\Phi_1, \Phi_2)^{\mathrm{T}}$. Again the numerical fluxes will satisfy the two conditions (22)–(23). Theorem 3.1 ensures that the resulting magnetic field $\mathcal{B}_h^{n+1}$ is in $\mathcal{M}_h^k$ and hence globally divergence-free. (One should refer to equations (5.4) and (5.6) in [10] for a more direct analysis.)

Even though the reformulation of the schemes in this subsection is more straightforward, in the presence of strong discontinuities in the solutions, nonlinear limiters need to be applied to all unknowns, including the magnetic field (see Sect. 5 and the numerical example of cloud–shock interaction in Sect. 6.2.5). When nonlinear limiters are needed for the magnetic field, it is more flexible to work with the schemes in the formulation as in Sect. 3.2.1–3.2.3, so the limiters are applied *before* the reconstruction or a revised reconstruction, in order to still have a globally divergence-free approximation for the magnetic field.

4 How to Choose Electric Field Flux Approximations?

Theorem 3.1 suggests that electric field flux approximations used in the different parts of the proposed schemes (19)–(21) need to be single-valued at vertices and share the same formulas on the element interfaces. Just as in standard DG methods, choices of numerical fluxes are crucial for accuracy and robustness of the schemes. In this section, we want to investigate numerically and analytically on the choices of the electric field flux approximations. To this end, we will focus on the following equation for the magnetic field

$$\frac{\partial\mathcal{B}}{\partial t} + \widehat{\nabla} \times E_z(\mathbf{U}, \mathcal{B}) = 0. \tag{31}$$

Here $E_z = u_y B_x - u_x B_y$, with a constant velocity field (u_x, u_y) that is given. This system will be referred to as the induction equation. We will adapt the proposed schemes in Sect. 3.2 to the induction equation, and investigate numerically and analytically in next two subsections how different choices of electric field flux approximations affect accuracy and numerical

stability. Based on such study, in Sect. 4.3 we will specify our choices of the numerical fluxes in the proposed schemes (19)–(21) to compute the magnetic field.

4.1 Numerical Study

Adapting from the proposed schemes (19)–(21) and following the two required conditions (22)–(23) in Theorem 3.1, we consider the following schemes for the induction equation on the element interfaces, that is: look for $b_{ij}^x(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$, such that for any $\varphi(y) \in P^k([y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}])$

$$\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{ij}^x(y)\varphi(y)dy = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_x^n(x_{i+\frac{1}{2}}, y)\varphi(y)dy - \Delta t \left(\widehat{E_z}(x_{i+\frac{1}{2}}, y)\varphi(y)\Big|_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \widetilde{\widetilde{-E_z}}(x_{i+\frac{1}{2}}, y)\frac{\partial \varphi(y)}{\partial y}dy \right), \tag{32}$$

and look for $b_{ij}^y(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$, such that for any $\varphi(x) \in P^k([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}])$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b_{ij}^y(x)\varphi(x)dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_y^n(x, y_{j+\frac{1}{2}})\varphi(x)dx - \Delta t \Big(- \widehat{E_z}(x, y_{j+\frac{1}{2}})\varphi(x)\Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \widetilde{E_z}\left(x, y_{j+\frac{1}{2}}\right)\frac{\partial \varphi(x)}{\partial x}dx \Big). \tag{33}$$

Corresponding to (21), the induction equation is further discretized as a two-dimensional system when $k \geq 2$: look for $\widetilde{\mathcal{B}}_h \in [P^{k-2}(I_{ij})]^2$ such that for any $\Phi = (\Phi_1, \Phi_2)^2 \in [P^{k-2}(I_{ij})]^2$,

$$\begin{aligned}
\int_{I_{ij}} \widetilde{\mathcal{B}}_h \cdot \Phi dxdy &= \int_{I_{ij}} \mathcal{B}_h^n \cdot \Phi dxdy \\
&- \Delta t \Big(\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j+\frac{1}{2}})dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \big(\widetilde{E_z}\Phi_1\big)(x, y_{j-\frac{1}{2}})dx \\
&+ \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i+\frac{1}{2}}, y)dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \big(\widetilde{\widetilde{-E_z}}\Phi_2\big)(x_{i-\frac{1}{2}}, y)dy \\
&- \int_{I_{ij}} \left(E_z \frac{\partial \Phi_1}{\partial y} - E_z \frac{\partial \Phi_2}{\partial x} \right) dxdy \Big).
\end{aligned} \tag{34}$$

To help with the presentation, we illustrate the notations of states around a vertex **P**, its neighboring elements, and the connected edges in Fig. 1. In (34), also in (32)–(33), $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ are taken as a one-dimensional Lax–Friedrichs flux. Namely, along an x-direction edge,

$$\widetilde{E_z} = \frac{E_z^{LD} + E_z^{LU}}{2} - \frac{\alpha_y}{2}\left(B_x^{LU} - B_x^{LD}\right), \tag{35}$$

and along a y-direction edge,

$$\widetilde{\widetilde{-E_z}} = \frac{\left(-E_z^{RD} - E_z^{LD}\right)}{2} - \frac{\alpha_x}{2}\left(B_y^{RD} - B_y^{LD}\right). \tag{36}$$

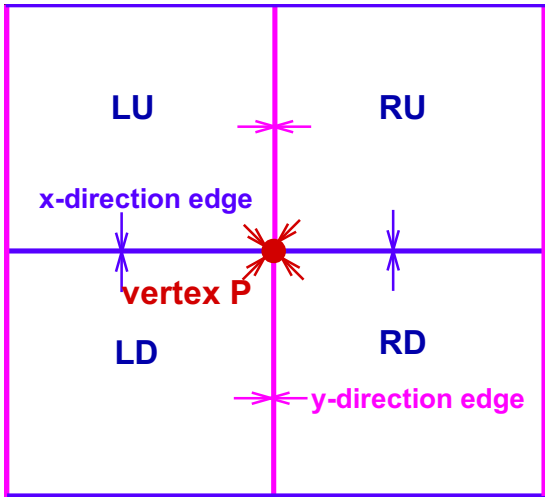


Fig. 1 The notations of states around a vertex **P**, its neighboring elements, and the connected edges

Here

$$\alpha_x = |u_x|, \quad \alpha_y = |u_y|, \tag{37}$$

and they are the largest absolute-value of eigenvalues of the Jacobian $\frac{\partial(0,-E_z)^{\mathrm{T}}}{\partial(B_x,B_y)^{\mathrm{T}}}$ and $\frac{\partial(E_z,0)^{\mathrm{T}}}{\partial(B_x,B_y)^{\mathrm{T}}}$, respectively.

Just as in [3,7,8,20], we use flux interpolations or approximate Riemann solvers to obtain the single-valued electric field flux $\widehat{E_z}$ at vertex **P** used in (32)–(33). Particularly, we take

$$\begin{aligned}
\widehat{E_z} &= \frac{1}{4}\left(\frac{E_z^{LD}+E_z^{LU}}{2} - \frac{\beta}{2}(B_x^{LU}-B_x^{LD})\right) \\
&\quad + \frac{1}{4}\left(\frac{E_z^{RD}+E_z^{RU}}{2} - \frac{\beta}{2}(B_x^{RU}-B_x^{RD})\right) \\
&\quad + \frac{1}{4}\left(\frac{E_z^{LD}+E_z^{RD}}{2} + \frac{\alpha}{2}(B_y^{RD}-B_y^{LD})\right) \\
&\quad + \frac{1}{4}\left(\frac{E_z^{LU}+E_z^{RU}}{2} + \frac{\alpha}{2}(B_y^{RU}-B_y^{LU})\right) \\
&= \frac{1}{4}\left(E_z^{LU}+E_z^{RU}+E_z^{LD}+E_z^{RD}\right) \\
&\quad - \frac{\beta}{4}\left(\frac{B_x^{LU}+B_x^{RU}}{2} - \frac{B_x^{LD}+B_x^{RD}}{2}\right) \\
&\quad + \frac{\alpha}{4}\left(\frac{B_y^{RD}+B_y^{RU}}{2} - \frac{B_y^{LD}+B_y^{LU}}{2}\right).
\end{aligned} \tag{38}$$

Here $\alpha = \sigma\alpha_x$ and $\beta = \sigma\alpha_y$, with the constant σ measures the amount of dissipation introduced by the numerical flux $\widehat{E_z}$, and α_x, α_y from (37). When $\alpha = \alpha_x$, $\beta = \alpha_y$, $\widehat{E_z}$ is the arithmetic average of the one-dimensional Lax–Friedrichs flux, namely, an average with equal weight, 1/4, of the numerical fluxes in (35)–(36) from four edges connected to the vertex **P**. When $\alpha = 1.2\alpha_x$ and $\beta = 1.2\alpha_y$, $\widehat{E_z}$ turns out to be the multi-dimensional HLL Riemann solver restricted at the vertex **P**, while $\widehat{E_z}$ with $\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$ is the

multi-dimensional Lax–Friedrichs Riemann solver restricted at **P**. Both multi-dimensional Riemann solvers were used in [7].

Next we want to investigate numerically how the different choices of $\widehat{E_z}$ will affect the performance of the numerical schemes (32)–(34) for the induction equation. We consider the same example as in [39], with the initial condition

$$(B_x, B_y) = (-\sin(2\pi y), \sin(2\pi x)),$$

and the constant velocity field is $(u_x, u_y) = (1, 1)$. Periodic boundary conditions are used. This example is computed on the domain $[0, 1] \times [0, 1]$ based on P^k approximations with $k = 0, 1, 2$. The third order TVD Runge–Kutta time discretization in (66) [21] is applied in time. The time step is determined as $\Delta t \leq CFL/\left(\frac{|u_x|}{\Delta x} + \frac{|u_y|}{\Delta y}\right)$ where the Courant–Friedrichs–Lewy (CFL) number CFL is taken to be 0.5, 0.2, 0.1 for $k = 0, 1, 2$, respectively. Table 1 shows the L^2 errors and orders of accuracy for the magnetic field component B_x at $t = 1.0$ and $t = 10$, computed by the methods (32)–(34) with different choices of the numerical fluxes. More specifically, $\widehat{E_z}$ in (32)–(33) is evaluated as (38) with $\alpha = \sigma\alpha_x$ and $\beta = \sigma\alpha_y$, where $\alpha_x = \alpha_y = 1$, and $\sigma = 1$, 1.2, and 2. And $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ in (32)–(34) are from the one-dimensional Lax–Friedrichs flux (35)–(37). It is observed from Table 1 that when $\alpha = \alpha_x$, $\beta = \alpha_y$, the scheme is stable and first order accurate with P^0 approximation; With P^1 approximation, the scheme is only first order accurate which is suboptimal; while the scheme with P^2 approximation starts to be optimally accurate with third order accuracy and then shows instability over long time simulation. When $\alpha = 1.2\alpha_x$ and $\beta = 1.2\alpha_y$, the schemes have optimal accuracy with P^0 and P^1 approximations, yet with P^2 approximation the scheme becomes unstable at $t = 10$. When $\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$, the schemes have optimal accuracy and are stable over the time period we examine. Even though the results are not reported here, we have also tested the schemes with the central flux or upwind flux on the element interfaces for $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ and their arithmetic average for $\widehat{E_z}$ from the four edges connecting to a vertex. We have learned from the numerical experiments that if $\widehat{E_z}$ of the form (38) is used at vertices, it is important to have sufficient numerical dissipation. For instance, $\widehat{E_z}$ based on the multi-dimensional Lax–Friedrichs flux with $\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$ leads to a stable scheme with optimal accuracy, yet $\widehat{E_z}$ based on either the one-dimensional Lax–Friedrichs flux with $\alpha = \alpha_x$ and $\beta = \alpha_y$, or the multi-dimensional HLL flux with $\alpha = 1.2\alpha_x$ and $\beta = 1.2\alpha_y$ leads to unstable schemes due to the insufficiency in numerical dissipation.

4.2 Fourier Analysis of the Scheme with P^0 Approximation

In this subsection, we will carry out the Fourier analysis for the scheme (32)–(33) with P^0 approximation. The goal is to further understand the role of the amount of the numerical dissipation in $\widehat{E_z}$ in the form of (38).

With the P^0 polynomial space, the scheme (32)–(33) becomes

$$\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{ij}^x(y)dy = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_x^n(x_{i+\frac{1}{2}}, y)dy - \Delta t\left(\widehat{E_z}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - \widehat{E_z}(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})\right), \quad (39)$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} b_{ij}^y(x)dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B_y^n(x, y_{j+\frac{1}{2}})dx + \Delta t\left(\widehat{E_z}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - \widehat{E_z}(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})\right), \quad (40)$$

Table 1 Errors and convergence orders of B_x for the induction equation, with the initial data $(B_x, B_y) = (-\sin(2\pi y), \sin(2\pi x))$ on the domain $\Omega = [0, 1] \times [0, 1]$ and at $t = 1.0, 10.0$. The velocity field is $(u_x, u_y) = (1, 1)$

N	L^2 error $t = 1$ $\alpha = 1, \beta = 1$	Order	L^2 error $t = 10$	Order	L^2 error $t = 1$ $\alpha = 1.2, \beta = 1.2$	Order	L^2 error $t = 10$	Order	L^2 error $t = 1$ $\alpha = 2, \beta = 2$	Order	L^2 error $t = 10$	Order
P^0												
16	1.18E−01	–	5.81E−01	–	2.02E−01	–	5.82E−01	–	3.61E−01	–	5.89E−01	–
32	9.21E−02	0.98	4.72E−01	0.30	1.06E−01	0.93	5.01E−01	0.22	2.21E−01	0.71	5.84E−01	0.01
64	4.69E−02	0.97	3.19E−01	0.57	5.48E−02	0.95	3.57E−01	0.49	1.23E−01	0.85	4.03E−01	0.40
128	2.37E−02	0.98	1.89E−01	0.75	2.79E−02	0.97	2.19E−01	0.71	6.49E−02	0.92	2.59E−01	0.64
256	1.20E−02	0.99	1.03E−01	0.87	1.41E−02	0.99	1.22E−01	0.84	3.34E−02	0.96	1.48E−01	0.81
P^1												
16	1.52E−01	–	5.60E−01	–	6.88E−02	–	4.76E−01	–	6.06E−03	–	3.05E−02	–
32	8.36E−02	0.87	4.61E−01	0.28	2.16E−04	1.67	2.02E−01	1.23	1.37E−04	2.14	1.71E−03	2.18
64	4.37E−02	0.94	3.16E−01	0.55	5.82E−03	1.89	5.75E−02	1.81	3.33E−04	2.04	5.94E−04	2.79
128	2.23E−02	0.97	1.89E−01	0.75	1.49E−03	1.97	1.48E−02	1.96	8.25E−05	2.01	1.04E−04	2.52
256	1.12E−02	0.99	1.03E−01	0.87	3.75E−04	1.99	3.71E−03	1.99	2.06E−05	2.00	2.21E−05	2.23
P^2												
16	9.45E−03	–	1.89E−02	–	7.54E−04	–	7.50E−03	–	1.46E−04	–	1.59E−04	–
32	1.18E−04	3.00	2.34E−03	3.01	9.44E−05	3.00	9.33E−04	3.01	1.83E−05	3.00	1.85E−05	3.09
64	1.47E−05	3.00	2.92E−04	3.00	1.18E−05	3.00	1.17E−04	3.00	2.28E−06	3.00	2.30E−06	3.01
128	1.84E−06	3.00	Unstable		1.48E−06	3.00	1.54E−05	2.92	2.86E−07	3.00	2.86E−07	3.00
256	2.30E−07	3.00	Unstable		1.85E−07	3.00	Unstable		3.57E−08	3.00	3.58E−08	3.00

and the electric field flux $\widehat{E_z}$ at the vertex $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ is given by

$$\begin{aligned}\widehat{E_z}\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}\right) &= \frac{1}{4}\left(E_z\mid_{I_{ij}} + E_z\mid_{I_{i+1j}} + E_z\mid_{I_{ij+1}} + E_z\mid_{I_{i+1j+1}}\right)\\ &- \frac{\beta}{4}\left(\frac{B_x\mid_{I_{ij+1}} + B_x\mid_{I_{i+1j+1}}}{2} - \frac{B_x\mid_{I_{ij}} + B_x\mid_{I_{i+1j}}}{2}\right)\\ &+ \frac{\alpha}{4}\left(\frac{B_y\mid_{I_{i+1j}} + B_y\mid_{I_{i+1j+1}}}{2} - \frac{B_y\mid_{I_{ij}} + B_y\mid_{I_{ij+1}}}{2}\right).\end{aligned} \tag{41}$$

We replace E_z by $u_y B_x - u_x B_y$, and rewrite (41) into

$$\begin{aligned}\widehat{E_z}\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}\right) &= \frac{2u_y - \beta}{8}\left(B_x\mid_{I_{ij+1}} + B_x\mid_{I_{i+1j+1}}\right)\\ &+ \frac{2u_y + \beta}{8}\left(B_x\mid_{I_{ij}} + B_x\mid_{I_{i+1j}}\right) - \frac{2u_x - \alpha}{8}\left(B_y\mid_{I_{i+1j}} + B_y\mid_{I_{i+1j+1}}\right)\\ &- \frac{2u_x + \alpha}{8}\left(B_y\mid_{I_{ij}} + B_y\mid_{I_{ij+1}}\right).\end{aligned} \tag{42}$$

Based on the divergence-free reconstruction procedure, we know $B_x\mid_{I_{ij}} = B_x\mid_{I_{i+1j}} = b^x_{ij}$ and $B_y\mid_{I_{ij}} = B_y\mid_{I_{ij+1}} = b^y_{ij}$. Therefore (41) is indeed

$$\widehat{E_z}\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}\right) = \frac{2u_y - \beta}{4} b^x_{ij+1} + \frac{2u_y + \beta}{4} b^x_{ij} - \frac{2u_x - \alpha}{4} b^y_{i+1j} - \frac{2u_x + \alpha}{4} b^y_{ij}, \tag{43}$$

and our scheme (39)–(40) can be formulated more explicitly: look for $b^{x,n+1}_{ij}$ and $b^{y,n+1}_{ij}$ in $\mathbb{R}$, satisfying

$$\begin{aligned}b^{x,n+1}_{ij} &= b^{x,n}_{ij} - \frac{\Delta t}{\Delta y}\left(\frac{2u_y - \beta}{4}(b^{x,n}_{ij+1} - b^{x,n}_{ij}) + \frac{2u_y + \beta}{4}(b^{x,n}_{ij} - b^{x,n}_{ij-1})\right)\\ &+ \frac{\Delta t}{\Delta y}\left(\frac{2u_x - \alpha}{4}(b^{y,n}_{i+1j} - b^{y,n}_{i+1j-1}) + \frac{2u_x + \alpha}{4}(b^{y,n}_{ij} - b^{y,n}_{ij-1})\right),\end{aligned} \tag{44}$$

$$\begin{aligned}b^{y,n+1}_{ij} &= b^{y,n}_{ij} + \frac{\Delta t}{\Delta x}\left(\frac{2u_y - \beta}{4}(b^{x,n}_{ij+1} - b^{x,n}_{i-1j+1}) + \frac{2u_y + \beta}{4}(b^{x,n}_{ij} - b^{x,n}_{i-1j})\right)\\ &- \frac{\Delta t}{\Delta x}\left(\frac{2u_x - \alpha}{4}(b^{y,n}_{i+1j} - b^{y,n}_{ij}) + \frac{2u_x + \alpha}{4}(b^{y,n}_{ij} - b^{y,n}_{i-1j})\right).\end{aligned} \tag{45}$$

Additionally, the divergence-free property of the numerical solution can be translated into the following relation, for any i, j, n,

$$\Delta y(b^{x,n}_{ij} - b^{x,n}_{i-1j}) + \Delta x(b^{y,n}_{ij} - b^{y,n}_{ij-1}) = 0. \tag{46}$$

The parameters α and β in (41) are taken as

$$\alpha = \sigma|u_x|, \beta = \sigma|u_y|, \tag{47}$$

and σ is a constant that measures the amount of numerical dissipation introduced through the numerical flux $\widehat{E_z}$. In the next Theorem, we will study the role of this constant σ to the numerical stability of the scheme. The stability condition for $\sigma = 2$ was previously given in [7].

Theorem 4.1 *The scheme* (44)–(45) *with* (46)–(47) *is stable under the following condition on the time step size* Δt*:*

1. *When* $\sigma \leq 2$,

$$\Delta t \left(\frac{|u_x|}{\Delta x} + \frac{|u_y|}{\Delta y} \right) \leq \frac{\sigma}{2}; \tag{48}$$

2. *when* $\sigma > 2$

$$\Delta t \left(\frac{|u_x|}{\Delta x} + \frac{|u_y|}{\Delta y} \right) \leq \frac{2}{\sigma}. \tag{49}$$

And the maximum of the upper bound of both formulas is 1*, that is,* $\max_{\sigma \geq 0}(\frac{\sigma}{2}, \frac{2}{\sigma}) = 1$*, and it is attained at* $\sigma = 2$.

Proof To carry out the Fourier analysis, let

$$(b^{x,n}, b^{y,n}) = (\widehat{b}_x^n, \widehat{b}_y^n) e^{i(k_1 x + k_2 y)}, \tag{50}$$

with k_1, k_2 being arbitrary integer. With (50), the Eq. (44) becomes

$$\begin{aligned} \widehat{b}_x^{n+1} = \widehat{b}_x^n &- \frac{\Delta t}{\Delta y} \left(\frac{2u_y - \beta}{4} (e^{ik_2 \Delta y} - 1) + \frac{2u_y + \beta}{4} (1 - e^{-ik_2 \Delta y}) \right) \widehat{b}_x^n \\ &+ \frac{\Delta t}{\Delta y} \left(\frac{2u_x - \alpha}{4} \left(e^{\frac{ik_1 \Delta x}{2} + \frac{ik_2 \Delta y}{2}} - e^{\frac{ik_1 \Delta x}{2} - \frac{ik_2 \Delta y}{2}} \right) \right. \\ &\left. + \frac{2u_x + \alpha}{4} \left(e^{-\frac{ik_1 \Delta x}{2} + \frac{ik_2 \Delta y}{2}} - e^{\frac{-ik_1 \Delta x}{2} - \frac{ik_2 \Delta y}{2}} \right) \right) \widehat{b}_y^n, \end{aligned} \tag{51}$$

and the divergence-free condition (46) becomes

$$\Delta y \left(e^{\frac{ik_1 \Delta x}{2}} - e^{\frac{-ik_1 \Delta x}{2}} \right) \hat{b}_x^n + \Delta x \left(e^{\frac{ik_2 \Delta y}{2}} - e^{\frac{-ik_2 \Delta y}{2}} \right) \hat{b}_y^n = 0,$$

i.e.

$$\Delta y \sin \left(\frac{k_1 \Delta x}{2} \right) \hat{b}_x^n + \Delta x \sin \left(\frac{k_2 \Delta y}{2} \right) \hat{b}_y^n = 0. \tag{52}$$

Combining (51) and (52), we get

$$\widehat{b}_x^{n+1} = Q \widehat{b}_x^n, \tag{53}$$

where the amplification factor Q is

$$\begin{aligned} Q = 1 &- \frac{\Delta t}{\Delta y} \left(\frac{2u_y - \beta}{4} \left(e^{ik_2 \Delta y} - 1 \right) + \frac{2u_y + \beta}{4} \left(1 - e^{-ik_2 \Delta y} \right) \right) \\ &- \frac{\Delta t}{\Delta x} \left(\frac{2u_x - \alpha}{4} e^{\frac{ik_1 \Delta x}{2}} + \frac{2u_x + \alpha}{4} e^{-\frac{ik_1 \Delta x}{2}} \right) 2i \sin \left(\frac{k_1 \Delta x}{2} \right). \end{aligned} \tag{54}$$

One can easily check that (45) and the divergence-free relation (46) will lead to the same amplification factor Q. Without loss of generality, we assume $u_x \geq 0, u_y \geq 0$. Let $c_1 = \frac{\Delta t u_x}{\Delta x}$

and $c_2 = \frac{\Delta t u_y}{\Delta y}$, and with σ defined in (47), we have

$$\begin{aligned} Q &= 1 - c_2 \left(\frac{2-\sigma}{4}(e^{ik_2\Delta y} - 1) + \frac{2+\sigma}{4}(1 - e^{-ik_2\Delta y}) \right) \\ &\quad - c_1 \left(\frac{2-\sigma}{4} e^{\frac{ik_1\Delta x}{2}} + \frac{2+\sigma}{4} e^{-\frac{ik_1\Delta x}{2}} \right) 2i \sin\left(\frac{k_1\Delta x}{2}\right) \\ &= 1 - \frac{\sigma c_1}{2}(1 - \cos(k_1\Delta x)) - \frac{\sigma c_2}{2}(1 - \cos(k_2\Delta y)) \\ &\quad - i\,(c_1 \sin(k_1\Delta x) + c_2 \sin(k_2\Delta y))\,. \end{aligned} \tag{55}$$

Next, we want to obtain the condition on the time step size to ensure $|Q| \le 1$. To this end,

$$\begin{aligned} |Q|^2 =& \left(1 - \frac{\sigma c_1}{2}(1 - \cos(k_1\Delta x)) - \frac{\sigma c_2}{2}(1 - \cos(k_2\Delta y))\right)^2 \\ &+ (c_1 \sin(k_1\Delta x) + c_2 \sin(k_2\Delta y))^2 \\ =& 1 + \frac{\sigma^2}{4}(c_1 + c_2)^2 + c_1^2 + c_2^2 - \sigma(c_1 + c_2) \\ &+ \frac{\sigma^2}{4}(c_1 \cos(k_1\Delta x) + c_2 \cos(k_2\Delta y))^2 - c_1^2 \cos^2(k_1\Delta x) - c_2^2 \cos^2(k_2\Delta y) \\ &+ \left(\sigma c_1 - \frac{\sigma^2 c_1^2}{2} - \frac{\sigma^2}{2} c_1 c_2\right) \cos(k_1\Delta x) + \left(\sigma c_2 - \frac{\sigma^2 c_2^2}{2} - \frac{\sigma^2}{2} c_1 c_2\right) \cos(k_2\Delta y) \\ &+ 2c_1 c_2 \sin(k_1\Delta x) \sin(k_2\Delta y). \end{aligned} \tag{56}$$

To handle the last term in (56), we will use

$$\begin{aligned} \sin(k_1\Delta x) \sin(k_2\Delta y) &= \cos(k_1\Delta x - k_2\Delta y) - \cos(k_1\Delta x)\cos(k_2\Delta y) \\ &\le 1 - \cos(k_1\Delta x)\cos(k_2\Delta y). \end{aligned}$$

Note that this inequality becomes an equality when $k_1\Delta x = k_2\Delta y + 2\pi n$ for some $n \in \mathbb{Z}$. Now with $s = \cos(k_1\Delta x) \in [-1, 1]$ and $t = \cos(k_2\Delta y) \in [-1, 1]$, (56) turns to

$$\begin{aligned} |Q|^2 &\le 1 + \frac{\sigma^2}{4}(c_1 + c_2)^2 + c_1^2 + c_2^2 - \sigma(c_1 + c_2) + \frac{\sigma^2}{4}(c_1 s + c_2 t)^2 - c_1^2 s^2 - c_2^2 t^2 \\ &\quad + \frac{\sigma c_1}{2}(2 - \sigma c_1 - \sigma c_2)s + \frac{\sigma c_2}{2}(2 - \sigma c_2 - \sigma c_1)t + 2c_1 c_2 - 2c_1 c_2 s t \\ &= 1 + \left(\frac{\sigma^2}{4} + 1\right)(c_1 + c_2)^2 + \left(\frac{\sigma^2}{4} - 1\right)(c_1 s + c_2 t)^2 \\ &\quad - \sigma(c_1 + c_2) + \sigma(c_1 s + c_2 t) - \frac{\sigma^2}{2}(c_1 + c_2)(c_1 s + c_2 t). \end{aligned} \tag{57}$$

We further set $A = c_1 + c_2$ and $B = c_1 s + c_2 t$, and

$$\begin{aligned} |Q|^2 &\le 1 + \left(\frac{\sigma^2}{4} + 1\right) A^2 + \left(\frac{\sigma^2}{4} - 1\right) B^2 - \frac{\sigma^2}{2} AB - \sigma(A - B) \\ &= 1 + (A - B)\left(\left(\frac{\sigma^2}{4} + 1\right) A + \left(1 - \frac{\sigma^2}{4}\right) B - \sigma\right). \end{aligned}$$

From the definitions of A and B, we know $A - B = c_1(1 - s) + c_2(1 - t) \geq 0$. Hence $|Q|^2 \leq 1$ if

$$\left(\frac{\sigma^2}{4} + 1\right) A + \left(1 - \frac{\sigma^2}{4}\right) B - \sigma \leq 0. \tag{58}$$

There are two cases:

Case 1 When $\sigma \leq 2$, we have $1 - \frac{\sigma^2}{4} \geq 0$. Therefore with $A \geq B$, it is sufficient to require

$$\left(\frac{\sigma^2}{4} + 1\right) A + \left(1 - \frac{\sigma^2}{4}\right) A - \sigma \leq 0,$$

that is, $A \leq \frac{\sigma}{2}$. This can not be further improved, since $A = B$ when $s = t = \cos(k_1 \Delta x) = \cos(k_2 \Delta y) = 1$.

Case 2 When $\sigma > 2$, we have $1 - \frac{\sigma^2}{4} < 0$. Therefore with $A + B = c_1(1+s) + c_2(1+t) \geq 0$, it is sufficient to require

$$\left(\frac{\sigma^2}{4} + 1\right) A - \left(1 - \frac{\sigma^2}{4}\right) A - \sigma \leq 0,$$

that is, $A \leq \frac{2}{\sigma}$. Again, this can not be further improved, since $B = -A$ when $s = t = \cos(k_1 \Delta x) = \cos(k_2 \Delta y) = -1$.

In summary,

$$\Delta t \left(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y}\right) = A \leq \begin{cases} \frac{\sigma}{2}, & \text{if } \sigma \leq 2, \\ \frac{2}{\sigma}, & \text{if } \sigma > 2. \end{cases} \tag{59}$$

Finally one can see the maximum of $\{\sigma/2, 2/\sigma\}$ is 1 when $\sigma = 2$. □

The theorem above implies that the scheme (44)–(45) with the multi-dimensional Lax–Friedrichs numerical flux for $\widehat{E_z}$, with $\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$, has the largest stability region for our scheme with the P^0 approximation. This is also illustrated by Fig. 2, where comparison is given between the stability regions of the schemes with the multi-dimensional Lax–Friedrichs numerical flux ($\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$) on the right, and one-dimensional Lax–Friedrichs numerical flux ($\alpha = \alpha_x$ and $\beta = \alpha_y$) on the left.

4.3 Our Choices of the Numerical Fluxes in (19)–(21)

Based on the numerical and theoretical studies in previous two subsections for the induction equation, the electric field flux approximations in our proposed schemes (19)–(21) to update the magnetic field in the full ideal MHD simulations will be chosen as follows.

(1) They satisfy the two conditions in (22)–(23);
(2) The singled-valued electric field flux $\widehat{E_z}$ at a vertex is determined by the average of the multi-dimensional Lax–Friedrichs numerical fluxes on four edges connecting to this vertex, given by (38) with $\alpha = 2\alpha_x$ and $\beta = 2\alpha_y$;
(3) On an element interface, the standard one-dimensional Lax–Friedrichs numerical flux (35)–(36) will be applied for both $\widetilde{E_z}$ and $\widetilde{\widetilde{-E_z}}$ with parameters α_x and α_y.

Both α_x and α_y in (2) and (3) represent the local speeds of the entire MHD system, and are taken as the largest absolute-value of eigenvalues of the Jacobian $\frac{\partial \mathbf{F}(\mathbf{U},\mathcal{B})\cdot\mathbf{n}}{\partial(\mathbf{U},\mathcal{B})}$, with $\mathbf{n} = (0, 1)^{\mathrm{T}}$, $(1, 0)^{\mathrm{T}}$ respectively, in the neighborhood of the relevant edge. Note that these local speeds are different from that in (37) for the induction equation.

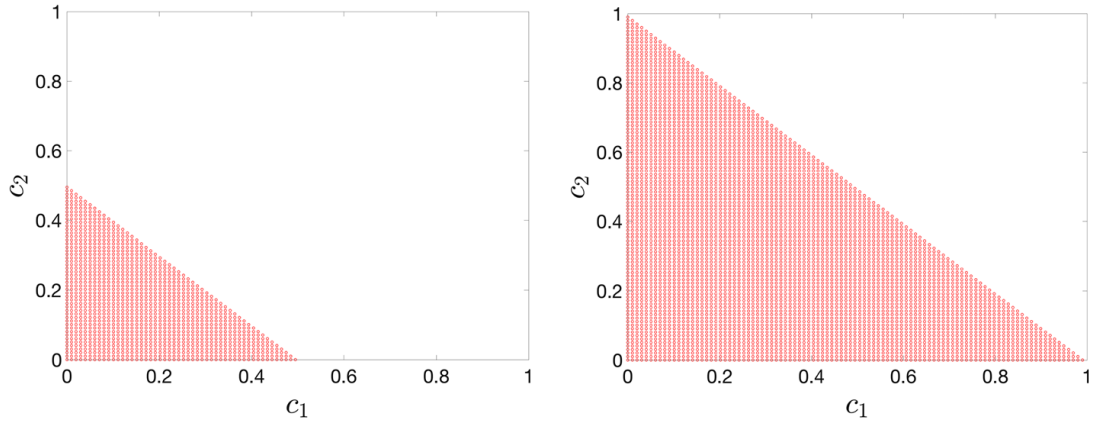


Fig. 2 The stability region of the scheme (32)–(33) for the P^0 approximation with $\widehat{E_z}$ in (38), and $\alpha = \sigma\alpha_x$ and $\beta = \sigma\alpha_y$. Here c_1 is $\frac{\Delta t|u_x|}{\Delta x}$, c_2 is $\frac{\Delta t|u_y|}{\Delta y}$. **a** $\sigma = 1$, **b** $\sigma = 2$

5 Nonlinear Limiter, a Revisit to the Reconstruction

In this section, we will discuss the use of nonlinear limiters to enhance numerical stability of the proposed schemes. Similar to high order DG methods for nonlinear hyperbolic conservation laws, nonlinear limiters are also needed for numerical stability of our methods. In this paper, the *minmod* total variation bounded (TVB) slope limiter in [16] is applied. This limiter involves a non-negative parameter M, and its value is often chosen for each example in actual implementation [14,31]. This limiter can be applied to component-wise variables or in local characteristic fields with respect to the 7×7-eigen system in [23].

Following the work in [26,27] with globally divergence-free central DG methods, for non-smooth solutions, we first apply the limiter only to the hydrodynamic variables $\mathbf{U}_h$, not to $\mathcal{B}_h$, $\widetilde{\mathcal{B}}_h$ or (b^x_{ij}, b^y_{ij}). This works well for the schemes with P^1 approximations, and when the discontinuities in the solutions are not strong.

It is known that central type schemes are in general more dissipative hence more stable than upwind type schemes. Therefore it is not unexpected that when our proposed DG methods are used to examples with strong shocks, it seems necessary to apply the nonlinear limiter to both $\mathbf{U}_h$ and the magnetic field $\mathcal{B}_h$ in order to effectively control numerical oscillations. One needs to be careful, though, about how to implement this without losing the globally divergence-free property of the computed magnetic field. A straightforward implementation will break the intrinsic relation between the data $\widetilde{\mathcal{B}}_h$ and (b^x_{ij}, b^y_{ij}) used in the reconstruction (see the proof of Theorem 3.1). On the other hand, for the methods we will focus on in this paper with P^1 and P^2 approximations, an alternative but equivalent way was presented in [27] to reconstruct I_{ij}, look for $\mathcal{B}_h^{n+1} \in \mathcal{W}^k(I_{ij})$ such that

1. $B_x^{n+1}(x_{l+\frac{1}{2}}, y) = b^x_{lj}(y)$ for $l = i-1, i$ and $y \in [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$,
2. $B_y^{n+1}(x, y_{l+\frac{1}{2}}) = b^y_{il}(x)$ for $l = j-1, j$ and $x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$,
3. $\nabla \cdot \mathcal{B}_h^{n+1}|_{I_{ij}} = 0$.

One can refer to [26] for the proof of the equivalency. An important feature of this equivalent reconstruction is that only the interface data (b^x_{ij}, b^y_{ij}) is needed. Now when the nonlinear limiter needs to be applied to the magnetic field, the normal components of the magnetic field $\{b^x_{ij}\}_{ij}, \{b^y_{ij}\}_{ij}$ will be limited first (this will be discussed in details next), then the equivalent

reconstruction given above will be used to obtain the globally divergence-free magnetic field based on the limited normal components of the magnetic field.

We here will use $k = 2$ as an example to illustrate how to apply the *minmod* TVB limiter to $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$. The quadratic polynomial b_{ij}^x can be written as

$$b_{ij}^x(y) = \overline{b_{ij}^x} + c_y Y + c_{yy}\left(Y^2 - \frac{1}{3}\right), \tag{60}$$

with $Y = \frac{y-y_j}{\Delta y/2}$, and $\overline{b_{ij}^x}$ is the edge average of b_{ij}^x, namely,

$$\overline{b_{ij}^x} = \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} b_{ij}^x(y)dy. \tag{61}$$

We compute $\widetilde{c_y}$ according to

$$\widetilde{c_y} = \widetilde{m}\left(c_y, \Delta_+\overline{b_{ij}^x}, \Delta_-\overline{b_{ij}^x}\right). \tag{62}$$

Here $\Delta_+\overline{b_{ij}^x} = \overline{b_{ij+1}^x} - \overline{b_{ij}^x}$, $\Delta_-\overline{b_{ij}^x} = \overline{b_{ij}^x} - \overline{b_{ij-1}^x}$, and the corrected *minmod* TVB function $\widetilde{m}$ is

$$\widetilde{m}(a_1, a_2, a_3) = \begin{cases} a_1, & \text{if } |a_1| \le M(\Delta y)^2; \\ m(a_1, a_2, a_3), & \text{otherwise,} \end{cases} \tag{63}$$

with the *minmod* function m defined as

$$m(a_1, a_2, a_3) = \begin{cases} s\min\left(|a_1|, |a_2|, |a_3|\right), & \text{if } s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3); \\ 0, & \text{otherwise.} \end{cases} \tag{64}$$

If $|\widetilde{c_y} - c_y| > 10^{-6}$, we apply the limiter by setting $c_y = \widetilde{c_y}$ and $c_{yy} = 0$ in (60). Otherwise, no modification is made to (60). The treatment for b_{ij}^y is very similar. It is important to know that the limiter does not change the edge averages $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$, hence a necessary compatible condition for the exactly divergence-free reconstruction, namely,

$$\int_{I_{ij}} \nabla \cdot \mathcal{B}_h^n dxdy = \Delta y\left(\overline{b_{ij}^{x,n}} - \overline{b_{i-1j}^{x,n}}\right) + \Delta x\left(\overline{b_{ij}^{y,n}} - \overline{b_{ij-1}^{y,n}}\right) = 0 \tag{65}$$

still holds.

Finally in Algorithm 1, we provide the flow chart of the proposed globally divergence-free methods when they are applied to ideal MHD equations. The time discretization is taken to be the forward Euler method.

6 Numerical Results

In this section, numerical examples are presented to illustrate the accuracy and stability of the proposed globally divergence-free methods with P^1 and P^2 approximations for the ideal MHD equations. They include two smooth examples and five non-smooth examples. In our simulations, uniform rectangular meshes with $N \times N$ elements are used. The initial numerical solution $\mathbf{U}_h \in \mathcal{V}_h^k$ is obtained through the L^2 projection, and $\mathcal{B}_h \in \mathcal{M}_h^k$ is by the BDM projection [10]. In time, a third order TVD Runge–Kutta method is applied [21].

Algorithm 1 The algorithm of the globally divergence-free DG methods for ideal MHD equations, with the forward Euler method as the time discretization.

Initialization:

Initialize $\mathbf{U}_h^0$ via the L^2 projection and $\mathcal{B}_h^0$ via the BDM projection. If the example is non-smooth, apply the TVB limiter to $\mathbf{U}_h^0$.

Time evolution:

With the numerical solutions available at time t_n for $n \geq 0$, namely $(\mathbf{U}_h^n, \mathcal{B}_h^n) \in \mathcal{V}_h^k \times \mathcal{M}_h^k$ with $\mathcal{B}_h^n = (B_{x,h}^n, B_{y,h}^n)^{\mathrm{T}}$, update $(\mathbf{U}_h^{n+1}, \mathcal{B}_h^{n+1}) \in \mathcal{V}_h^k \times \mathcal{M}_h^k$ with $\mathcal{B}_h^{n+1} = (B_{x,h}^{n+1}, B_{y,h}^{n+1})^{\mathrm{T}}$ at $t_{n+1} = t_n + \Delta t$;

1: Compute the time step Δt based on the maximum value α_x, α_y;

2: Impose boundary conditions;

3: Pre-compute the numerical solutions at t_{n+1}:

- for each element I_{ij}, update $\mathbf{U}_h^{n+1}$ by scheme (14);
- for each y-direction element interface, compute $\{b_{ij}^x\}_{ij}$ by scheme (19);
- for each x-direction element interface, compute $\{b_{ij}^y\}_{ij}$ by scheme (20);
- if $k \geq 2$, compute $\widetilde{\mathcal{B}}_h$ on each element I_{ij} by scheme (21);

4: If the example is non-smooth, apply the TVB limiter to $\mathbf{U}_h^{n+1}$; for challenging non-smooth examples (such as the cloud–shock example), also apply the limiter as in Sect. 5 to $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$;

5: Reconstruction on each element: if the limiter is not applied to $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$, reconstruct $\mathcal{B}_h^{n+1}$ following **R1**-**R3** in Sect. 3.2.3; otherwise, reconstruct $\mathcal{B}_h^{n+1}$ following the procedure given in Sect. 5;

6: Return $(\mathbf{U}_h^{n+1}, \mathcal{B}_h^{n+1}) \in \mathcal{V}_h^k \times \mathcal{M}_h^k$.

That is, to solve $u_t = L(u, t)$, given the numerical solution u^n at t_n, we compute u^{n+1} at $t_{n+1} = t_n + \Delta t$ as follows,

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u_n, t_n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L\left(u^{(1)}, t_n + \Delta t\right), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L\left(u^{(2)}, t_n + \frac{1}{2}\Delta t\right). \end{aligned} \tag{66}$$

The time step is determined by

$$\Delta t = \frac{CFL}{\alpha_x/\Delta x + \alpha_y/\Delta y}, \tag{67}$$

where α_x and α_y are the largest absolute eigenvalues of Jacobian $\frac{\partial F_1(\mathbf{U},\mathcal{B})}{\partial(\mathbf{U},\mathcal{B})}$ and $\frac{\partial F_2(\mathbf{U},\mathcal{B})}{\partial(\mathbf{U},\mathcal{B})}$, respectively. We take $CFL = 0.2$ for $k = 1$ and $CFL = 0.1$ for $k = 2$ similar as for the standard DG methods. The numerical fluxes in the schemes to update the magnetic field follow the strategies summarized in Sect. 4.3. The *minmod* TVB slope limiter is applied for non-smooth examples with $M = 1$.

6.1 Smooth Examples

6.1.1 The Smooth Vortex Problem

The first example we consider is the smooth vortex example which was introduced in [2], and it models a smooth vortex propagating with speed (1, 1) in a two-dimensional domain. The initial condition is given by

$$(\rho, u_x, u_y, u_z, B_x, B_y, B_z, p) = \left(1, 1 + \delta u_x, 1 + \delta u_y, 0, \delta B_x, \delta B_y, 0, 1 + \delta p\right),$$

Table 2 L^2 errors and orders of accuracy of ρ, u_x, B_x and p for smooth vortex problem at $t = 20$. The computational domain is $[-10, 10] \times [-10, 10]$

N	ρ		u_x		p		B_x	
	L^2 error	Order	L^2 error	Order	L^2 error	Order	L^2 error	Order
P^1								
32	3.98E−05	–	9.18E−03	–	1.22E−03	–	7.51E−03	–
64	2.44E−05	0.71	3.45E−03	1.41	5.44E−04	1.17	2.75E−03	1.45
128	8.23E−06	1.58	6.80E−04	2.34	1.20E−04	2.19	5.35E−04	2.36
256	1.84E−06	2.16	9.45E−05	2.85	1.98E−05	2.60	7.39E−05	2.87
P^2								
32	1.50E−04	–	1.96E−03	–	1.02E−03	–	7.10E−03	–
64	6.62E−05	1.18	8.76E−04	1.16	4.90E−04	1.06	2.56E−03	1.47
128	1.30E−05	2.35	1.70E−04	2.37	9.76E−05	2.33	4.63E−04	2.47
256	1.76E−06	2.88	2.31E−05	2.88	1.33E−05	2.87	6.21E−05	2.90

where

$$(\delta u_x, \delta u_y) = \frac{\xi}{2\pi}\widehat{\nabla} \times \exp\{0.5(1-r^2)\}, \quad (\delta B_x, \delta B_y) = \frac{\eta}{2\pi}\widehat{\nabla} \times \exp\{0.5(1-r^2)\},$$
$$\delta p = \frac{\eta^2(1-r^2-\xi^2)}{8\pi^2}\exp(1-r^2).$$

Here $r = \sqrt{x^2+y^2}$, $\xi = \eta = 1$ and $\gamma = 5/3$. The computational domain is taken as $[-10, 10] \times [-10, 10]$. Even though the problem is not-periodic, periodic boundary conditions are used in our simulation. This will introduce an error of size $O(10^{-22})$ which is negligible with respect to the resolution of the numerical solutions. In Table 2, L^2 errors and orders of accuracy are presented for the variables ρ, u_x, B_x and pressure p at $t = 20$, right after one time period, by which the vortex returns to its initial location. The results show that our numerical schemes have $(k+1)$th order accuracy for $k = 1, 2$. For this smooth example, no nonlinear limiter is needed.

6.1.2 The Smooth Alfvén Wave

The second smooth example is the smooth Alfvén wave problem, which describes a circularly polarized Alfvén wave moving in the domain $\Omega = [0, 1/\cos\alpha] \times [0, 1/\sin\alpha]$ [28,37]. Here, α represents the angle of the wave propagation with respect to x-axis, and it is set to be $\pi/4$. The same initial data as in [28] is taken

$$\rho = 1, u_\parallel = 0, u_\perp = 0.1\sin(2\pi\beta), u_z = 0.1\cos(2\pi\beta),$$
$$B_\parallel = 1, B_\perp = u_\perp, B_z = u_z, p = 0.1,$$

where $\beta = x\cos\alpha + y\sin\alpha$. The subscripts $\parallel$ and $\perp$ denote the directions parallel and perpendicular to the wave propagation direction, respectively. Periodic boundary conditions are used and $\gamma = 5/3$. The Alfvén wave travels at a constant Alfvén speed $B_\parallel/\sqrt{\rho} = 1$. The solution returns to its initial configuration when time t is an integer. In Table 3, we present the L^2 errors and orders of accuracy for u_x, u_z, B_x and p at time $t = 2$. From the results, we can see that the P^k approximations with $k = 1, 2$ are $(k+1)$th order accurate, and they are optimal. No nonlinear limiter is applied.

Table 3 L^2 errors and orders of accuracy for u_x, u_z, p and B_x for smooth Alfvén wave problem at $t = 2$. The computational domain is $[0, \sqrt{2}] \times [0, \sqrt{2}]$

N	u_x		u_z		p		B_x	
	L^2 error	order	L^2 error	order	L^2 error	order	L^2 error	order
P^1								
16	2.99E−03	–	3.91E−03	–	7.41E−04	–	2.44E−03	–
32	4.27E−04	2.81	5.69E−04	2.78	1.14E−04	2.71	3.35E−04	2.87
64	6.82E−05	2.65	9.47E−05	2.59	1.96E−05	2.54	4.87E−05	2.78
128	1.34E−05	2.34	1.94E−05	2.29	4.10E−06	2.25	8.59E−06	2.50
P^2								
16	3.88E−03	–	7.17E−04	–	3.64E−03	–	2.08E−03	–
32	3.50E−04	3.47	5.38E−05	3.73	3.27E−04	3.48	2.01E−04	3.37
64	2.56E−05	3.78	2.81E−06	4.26	1.71E−05	4.26	1.78E−05	3.50
128	2.81E−06	3.19	2.36E−07	3.57	1.39E−06	3.62	2.09E−06	3.09

6.2 Non-smooth Examples

6.2.1 The Field Loop Advection

In this subsection, we consider the magnetic field loop advection problem originally introduced in [20]. The same initial data as in [27] is taken, with $(\rho, u_x, u_y, u_z, B_z, p) = (1, 2, 1, 1, 0, 1)$, and $(B_x, B_y) = \widehat{\nabla} \times A_z$. Here A_z is the z-component of the magnetic potential

$$A_z = \begin{cases} A_0(R - r) & \text{if } r \le R, \\ 0 & \text{if } r > R, \end{cases}$$

with $A_0 = 10^{-3}$, $R = 0.3$ and $r = \sqrt{x^2 + y^2}$. This problem is computed on the domain $[-1, 1] \times [-0.5, 0.5]$ with a 200×100 mesh. Periodic boundary conditions are used and $\gamma = 5/3$.

In Fig. 3, we report the gray-scale images of the magnetic pressure $B_x^2 + B_y^2$ (left) and the magnetic field lines (right) at time $t = 0$, $t = 2$ and $t = 10$. With the globally divergence-free magnetic field, the magnetic field lines are plotted by contouring the z-component of the numerical magnetic potential A_z. The magnetic pressure is convected across the domain periodically, and this is confirmed by our numerical results based on P^2 approximation. The component-wise *minmod* TVB limiter is applied to $\mathbf{U}_h$ with the parameter $M = 1$. There is no visible difference in the numerical results when the limiter is applied to the local characteristic fields. Overall our schemes capture the field loop well. From the images on the left in Fig. 3, one can observe numerical dissipation around the center and the boundary of the loop, similar to the observation in [20,26,27]. There is no obvious oscillations in our solutions even at later time $t = 10$ unlike in some numerical results commented in [20,28]. From the images on the right of Fig. 3, symmetry can be seen in the magnetic field lines, with some distortion at $t = 10$ due to the accumulated numerical dissipation over long time simulation.

Note that the initial data is discontinuous, and one needs to pay special attention to the initialization to ensure the magnetic field being divergence-free at $t = 0$. For example,

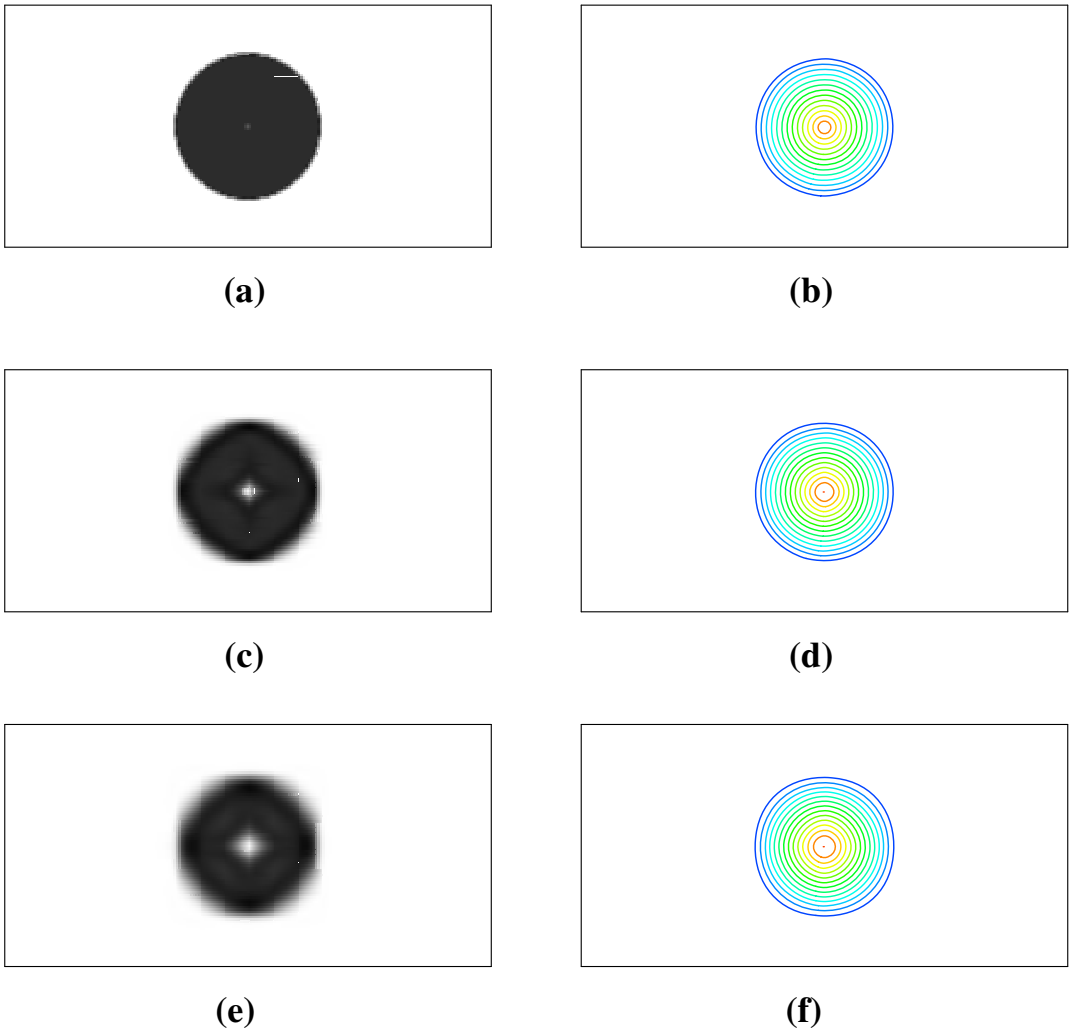


Fig. 3 The magnetic pressure $B_x^2 + B_y^2$ (left) and magnetic field lines (right) of the field loop advection. P^2 approximation on 200×100 mesh. The magnetic field lines are plotted with the same range. **a** Magnetic pressure at $t = 0$, **b** magnetic field lines at $t = 0$, **c** magnetic pressure at $t = 2$, **d** magnetic field lines at $t = 2$, **e** magnetic pressure at $t = 10$, **f** magnetic field lines at $t = 10$

to apply the BDM projection to the initial magnetic field, one needs to compute the first order coefficient $B_x^0 := \frac{1}{\Delta x \Delta y} \int_{I_{ij}} B_x dx dy$ with $B_x = \partial A_z / \partial y$. If a numerical quadrature is applied without taking into account the discontinuity in B_x, then nonzero divergence will be introduced to the magnetic field approximation. Instead, we will evaluate B_x^0 as follows,

$$B_x^0 = \frac{1}{\Delta x \Delta y} \int_{I_{ij}} \frac{\partial A_z}{\partial y} dx dy = \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[A_z(x, y_{j+\frac{1}{2}}) - A_z(x, y_{j-\frac{1}{2}}) \right] dx. \quad (68)$$

Similarly, to evaluate $a_{0R} := \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} B_x(x_{i+\frac{1}{2}}, y) dy$, we will follow

$$a_{0R} = \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \frac{\partial A_z}{\partial y} dy = \frac{1}{\Delta y} \left(A_z(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - A_z(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) \right). \quad (69)$$

This will lead to an exactly divergence-free magnetic field approximation at $t = 0$.

6.2.2 Orszag–Tang Vortex Problem

In this subsection, we test the Orszag–Tang vortex problem, whose solution involves the formation and interaction of multiple shocks as the nonlinear system evolves in time. The same initial date as in [25] is taken, namely,

$$\rho = \gamma^2, \quad u_x = -\sin y, \quad u_y = \sin x, \quad u_z = 0,$$
$$B_x = -\sin y, \quad B_y = \sin 2x, \quad B_z = 0, \quad p = \gamma.$$

This problem is computed on the domain $[0, 2\pi] \times [0, 2\pi]$ with a 192×192 mesh based on P^1 and P^2 approximations. Periodic boundary conditions are used with $\gamma = 5/3$. Figures 4 and 5 demonstrate the time evolutions of the density ρ at times $t = 3, 4$ with P^1 and P^2 approximations, respectively. The component-wise *minmod* TVB limiter is applied to $\mathbf{U}_h$ with the parameter $M = 1$. The results show that our schemes work well for this problem and they are in good agreement with the results in literature [23,25,27].

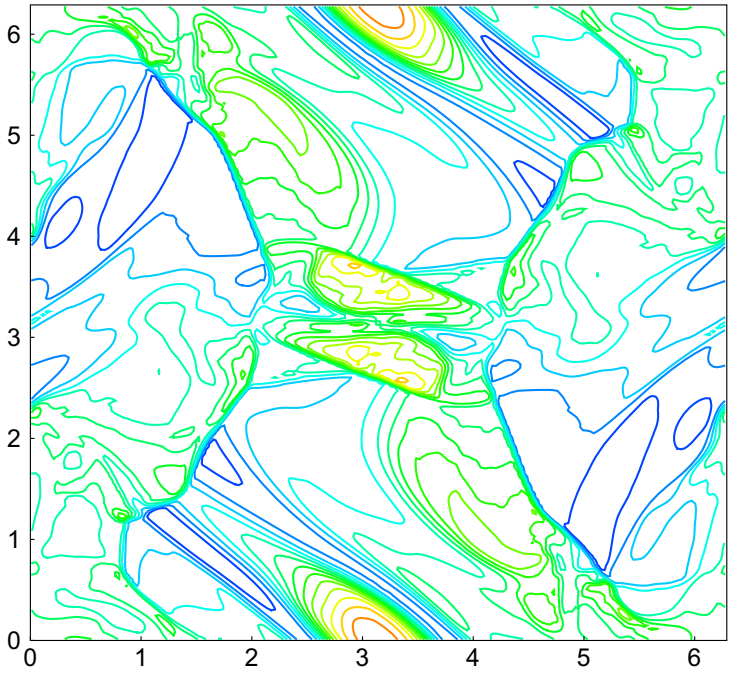
As observed in [23,25,29], different numerical methods can demonstrate different levels of stability for this example, (partially) depending on their ability to control the divergence error in the computed magnetic field. Standard numerical methods that work well for nonlinear hyperbolic conservation laws can show instability when simulating this example, if the divergence error is not sufficiently controlled. Our proposed exactly divergence-free DG methods display very good stability over long time simulation, for example the schemes with P^1 and P^2 approximations are stable up to $t = 25$ (the maximum time we run) on the 192×192 mesh when the *minmod* TVB limiter is applied in local characteristic fields. In addition to the divergence error, as indicated in [37] the choices of the limiters can also affect the numerical stability. When the component-wise *minmod* TVB limiter is applied, the simulation will break down at $t = 7.4$ with the P^2 approximation. Again, the limiters are only applied to $\mathbf{U}_h$.

For this example, we further perform a convergence study for the methods with P^2 approximation. In Fig. 6, we plot the pressure p (left) at $y = 1.99635$ and $t = 2$, and the magnetic variable B_x at $x = \pi$ and $t = 3$, computed with the 192×192 (circle) and 384×384 (line) meshes. With shocks developed in the solution, convergence is observed. The pressure lines and magnetic field lines are comparable to the results by the locally divergence-free DG methods in [25] and exactly divergence-free central DG methods in [26,27]. As in [26,27], there is no negative pressure produced throughout the simulation.

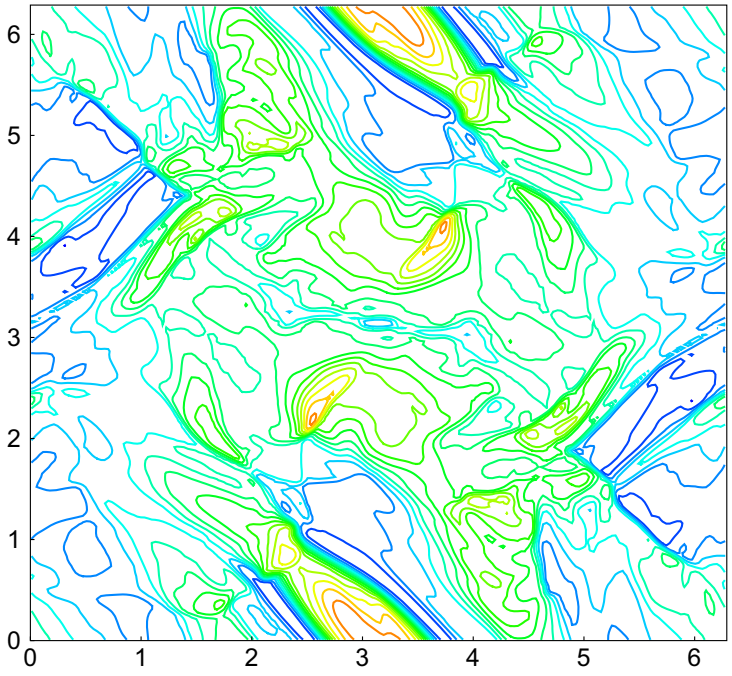
6.2.3 The Rotor Problem

In this subsection, a rotor problem is considered which was first documented in [8]. This problem describes a dense disk of fluid rapidly spinning in a light ambient fluid. To reduce the initial transition, a "taper" function is used to bridge these two areas. We take the same initial data as in [26,37], that is,

$$(u_z, B_x, B_y, B_z, p) = \left(0, 2.5/\sqrt{4\pi}, 0, 0, 0.5\right),$$

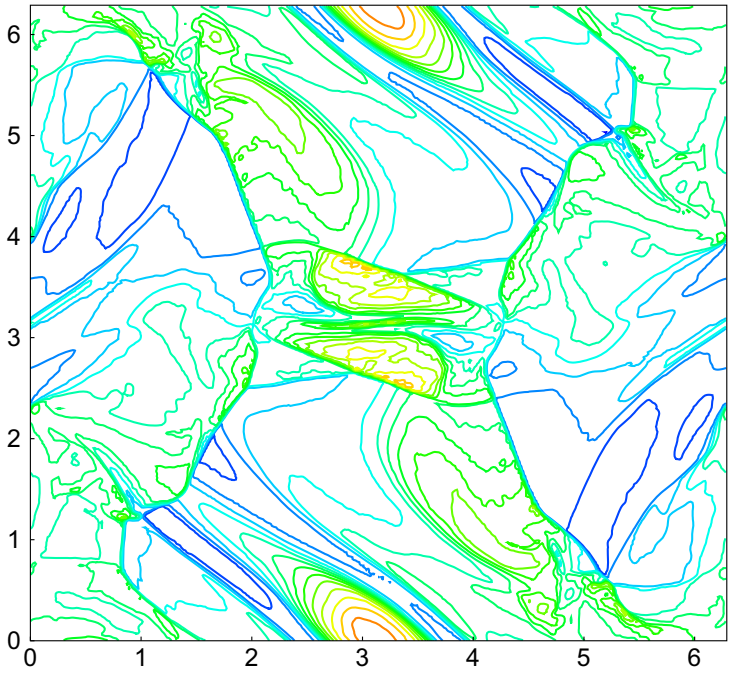


(a)

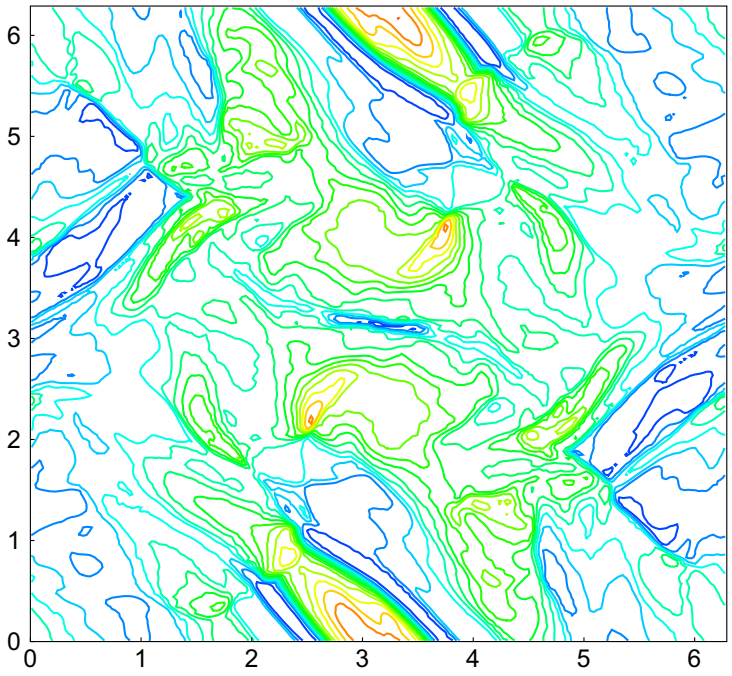


(b)

Fig. 4 Development of the density ρ in Orszag–Tang vortex problem with P^1 approximation at $t = 3$, $t = 4$ on 192×192 mesh. 15 equally spaced contours with ranges $[1.144, 6.134]$, $[1.179, 5.813]$ respectively. **a** $t = 3$, **b** $t = 4$



(a)



(b)

Fig. 5 Development of the density ρ in the Orszag–Tang vortex problem with P^2 approximation at $t = 3$, $t = 4$ on 192×192 mesh. 15 equally spaced contours with ranges [1.122, 6.161], [1.127, 5.857], respectively. **a** $t = 3$, **b** $t = 4$

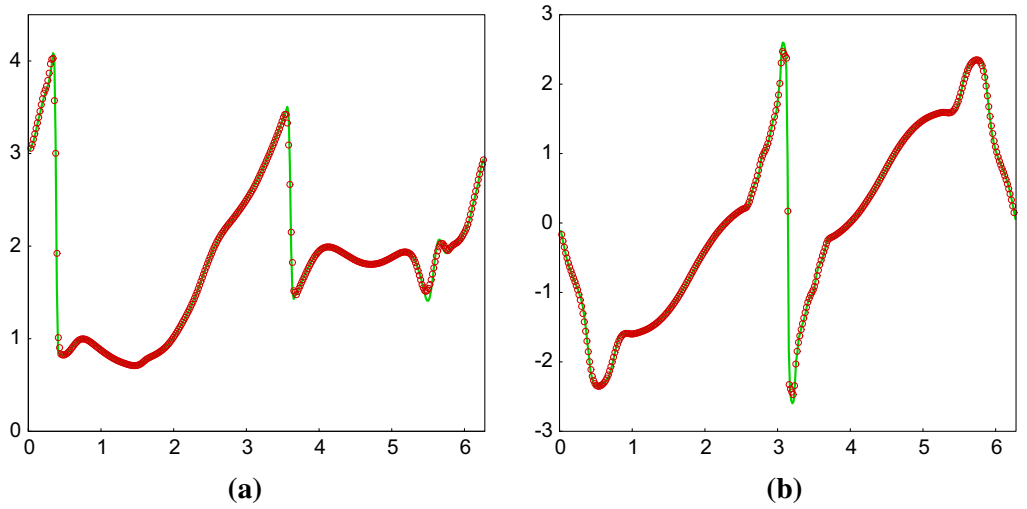


Fig. 6 The P^2 approximation for the Orszag–Tang vortex problem on 192×192 (circle) and 384×384 (solid line) meshes. **a** p with $y = 1.9635$ at $t = 2$, **b** B_x with $x = \pi$ at $t = 3$

and

$$(\rho, u_x, u_y) = \begin{cases} (10, -(y-0.5)/r_0, (x-0.5)/r_0) & r < r_0 \\ (1+9\lambda, -\lambda(y-0.5)/r, \lambda(x-0.5)/r) & r_0 < r < r_1 \\ (1, 0, 0) & r > r_1 \end{cases}$$

where $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$, $r_0 = 0.1$, $r_1 = 0.115$ and $\lambda = (r_1 - r)/(r_1 - r_0)$. We simulate the problem in the domain $[0, 1] \times [0, 1]$. Periodic boundary conditions are used and $\gamma = 5/3$.

In Figs. 7 and 8, we present the results of density ρ, pressure p, the hydrodynamic Mach number $|\mathbf{u}|/c$ with the sound speed $c = \sqrt{\gamma p/\rho}$, and the magnetic pressure $|\mathbf{B}|^2/2$ at $t = 0.295$, based on P^1 and P^2 approximations on the 200×200 mesh. The *minmod* TVB limiter is applied in the characteristic fields and only to $\mathbf{U}_h$. Compared with the results in [25, 37], our methods also resolve this problem well. When divergence error is not sufficiently controlled in the magnetic field by some numerical methods, "distortion" can develop in Mach number [25,37]. In Fig. 9, we zoom in the central part of the Mach number, and no "distortion" is observed.

As in [26,27], we examine the convergence of the methods with P^2 approximation. In Fig. 10, we present the Mach number with $x = 0.413$ (left) and the magnetic field B_x (right) with $x = 0.25$ at $t = 0.295$ on 400×400 (circle) and 600×600 (solid) meshes. Convergence of the method is observed, with the shocks being captured in the numerical solution. The cut lines in Fig. 10 are very close to the results in [26], and there is no significant oscillation in the solutions. In our simulation, negative pressure is not observed.

6.2.4 The Blast Problem

In this subsection, we consider the blast problem as in [8]. There are strong magnetosonic shocks in the solution. The initial condition is taken as

$$\left(\rho, u_x, u_y, u_z, B_x, B_y, B_z, p\right) = \begin{cases} \left(1, 0, 0, 0, \frac{100}{\sqrt{4\pi}}, 0, 0, 1000\right), & r \le R, \\ \left(1, 0, 0, 0, \frac{100}{\sqrt{4\pi}}, 0, 0, 0.10\right), & r > R, \end{cases}$$

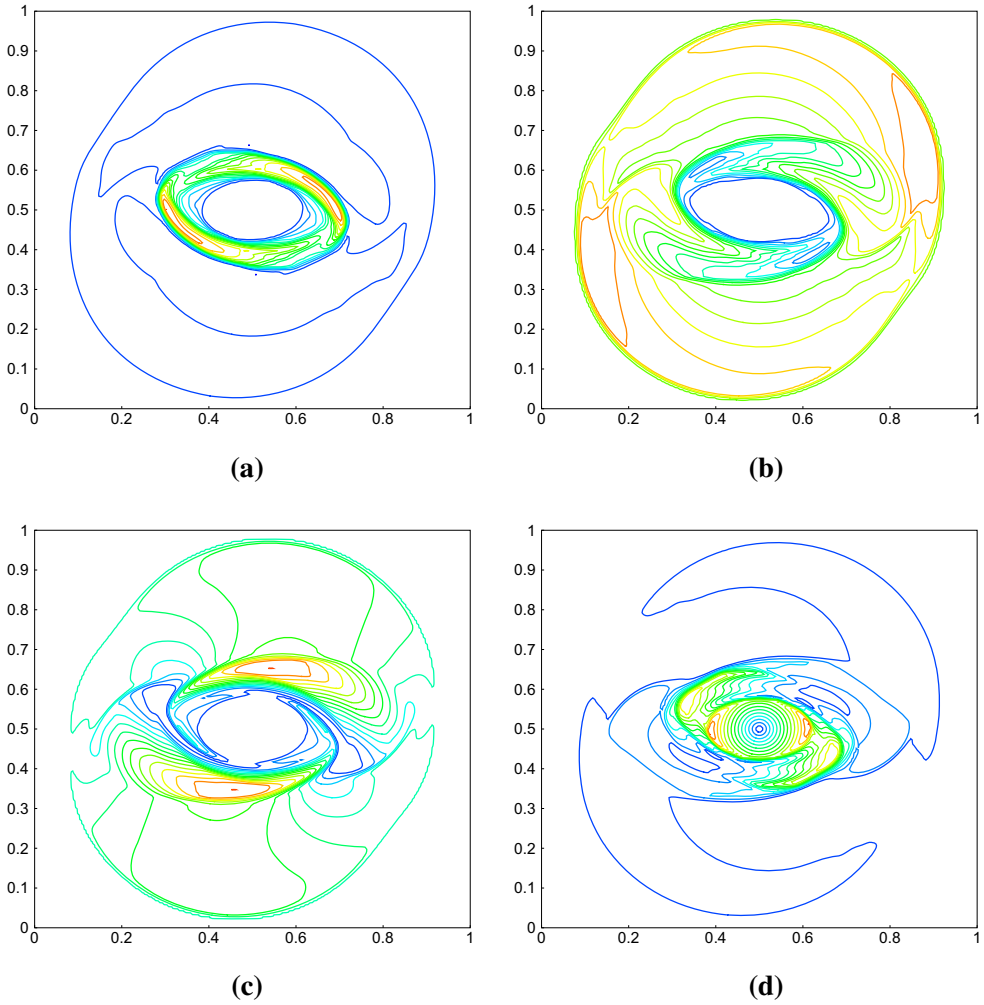


Fig. 7 P^1 approximation for the rotor problem on the 200×200 mesh at $t = 0.295$. 15 equally spaced contours. **a** $\rho \in [0.507, 8.837]$, **b** $p \in [0.010, 0.774]$, **c** $|\mathbf{B}|^2/2 \in [0.012, 0.676]$, **d** $|\mathbf{u}|/c \in [0, 2.673]$

with $r = \sqrt{x^2 + y^2}$ and $R = 0.1$. With this setup, the fluid pulse has very small plasma beta, namely, $\beta = \frac{p}{(B_x^2+B_y^2)/2} = 2.513E{-}04$, in the region outside the initial pressure pulse. We carry out the simulation in the domain $[-0.5, 0.5] \times [-0.5, 0.5]$ with a 200×200 mesh. Outgoing boundary conditions are used and $\gamma = 1.4$.

In Figs. 11 and 12, we report the numerical results at time $t = 0.01$ based on P^1 and P^2 approximations for density ρ, pressure p, square of total velocity $u_x^2 + u_y^2$, and the magnetic pressure $B_x^2 + B_y^2$, respectively. As pointed out in [8,26–28], this is a stringent problem to solve. In our simulation, negative pressure is observed near the shock front, similar as in many other methods when positivity preserving techniques are not applied to pressure [26–28]. In Fig. 13, we plot the negative part of pressure, $\min(0, p)$, based on the P^1 and P^2 approximations. The minimum of pressure in the P^2 approximation is -16.295, and it is more negative than -4.369, the minimum of the pressure in the P^1 approximation. These results are obtained when the component-wise *minmod* TVB limiter is applied only to $\mathbf{U}_h$.

To further improve the numerical stability, we run the simulation by applying the *minmod* TVB limiter to both the hydrodynamic variables $\mathbf{U}_h$ and the normal component of the magnetic field $\{b_{ij}^x\}_{ij}$ and $\{b_{ij}^y\}_{ij}$ (see Sect. 5 for details of the limiter and the reconstruction). In

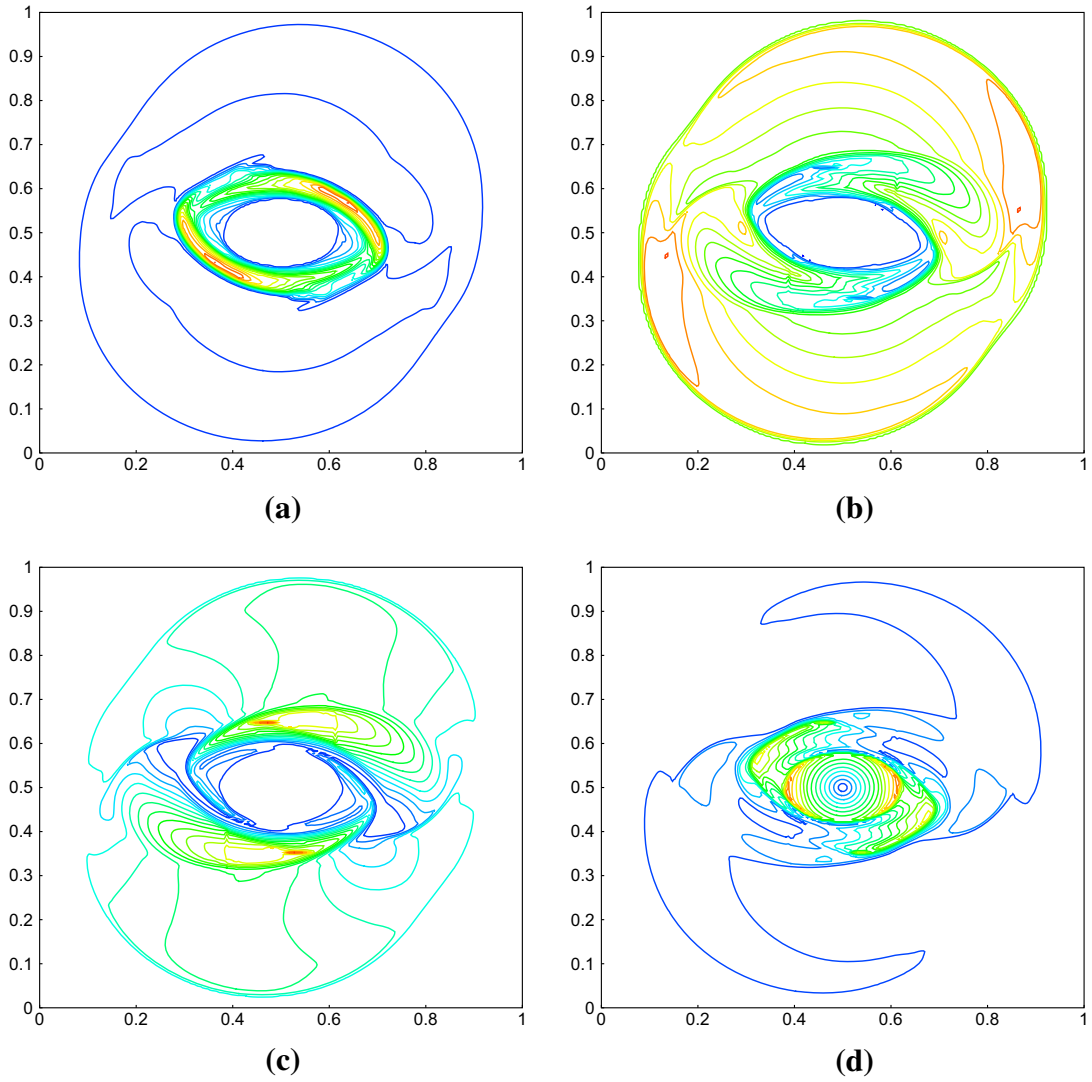


Fig. 8 P^2 approximation for the rotor problem on the 200×200 mesh at $t = 0.295$. 15 equally spaced contours. **a** $\rho \in [0.551, 9.910]$, **b** $p \in [0.008, 0.776]$, **c** $|\mathbf{B}|^2/2 \in [0.012, 0.847]$, **d** $|\mathbf{u}|/c \in [0, 3.033]$

Fig. 14, the results are shown for density ρ, pressure p, square of total velocity $u_x^2 + u_y^2$, and the magnetic pressure $B_x^2 + B_y^2$, respectively, at $t = 0.01$ based on P^2 approximation. With the magnetic field being limited, the minimum of pressure is now -7.347 which is greatly improved, hence the schemes with all unknowns being limited are more robust.

Remark 6.1 Following Zhang and Shu's important work in [41] to design positivity-preserving limiters for high order numerical methods, similar limiters were developed in [12] for DG and central DG methods to simulate ideal MHD equations. Locally divergence-free approximations can be easily used for the methods in [12] without affecting the positivity-preserving property of the overall algorithms. Unfortunately, such limiters can not be applied to the proposed methods in this paper, as they will destroy the globally divergence-free property of the numerical solutions.

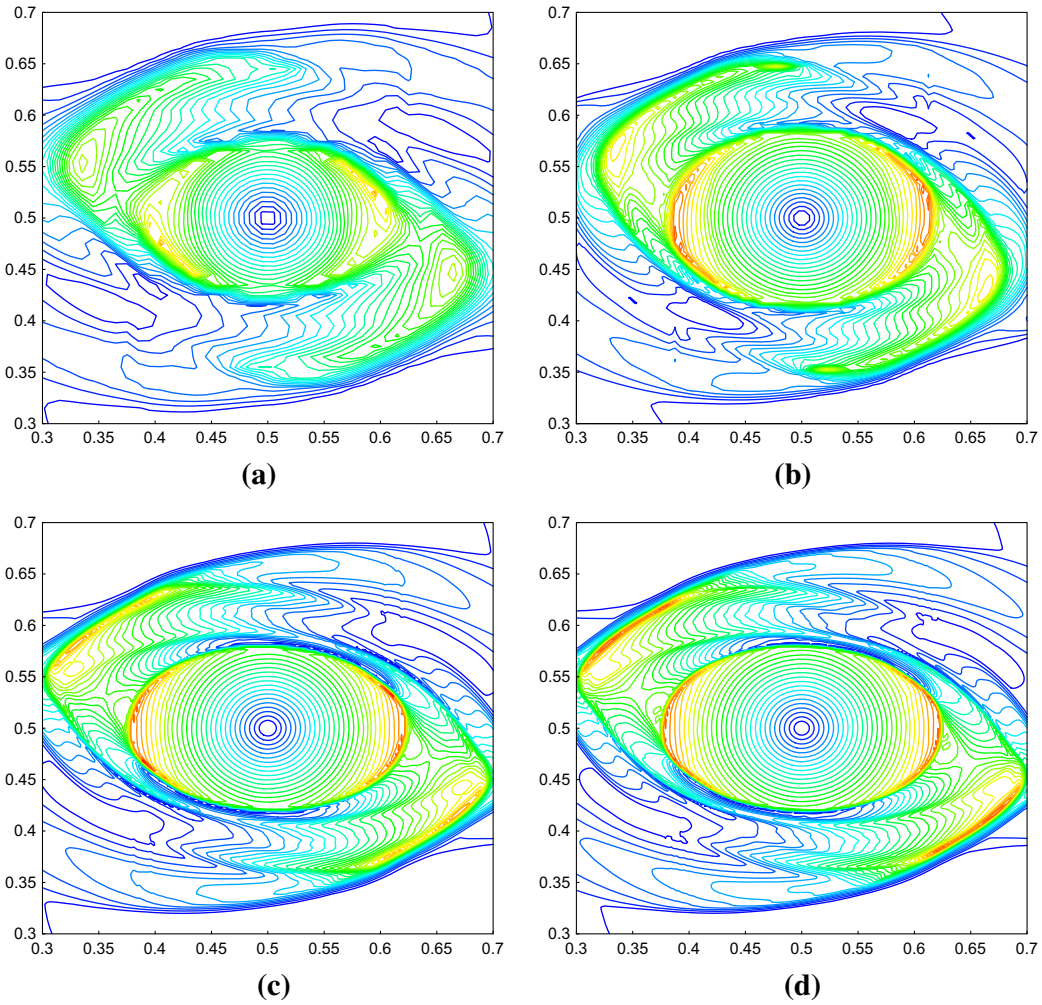


Fig. 9 Zoom-in central part of Mach number $|\mathbf{u}|/c$ with P^2 approximation in the rotor problem at $t = 0.295$. 30 equally spaced contours with range [0.18, 3.12]. **a** 100×100 mesh, **b** 200×200 mesh, **c** 400×400 mesh, **d** 600×600 mesh

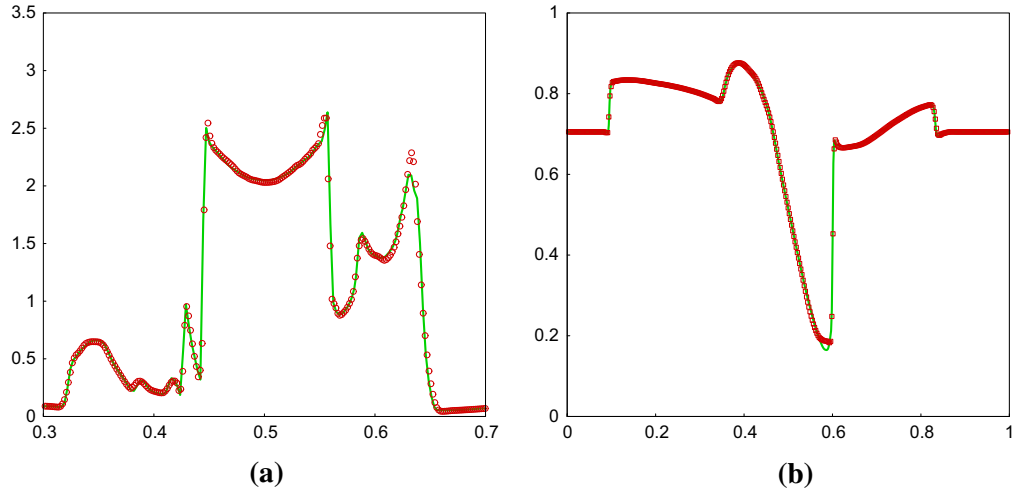


Fig. 10 The Mach number $|\mathbf{u}|/c$ and magnetic filed B_x of the rotor problem with P^2 approximation at $t = 0.295$ on 400×400 (circle) and 600×600 (solid line) meshes. **a** $|\mathbf{u}|/c$ with $x = 0.41$, **b** B_x with $x = 0.25$

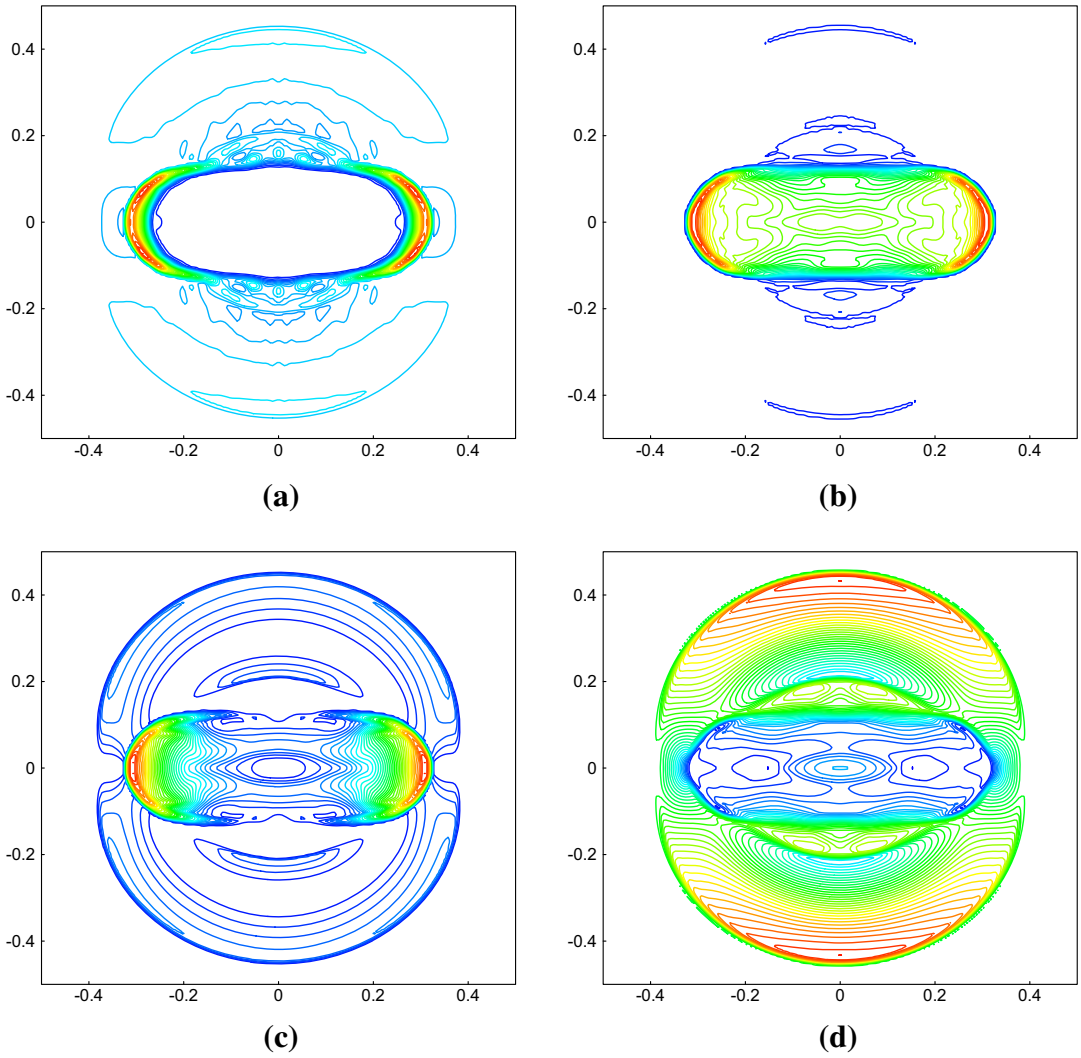


Fig. 11 P^1 approximation for the blast problem on the 200×200 mesh at $t = 0.01$. 40 equally spaced contours are plotted. **a** $\rho \in [0.184, 4.602]$, **b** $p \in [-4.369, 259.297]$, **c** $u_x^2 + u_y^2 \in [0, 288.251]$, **d** $B_x^2 + B_y^2 \in [431.002, 1186.060]$

6.2.5 The Cloud–Shock Interaction

The last example we consider is a cloud–shock interaction problem which involves strong MHD shocks interacting with a dense cloud. We take the same initial data as in [26,27]. The computational domain, $\Omega = [0, 2] \times [0, 1]$, is divided into three regions initially: the post-shock region $\Omega_1 = \{(x, y): 0 \le x \le 1.2, 0 \le y \le 1\}$, the pre-shock region $\Omega_2 = \{(x, y): 1.2 \le x \le 2, 0 \le y \le 1, \sqrt{(x-1.4)^2 + (y-0.5)^2} \ge 0.18\}$ and the cloud region $\Omega_3 = \{(x, y): \sqrt{(x-1.4)^2 + (y-0.5)^2} < 0.18\}$. The initial data in Ω_1, Ω_2 and Ω_3 for $(\rho, u_x, u_y, u_z, B_x, B_y, B_z, p)$ is given by $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_3$, respectively, with

$$\mathbf{U}_1 = (3.88968, 0, 0, -0.05234, 1, 0, 3.9353, 14.2641)\,,$$
$$\mathbf{U}_2 = (1, -3.3156, 0, 0, 1, 0, 1, 0.04)\,,$$
$$\mathbf{U}_3 = (5, -3.3156, 0, 0, 1, 0, 1, 0.04)\,.$$

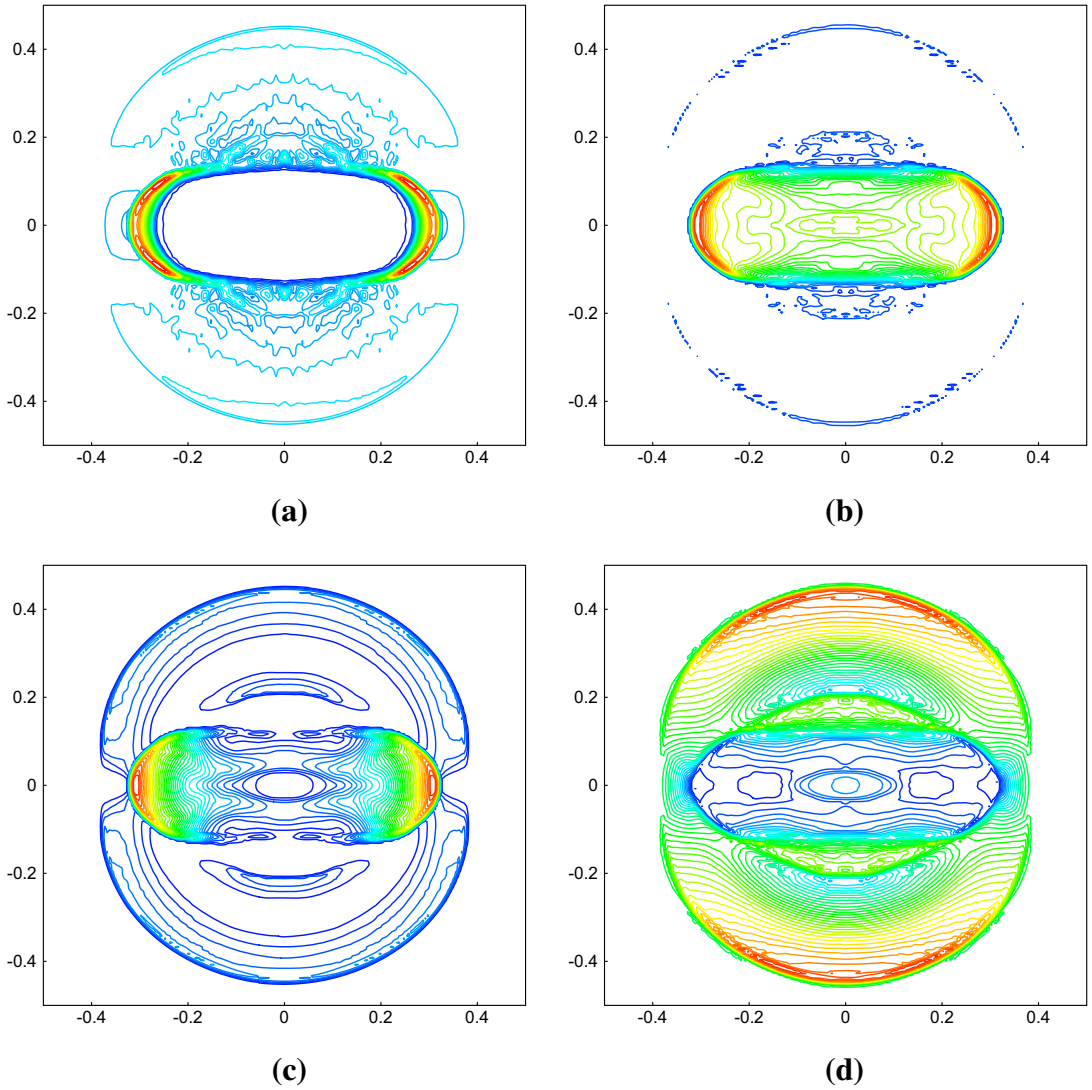


Fig. 12 P^2 approximation for the blast problem on the 200×200 mesh at $t = 0.01$. 40 equally spaced contours are plotted. **a** $\rho \in [0.191, 4.769]$, **b** $p \in [-16.397, 256.291]$, **c** $u_x^2 + u_y^2 \in [0, 288.838]$, **d** $B_x^2 + B_y^2 \in [426.407, 1236.830]$

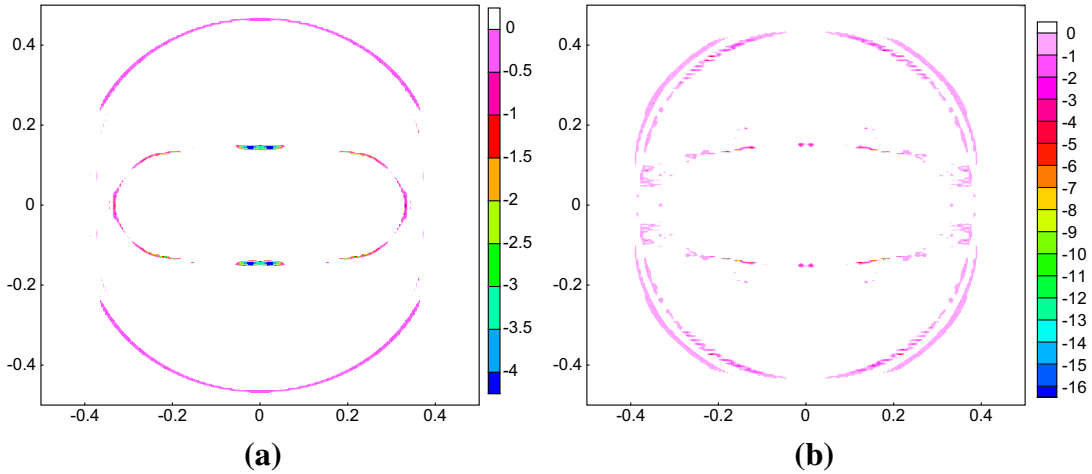


Fig. 13 Negative part of the pressure, $\min(0, p)$, in the blast problem with P^1 and P^2 approximations at $t = 0.01$ on the 200×200 mesh. **a** P^1, **b** P^2

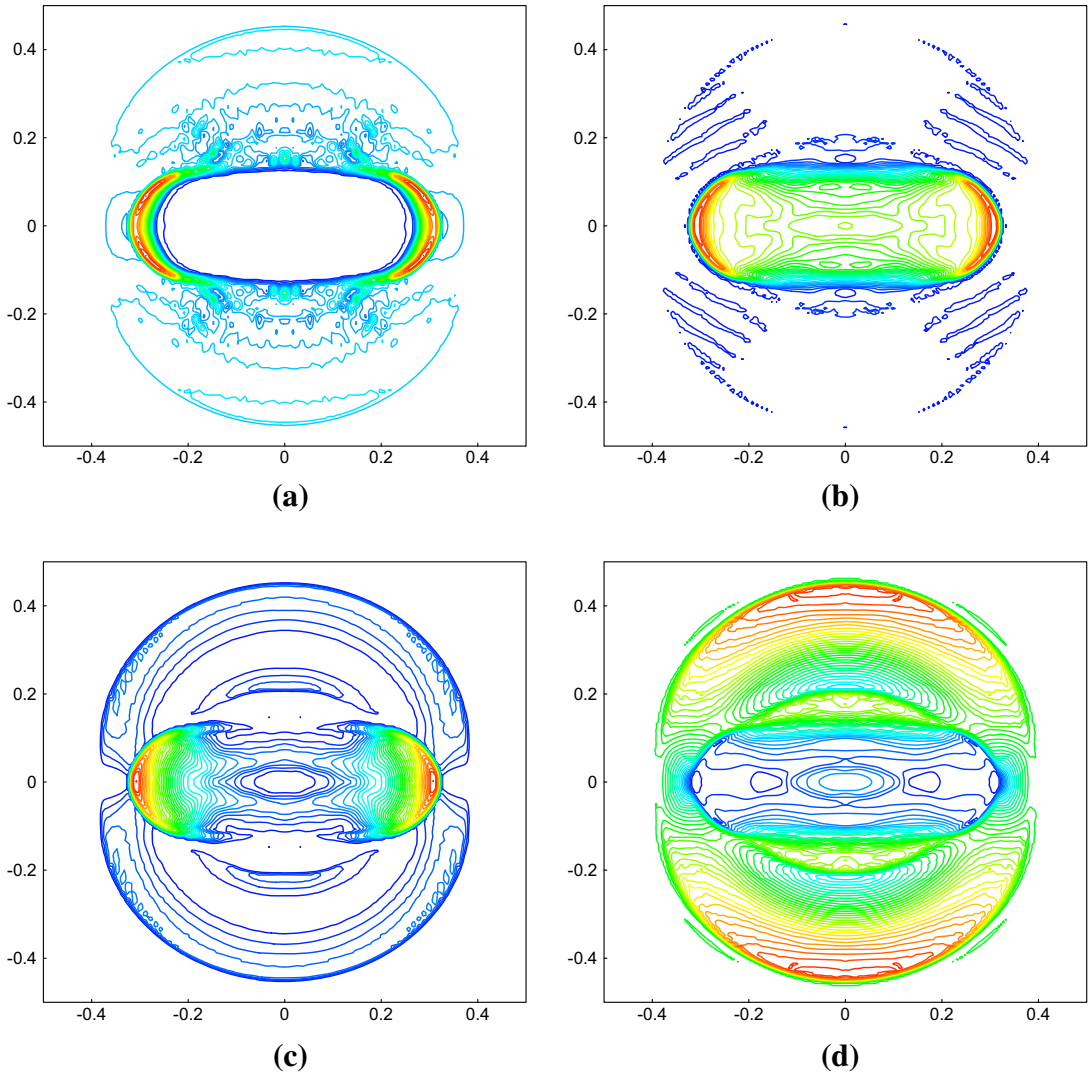


Fig. 14 P^2 approximation for the blast problem on the 200×200 mesh at $t = 0.01$. Nonlinear limiter is applied to both $\mathbf{U}_h$ and $\{b_{ij}^x\}_{ij}$, $\{b_{ij}^y\}_{ij}$. 40 equally spaced contours are used. **a** $\rho \in [0.182, 4.573]$, **b** $p \in [-7.347, 254.906]$, **c** $u_x^2 + u_y^2 \in [0, 287.389]$, **d** $B_x^2 + B_y^2 \in [422.866, 1188.86]$

The cloud in the region Ω_3 is five times denser than its surrounding. Outgoing boundary conditions are used and $\gamma = 5/3$. We run the simulation up to $t = 0.6$.

In Fig. 15, we show the gray-scale images of the P^1 approximations for density ρ, pressure p and magnetic field component B_x, B_y on the 600×300 mesh. The white area represents relatively larger value. The numerical results are fairly close to those by exactly divergence-free central DG methods in [26,27]. The *minmod* TVB limiter is implemented in the local characteristic fields and is only applied to $\mathbf{U}_h$.

In Fig. 16, gray-scale images of P^2 approximations are shown for density ρ, pressure p and magnetic field component B_x, B_y on the 600×300 mesh. In Fig. 17, we further plot the cut lines of density ρ based on P^2 approximation with $y = 0.6$ and $x = 1.0$ on the 600×300 and 800×400 meshes. The convergence of the methods is confirmed. With P^2 approximation, it is not sufficient to just apply the nonlinear limiter to $\mathbf{U}_h$ for numerical stability. And the results presented here are obtained when the limiter is applied to both $\mathbf{U}_h$ and $\{b_{ij}^x\}_{ij}$, $\{b_{ij}^y\}_{ij}$. In order to see the necessity to limit the magnetic field is related to the

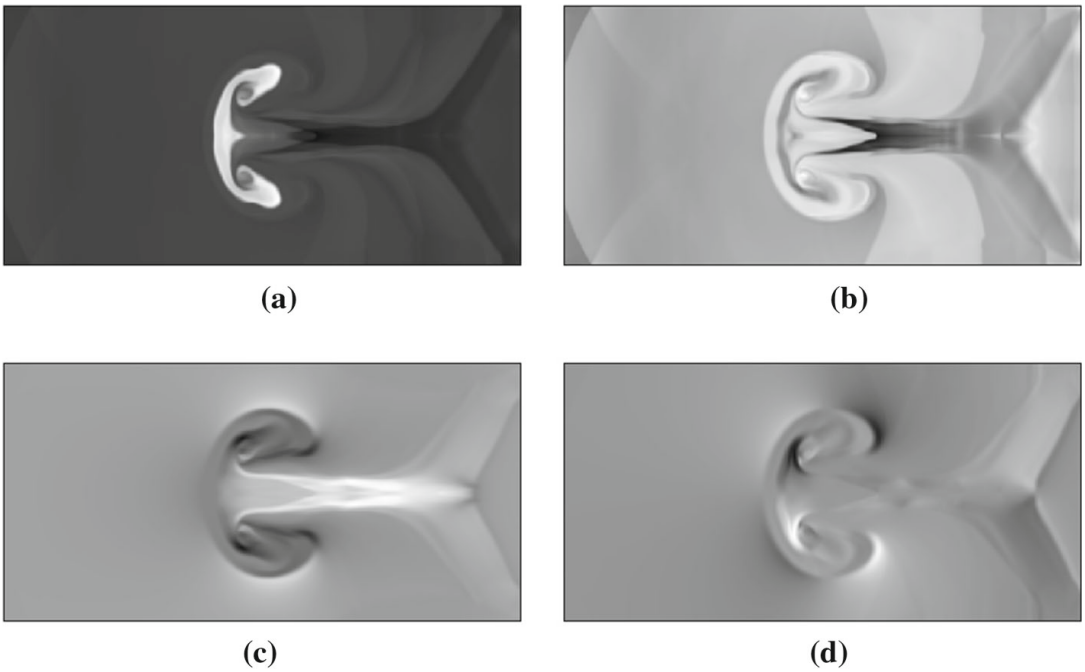


Fig. 15 P^1 approximation of the cloud–shock interaction problem at $t = 0.6$ on the 600×300 mesh. **a** $\rho \in [1.804, 11.638]$, **b** $p \in [6.295, 15.567]$, **c** $B_x \in [-3.073, 4.355]$, **d** $B_y \in [-3.299, 3.265]$

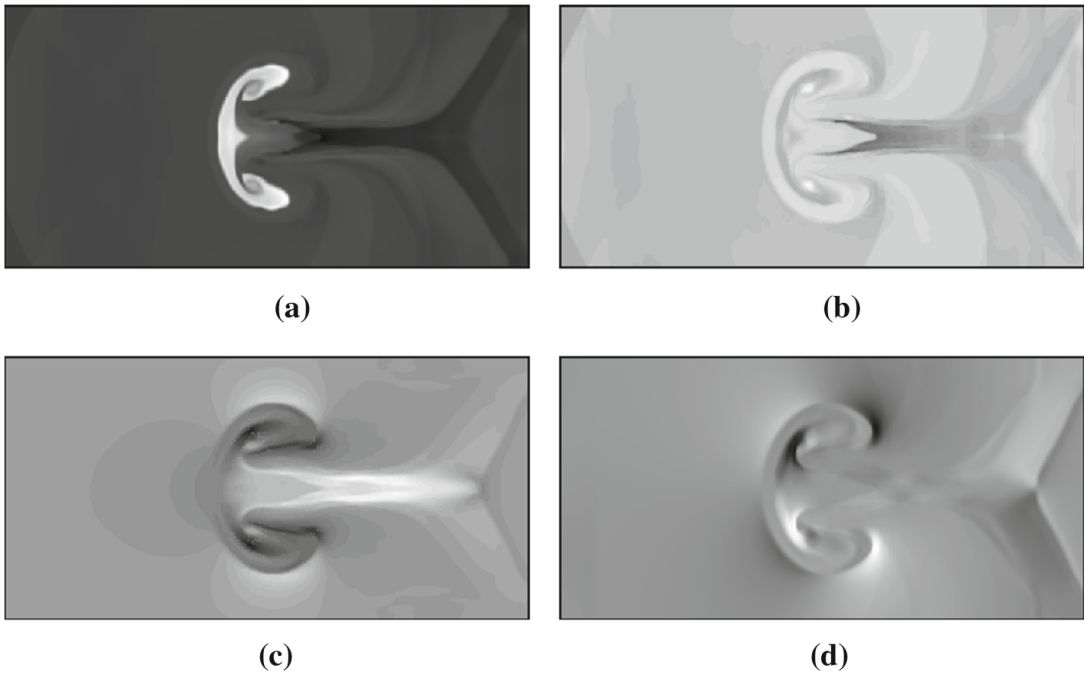


Fig. 16 P^2 approximation of the cloud–shock interaction problem at $t = 0.6$ on the 600×300 mesh. **a** $\rho \in [1.777, 11.655]$, **b** $p \in [1.028, 16.734]$, **c** $B_x \in [-2.922, 4.472]$, **d** $B_y \in [-3.027, 2.961]$

strength of the discontinuity, we also simulate a similar clock–shock interaction example, with the cloud in region Ω_3 two times denser than its surrounding at $t = 0$. As expected, our methods are stable for this modified example when the limiter is applied only to $\mathbf{U}_h$. The numerical results are not included here.

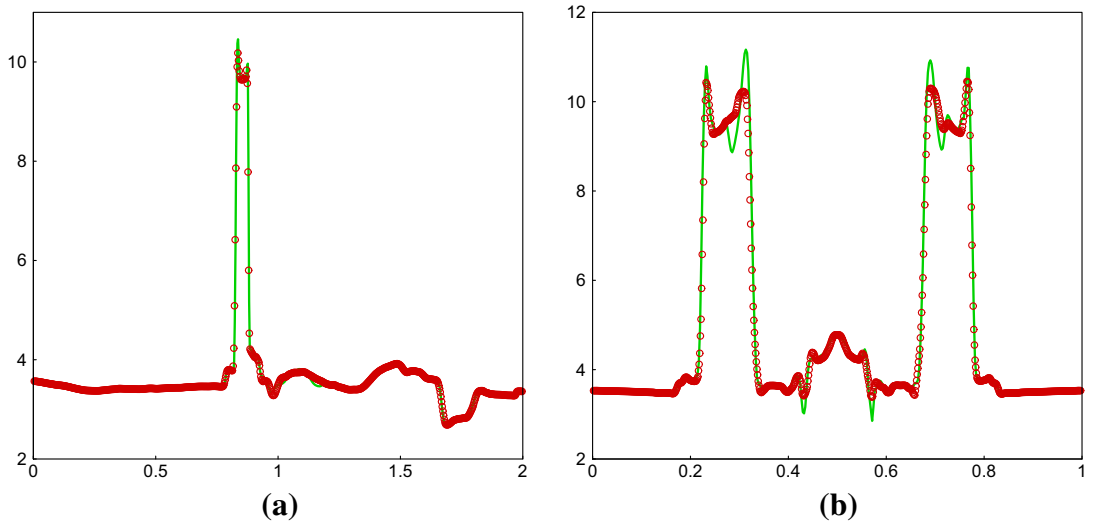


Fig. 17 The P^2 approximation of ρ in the cloud–shock interaction problem at $t = 0.6$ on 600×300 (circle) and 800×400 (solid) meshes. **a** $y = 0.6$, **b** $x = 1.0$

7 Concluding Remarks

In this paper, we propose second and third order globally divergence-free discontinuous Galerkin methods for ideal MHD equations on structured meshes in two dimensions. The main technical aspect is on the choices of numerical fluxes used in the different parts of the algorithms. Analysis is presented to identify conditions on numerical fluxes to ensure the exactly divergence-free property of the approximated magnetic field. A careful numerical and analytical study was carried out to find good choices of numerical fluxes for the accuracy and numerical stability of the methods. A set of smooth and non-smooth numerical examples are presented to illustrate the performance of the proposed methods. Our future efforts will include the extension of the methods to high order accuracy, three dimensions, and unstructured meshes.

References

1. Balsara, D.S.: Divergence-free adaptive mesh refinement for magnetohydrodynamics. J. Comput. Phys. **174**(2), 614–648 (2001)
2. Balsara, D.S.: Second-order-accurate schemes for magnetohydrodynamics with divergence-free reconstruction. Astrophys. J. Suppl. Ser. **151**(1), 149–184 (2004)
3. Balsara, D.S.: Divergence-free reconstruction of magnetic fields and WENO schemes for magnetohydrodynamics. J. Comput. Phys. **228**(14), 5040–5056 (2009)
4. Balsara, D.S.: Multidimensional HLLE Riemann solver: application to Euler and magnetohydrodynamic flows. J. Comput. Phys. **229**(6), 1970–1993 (2010)
5. Balsara, D.S., Dumbser, M.: Divergence-free MHD on unstructured meshes using high order finite volume schemes based on multidimensional Riemann solvers. J. Comput. Phys. **299**, 687–715 (2015)
6. Balsara, D.S., Dumbser, M., Abgrall, R.: Multidimensional HLLC Riemann solver for unstructured meshes with application to Euler and MHD flows. J. Comput. Phys. **261**, 172–208 (2014)
7. Balsara, D.S., Käppeli, R.: Von Neumann stability analysis of globally divergence-free RKDG schemes for the induction equation using multidimensional Riemann solvers. J. Comput. Phys. **336**, 104–127 (2017)
8. Balsara, D.S., Spicer, D.S.: A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. J. Comput. Phys. **149**(2), 270–292 (1999)

9. Brackbill, J., Barnes, D.: The effect of nonzero $\nabla\cdot\mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations. J. Comput. Phys. **35**(3), 426–430 (1980)
10. Brezzi, F., Douglas, J., Marini, L.D.: Two families of mixed finite elements for second order elliptic problems. Numer. Math. **47**(2), 217–235 (1985)
11. Brezzi, F., Fortin, M., Marini, L.D., et al.: Efficient rectangular mixed finite elements in two and three space variables. ESAIM Math. Model. Numer. Anal. **21**(4), 581–604 (1987)
12. Cheng, Y., Li, F., Qiu, J., Xu, L.: Positivity-preserving DG and central DG methods for ideal MHD equations. J. Comput. Phys. **238**, 255–280 (2013)
13. Cockburn, B., Hou, S., Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. Math. Comput. **54**(190), 545–581 (1990)
14. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. J. Comput. Phys. **84**(1), 90–113 (1989)
15. Cockburn, B., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. Math. Comput. **52**(186), 411–435 (1989)
16. Cockburn, B., Shu, C.-W.: The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**(2), 199–224 (1998)
17. Dai, W., Woodward, P.R.: A simple finite difference scheme for multidimensional magnetohydrodynamical equations. J. Comput. Phys. **142**(2), 331–369 (1998)
18. Dedner, A., Kemm, F., Kröner, D., Munz, C.D., Schnitzer, T., Wesenberg, M.: Hyperbolic divergence cleaning for the MHD equations. J. Comput. Phys. **175**(2), 645–673 (2002)
19. Evans, C.R., Hawley, J.F.: Simulation of magnetohydrodynamic flows: a constrained transport method. Astrophys. J. **332**, 659–677 (1988)
20. Gardiner, T.A., Stone, J.M.: An unsplit Godunov method for ideal MHD via constrained transport. J. Comput. Phys. **205**(2), 509–539 (2005)
21. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM Rev. **43**(1), 89–112 (2001)
22. Hesthaven, J.S., Warburton, T.: Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer, Berlin (2007)
23. Jiang, G.-S., Wu, C.: A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics. J. Comput. Phys. **150**(2), 561–594 (1999)
24. Li, B.Q.: Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer. Springer, Berlin (2005)
25. Li, F., Shu, C.-W.: Locally divergence-free discontinuous Galerkin methods for MHD equations. J. Sci. Comput. **22**(1), 413–442 (2005)
26. Li, F., Xu, L.: Arbitrary order exactly divergence-free central discontinuous Galerkin methods for ideal MHD equations. J. Comput. Phys. **231**(6), 2655–2675 (2012)
27. Li, F., Xu, L., Yakovlev, S.: Central discontinuous Galerkin methods for ideal MHD equations with the exactly divergence-free magnetic field. J. Comput. Phys. **230**(12), 4828–4847 (2011)
28. Li, S.: High order central scheme on overlapping cells for magneto-hydrodynamic flows with and without constrained transport method. J. Comput. Phys. **227**(15), 7368–7393 (2008)
29. Li, S.: A fourth-order divergence-free method for MHD flows. J. Comput. Phys. **229**(20), 7893–7910 (2010)
30. Powell, K.G.: An Approximate Riemann Solver for Magnetohydrodynamics (that works in more than one dimension). ICASE report No. 94-24, Langley (1994)
31. Qiu, J., Shu, C.-W.: Runge–Kutta discontinuous Galerkin method using WENO limiters. SIAM J. Sci. Comput. **26**(3), 907–929 (2005)
32. Raviart, P.A., Thomas, J.M.: A mixed finite element method for 2-nd order elliptic problems. In: Dold, A., Eckmann, B. (eds.) Mathematical Aspects of Finite Element Methods. Proceedings of the Conference Held in Rome, 10-12 Dec, 1975. Lecture Notes in Mathematics, vol. 606 (1977). Springer, Berlin, Heidelberg (1977)
33. Reed, W. H., Hill, T. R.: Triangular Mesh Methods for the Neutron Transport Equation, Technical Report LA-UR-73-479. Los Alamos Scientific Laboratory (1973)
34. Riviere, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. SIAM, Philadelphia (2008)
35. Rossmanith, J. A.: High-order discontinuous Galerkin finite element methods with globally divergence-free constrained transport for ideal MHD. arXiv preprint arXiv:1310.4251, (2013)
36. Shu, C.-W.: TVB uniformly high-order schemes for conservation laws. Math. Comput. **49**(179), 105–121 (1987)

37. Tóth, G.: The $\nabla \cdot \mathbf{B}$ constraint in shock-capturing magnetohydrodynamics codes. J. Comput. Phys. **161**(2), 605–652 (2000)
38. Yakovlev, S., Xu, L., Li, F.: Locally divergence-free central discontinuous Galerkin methods for ideal MHD equations. J. Comput. Sci. **4**(1), 80–91 (2013)
39. Yang, H., Li, F.: Stability analysis and error estimates of an exactly divergence-free method for the magnetic induction equations. ESAIM Math. Model. Numer. Anal. **50**(4), 965–993 (2016)
40. Yee, K.S.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antennas Propag. **14**(3), 302–307 (1966)
41. Zhang, X., Shu, C.-W.: On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. J. Comput. Phys. **229**(23), 8918–8934 (2010)

SIAM J. SCI. COMPUT.
Vol. 41, No. 3, pp. A2037–A2063

POSITIVITY PRESERVING LIMITERS FOR TIME-IMPLICIT HIGHER ORDER ACCURATE DISCONTINUOUS GALERKIN DISCRETIZATIONS*

J. J. W. VAN DER VEGT†, YINHUA XIA‡, AND YAN XU§

Abstract. Currently, nearly all positivity preserving discontinuous Galerkin (DG) discretizations of partial differential equations are coupled with explicit time integration methods. Unfortunately, for many problems this can result in severe time-step restrictions. The techniques used to develop explicit positivity preserving DG discretizations cannot, however, easily be combined with implicit time integration methods. In this paper, we therefore present a new approach. Using Lagrange multipliers, the conditions imposed by the positivity preserving limiters are directly coupled to a DG discretization combined with a diagonally implicit Runge–Kutta time integration method. The positivity preserving DG discretization is then reformulated as a Karush–Kuhn–Tucker (KKT) problem, which is frequently encountered in constrained optimization. Since the limiter is only active in areas where positivity must be enforced, it does not affect the higher order DG discretization elsewhere. The resulting nonsmooth nonlinear algebraic equations have, however, a different structure compared to most constrained optimization problems. We therefore develop an efficient active set semismooth Newton method that is suitable for the KKT formulation of time-implicit positivity preserving DG discretizations. Convergence of this semismooth Newton method is proven using a specially designed quasi-directional derivative of the time-implicit positivity preserving DG discretization. The time-implicit positivity preserving DG discretization is demonstrated for several nonlinear scalar conservation laws, which include the advection, Burgers, Allen–Cahn, Barenblatt, and Buckley–Leverett equations.

Key words. positivity preserving, maximum principle, Karush–Kuhn–Tucker equations, discontinuous Galerkin methods, implicit time integration methods, semismooth Newton methods

AMS subject classifications. 65M60, 65K15, 65N22

DOI. 10.1137/18M1227998

1. Introduction.

The solution of many partial differential equations frequently must satisfy a maximum principle, or, more generally, certain variables must obey a lower and/or upper bound. In this paper, we will denote all these cases with positivity preserving. In particular, if the partial differential equations model physical processes, then these bounds are also crucial to obtain a meaningful physical solution. For example, a density, concentration, or pressure in fluid flow must be nonnegative, and a probability distribution should be in the range $[0, 1]$. A numerical solution should therefore strictly obey the bounds on the exact solution; otherwise, the problem can become ill-posed and the solution would be meaningless. Also, the numerical

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 21, 2018; accepted for publication (in revised form) April 4, 2019; published electronically June 25, 2019.

http://www.siam.org/journals/sisc/41-3/M122799.html

Funding: The first author's research was supported by the University of Science and Technology of China (USTC). The second author's research was supported by NSFC grants 11471306 and 11871449 and a grant from the Science & Technology on Reliability & Environmental Engineering Laboratory (6142A0502020817). The third author's research was supported by NSFC grants 11722112 and 91630207.

†Department of Applied Mathematics, Mathematics of Computational Science Group, University of Twente, Enschede, 7500 AE, The Netherlands (j.j.w.vandervegt@utwente.nl).

‡School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, People's Republic of China (yhxia@ustc.edu.cn).

§Corresponding author. School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, People's Republic of China (yxu@ustc.edu.cn).

algorithm can easily become unstable and lack robustness if the numerical solution violates these essential bounds.

In recent years, the development of positivity preserving discontinuous Galerkin (DG) finite element methods therefore has been a very active area of research. The standard approach to ensure that the numerical solution satisfies the bounds imposed by the partial differential equations is to use limiters, but this can easily result in loss of accuracy, especially for higher order accurate discretizations.

In a seminal paper, Zhang and Shu [34] showed how to design maximum principle and positivity preserving higher order accurate DG methods for first order scalar conservation laws. Their algorithm consists of a several important steps: (i) starting from a bounds preserving solution at time t_n, ensure that the element average of the solution satisfies the bounds at the next time level t_{n+1} by selecting a suitable time step in combination with a monotone first order scheme; (ii) limit the higher order accurate polynomial solution at the quadrature points in each element without destroying the higher order accuracy; (iii) higher order accuracy in time can then be easily obtained using explicit SSP Runge–Kutta methods [31]. This algorithm has been subsequently extended in many directions, e.g., various element shapes, the convection-diffusion equation, Euler and Navier–Stokes equations, and relativistic hydrodynamics [37, 38, 35, 36, 33, 29]. Other approaches to obtain higher order positivity preserving DG discretizations can be found in, e.g., [5, 13, 12].

All these DG discretizations use, however, an explicit time integration method. For many partial differential equations, this results in an efficient numerical discretization, where to ensure stability the time step is restricted by the Courant–Friedrichs–Lewy (CFL) condition. On locally dense meshes and for higher order partial differential equations, which often have a time step constraint $\triangle t \leq Ch^p$, with $p > 1$ and h the mesh size, these time-explicit algorithms can become computationally very costly. The alternative is to resort to implicit time integration methods, but positivity preserving time-implicit DG discretizations are still very much in their infancy. Meister and Ortleb developed in [22] a positivity preserving DG discretization for the shallow water equations using the Patankar technique [26]. Qin and Shu [28] extended the framework in [34, 35] to implicit positivity preserving DG discretizations of conservation laws in combination with an implicit Euler time integration method. An interesting result of the analysis in [28] is that to ensure positivity in the algorithm of Qin and Shu a lower bound on the time step is required. The approaches in [22, 28] require, however, a detailed analysis of the time-implicit DG discretization to ensure that the bounds are satisfied and are not so easy to extend to other classes of problems.

In this paper, we will present a very different approach to develop positivity preserving higher order accurate DG discretizations that are combined with a diagonally implicit Runge–Kutta (DIRK) time integration method. In analogy with obstacle problems, we consider the bounds imposed by a maximum principle or positivity constraint as a restriction on the DG solution space. The constraints are then imposed using a limiter and directly coupled to the time-implicit higher order accurate DG discretization using Lagrange multipliers. The resulting equations are the well-known Karush–Kuhn–Tucker (KKT) equations, which are frequently encountered in constrained optimization and solved with a semismooth Newton method [11, 17], and also used in constrained optimization-based discretizations of partial differential equation in, e.g., [3, 8, 10, 20]. The key benefit of the approach discussed in this paper, which we denote by KKT-Limiter and so far has not been applied to positivity preserving time-implicit DG discretizations, is that no detailed analysis is required to ensure that the DG discretization preserves the bounds for a particular partial differ-

ential equation. They are imposed explicitly and not part of the DG discretization. Also, since the limiter is only active in areas where positivity must be enforced, it does not affect the higher order DG discretization elsewhere since the Lagrange multipliers will be zero there. The approach discussed in this paper presents a general framework for how to couple DG discretizations with limiters and, very importantly, how to efficiently solve the resulting nonlinear algebraic equations.

The algebraic equations resulting from the KKT formulation of the positivity preserving time-implicit DG discretization are only semismooth. This excludes the use of standard Newton methods since they require C^1 continuity [9]. The obvious choice would be to use one of the many semismooth Newton methods available for nonlinear constrained optimization problems [11, 17], but the algebraic equations for the positivity preserving time-implicit DG discretization have a structure different from that for most constrained optimization problems. For instance, the conditions to ensure a nonsingular Jacobian [11] for methods based on the Fischer–Burmeister or related complementarity functions [23, 4] are not met by the KKT-Limiter in combination with a time-implicit DG discretization. This frequently results in nearly singular Jacobian matrices, poor convergence, and lack of robustness. We therefore developed an efficient active set semismooth Newton method that is suitable for the KKT formulation of time-implicit positivity preserving DG discretizations. Convergence of this semismooth Newton method can be proven using a specially designed quasi-directional derivative, as outlined in [15]; see also [17, 18].

The organization of this paper is as follows. In section 2, we formulate the KKT-equations, followed in section 3 by a discussion of an active set semismooth Newton method that is suitable to solve the nonlinear algebraic equations resulting from the positivity preserving time-implicit DG discretization. Special attention will be given to the quasi-directional derivative, which is an essential part to ensure convergence of the semismooth Newton method. In section 4, we discuss the DG discretization in combination with a DIRK time integration method and positivity constraints. In section 5, numerical experiments for the advection, Burgers, Allen–Cahn, Barenblatt, and Buckley–Leverett equations are provided. Conclusions are drawn in section 6. In Appendix B, more details on the quasi-directional derivative are given.

2. KKT limiting approach. In this section, we will directly couple the bounds preserving limiter to the time-implicit discontinuous Galerkin discretization using Lagrange multipliers. We will denote this approach as the KKT-Limiter.

Define the set

$$K := \{x \in \mathbb{R}^n \mid h(x) = 0,\ g(x) \leq 0\},$$

where $h : \mathbb{R}^n \to \mathbb{R}^l$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable functions denoting, respectively, the l equality and m inequality constraints to be imposed on the DG discretization. The variable x denotes the degrees of freedom and n the number of degrees of freedom in the unlimited DG discretization. For the continuously differentiable function $L : \mathbb{R}^n \to \mathbb{R}^n$, representing the unlimited discontinuous Galerkin discretization, the KKT-equations are

$$\mathcal{L}(x, \mu, \lambda) := L(x) + \nabla h(x)^T \mu + \nabla g(x)^T \lambda = 0, \tag{2.1a}$$

$$-h(x) = 0, \tag{2.1b}$$

$$0 \geq g(x) \perp \lambda \geq 0, \tag{2.1c}$$

with $\mu \in \mathbb{R}^l$, $\lambda \in \mathbb{R}^m$ the Lagrange multipliers. The compatibility condition (2.1c) is componentwise equal to

$$0 \geq g_j(x), \qquad \lambda_j \geq 0 \quad \text{and} \quad g_j(x)\lambda_j = 0, \quad j = 1, \ldots, m,$$

which is equivalent to

$$\min(-g(x), \lambda) = 0,$$

where the min-function is applied componentwise. The KKT-equations, with $F(z) \in \mathbb{R}^{n+l+m}$, can now be formulated as

$$0 = F(z) := \begin{pmatrix} \mathcal{L}(x, \mu, \lambda) \\ -h(x) \\ \min(-g(x), \lambda) \end{pmatrix}, \tag{2.2}$$

where $z := (x, \mu, \lambda)$. In the next section, we will discuss a global active set semismooth Newton method suitable for the efficient solution of (2.2) in combination with a DIRK-DG discretization. In section 4, the DG discretization and KKT-Limiter will be presented for a number of scalar conservation laws.

3. Semismooth Newton method. Standard Newton methods assume that $F(z)$ is continuously differentiable [9], but $F(z)$ given by (2.2) is only semismooth [11]. In this section, we will present a robust active set semismooth Newton method for (2.2) that is suitable for the efficient solution of the KKT-equations resulting from a higher order DG discretization combined with positivity preserving limiters and a DIRK time integration method [14].

3.1. Differentiability concepts. For the definition of the semismooth Newton method, we need several more general definitions of derivatives, which will be discussed in this section. For more details, we refer the reader to, e.g., [6, 11, 17, 30]. Since we use the semismooth Newton method directly on the algebraic equations of the limited DIRK-DG discretization, we only consider finite-dimensional spaces here.

Let $D \subseteq \mathbb{R}^m$ be an open subset in $\mathbb{R}^m$. Given $d \in \mathbb{R}^m$, the directional derivative of $F : D \to \mathbb{R}^n$ at $x \in D$ in the direction d is defined as

$$F'(x; d) := \lim_{t \downarrow 0^+} \frac{F(x + td) - F(x)}{t}. \tag{3.1}$$

A function $F : D \to \mathbb{R}^n$ is locally Lipschitz continuous if for every $x \in D$ there exist a neighborhood $N_x \subseteq D$ and a constant C_x, such that

$$|F(y) - F(z)| \leq C_x |y - z| \qquad \text{for all } y, z \in N_x.$$

If F is locally Lipschitz on D, then according to Rademacher's theorem, F is differentiable almost everywhere with derivative $F'(x)$. The B-subdifferential $\partial_B F(x)$ of $F(x)$ is then defined as

$$\partial_B F(x) := \lim_{\bar{x} \to x, \bar{x} \in D_F} F'(\bar{x}),$$

with D_F the points where F is differentiable, and the generalized derivative in the sense of Clarke is defined as

$$\partial F(x) := \text{ convex hull of } \partial_B F(x).$$

For example, $F(x) = |x|$ at $x = 0$ has $\partial_B F(0) = \{-1, 1\}$ and $\partial F(0) = [-1, 1]$. A function $F : D \to \mathbb{R}^n$ is called semismooth if [27]

$$\lim_{V \in \partial F(x+td'), d' \to d, t \downarrow 0^+} Vd' \quad \text{exists for all } d \in \mathbb{R}^m.$$

A function $F : D \to \mathbb{R}^n$ is Bouligand-differentiable (B-differentiable) at $x \in D$ if it is directionally differentiable at x and

$$\lim_{d \to 0} \frac{F(x+d) - F(x) - F'(x;d)}{|d|} = 0.$$

A locally Lipschitz continuous function F is B-differentiable at x if and only if it is directionally differentiable at x [30].

Given $d \in \mathbb{R}^m$, the Clarke generalized directional derivative of $F : D \to \mathbb{R}^n$ at $x \in D$ in the direction of d is defined by [6]

$$F^0(x;d) := \lim_{y \to x} \sup_{t \downarrow 0^+} \frac{F(y+td) - F(y)}{t}.$$

3.2. Global active set semismooth Newton method. For the construction of a global semismooth Newton method for (2.2), we will use the merit function $\theta(z) = \frac{1}{2}|F(z)|^2$, with $z = (x, \mu, \lambda)$. The Clarke directional derivatives of θ and F have the following relation.

Let $F : D \subseteq \mathbb{R}^p \to \mathbb{R}^p$, with D an open set and $p = n + l + m$, be a locally Lipschitz continuous function; then the Clarke generalized directional derivative of $\theta(z)$ can be expressed as [17]

$$\theta^0(z;d) = \limsup_{y \to z, t \downarrow 0^+} \frac{(F(z), (F(y+td) - F(y))}{t}, \tag{3.2}$$

and there exists an $F^0 : D \times \mathbb{R}^p \to \mathbb{R}^p$ such that

$$\theta^0(z;d) = (F(z), F^0(z;d)) \qquad \text{for } (z,d) \in D \times \mathbb{R}^p. \tag{3.3}$$

Here $(\cdot, \cdot)$ denotes the Euclidean inner product. The crucial point in designing a Newton method is to obtain proper descent directions for the Newton iterations. A possible choice is to use the Clarke derivative ∂F as the generalized Jacobian [11, 17], but this derivative is in general difficult to compute. In [24, 25], it was proposed to use d as the solution of

$$F(z) + F'(z;d) = 0, \tag{3.4}$$

which for the KKT-equations results in a mixed linear complementarity problem [25]. Unfortunately, (3.4) does not always have a solution, unless additional conditions are imposed. A better alternative is to use the quasi-directional derivative G of F [15, 17, 18].

Let $F : D \subseteq \mathbb{R}^p \to \mathbb{R}^p$ be directionally differentiable and locally Lipschitz continuous. Assume that $S = \{z \in D \mid |F(z)| \leq |F(z^0)|\}$ is bounded. Then $G : S \times \mathbb{R}^p \to \mathbb{R}^p$ is called the quasi-directional derivative of F on $S \subset \mathbb{R}^p$ if for all $z, \bar{z} \in S$ the following

conditions hold [15, 17, 18]:

$$(F(z), F'(z;d)) \le (F(z), G(z;d)), \tag{3.5a}$$

$$G(z;td) = tG(z;d) \quad \text{for all } d \in \mathbb{R}^p,\ z \in S, \text{ and } t \ge 0, \tag{3.5b}$$

$$(F(\bar{z}), F^0(\bar{z};\bar{d})) \le \limsup_{z\to\bar{z},d\to\bar{d}} (F(z), G(z;d)) \quad \text{for all } z \to \bar{z}, d \to \bar{d}. \tag{3.5c}$$

The search direction d in the semismooth Newton method is now the solution of

$$F(z) + G(z;d) = 0, \quad \text{with } z \in S, d \in \mathbb{R}^p, \tag{3.6}$$

which results for the KKT-equations (2.2) in a mixed linear complementarity problem. Using (3.3), (3.5c), and (3.6) this immediately results in the bound

$$\theta^0(\bar{z};\bar{d}) \le \limsup_{z\to\bar{z},d\to\bar{d}} (F(z), G(z;d)) = -\lim_{z\to\bar{z}} |F(z)|^2 = -2\theta(\bar{z}).$$

Hence the search direction d obtained from (3.6) always provides a descent direction for the merit function $\theta(z)$. The merit function $\theta(z)$ and the quasi-directional derivative $G(z,d)$ can therefore be used to define a global line search semismooth Newton algorithm, which is stated in Algorithm 3.1. The key benefit of using the quasi-directional derivative G in this Newton algorithm is that, under the additional assumption $\|G(z;d)\| \ge L\|d\|$, with $L > 0$ constant, we immediately obtain a proof of the convergence of this algorithm, given by [15, Theorem 1].

In the next section, we will present the quasi-directional derivative G for the KKT-equations (2.2) and define the active sets used to solve (3.6) with the semismooth Newton algorithm presented in section 3.4. In section 4, Algorithm 3.1 will then be used to solve the nonlinear equations resulting from the DG discretization using a KKT-limiter in combination with a DIRK method.

3.3. Quasi-directional derivative. In order to compute the quasi-directional derivative G, satisfying the conditions stated in (3.5), we first need to compute the directional and Clarke generalized directional derivatives of the function $F(z)$ defined in (2.2).

Define $z \in \mathbb{R}^p$, with $p = n + l + m$ as $z = (x, \mu, \lambda)$ with $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^l$, $\lambda \in \mathbb{R}^m$. Define $d \in \mathbb{R}^p$ as $d = (u, v, w)$, with $u \in \mathbb{R}^n$, $v \in \mathbb{R}^l$, $w \in \mathbb{R}^m$. The directional derivative $F'(z;d) \in \mathbb{R}^p \times \mathbb{R}^p$ of $F(z)$ defined in (2.2) in the direction d is equal to

$$F_i'(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, \qquad i \in N_n, \tag{3.7a}$$

$$F_{i+n}'(z;d) = -D_x h_i(x)\cdot u, \qquad i \in N_l, \tag{3.7b}$$

$$F_{i+n+l}'(z;d) = -D_x g_i(x)\cdot u, \qquad i \in \alpha(z), \tag{3.7c}$$

$$= \min(-D_x g_i(x)\cdot u, w_i), \qquad i \in \beta(z), \tag{3.7d}$$

$$= w_i, \qquad i \in \gamma(z), \tag{3.7e}$$

where the following sets are used:

$$\begin{aligned} N_q &= \{j \in \mathbb{N} \mid 1 \le j \le q\}, \\ \alpha(z) &= \{j \in \mathbb{N}_m \mid \lambda_j > -g_j(x)\}, \\ \beta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j = -g_j(x)\}, \\ \gamma(z) &= \{j \in \mathbb{N}_m \mid \lambda_j < -g_j(x)\}, \end{aligned}$$

with $q = n$ or $q = l$. The calculation of most of the terms in (3.7) is straightforward, except (3.7d), which can be computed using a Taylor series expansion of the arguments of $\min(-g_i(x), \lambda_i)$ in the limit of the directional derivative (3.1), combined with the relation $\min(a + b, a + d) - \min(a, a) = \min(b, d)$ and the fact that $i \in \beta(z)$.

The Clarke generalized derivative of $F(z)$ can be computed using the relations (3.2)–(3.3) and is equal to

$$
\begin{aligned}
&F_i^0(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, && i \in N_n, && \text{(3.8a)}\\
&F_{i+n}^0(z;d) = -D_x h_i(x)\cdot u, && i \in N_l, && \text{(3.8b)}\\
&F_{i+n+l}^0(z;d) = -D_x g_i(x)\cdot u, && i \in \alpha(z), && \text{(3.8c)}\\
&\qquad = \max(-D_x g_i(x)\cdot u, w_i), && i \in \beta(z), F_{i+n+l}(z) > 0, && \text{(3.8d)}\\
&\qquad = \min(-D_x g_i(x)\cdot u, w_i), && i \in \beta(z), F_{i+n+l}(z) \le 0, && \text{(3.8e)}\\
&\qquad = w_i, && i \in \gamma(z). && \text{(3.8f)}
\end{aligned}
$$

The calculation of (3.8d) and (3.8e) in $F^0(z;d)$ is nontrivial and is detailed in Appendix A.

Using the results for the directional derivative and the Clarke generalized directional derivative, we can now state a quasi-directional derivative $G : D \times \mathbb{R}^p \to \mathbb{R}^p$, satisfying the conditions (3.5), which for any $\delta > 0$ is equal to

$$
\begin{aligned}
&G_i(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, && i \in N_n, && \text{(3.9a)}\\
&G_{i+n}(z;d) = -D_x h_i(x)\cdot u, && i \in N_l, && \text{(3.9b)}\\
&G_{i+n+l}(z;d) = -D_x g_i(x)\cdot u, && i \in \alpha_\delta(z), && \text{(3.9c)}\\
&\qquad = \max(-D_x g_i(x)\cdot u, w_i), && i \in \beta_\delta(z), F_{i+n+l}(z) > 0, && \text{(3.9d)}\\
&\qquad = \min(-D_x g_i(x)\cdot u, w_i), && i \in \beta_\delta(z), F_{i+n+l}(z) \le 0, && \text{(3.9e)}\\
&\qquad = w_i, && i \in \gamma_\delta(z), && \text{(3.9f)}
\end{aligned}
$$

with the sets

$$
\begin{aligned}
\alpha_\delta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j > -g_j(x) + \delta\},\\
\beta_\delta(z) &= \{j \in \mathbb{N}_m \mid -g_j(x) - \delta \le \lambda_j \le -g_j(x) + \delta\},\\
\gamma_\delta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j < -g_j(x) - \delta\}.
\end{aligned}
$$

The main benefit of introducing the δ-dependent sets is that in practice it is hard to test for the set $\beta(z)$, which would generally be ignored in real computations due to rounding errors. One would then miss a number of important components in the quasi-directional derivative, which can significantly affect the performance of the Newton algorithm. The set β_δ gives, however, a computational well-defined quasi-directional derivative $G(z;d)$. In Appendix B, a proof is given that $G(z;d)$ satisfies the conditions stated in (3.5), which is the condition required in [15, Theorem 1], to ensure convergence of the semismooth Newton method.

The formulation of the quasi-directional derivative G (3.9) is, however, not directly useful as a Jacobian in the semismooth Newton method due to the max and min functions. In order to eliminate these functions, we introduce the sets

$$
\begin{aligned}
I_{\beta_\delta}^{11}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) > 0, -D_x g_i(x)\cdot u > w_i\},\\
I_{\beta_\delta}^{12}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) > 0, -D_x g_i(x)\cdot u \le w_i\},\\
I_{\beta_\delta}^{21}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) \le 0, -D_x g_i(x)\cdot u > w_i\},\\
I_{\beta_\delta}^{22}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) \le 0, -D_x g_i(x)\cdot u \le w_i\}
\end{aligned}
$$

and define

$$I^1_\delta(z,d) := \alpha_\delta(z) \cup I^{11}_{\beta_\delta}(z,d) \cup I^{22}_{\beta_\delta}(z,d), \tag{3.10a}$$

$$I^2_\delta(z,d) := \gamma_\delta(z) \cup I^{12}_{\beta_\delta}(z,d) \cup I^{21}_{\beta_\delta}(z,d). \tag{3.10b}$$

The quasi-directional derivative $G(z;d)$ can now be written in a form suitable to serve as a Jacobian in the active set semismooth Newton method defined in Algorithm 3.1 to solve (2.2):

$$G(z;d) = \widehat{G}(z)d,$$

with

$$\widehat{G}(z) = \begin{pmatrix} D_x\mathcal{L}_i(z)|_{i\in N_n} & D_\mu\mathcal{L}_i(z)|_{i\in N_n} & D_\lambda\mathcal{L}_i(z)|_{i\in N_n} \\ -D_x h_i(x)|_{i\in N_l} & 0 & 0 \\ -D_x g_i(x)|_{i\in I^1_\delta(z,d)} & 0 & \delta_{ij}|_{i,j\in I^2_\delta(z,d)} \end{pmatrix} \in \mathbb{R}^{p\times p}, \tag{3.11}$$

with δ_{ij} the Kronecker symbol. By updating the sets $I^1_\delta(z;d)$ and $I^2_\delta(z;d)$ as part of the Newton method, the complementary problem (3.6) is simultaneously solved with the solution of (2.2). In general, after a few iterations the proper sets $I^{1,2}_\delta(z;d)$ will be found and the semismooth Newton method then converges like a regular Newton method. Also, one should note that *only* the contribution $D_x\mathcal{L}_i(z)$ in (3.11) depends on the DG discretization in $\mathcal{L}_i(z)$. Hence, the KKT-Limiter provides a general framework to impose limiters on time-implicit numerical discretizations and could, for instance, also be applied to time-implicit finite volume discretizations.

3.4. Active set semismooth Newton algorithm. As default values we use in Algorithm 3.1 $\bar{\alpha} = 10^{-12}$, $\beta = \gamma = \frac{1}{2}$, $\sigma = 10^{-9}$, $\delta = 10^{-12}$, and $\epsilon = 10^{-8}$.

An important aspect of Algorithm 3.1 is that we simultaneously solve the mixed linear complementarity equations (3.6) for the search direction d as part of the global Newton method using an active set technique. This was motivated by [16] and will reduce the mixed linear complementarity problem (3.6) into a set of linear equations. The use of the active set technique is also based on the observation in [18] of the close relation between an active set Newton method and a semismooth Newton method. After the proper sets $I^1_\delta(z;d)$, $I^2_\delta(z;d)$ are obtained for the quasi-directional derivative $G(z;d)$, the difference with a Newton method for smooth problems [9] will be rather small. The mixed linear complementarity problem can, however, have one, multiple, or no solutions, and, in order to deal also with cases where the matrix G is poorly conditioned, we will use a minimum norm least squares or Gauss–Newton method to solve the algebraic equations (3.12).

For the performance of a Newton algorithm, proper scaling of the variables is crucial. Here we use the approach outlined in [9] and the Newton method is applied directly to the scaled variables. Also, the matrix $\widehat{G}_k^T\widehat{G}_k + \bar{\alpha}\|F(z^k)/F(z^0)\|I$ in the Newton method will have a much larger condition number than the matrix $\widehat{G}_k$. In order to improve the conditioning of this matrix, we use simultaneous iterative row and column scaling in the L^∞-matrix norm, as described in [2]. This algorithm very efficiently scales the rows and columns such that an L^∞-matrix norm approximately equal to one is obtained. This gives a many orders of magnitude reduction in the matrix condition number and generally reduces the condition number of the matrix (3.12) to the same order as the condition number of the original matrix $\widehat{G}_k$.

Algorithm 3.1 Active set semismooth Newton method.

(A.0) (*Initialization*) Let $\bar{\alpha} \geq 0$, $\beta, \gamma \in (0,1)$, $\sigma \in (0, \bar{\sigma})$, $\delta > 0$, and $b > C \in \mathbb{R}^+$ arbitrarily large, but bounded. Choose $z^0, d^0 \in \mathbb{R}^p$ and tolerance ϵ.
(A.1) Scale z^0.
(A.2) (*Newton method*)
for $k = 0, 1, \ldots$ until $\|F(z^k)\| \leq \epsilon$ **and** $\|d^k\| \leq \epsilon$ **do**
    Compute the quasi-directional derivative matrix $\widehat{G}_k := \widehat{G}(z^k)$ given by (3.11) and the active sets $I^1_\delta(z;d)$, $I^2_\delta(z;d)$ of $\widehat{G}_k$ given by (3.10).
    Apply row-column scaling to $(\widehat{G}_k^T \widehat{G}_k + \bar{\alpha}\|F(z^k)/F(z^0)\|I)$, with I the identity matrix, such that the matrix has a norm $\|\cdot\|_{L^\infty} \cong 1$.
    if there exists a solution h^k to

$$(\widehat{G}_k^T \widehat{G}_k + \bar{\alpha}\|F(z^k)/F(z^0)\|I)h^k = -\widehat{G}_k^T F(z^k), \tag{3.12}$$

    with $|h^k| \leq b|F(z^k)|$ **and**

$$|F(z^k + h^k)| < \gamma |F(z^k)|,$$

    then
        Set $d^k = h^k$, $z^{k+1} = z^k + d^k$, $\alpha_k = 1$, and $m_k = 0$.
    else
        Choose $d^k = h^k$.
        Compute $\alpha_k = \beta^{m_k}$, where m_k is the first positive integer m for which

$$\theta(z^k + \beta^{m_k} d^k) - \theta(z^k) \leq -\sigma \beta^m \theta(z^k).$$

        Set $z^{k+1} = z^k + \alpha_k d^k$.
    end if
end for

4. KKT-Limiter DG discretization. Given a domain $\Omega \subseteq \mathbb{R}^d$, $d = \dim(\Omega)$, $d = 1, 2$, with Lipschitz continuous boundary $\partial\Omega$. As a general model problem we consider the following second order nonlinear scalar equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) + G(u) - \nabla \cdot (\nu(u)\nabla u) = 0, \tag{4.1}$$

with $u(x,t) : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}$ a scalar quantity, $F(u) : \mathbb{R} \to \mathbb{R}^d$ the flux, $G(u) : \mathbb{R} \to \mathbb{R}$ a reaction term, and $\nu(u) : \mathbb{R} \to \mathbb{R}^+$ a nonlinear diffusion term. By selecting different functions F, G, and ν in (4.1) we will demonstrate in section 5 the KKT-Limiter on various model problems that impose different positivity constraints on the solution.

For the DG discretization, we introduce the auxiliary variable $Q \in \mathbb{R}^d$ and rewrite (4.1) as a first order system of conservation laws

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) + G(u) - \nabla \cdot (\nu(u)Q) = 0, \tag{4.2a}$$

$$Q - \nabla u = 0. \tag{4.2b}$$

4.1. DG discretization. Let $\mathcal{T}_h$ be a tessellation of the domain Ω with shape regular line or quadrilateral elements K with maximum diameter $h > 0$. The total number of elements in $\mathcal{T}_h$ is N_K. We denote the union of the set of all boundary faces ∂K, $K \in \mathcal{T}_h$, as $\mathcal{F}_h$, denote all internal faces $\mathcal{F}_h^i$ and the boundary faces as $\mathcal{F}_h^b$, and hence get $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^b$. The elements connected to each side of a face

$S \in \mathcal{F}_h$ are denoted by the indices L and R, respectively. For the KKT-Limiter, it is important to use orthogonal basis functions; see section 4.2. In this paper, $\mathcal{P}_p(K)$ represent tensor product Legendre polynomials of degree p on d-dimensional rectangular elements $K \in \mathcal{T}_h$, when K is mapped to the reference element $(-1,1)^d$. For general elements, one can use Jacobi polynomials with proper weights to obtain an orthogonal basis; see [19, section 3.2]. Next, we define the finite element spaces

$$V_h^p := \Big\{ v \in L^2(\Omega) \mid v|_K \in \mathcal{P}_p(K) \ \forall K \in \mathcal{T}_h \Big\},$$
$$W_h^p := \Big\{ v \in (L^2(\Omega))^d \mid v|_K \in (\mathcal{P}_p(K))^d \ \forall K \in \mathcal{T}_h \Big\},$$

with $L^2(\Omega)$ the Sobolev space of square integrable functions. Equation (4.2) is discretized using the local discontinuous Galerkin discretization from [7]. Define $L_h^1 : V_h^p \times W_h^p \times V_h^p \to \mathbb{R}$ and $L_h^2 : V_h^p \times W_h^p \to \mathbb{R}$ as

$$\begin{aligned} L_h^1(u_h, Q_h; v) :=& -\big(F(u_h) - \nu(u_h)Q_h, \nabla_h v\big)_\Omega + \big(G(u_h), v\big)_\Omega \\ &+ \sum_{S \in \mathcal{F}_h^i} \big(H(u_h^L, u_h^R; n^L) - \widehat{\nu(u_h)} n^L \cdot \widehat{Q_h}, v^L - v^R\big)_S \\ &+ \sum_{S \in \mathcal{F}_h^b} \big(H(u_h^L, u_h^b; n^L) - \widehat{\nu(u_h)} n^L \cdot Q_h^b, v^L\big)_S, \end{aligned} \tag{4.3}$$

$$\begin{aligned} L_h^2(u_h; w) :=& \big(u_h, \nabla_h \cdot w\big)_\Omega - \sum_{S \in \mathcal{F}_h^i} \big(\widehat{u_h} n^L, w^L - w^R\big)_S \\ &- \sum_{S \in \mathcal{F}_h^b} \big(u_h^b n^L, w^L\big)_S, \end{aligned}$$

where $(\cdot,\cdot)_D$ is the $L^2(D)$ inner product, ∇_h is the elementwise ∇ operator, and the superscript b refers to boundary data. Here $n^L \in \mathbb{R}^d$ is the exterior unit normal vector at the boundary of the element $L \in \mathcal{T}_h$ that is connected to face S. The numerical flux H is the Lax–Friedrichs flux

$$H(u_h^L, u_h^R; n) = \frac{1}{2}\big(n \cdot (F(u_h^L) + F(u_h^R)) - C_{LF}(u_h^R - u_h^L)\big),$$

with Lax–Friedrichs coefficient $C_{LF} = \sup_{u_h \in [u_h^L, u_h^R]} |\frac{\partial}{\partial u_h}(n \cdot F(u_h))|$. For $\widehat{Q_h}$ and $\widehat{u_h}$, we use the alternating fluxes

$$\widehat{Q_h} = (1-\alpha)Q_h^L + \alpha Q_h^R, \tag{4.4a}$$
$$\widehat{u_h} = \alpha u_h^L + (1-\alpha)u_h^R, \tag{4.4b}$$

with $0 \le \alpha \le 1$. The numerical flux for the nonlinear diffusion is defined as

$$\widehat{\nu(u_h)} = \frac{1}{2}(\nu(u_h^L) + \nu(u_h^R)).$$

For $t \in (0, T]$, the semidiscrete DG formulation for (4.2) now can be expressed as follows: Find $u_h(t) \in V_h^p$, $Q_h(t) \in W_h^p$, such that for all $v \in V_h^p$, $w \in W_h^p$,

$$\left(\frac{\partial u_h}{\partial t}, v\right)_\Omega + L_h^1(u_h, Q_h; v) = 0, \tag{4.5a}$$
$$(Q_h, w)_\Omega + L_h^2(u_h; w) = 0. \tag{4.5b}$$

These equations are discretized in time with a DIRK method [14]. The main benefit of the DIRK method is that the RK stages can be computed successively, which significantly reduces the computational cost and memory overhead.

We represent u_h and Q_h in each element $K \in \mathcal{T}_h$, respectively, as $u_h|_K = \sum_{j=1}^{N_u} \widehat{U}_j^K \phi_j^K$ and $Q_h|_K = \sum_{j=1}^{N_Q} \widehat{Q}_j^K \psi_j^K$, with basis functions $\phi_j^K \in \mathcal{P}_p(K)$, $\psi_j^K \in \big(\mathcal{P}_p(K)\big)^d$ and DG coefficients $\widehat{U}_j^K \in \mathbb{R}$, $\widehat{Q}_j^K \in \mathbb{R}^d$. After replacing the test functions $v \in V_h^p$ in (4.5a) and $w \in W_h^p$ (4.5b) with, respectively, the independent basis functions $\phi_i^K \in \mathcal{P}_p(K)$, $i = 1, \dots, N_u$, and $\psi_i^K \in \big(\mathcal{P}_p(K)\big)^d$, $i = 1, \dots, N_Q$, we obtain the algebraic equations for the DG discretization.

In order to simplify notation, we introduce $\widehat{L}_h^1(\widehat{U}, \widehat{Q}) = L_h^1(u_h, Q_h; \phi) \in \mathbb{R}^{N_u N_K}$ and $\widehat{L}_h^2(\widehat{U}) = L_h^2(u_h; \psi) \in \mathbb{R}^{dN_Q N_K}$, with N_K the number of elements in $\mathcal{T}_h$ and $\phi = \phi_i^K$, $\psi = \psi_i^K$ the basis functions in element K. The algebraic equations for the DIRK stage vector $\widehat{K}^{(i)} \in \mathbb{R}^{N_u N_K}$, $i = 1, \dots, s$, with the DG coefficients can then be expressed as

$$\widehat{L}_h(\widehat{K}^{(i)}) := M_1\big(\widehat{K}^{(i)} - \widehat{U}^n\big) + \triangle t \sum_{j=1}^{i} a_{ij} \widehat{L}_h^1\big(\widehat{K}^{(j)}, -M_2^{-1} \widehat{L}_h^2(\widehat{K}^{(j)})\big) = 0. \tag{4.6}$$

Here we eliminated the DG coefficients for the auxiliary variable Q_h using (4.5b). The matrices $M_1 \in \mathbb{R}^{N_u N_K \times N_u N_K}$, $M_2 \in \mathbb{R}^{dN_Q N_K \times dN_Q N_K}$ are block-diagonal mass matrices since we use orthogonal basis functions and n denotes the index of time level $t = t_n$.

The coefficients a_{ij} are the coefficients in the Butcher tableau, which determine the properties of the RK method [14]. For DIRK methods, $a_{ij} = 0$ if $j > i$. The following DIRK methods are used: for basis functions with polynomial order $p = 1$ [1, page 1012, Theorem 5, first method with $\alpha = 1 - \frac{1}{2}$]; $p = 2$ [32, page 2117 (top)]; $p = 3$ [1, page 1012, Theorem 5, second method]; see also [32, page 2117 (top)]. The order of accuracy of these DIRK methods is $p + 1$, and their coefficients in the Butcher tableau satisfy $a_{sj} = b_j$, $j = 1, \dots, s$, which implies that these methods are stiffly accurate (see [14, section IV.6]), and the solution of the last DIRK stage is equal to the solution at the new time step

$$\widehat{U}^{n+1} = \widehat{K}^{(s)}.$$

Since each DIRK stage vector must satisfy the positivity constraints, this then also immediately applies to the solution at time t_{n+1}.

The Jacobian $D_x \mathcal{L}(\widehat{K}^{(i)}) \in \mathbb{R}^{N_u N_K \times N_u N_K}$, with $x = \widehat{K}^{(i)}$, in the quasi-directional derivative G (3.11) of DIRK stage i of the unlimited DIRK-DG discretization (4.6) is now equal to

$$D_x \mathcal{L}(\widehat{K}^{(i)}) = M_1 + \triangle t a_{ii} \left(\frac{\partial L_h^1}{\partial \widehat{K}^{(i)}} - \frac{\partial L_h^1}{\partial \widehat{Q}^{(i)}} M_2^{-1} \frac{\partial L_h^2}{\partial \widehat{K}^{(i)}} \right).$$

4.2. Limiter constraints. The limiter constraints for the DG discretization can be imposed directly by defining the inequality constraints in the KKT-equations. In each element $K \in \mathcal{T}_h$, we apply for each DIRK-stage $i = 1, \dots, s$ the following inequality constraints:

(i) *Positivity constraint*:

$$g_{1,k}^K(\widehat{K}^{K,(i)}) = u_{\min} - \sum_{q=1}^{N_u} \widehat{K}_q^{K,(i)} \phi_q^K(x_k), \quad k = 1, \dots, N_p. \tag{4.7}$$

(ii) *Maximum constraint*:

$$g_{2,k}^K(\widehat{K}^{K,(i)}) = \sum_{q=1}^{N_u} \widehat{K}_q^{K,(i)} \phi_q^K(x_k) - u_{\max}, \quad k = 1, \ldots, N_p. \tag{4.8}$$

Here the superscript K refers to element $K \in \mathcal{T}_h$, and (i) is the ith DIRK stage. The points x_k, $k = 1, \ldots, N_p$, are the points in element K where the inequality constraints are imposed and $u_{\min}$ and $u_{\max}$ denote, respectively, the allowed minimum and maximum values of u. The inequality constraints are imposed using the Lagrange multiplier λ; see (2.1c).

(iii) *Conservation constraint*:

Since the basis functions ϕ_j^K, $j = 1, \ldots, N_u$, are orthogonal in each element K, we have $(1, \phi_j^K)_K = 0$ for $j = 2, \ldots, N_u$. Hence, at each RK stage i, limiting the DG coefficients $\widehat{K}_j^{K,(i)}$, with $j = 2, \ldots, N_u$, has no effect on the element average $\bar{u}_h^{K,(i)} = \frac{1}{|K|}(u_h^{(i)}, 1)_K = \widehat{K}_1^{K,(i)}$, with $u_h^{(i)}$ the solution at stage i, and therefore does not influence the conservation properties of the DG discretization.

Limiting the DG coefficients $\widehat{K}_1^{K,(i)}$ can, however, affect the conservation properties of the DG discretization since $\bar{u}_h^{K,(i)} = \widehat{K}_1^{K,(i)}$. In order to ensure local conservation, we therefore need to impose in each element the local conservation constraint

$$\begin{aligned} h^K\big(\widehat{K}^{K,(i)}\big) &= \widehat{L}_{h,1}^K(\widehat{K}^{(i)}) \\ &= |K|\big(\widehat{K}_1^{K,(i)} - \widehat{U}_1^n\big) + (G(u_h^{(i)}), \phi_1^K)_K \\ &+ \sum_{S \in \mathcal{F}_h^i \cap \partial K} \big(H(u_h^{L,(i)}, u_h^{R,(i)}; n^L) \\ &- \widehat{\nu(u_h)} n^L \cdot ((1-\alpha) Q_h^{L,(i)} + \alpha Q_h^{R,(i)}), \phi_1^L - \phi_1^R\big)_S \\ &+ \sum_{S \in \mathcal{F}_h^b \cap \partial K} \big(H(u_h^{L,(i)}, u_h^b; n^L) - \widehat{\nu(u_h)} n^L \cdot Q_h^b, \phi_1^L\big)_S, \end{aligned} \tag{4.9}$$

with $\widehat{L}_{h,1}^K$ the equation for the element mean in element K in (4.6). The conservation constraint (4.9) is imposed using the Lagrange multiplier μ; see (2.1b). The conservation constraint explicitly ensures that at each RK stage the equation for the element mean $\bar{u}_h^{K,(i)}$ is exactly preserved in each element, and hence the KKT-Limiter does not affect the conservation properties of the DG discretization.

The remaining Jacobians $D_x h_i(x) \in \mathbb{R}^{N_K \times N_u N_K}$, $D_x g_i(x) \in \mathbb{R}^{N_p N_K \times N_u N_K}$ and $D_\mu \mathcal{L}_i(z) \in \mathbb{R}^{N_u N_K \times N_K}$, $D_\lambda \mathcal{L}_i(z) \in \mathbb{R}^{N_u N_K \times N_p N_K}$, with $x = \widehat{K}^{(i)}$, in the quasi-directional derivative matrix $\widehat{G}$ (3.11) are now straightforward to calculate.

It is important to ensure that the initial solution also satisfies the positivity constraints. An L^2-projection of the solution will in general not satisfy these constraints for a nonsmooth solution. To ensure that the initial solution also satisfies the positivity constraints, we apply a constrained projection using the active set semismooth Newton method given by Algorithm 3.1. The only difference is now that instead of

(4.6) we use L^2-projection

$$\widehat{L}_{hi}(\widehat{U}^0) = M^1\widehat{U}^0 - (u_0, \phi_i)_\Omega$$

and combine this with the positivity constraints (4.7)–(4.8). Here u_0 denotes the initial solution. As the initial solution for the constrained projection we use in Algorithm 3.1 the standard L^2-projection without constraints.

The positivity constraints are imposed at all element quadrature points since only the solution at these quadrature points is used in the DG discretization. In one dimension we use Gauss–Lobatto quadrature rules and in two dimensions product Gauss–Legendre quadrature rules. Since the number of quadrature points in an element is generally larger than the number of degrees of freedom in an element, this will result in an overdetermined set of algebraic equations and a rank deficit Jacobian matrix if the number of active constraints in an element is larger than the degrees of freedom N_u in the element. In order to obtain in Algorithm 3.1 accurate search directions h^k, we use the Gauss–Newton method given by (3.12). This approach can efficiently deal with the possible rank deficiency of the Jacobian matrix.

In practice, it will not be necessary to apply the inequality constraints in all elements, and one can significantly reduce the computational cost and memory overhead by excluding those elements for which it is obvious that they will meet the constraints anyway.

5. Numerical experiments. In this section, we will discuss a number of numerical experiments to demonstrate the performance of the DIRK-DG scheme with the positivity preserving KKT-Limiter. All computations were performed using the default values for the coefficients listed for Algorithm 3.1, except that for the accuracy tests discussed in section 5.1 we use $\epsilon = 10^{-10}$. The upwind coefficient α in (4.4) is set to $\alpha = 1$. In all 1D computations, the local conservation constraint is imposed and satisfied with an error less than 10^{-12}.

5.1. Accuracy tests. It is important to investigate whether the KKT-Limiter negatively affects the accuracy of the DG discretization in case the exact solution is smooth, but where also a positivity preserving limiter is required to ensure that the numerical solution stays within the bounds. To investigate this, we conduct the same accuracy tests as conducted in Qin and Shu [28, section 5.1]. Both the linear advection and the inviscid Burgers' equation are considered, which are obtained by setting $F(u) = u$ and $F(u) = \frac{1}{2}u^2$, respectively, and $G(u) = \nu(u) = 0$ in (4.1).

Example 5.1 (steady state solution to the linear advection equation). We consider

$$u_t + u_x = \sin^4 x, \qquad u(x,0) = \sin^2 x, \quad u(0,t) = 0, \tag{5.1}$$

with an outflow boundary condition at $x = 2\pi$. The exact solution $u(x,t)$ is positive for all $t > 0$; see [28]. As the steady state solution we use the solution at $t = 500$, when all residuals are approximately 10^{-16}. During the computations, the CFL number is dynamically adjusted between 10 and 89. For the time integration, an implicit Euler method is used. In Tables 1 and 2, the results of the accuracy tests, without and with the KKT-Limiter, are shown. The results in Table 2 show that the KKT-Limiter does not negatively affect the accuracy. For all test cases, the optimal accuracy in the L^2- and L^∞-norms is obtained. Also, the limiter is necessary, as can be seen from Table 1, and preserves the imposed positivity bound $u_{h\,\min} = 10^{-14}$ for the numerical solution.

TABLE 1
Error table for steady state linear advection equation (5.1) *without limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	1.461068e-02	-	2.044253e-02		–5.169578e-03
	40	3.702581e-03	1.98	5.287628e-03	1.95	–2.883487e-04
1	80	9.288342e-04	2.00	1.331962e-03	1.99	–1.208793e-05
	160	2.324090e-04	2.00	3.336614e-04	2.00	–4.036603e-07
	320	5.811478e-05	2.00	8.345620e-05	2.00	–1.282064e-08
	20	9.287703e-04	-	1.776878e-03	-	–4.952018e-05
	40	1.177042e-04	2.98	2.489488e-04	2.84	–1.627459e-06
2	80	1.476405e-05	3.00	3.200035e-05	2.96	–5.149990e-08
	160	1.847107e-06	3.00	4.027944e-06	2.99	–1.614420e-09
	320	2.309385e-07	3.00	5.043677e-07	3.00	–5.049013e-11
	20	5.653820e-05	-	1.230308e-04	-	–3.877467e-05
	40	3.583918e-06	3.98	7.803741e-06	3.98	–1.326415e-06
3	80	2.247890e-07	3.99	4.950122e-07	3.98	–4.237972e-08
	160	1.406175e-08	4.00	3.090593e-08	4.00	–1.331692e-09
	320	8.790539e-10	4.00	1.935324e-09	4.00	–4.167274e-11

TABLE 2
Error table for steady state linear advection equation (5.1) *with limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	1.464990e-02	-	2.044253e-02	-	9.998946e-15
	40	3.702367e-03	1.98	5.287628e-03	1.95	9.999813e-15
1	80	9.288338e-04	2.00	1.331962e-03	1.99	1.000000e-14
	160	2.324090e-04	2.00	3.336614e-04	2.00	1.000000e-14
	320	5.811478e-05	2.00	8.345620e-05	2.00	1.000000e-14
	20	9.290268e-04	-	1.776878e-03	-	1.000000e-14
	40	1.177053e-04	2.98	2.489488e-04	2.84	1.000000e-14
2	80	1.476406e-05	3.00	3.200035e-05	2.96	1.000000e-14
	160	1.847107e-06	3.00	4.027944e-06	2.99	1.000000e-14
	320	2.309385e-07	3.00	5.043677e-07	3.00	1.000000e-14
	20	5.742649e-05	-	1.230309e-04	-	9.999990e-15
	40	3.592170e-06	4.00	7.803745e-06	3.98	1.000000e-14
3	80	2.248562e-07	4.00	4.950122e-07	3.98	1.000000e-14
	160	1.406228e-08	4.00	3.090593e-08	4.00	1.000000e-14
	320	8.790580e-10	4.00	1.935323e-09	4.00	1.000000e-14

Example 5.2 (steady state solution to the inviscid Burgers' equation). We consider the inviscid Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = \sin^3\left(\frac{x}{4}\right), \qquad u(x,0) = \sin^2\left(\frac{x}{4}\right), \quad u(0,t) = 0, \tag{5.2}$$

with an outflow boundary condition at $x = 2\pi$. The exact solution $u(x,t)$ is positive for all $t > 0$; see [28]. As the steady state solution we use the solution at $t = 20.000$, when all residuals are approximately 10^{-16}. During the computations, the CFL number is dynamically adjusted between 10 and 954. For the time integration, an implicit Euler method is used. In Tables 3 and 4, the results of the accuracy tests, without and with the KKT-Limiter, show that the KKT-Limiter does not negatively

affect the accuracy. For all test cases, optimal accuracy in the L^2- and L^∞-norms is obtained. Also, the limiter is necessary and preserves the imposed positivity bound $u_{h\,\min} = 10^{-14}$ for the numerical solution.

TABLE 3
Error table for the steady state inviscid Burgers' equation (5.2) *without limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	2.110016e-03	-	3.387013e-03	-	–2.347303e-03
	40	5.230241e-04	2.01	8.577912e-04	1.98	–5.865522e-04
1	80	1.297377e-04	2.01	2.151386e-04	2.00	–1.466204e-04
	20	2.122765e-05	-	3.024868e-05	-	–1.048636e-05
	40	2.623666e-06	3.02	3.731754e-06	3.02	–6.681764e-07
2	80	3.266401e-07	3.01	4.634046e-07	3.01	–4.196975e-08
	20	2.985321e-07	-	1.895437e-06	-	1.895437e-06
	40	1.452601e-08	4.36	1.196963e-07	3.99	1.196963e-07
3	80	7.368455e-10	4.30	7.500564e-09	4.00	7.500564e-09
	160	3.948207e-11	4.22	4.346084e-10	4.11	4.346084e-10

TABLE 4
Error table for steady state inviscid Burgers' equation (5.2) *with limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	2.208009e-03	-	3.637762e-03	-	9.999813e-15
	40	5.358952e-04	2.04	9.282398e-04	1.97	1.000003e-14
1	80	1.313948e-04	2.03	2.339566e-04	1.99	1.000003e-14
	20	2.116746e-05	-	3.024864e-05	-	1.000003e-14
	40	2.622584e-06	3.01	3.731752e-06	3.02	1.000139e-14
2	80	3.266221e-07	3.01	4.634046e-07	3.01	1.000040e-14
	20	2.985321e-07	-	1.895437e-06	-	1.895437e-06
	40	1.452601e-08	4.36	1.196963e-07	3.99	1.196963e-07
3	80	5.610147e-10	4.70	1.574760e-09	6.25	1.000105e-14
	160	3.232240e-11	4.11	9.038604e-11	4.12	1.000017e-14

5.2. Time-dependent tests. In this section, we will present results of simulations of the linear advection, Allen–Cahn, Barenblatt, and Buckley–Leverett equations. The order of accuracy of the DIRK time integration method is always $p+1$, with p the polynomial order of the spatial discretization. The minimum value of the residual $F(z)$ and Newton update d in Algorithm 3.1 to stop the Newton iterations is $\epsilon = 10^{-8}$ for each DIRK stage. This is a quite strong stopping criterion, and in practice the values are often smaller at the end of each DIRK stage. It is also important to make sure that the Newton stopping criterion is in balance with the accuracy required for the constraints. If the algebraic equations are not solved sufficiently accurate, then it is not likely that the KKT-constraints will be satisfied.

The time step for the DIRK method is dynamically computed, based on the CFL or diffusion number. If the Newton method does not converge within a predefined number of iterations, then the computation for the time step will be restarted with $\triangle t/2$. This is generally more efficient than conducting many Newton iterations. In the next time step, the time step will then be increased to $1.2\triangle t$, until the maximum

CFL number is obtained. In practice, depending on the severity of the nonlinearity, the time step will be constantly adjusted during the computations.

Example 5.3 (1D linear advection equation). We consider (5.1) with a zero right-hand side in the domain $\Omega = [0, 10]$ and periodic boundary conditions. The exact solution is

$$u(x,t) = \max(\cos(2\pi(x-t)/10), 0) \quad \text{for } x \in \Omega,\ t \in [0,T].$$

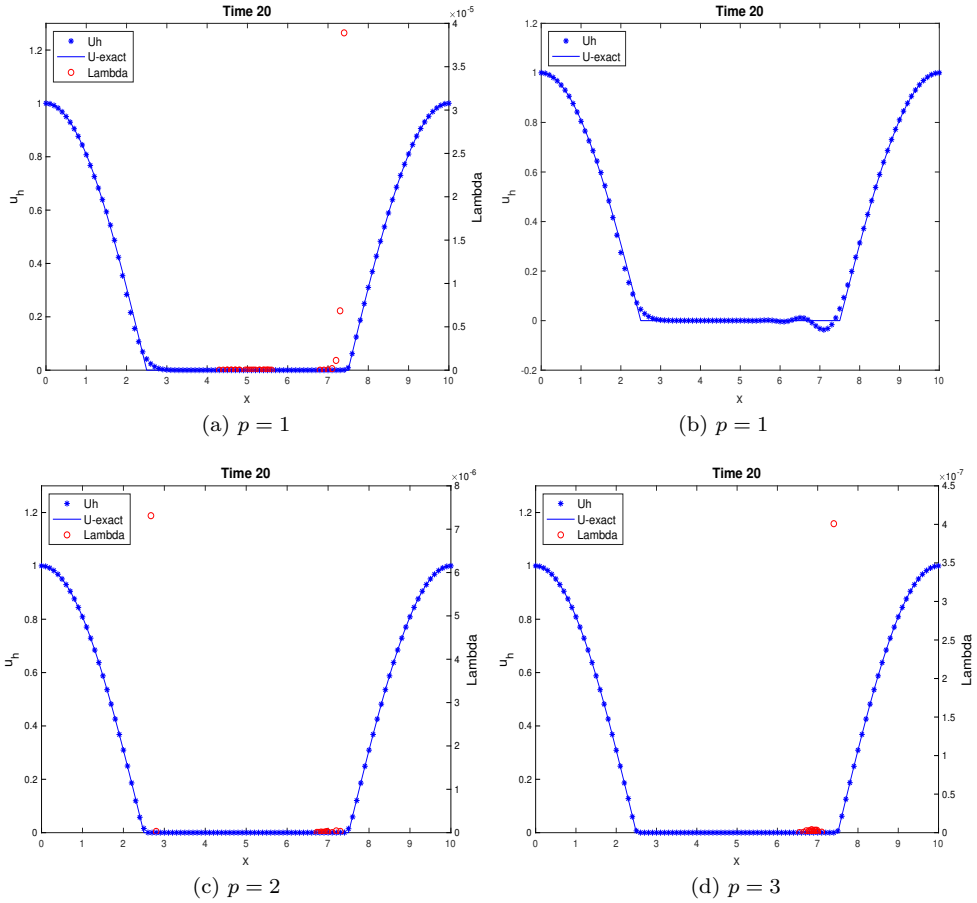
A constrained projection of $u(x,0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x,0)$. The computational mesh contains 100 elements, and the maximum CFL number is 1. In Figures 1a, 1c, and 1d, the exact and numerical solutions at time $t = 20$ are plotted for, respectively, polynomial orders 1, 2, and 3. At this time the wave has traveled twice through the domain and the numerical solution matches very well with the exact solution. Also plotted is the value of the Lagrange multipliers used to impose the positivity constraint $u_{h\,\min} = 10^{-10}$. These plots clearly show that the limiter is only active at locations where the constraint must be imposed and not in the smooth part of the solution. In Figure 1b, the solution for polynomial order $p = 1$ without the KKT-Limiter is plotted, which clearly shows that without the limiter the solution is significantly below the $u = 0$ minimum of the exact solution $u(x,t)$.

Example 5.4 (2D linear advection equation). The KKT-Limiter is also tested on a 2D linear advection equation, which is obtained by setting $F(u) = cu$, with $c = (-1,-2)$, and $G(u) = \nu(u) = 0$ in (4.1). The domain $\Omega = [0,3]^2$ with periodic boundary conditions is used in the computations. The computational mesh contains 30×30 elements. The exact solution is

$$u(x,t) = \max(\cos(2\pi(x+t)/3)\cos(2\pi(y+2t)/3), 0) \quad \text{for } x \in \Omega, t \in [0,T].$$

A constrained projection of $u(x,0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x,0)$. The maximum CFL number is 1. In Figure 2a the numerical solution is shown at $t = 6.3428$ and in Figure 2b the values of the Lagrange multipliers used to enforce the positivity constraint $u_{h\,\min} = 10^{-10}$. Comparing Figures 2a and 2b clearly shows that the KKT-Limiter is only active in those parts of the domain where the solution needs to satisfy the positivity constraint and not in the smooth part.

Example 5.5 (1D Burgers' equation). In order to test the KKT-Limiter on problems with time-dependent shocks, we consider the 1D Burgers' equation on a domain $\Omega = [-1,1]$ with initial condition $u_0 = \max(\cos(\pi x), 0)$ and periodic boundary conditions. The polynomial order is $p = 3$. As lower and upper bounds in the positivity preserving limiter we use, respectively, $u_{h\,\min} = 10^{-10}$ and $u_{h\,\max} = 1$, and no monotonicity constraint is imposed. The initially smooth part of the solution develops into a shock. The onset of the shock is shown in Figure 3a and the later stages of the shock at $t = 0.65$ in Figure 3b. Figure 3c shows the solution when the conservation constraint (4.9) is not explicitly enforced. The difference in the shock solution for the discretizations with and without the explicitly imposed conservation constraint is very small. The main reason for this is that the KKT-Limiter is only active in regions where the constraints must be imposed and does not affect the discretization at other places in the domain. This can be seen from the values of the Lagrange multipliers that are used to impose the positivity constraints, which are indicated with red



(a) $p = 1$ (b) $p = 1$

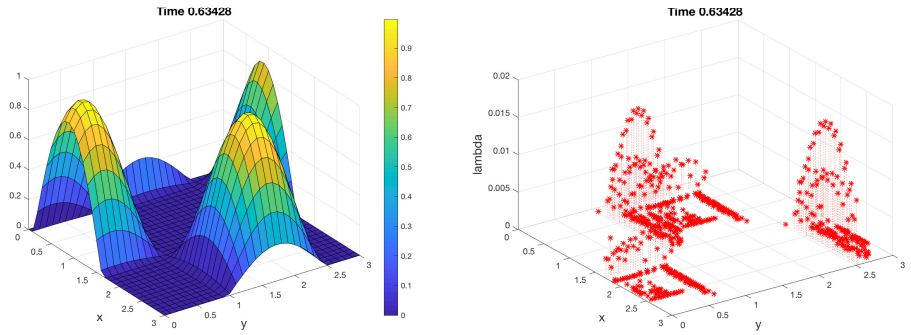
(c) $p = 2$ (d) $p = 3$

FIG. 1. *Example* 5.3, 1*D advection equation:* (a), (c), (d) *numerical solution* u_h *with positivity preserving limiter, polynomial order, respectively,* $p = 1$, 2, *and* 3*;* (b) *numerical solution* u_h *without positivity preserving limiter, polynomial order* $p = 1$. *Computational mesh* 100 *elements. Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (a), (c), *and* (d) *with a red (open) circle.*

circles, and are only nonzero in the vicinity of the shock and at locations where the solution has a discontinuous derivative. The KKT-Limiter to ensure the positivity constraints therefore has a very small effect on the conservation properties of the DG discretization, as can be seen by comparing Figures 3b and 3c.

Example 5.6 (Allen–Cahn equation). The Allen–Cahn equation is a reaction-diffusion equation that describes phase transition. The Allen–Cahn equation is obtained by setting $G(u) = u^3 - u$, $\nu(u) = \bar{\nu}$, and $F(u) = 0$ in (4.1). The solution of the Allen–Cahn equation should stay within the range $[0, 1]$. Hence, we apply both the positivity and the maximum preserving limiters, respectively, (4.7)–(4.8) with bounds $u_{h\min} = 10^{-14}$ and $u_{h\max} = 1 - 10^{-10}$. A constrained projection of $u(x, 0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x, 0)$.

Example 5.6a (1D Allen–Cahn equation). As the test case we use the traveling



(a) (b)

FIG. 2. *Example* 5.4, 2*D advection equation:* (a) *solution* u_h, (b) *Lagrange multiplier. Computational mesh* 30×30 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

wave solution

$$u(x,t) = \frac{1}{2}\left(1 - \tanh\left(\frac{x - st}{2\sqrt{2\bar{\nu}}}\right)\right),$$

with wave velocity $s = 3\sqrt{\bar{\nu}/2}$. The computational domain is $\Omega = [-\frac{1}{2}, 2]$. If the mesh resolution is sufficiently dense such that the jump in the traveling wave solution is well resolved, then no limiter is required. For small values of the viscosity, the solution will, however, violate the positivity constraints, except on very fine meshes. In Figures 4a and 4b, respectively, the numerical solution u_h and its derivative Q_h and the exact solutions are shown for the viscosity $\bar{\nu} = 10^{-5}$ on a mesh with 100 elements and polynomial order 3 for the basis functions. The values of the Lagrange multiplier used to impose the positivity constraints are also shown in Figure 4a. The solution has a very thin and steep transition region, but the wave speed is still correctly computed by the LDG scheme and the KKT limiter ensures that both the positivity and the maximum constraints are satisfied.

Example 5.6b (2D Allen–Cahn equation). For the 2D test case, the computational domain is $\Omega = [-\frac{1}{2}, 2]^2$ and the computational mesh contains 30×30 elements. The viscosity coefficient is selected as $\bar{\nu} = 10^{-4}$. As the test case we use the initial solution

$$u(x,0) = \frac{1}{4}\left(1 - \tanh\left(\frac{x}{2\sqrt{2\bar{\nu}}}\right)\right)\left(1 - \tanh\left(\frac{y}{2\sqrt{2\bar{\nu}}}\right)\right),$$

whose values are also used as boundary conditions for $t > 0$. At this mesh resolution a positivity preserving limiter is necessary. The numerical solution shown in Figure 5a has steep gradients, and the positivity preserving limiter ensures that the bounds are satisfied. The locations where the limiter are active can be seen in Figure 5b, which shows the values and locations of the Lagrange multipliers used to impose the bounds in the DG discretization.

Example 5.7 (Barenblatt equation). The Barenblatt equation, which models a porous medium, is obtained by setting $\nu(u) = mu^{m-1}$, $m > 1$, and $F(u) = 0$,

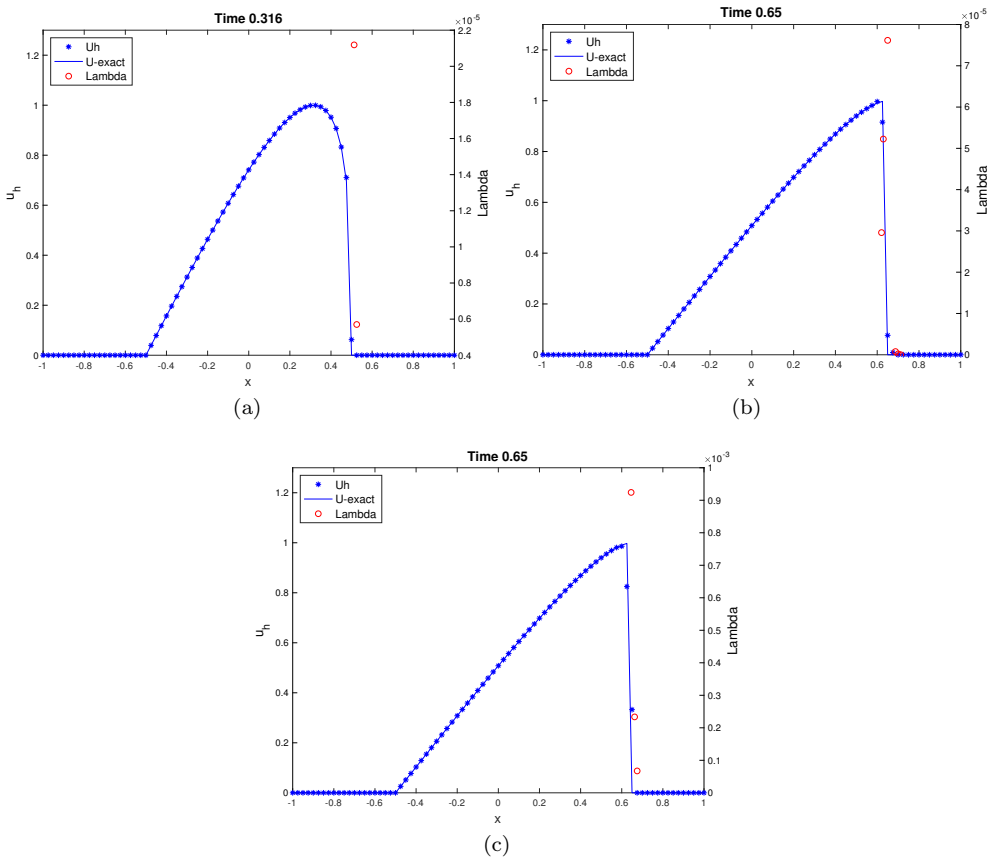


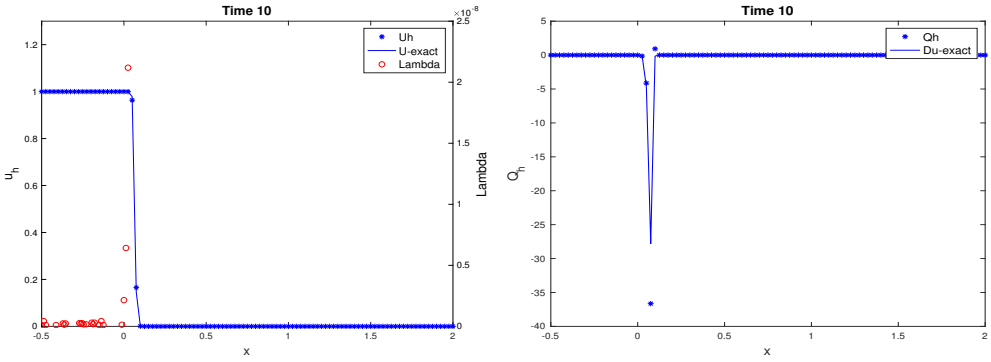
FIG. 3. *Example* 5.5, 1*D Burgers' equation:* (a)–(c) *solution* u_h *and Lagrange multiplier. The solution in* (a) *and* (b) *is computed with local conservation imposed as an explicit constraint, whereas* (c) *shows the solution without explicitly imposing local conservation. Computational mesh* 80 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated with a red (open) circle.*

$G(u) = 0$ in (4.1). The exact solution is

$$u(t,x) = t^{\alpha}\left(\left(C - \frac{\beta(m-1)}{2m}\frac{|x|^2}{t^{2\beta}}\right)_+\right)^{\frac{1}{m-1}},$$

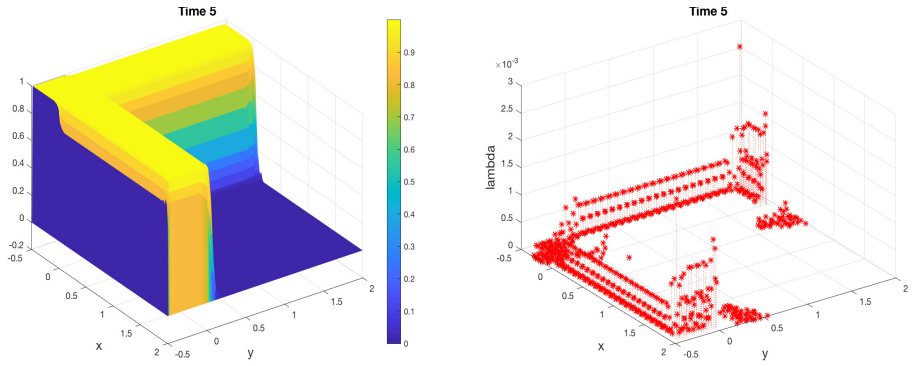
with $\alpha = \frac{n}{n(m-1)+2}$, $\beta = \frac{\alpha}{n}$, $n = \dim(\Omega)$, $(x)_+ = \max(x, 0)$, and $C > 0$. We selected $C = 1$ and $m = 8$. The solution should be positive or zero for $t > 0$. The initial solution for the computations is the constrained projection of $u(x, 1)$ onto the finite element space V_h^p. In the computations, Dirichlet boundary conditions are imposed, where the solution for $t > 0$ is fixed at the same level as the initial solution.

Example 5.7a (1D Barenblatt equation). We first consider the 1D Barenblatt equation on the domain $\Omega = [-7, 7]$ using a computational mesh of 100 elements. In Figure 6, the numerical solution without the use of a limiter is shown. It is clear that near the boundary of $u(t, x) > 0$, where the derivative of u becomes unbounded, significant negative values of u_h are obtained. These cause severe numerical problems



(a) u_h - with limiter (b) Q_h - with limiter

FIG. 4. *Example* 5.6a, *1D Allen–Cahn equation:* (a) *numerical solution* u_h *and exact solution* u, (b) *derivative of numerical solution* Q_h *and exact derivative* Du. *Computational mesh* 100 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated in* (a) *with a red (open) circle.*



(a) (b)

FIG. 5. *Example* 5.6b, *2D Allen–Cahn equation:* (a) *numerical solution* u_h *and* (b) *Lagrange multiplier. Computational mesh* 30×30 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

and do not allow the continuation of the computations.

Example 5.7b (2D Barenblatt equation). In Figures 7a and 7b, respectively, the numerical solution u_h of the 2D Barenblatt equation and the values of the Lagrange multiplier are shown at time $t = 2$ on a mesh of 50×50 elements. In these computations, the KKT-Limiter was used, which successfully prevents the numerical solution u_h from becoming negative, which is shown in Figure 7c. The imposed constraint is $u_{h\min} = 10^{-10}$. Figure 7c also shows an excellent agreement between the exact solution u and the numerical solution u_h.

Example 5.8 (1D Buckley–Leverett equation). The Buckley–Leverett equation models two phase flow in a porous medium. We consider two cases, respectively, with and without gravity. Since the solution has to be strictly inside the range $[0, 1]$, we use

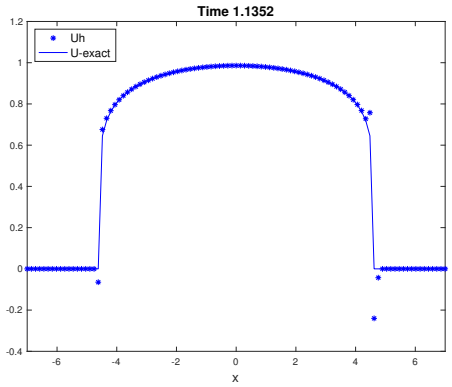


FIG. 6. *Example* 5.7a, *1D Barenblatt equation: numerical solution* u_h *without limiter and exact solution* u. *Computational mesh* 100 *elements, polynomial order* $p = 3$.

both the positivity and the maximum preserving limiters, with bounds $u_{h\min} = 10^{-10}$ and $u_{h\max} = 1 - 10^{-10}$, respectively. The computational domain is $\Omega = [0, 1]$. A Dirichlet boundary condition at $x = 0$, based on the initial solution, and an outflow boundary condition at $x = 1$ are imposed. The viscosity coefficient is $\bar{\nu} = 0.01$. Since we do not have an exact solution to compare with, we compute the numerical solution on two meshes, namely with 100 and 200 elements. The two test cases given by Examples 5.8a and 5.8b are also considered in [21].

Example 5.8a (1D Buckley–Leverett equation without gravity). The 1D Buckley–Leverett equation without gravity is obtained by setting $G(u) = 0$, and $\nu(u)$ and $F(u) = f(u)$, respectively, as

$$\nu(u) = \begin{cases} 4\bar{\nu}u(1-u) & \text{if } 0 \le u \le 1, \\ 0 & \text{otherwise;} \end{cases}$$

$$f(u) = \begin{cases} 0 & \text{if } u < 0, \\ \frac{u^2}{u^2+(1-u)^2} & \text{if } 0 \le u \le 1, \\ 1 & \text{if } u > 1. \end{cases} \tag{5.3}$$

The initial condition is

$$u(x, 0) = \begin{cases} 0.99 - 3x, & 0 \le x \le 0.33, \\ 0, & \frac{1}{3} < x \le 1. \end{cases}$$

The numerical solution u_h and its derivative Q_h are shown in, respectively, Figures 8a and 8b. Also, the values of the Lagrange multiplier used to enforce the constraints are shown in Figure 8a. The limiter is only active in the thin layer between the phases and is crucial to obtain sensible physical solutions. The results of 100 and 200 elements match well.

Example 5.8b (1D Buckley–Leverett equation with gravity). A much more difficult test case is provided by the Buckley–Leverett equation with gravity, which is obtained

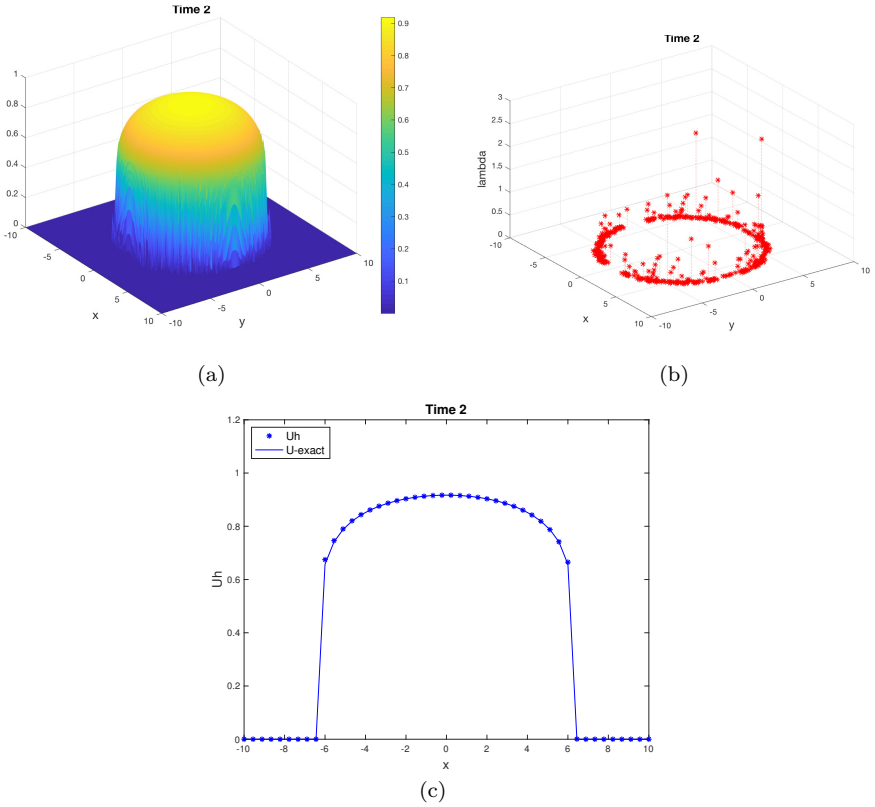


FIG. 7. *Example* 5.7b, *2D Barenblatt equation:* (a) *solution* u_h, (b) *Lagrange multiplier,* (c) *numerical solution* u_h *and exact solution* u *in cross-section at* $y = 0$. *Computational mesh* 50×50 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

by modifying the flux $F(u)$ as

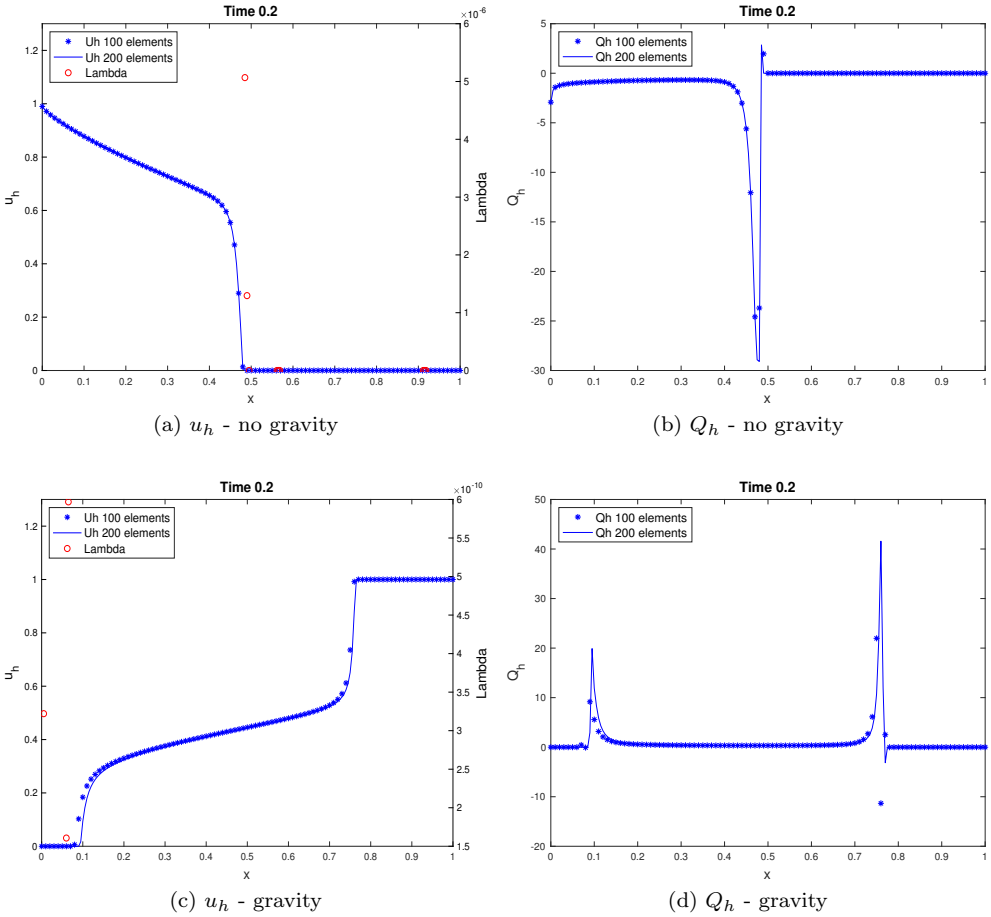
$$F(u) = \begin{cases} f(u)(1 - 5(1-u)^2), & u \le 1, \\ 1 & u > 1, \end{cases}$$

with $f(u)$ given by (5.3). The initial solution is

$$u(x,0) = \begin{cases} 0, & 0 \le x \le a, \\ \frac{1}{mh}(x-a), & a < x \le 1 - \frac{1}{\sqrt{2}}, \\ 1, & 1 - \frac{1}{\sqrt{2}} < x \le 1, \end{cases}$$

with $a = 1 - \frac{1}{\sqrt{2}} - mh$, h the mesh size, and $m = 3$. The linear transition for x in the range $[a, 1 - \frac{1}{\sqrt{2}}]$ is used to remove the infinite value in the derivative, which would otherwise result in unbounded values of Q_h at $t = 0$. The Buckley–Leverett equations with gravity result in a strongly nonlinear problem where the equations change type and are a severe test for the KKT-Limiter and semismooth Newton algorithm. The solution u_h and values of the Lagrange multiplier are shown in Figure 8c and the derivative Q_h in Figure 8d. The results on the two meshes compare well, and the

limiter ensures that the positivity and maximum bounds are satisfied.



(a) u_h - no gravity (b) Q_h - no gravity

(c) u_h - gravity (d) Q_h - gravity

FIG. 8. *Example* 5.8a, *2D Buckley–Leverett equation without gravity:* (a) *numerical solution* u_h, (b) *numerical solution derivative* Q_h. *Example* 5.8b, *1D Buckley–Leverett equation with gravity:* (c) *numerical solution* u_h, (d) *numerical solution derivative* Q_h. *Computational meshes* 100 *and* 200 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated with a red (open) circle in* (a) *and* (c).

The number of Newton iterations necessary to obtain a minimum value 10^{-8} for the residual $F(z)$ and Newton update d in Algorithm 3.1 to stop the Newton iterations for each DIRK stage strongly varies. It depends on the type of equation, time step, and nonlinearity. In general, the time step is chosen such that the number of Newton iterations for each DIRK stage is between 5 and 20. For most time-dependent problems, the CFL number is then close to one, which is necessary to ensure time accuracy. Only for the Buckley–Leverett equation with gravity did the time step frequently have to be less than one in order to deal with the strong nonlinearity of the problem. In the computations, we did not observe a minimum time step to ensure positivity, as noticed in [28].

6. Conclusions. In this paper, we present a novel framework to combine positivity preserving limiters for DG discretizations with implicit time integration methods.

This approach does not depend on the specific type of DG discretization and is also applicable to, e.g., finite volume discretizations. The key features of the numerical method are the formulation of the positivity constraints as a KKT-problem and the development of an active set semismooth Newton method that accounts for the nonsmoothness of the algebraic equations. The algorithm was successfully tested on a number of increasingly difficult test cases, which required that the positivity constraints are satisfied in order to obtain meaningful results. The KKT-Limiter does not negatively affect the accuracy for smooth problems and accurately preserves the positivity constraints. Future work will focus on the extension of the KKT-Limiter to ensure also monotonicity of the solution.

Appendix A. Derivation of Clarke directional derivative. For completeness, we give here a derivation of the terms (3.8d) and (3.8e) in the Clarke directional derivative of $F(z)$ in (2.2). We will follow the approach outlined in [17]. Define $z := (x, \mu, \lambda)$, $\bar{z} := (\bar{x}, \bar{\mu}, \bar{\lambda})$, $d := (u, v, w) \in \mathbb{R}^p$, with $p = n + l + m$. Consider $\bar{F}(z) = F_{i+n+l}(z)$, $i \in \beta(z)$. The other Clarke directional derivatives of F are straightforward to compute. If we consider (3.2) only for the contribution of $\bar{F}(z)$ to the merit function to $\theta(z)$ and use (2.2) and a Taylor expansion of $\bar{F}(z)$ around z, then we obtain

$$\begin{aligned}\bar{\theta}^0(z;d) &= \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(\bar{F}(z), \min(-g(\bar{x}+tu), \bar{\lambda}+tw) - \min(-g(\bar{x}), \bar{\lambda})\Big)\\ &= \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(\bar{F}(z), \min(-g(x) - J(\bar{x}+tu-x), \bar{\lambda}+tw)\\ &\qquad - \min\big(-g(x) - J(\bar{x}-x), \bar{\lambda}\big)\Big),\end{aligned}$$

with $J := D_x g(x) \in \mathbb{R}^{m\times n}$. Here higher order terms are omitted since they will become zero in the limit. Define $h(x) := -g(x) + Jx$; then

$$\begin{aligned}\bar{\theta}^0(z;d) = \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(&\bar{F}(z), \min(-J\bar{x} - tJu + h(x), \bar{\lambda}+tw)\\ &- \min(-J\bar{x} + h(x), \bar{\lambda})\Big).\end{aligned} \tag{A.1}$$

For $u \in \mathbb{R}^n, w \in \mathbb{R}^m$, define $r \in \mathbb{R}^m$ by

$$\begin{aligned}r_i < 0 \text{ on } S_1 :=& \{i \in \beta(z) \mid \bar{F}_i(z) > 0, -(Ju)_i > w_i\}\\ &\cup \{i \in \beta(z) \mid \bar{F}_i(z) \le 0, -(Ju)_i \le w_i\},\end{aligned} \tag{A.2a}$$

$$\begin{aligned}r_i > 0 \text{ on } S_2 :=& \{i \in \beta(z) \mid \bar{F}_i(z) > 0, -(Ju)_i \le w_i\}\\ &\cup \{i \in \beta(z) \mid \bar{F}_i(z) \le 0, -(Ju)_i > w_i\}.\end{aligned} \tag{A.2b}$$

Let $\bar{x} \in \mathbb{R}^n$ be such that

$$-J\bar{x} + h(x) = \bar{\lambda} + r. \tag{A.3}$$

Note that such an $\bar{x}$ exists for $i \in \beta(z)$ since (A.3) is equivalent to $-Ju = w + r$ with $u = \bar{x} - x$ and $w = \bar{\lambda} - \lambda$ as components of the search direction d. Choose $t \in (0, t_{\bar{x}})$ for $t_{\bar{x}} > 0$ such that

$$(-J\bar{x} + h(x) - tJu)_i < (\bar{\lambda} + tw)_i \quad \text{for } i \in S_1, \tag{A.4a}$$

$$(-J\bar{x} + h(x) - tJu)_i > (\bar{\lambda} + tw)_i \quad \text{for } i \in S_2. \tag{A.4b}$$

Note that such a $t_{\bar{x}}$ exists; see Remark A.1. We then obtain

$$\min((-J\bar{x}+h(x)-tJu)_i,(\bar{\lambda}+tw)_i)=\begin{cases}(-J\bar{x}+h(x)-tJu)_i & \text{for } i\in S_1,\\ (\bar{\lambda}+tw)_i & \text{for } i\in S_2.\end{cases}$$

Use now (A.3) and (A.2); then

$$\min((-J\bar{x}+h(x))_i,\bar{\lambda}_i)=\min(\bar{\lambda}_i+r_i,\bar{\lambda}_i)=\begin{cases}\bar{\lambda}_i+r_i & \text{for } i\in S_i,\\ \bar{\lambda}_i & \text{for } i\in S_2.\end{cases}$$

Combining the above results and using (A.3) again gives

$$\begin{aligned}\min((-J\bar{x}+h(x)-tJu)_i,(\bar{\lambda}+tw)_i)&-\min((-J\bar{x}+h(x))_i,\bar{\lambda}_i)\\ &=\begin{cases}-t(Ju)_i & \text{for } i\in S_1,\\ tw_i & \text{for } i\in S_2\end{cases}\\ &=\begin{cases}t\max(-(Ju)_i,w_i) & \text{if } \bar{F}_i(z)>0,\\ t\min(-(Ju)_i,w_i) & \text{if } \bar{F}_i(z)\le 0.\end{cases}\end{aligned}$$

Taking the limit in (A.1) and using (3.3) for $\bar{\theta}(z;d)$ then gives (3.8d) and (3.8e).

Remark A.1. Conditions (A.2) imply (A.4). Use $-J\bar{x}+h(x)=\bar{\lambda}+r$ in (A.4); then we obtain

$$(r-tJu)_i<tw_i \quad \text{for } i\in S_1, \tag{A.5}$$

$$(r-tJu)_i>tw_i \quad \text{for } i\in S_2. \tag{A.6}$$

I. If $i\in S_1$, $\bar{F}_i(z)>0$, then from (A.2a) we obtain $-(Ju)_i-w_i>0$ and (A.5) implies $r_i+t(-(Ju)_i-w_i)<0$. Choose $t<\frac{-r_i}{-(Ju)_i-w_i}=t_{\bar{x}}$. Since $r_i<0$ and $-(Ju)_i-w_i>0$ for $i\in S_1$, $\bar{F}_i(z)>0$, we obtain that $t_{\bar{x}}>0$.

II. If $i\in S_1$, $\bar{F}_i(z)\le 0$, then (A.2a) implies $-(Ju)_i-w_i\le 0$ and (A.5) gives $r_i+t(-(Ju)_i-w_i)<0$. Since both r_i and $-(Ju)_i-w_i<0$, any $t>0$ will imply (A.5).

The proof for $i\in S_2$ is completely analogous and is therefore omitted. Hence there exists a $t_{\bar{x}}>0$ for (A.4).

Appendix B. Verification of conditions for quasi-directional derivative. In this section, we show that the quasi-directional derivative (3.9) satisfies the conditions stated in (3.5), which are necessary to ensure convergence of the Newton algorithm defined in Algorithm 3.1.

Consider condition (3.5a): First note that

$$\begin{aligned}&F_i'(z;d)=F_i^0(z;d)=G_i(z;d), && i\in N_n,\\ &F_{i+n}'(z;d)=F_{i+n}^0(z;d)=G_{i+n}(z;d), && i\in N_l,\\ &F_{i+n+l}'(z;d)=F_{i+n+l}^0(z;d)=G_{i+n+l}(z;d), && i\in \alpha_\delta(z)\cup\gamma_\delta(z),\end{aligned}$$

since $\alpha_\delta(z)\cup\gamma_\delta(z)\subset\alpha(z)\cup\gamma(z)$. If $i\in\beta_\delta(z)$ and $F_{i+n+l}(z)\le 0$, then

$$\min(-(Ju)_i,w_i)\le -(Ju)_i,w_i.$$

Since $F_{i+n+l}(z) \leq 0$, this implies

$$F_{i+n+l}(z)\min(-(Ju)_i, w_i) \geq F_{i+n+l}(z)(-(Ju)_i), F_{i+n+l}(z)w_i.$$

If $i \in \beta_\delta(x)$ and $F_{i+n+l}(z) > 0$, then

$$-(Ju)_i, w_i \leq \max(-(Ju)_i, w_i).$$

Hence, since $F_{i+n+l}(z) > 0$, this implies

$$F_{i+n+l}(z)(-(Ju)_i), F_{i+n+l}(z)w_i \leq F_{i+n+l}(z)\max(-(Ju)_i, w_i).$$

Comparing all terms then immediately shows that $G(z; d)$ satisfies (3.5a) and (3.5c). Condition (3.5b) directly follows from the definition of G in (3.5).

Acknowledgment. We would like to acknowledge Mrs. Fengna Yan from USTC and the University of Twente for her contributions in testing the KKT-Limiter for several DG discretizations.

REFERENCES

[1] R. Alexander, *Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s*, SIAM J. Numer. Anal., 14 (1977), pp. 1006–1021, https://doi.org/10.1137/0714068.

[2] P. R. Amestoy, I. S. Duff, D. Ruiz, and B. Uçar, *A parallel matrix scaling algorithm*, in International Conference on High Performance Computing for Computational Science - VECPAR 2008, Springer, Berlin, Heidelberg, 2008, pp. 301–313.

[3] P. Bochev and D. Ridzal, *Optimization-based additive decomposition of weakly coercive problems with applications*, Comput. Math. Appl., 71 (2016), pp. 2140–2154, https://doi.org/10.1016/j.camwa.2015.12.032.

[4] J.-S. Chen, S. Pan, and T.-C. Lin, *A smoothing Newton method based on the generalized Fischer–Burmeister function for MCPs*, Nonlinear Anal., 72 (2010), pp. 3739–3758, https://doi.org/10.1016/j.na.2010.01.012.

[5] Z. Chen, H. Huang, and J. Yan, *Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes*, J. Comput. Phys., 308 (2016), pp. 198–217, https://doi.org/10.1016/j.jcp.2015.12.039.

[6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990, https://doi.org/10.1137/1.9781611971309.

[7] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463, https://doi.org/10.1137/S0036142997316712.

[8] M. D'Elia, M. Perego, P. Bochev, and D. Littlewood, *A coupling strategy for nonlocal and local diffusion models with mixed volume constraints and boundary conditions*, Comput. Math. Appl., 71 (2016), pp. 2218–2230, https://doi.org/10.1016/j.camwa.2015.12.006.

[9] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer, Heidelberg, 2011.

[10] J. A. Evans, T. J. Hughes, and G. Sangalli, *Enforcement of constraints and maximum principles in the variational multiscale method*, Comput. Methods Appl. Mech. Engrg., 199 (2009), pp. 61–76, https://doi.org/10.1016/j.cma.2009.09.019.

[11] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer Science & Business Media, New York, 2007.

[12] H. Guo and Y. Yang, *Bound-preserving discontinuous Galerkin method for compressible miscible displacement in porous media*, SIAM J. Sci. Comput., 39 (2017), pp. A1969–A1990, https://doi.org/10.1137/16M1101313.

[13] L. Guo and Y. Yang, *Positivity preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions*, J. Comput. Phys., 289 (2015), pp. 181–195, https://doi.org/10.1016/j.jcp.2015.02.041.

[14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II. *Stiff and Differential-Algebraic Problem*, Springer, Berlin, 2010.

[15] S.-P. Han, J.-S. Pang, and N. Rangaraj, *Globally convergent Newton methods for nonsmooth equations*, Math. Oper. Res., 17 (1992), pp. 586–607.
[16] P. T. Harker and J.-S. Pang, *A damped-Newton method for the linear complementarity problem*, in Computational Solution of Nonlinear Systems of Equations, Lectures in Appl. Math. 26, AMS, Providence, RI, 1990, pp. 265–284.
[17] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, SIAM, Philadelphia, 2008, https://doi.org/10.1137/1.9780898718614.
[18] K. Ito and K. Kunisch, *On a semi-smooth Newton method and its globalization*, Math. Program., 118 (2009), pp. 347–370, https://doi.org/10.1007/s10107-007-0196-3.
[19] G. Karniadakis and S. Sherwin, *Spectral/HP Element Methods for Computational Fluid Dynamics*, Oxford University Press, New York, 2013.
[20] P. Kuberry, P. Bochev, and K. Peterson, *An optimization-based approach for elliptic problems with interfaces*, SIAM J. Sci. Comput., 39 (2017), pp. S757–S781, https://doi.org/10.1137/16M1084547.
[21] A. Kurganov and E. Tadmor, *New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282, https://doi.org/10.1006/jcph.2000.6459.
[22] A. Meister and S. Ortleb, *On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows*, Internat. J. Numer. Methods Fluids, 76 (2014), pp. 69–94, https://doi.org/10.1002/fld.3921.
[23] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, and C. Kanzow, *The semismooth algorithm for large scale complementarity problems*, INFORMS J. Comput., 13 (2001), pp. 294–311, https://doi.org/10.1287/ijoc.13.4.294.9734.
[24] J.-S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.
[25] J.-S. Pang, *A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming, 51 (1991), pp. 101–131.
[26] S. Patankar, *Numerical Heat Transfer and Fluid Flow*, CRC Press, Boca Raton, FL, 1980.
[27] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.
[28] T. Qin and C.-W. Shu, *Implicit positivity-preserving high-order discontinuous Galerkin methods for conservation laws*, SIAM J. Sci. Comput., 40 (2018), pp. A81–A107, https://doi.org/10.1137/17M112436X.
[29] T. Qin, C.-W. Shu, and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, J. Comput. Phys., 315 (2016), pp. 323–347, https://doi.org/10.1016/j.jcp.2016.02.079.
[30] A. Shapiro, *On concepts of directional differentiability*, J. Optim. Theory Appl., 66 (1990), pp. 477–487.
[31] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[32] L. Skvortsov, *Diagonally implicit Runge-Kutta methods for stiff problems*, Comput. Math. Math. Phys., 46 (2006), pp. 2110–2123, https://doi.org/10.1134/S0965542506120098.
[33] X. Zhang, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations*, J. Comput. Phys., 328 (2017), pp. 301–343, https://doi.org/10.1016/j.jcp.2016.10.002.
[34] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, J. Comput. Phys., 229 (2010), pp. 3091–3120, https://doi.org/10.1016/j.jcp.2009.12.030.
[35] X. Zhang and C.-W. Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, J. Comput. Phys., 229 (2010), pp. 8918–8934, https://doi.org/10.1016/j.jcp.2010.08.016.
[36] X. Zhang and C.-W. Shu, *Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms*, J. Comput. Phys., 230 (2011), pp. 1238–1248, https://doi.org/10.1016/j.jcp.2010.10.036.
[37] X. Zhang, Y. Xia, and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes*, J. Sci. Comput., 50 (2012), pp. 29–62, https://doi.org/10.1007/s10915-011-9472-8.
[38] Y. Zhang, X. Zhang, and C.-W. Shu, *Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection–diffusion equations on triangular meshes*, J. Comput. Phys., 234 (2013), pp. 295–316, https://doi.org/10.1016/j.jcp.2012.09.032.