

Discontinuous Galerkin methods for convection-dominated problems

Bernardo Cockburn¹

School of Mathematics, University of Minnesota,
206 Church Street S.E., Minneapolis, MN 55455, USA

1 Introduction

1.1 The purpose of these notes

In these notes, we study the Runge Kutta Discontinuous Galerkin method for numerically 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; applications to Hamilton-Jacobi equations is another important example. 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. Most of the material of these notes has been presented in [17].

Acknowledgements The author would like to thank T.J. Barth for the invitation to give a series of lectures in the NATO special course on ‘Higher Order Discretization Methods in Computational Fluid Dynamics,’ the material of which is contained in these notes. He would also like to thank F. Bassi and S. Rebay, and I. Lomtev and G. Karniadakis for kindly supplying several of their figures, and to C. Hu and Chi-Wang Shu for providing their material on Hamilton-Jacobi equations. Thanks are also due to Rosario Grau for fruitful discussions concerning the numerical experiments of Chapter 2, to J.X. Yang for a careful proof-reading the appendix of Chapter 6, and to A. Zhou for bringing the author’s attention to several of his papers concerning the discontinuous Galerkin method.

1.2 A historical overview

The original Discontinuous Galerkin method The original discontinuous Galerkin (DG) finite element method was introduced by Reed and Hill [76] for solving the neutron transport equation

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

where σ is a real number and \bar{a} a constant vector. A remarkable advantage of this method is that, because of the linear nature of the equation, the approximate solution can be computed element by element when the elements are suitably ordered according to the characteristic direction.

LeSaint and Raviart [58] 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 Pitkaranta [52] proved a rate of convergence of $(\Delta x)^{k+1/2}$ for general triangulations and Peterson [75] numerically confirmed this rate to be optimal. Richter [77] obtained the optimal rate of convergence of $(\Delta x)^{k+1}$ for some structured two-dimensional non-Cartesian grids. In all the above papers, the exact solution is assumed to be very smooth. The case in which the solution admits discontinuities was treated by Lin and Zhou [60] who proved the convergence of the method. The issue of the interrelation between the mesh and the order of convergence of the method was explored by Zhou and Lin [93], case $k = 1$, and later by Lin, Yan, and Zhou [59], case $k = 0$, and optimal error estimates were proven under suitable assumptions on the mesh. Recently, several new results have been obtained. Thus, Falk and Richter [39] obtained a rate of convergence of $(\Delta x)^{k+1/2}$ for general triangulations for Friedrich systems; Houston, Schwab and Süli [42] analyzed the hp version of the discontinuous Galerkin method and showed its exponential convergence when the solution is piecewise analytic; and, finally, Cockburn, Luskin, Shu, and Süli [22] showed how to exploit the translation invariance of a grid to *double* the order of convergence of the method by a simple, local postprocessing of the approximate solution.

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 [13] constructed 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 [12] modified the scheme by introducing a suitably defined ‘slope limiter’ following the ideas introduced by van Leer in [88]. 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 [12] indicate convergence to the correct entropy solutions. However, 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 [26], where the first Runge Kutta Discontinuous Galerkin (RKDG) method was introduced. This method was constructed (i) by retaining the piecewise linear DG method for the space discretization, (ii) by using a special explicit TVD second order Runge-Kutta type discretization introduced by Shu and Osher in a finite difference framework [80], [81], and (iii) by modifying the ‘slope limiter’ to maintain the formal accuracy of the scheme 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 [26] indicate good convergence behavior: Second order in smooth regions including extrema, sharp shock transitions (usually in one or two elements) without oscillations, and convergence to entropy solutions even for non convex fluxes.

In [24], Cockburn and Shu extended this approach to construct (formally) high-order accurate RKDG methods for the scalar conservation law. To devise 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 [24], 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 [21].

- **The multidimensional case.**

The extension of the RKDG method to the multidimensional case was done in [20] 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 barriers 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 multi-dimensional 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 so in several space dimensions since Goodman and LeVeque [41] 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 [20], Cockburn, Hou and Shu devised a generalized ‘slope limiter’ that enforced a **local** maximum principle only since they are not incompatible with high-order accuracy. No other class of schemes has a proven maximum principle for general nonlinearities \mathbf{f} and arbitrary triangulations.

The extension of the RKDG methods to general multidimensional systems was started by Cockburn and Shu in [25] and has been recently completed in [28]. Bey and Oden [10], Bassi and Rebay [4], and more recently Baumann [6] and Baumann and Oden [9] have studied applications of the method to the Euler equations of gas dynamics. Recently, Kershaw *et al.* [56], from the Lawrence Livermore National Laboratory, extended the method to arbitrary Lagrangian-Eulerian fluid flows where the computational mesh can move to track the interface between the different material species.

- **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 [44], [49], [45], [46], [47], [48] (which has been analyzed by Johnson *et al.* in [53], [54], [55]), 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 [90], Lindquist [62], [61], Tong and Zheng [86], and Tong and Chen [85]. 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 have been performed by Biswas, Devine, and Flaherty [11], Devine, Flaherty, Loy, and Wheat [32], Devine and Flaherty [31], and more recently by Flaherty *et al.* [40]. Studies of load balancing related to conservation laws but not restricted to them can be found in the works by Devine, Flaherty, Wheat, and Maccabe [33], by deCougny *et al.* [30], and by Özturan *et al.* [74].

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.* [15], [14] 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 [15] and [14] were ‘lumped’ in order to maintain the high parallelizability of the method. Since in [15] and [14] 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.* [4]) 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 [27] 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 [51]. Moreover, error estimates (in the linear case) have been obtained in [27]. A recent analysis of this method is currently being carried out by Cockburn and Schwab [23] in the one dimensional case by taking into account the characterization of the viscous boundary layer of the exact solution.

The LDG methods [27] are very different from the so-called Discontinuous Galerkin (DG) method for parabolic problems introduced by Jamet [50] and studied by Eriksson, Johnson, and Thomée [38], Eriksson and Johnson [34], [35], [36], [37], and more recently by Makridakis and Babuška [68]. 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 [64], [65], [63], [66], and [91].

Another technique to discretize the diffusion terms have been proposed by Baumann [6]. The one-dimensional case was studied by Babuška, Baumann, and J.T. Oden [2] and the multidimensional case has been considered by Oden, Babuška, and Baumann [70]. The case of convection-diffusion in multidimensions was treated by Baumann and Oden in [7]. In [8], Baumann and Oden consider applications to the Navier-Stokes equations.

Finally, let us point bring the attention of the reader to the non-conforming staggered-grid Chebyshev spectral multidomain numerical method for the solution of the compressible Navier-Stokes equations proposed and studied by Kopriva [57]; this method is strongly related to discontinuous Galerkin methods.

1.3 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 [26], [24], [21], [20], and [28] in which the RKDG method was developed and on the paper by Cockburn and Shu [27] which is devoted to the LDG methods. We also include numerical results from the papers by Bassi and Rebay [4] and by Warburton, Lomtev, Kirby and Karniadakis [91] on the Euler equations of gas dynamics and from the papers by Bassi and Rebay [3] and by Lomtev and Karniadakis [63] on the compressible Navier-Stokes equations. Finally, we also use the material contained in the paper by Hu and Shu [43] in which the application of the RKDG method is extended to Hamilton-Jacobi equations.

The emphasis in these notes is on *how the above mentioned schemes were devised*. As a consequence, the chapters that follow reflect that development. Thus, Chapter 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 devising of the RKDG methods; chapter 3 contains its extension to one-dimensional Hamilton-Jacobi equations. In chapter 4, we extend the RKDG method to multidimensional systems and in Chapter 5, to multidimensional Hamilton-Jacobi equations. Finally, in chapter 6 we study the extension to convection-diffusion problems.

We would like to emphasize that the guiding principle in the devising 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 [11], Devine, Flaherty, Loy, and Wheat [32], Devine and Flaherty [31], and by Flaherty *et al.* [40] (see also the works by Devine, Flaherty, Whea, and Maccabe [33], by deCougny *et al.* [30], and by Özturan *et al.* [74]) are certainly very important. Another issue of importance is how to render the method computationally 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.

2 The scalar conservation law in one space dimension

2.1 Introduction

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

$$u_t + f(u)_x = 0, \quad \text{in } (0, 1) \times (0, T), \quad (2.1)$$

$$u(x, 0) = u_0(x), \quad \forall x \in (0, 1), \quad (2.2)$$

and periodic boundary conditions. This section has material drawn from [26] and [24].

2.2 The discontinuous Galerkin-space discretization

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 $j = 1, \dots, N$, and denote the quantity $\max_{1 \leq j \leq N} \Delta_j$ 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

$$V_h = V_h^k \equiv \{v \in L^1(0, 1) : v|_{I_j} \in P^k(I_j), j = 1, \dots, N\}, \quad (2.3)$$

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 (2.1) and (2.2) by arbitrary, smooth functions v and integrate over I_j , and get, after a simple formal integration by parts,

$$\begin{aligned} & \int_{I_j} \partial_t u(x, t) v(x) dx - \int_{I_j} f(u(x, t)) \partial_x v(x) dx \\ & + f(u(x_{j+1/2}, t)) v(x_{j+1/2}^-) - f(u(x_{j-1/2}, t)) v(x_{j-1/2}^+) = 0, \end{aligned} \quad (2.4)$$

$$\int_{I_j} u(x, 0) v(x) dx = \int_{I_j} u_0(x) v(x) dx. \quad (2.5)$$

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

$$h(u)_{j+1/2}(t) = h(u(x_{j+1/2}^-, t), u(x_{j+1/2}^+, t)), \quad (2.6)$$

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:

$$\begin{aligned} \forall j = 1, \dots, N, \quad \forall v_h \in P^k(I_j) : \\ \int_{I_j} \partial_t u_h(x, t) v_h(x) dx - \int_{I_j} f(u_h(x, t)) \partial_x v_h(x) dx \\ + h(u_h)_{j+1/2}(t) v_h(x_{j+1/2}^-) - h(u_h)_{j-1/2}(t) v_h(x_{j-1/2}^+) = 0, \quad (2.7) \\ \int_{I_j} u_h(x, 0) v_h(x) dx = \int_{I_j} u_0(x) v_h(x) dx. \quad (2.8) \end{aligned}$$

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*. The idea is that by *perturbing* the monotone schemes, we would achieve high-order accuracy while keeping their stability and convergence properties. Thus, 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

$$u_h(x, t) = u_j^0,$$

we can rewrite our weak formulation (2.7), (2.8) as follows:

$$\begin{aligned} \forall j = 1, \dots, N : \\ \partial_t u_j^0(t) + \{h(u_j^0(t), u_{j+1}^0(t)) - h(u_{j-1}^0(t), u_j^0(t))\} / \Delta_j = 0, \\ u_j^0(0) = \frac{1}{\Delta_j} \int_{I_j} u_0(x) dx, \end{aligned}$$

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:

$$h^G(a, b) = \begin{cases} \min_{a \leq u \leq b} f(u), & \text{if } a \leq b \\ \max_{b \leq u \leq a} f(u), & \text{otherwise.} \end{cases}$$

(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:

$$h^{LF}(a, b) = \frac{1}{2} [f(a) + f(b) - C(b - a)],$$

$$C = \max_{\inf u^0(x) \leq s \leq \sup u^0(x)} |f'(s)|;$$

(iv) The local Lax-Friedrichs flux:

$$h^{LLF}(a, b) = \frac{1}{2} [f(a) + f(b) - C(b - a)],$$

$$C = \max_{\min(a, b) \leq s \leq \max(a, b)} |f'(s)|;$$

(v) The Roe flux with ‘entropy fix’:

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

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.

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

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

to 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 (2.7), (2.8) takes the following simple form:

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

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

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

This shows that after discretizing in space the problem (2.1), (2.2) by the DG method, we obtain a system of ODEs for the degrees of freedom that we can rewrite as follows:

$$\frac{d}{dt} u_h = L_h(u_h), \quad \text{in } (0, T), \quad (2.9)$$

$$u_h(t = 0) = u_{0h}. \quad (2.10)$$

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.

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 (2.7), (2.8) 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

$$h(a, b) = c \frac{a + b}{2} - \frac{|c|}{2} (b - a). \quad (2.11)$$

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 1. (L²-stability) *We have,*

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

where

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

Note how the jumps of u_h are controlled by the L²-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 accuracy equal to $k + 1/2$ for the same smoothness of the initial condition.

Theorem 2. (First L²-error estimate) *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. (Second L²-error estimate) *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 .

Theorem 2 is a simplified version of a more general result proven in 1986 by Johnson and Pitkäranta [52] and Theorem 3 is a simplified version of a more general result proven in 1974 by LeSaint and Raviart [58]. To provide a simple introduction to the techniques used in these general results, we give *new* proofs of Theorems 2 and 3 in an appendix to this chapter.

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.

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:

$$w_t + f(w)_x = (\nu(w) w_x)_x, \quad (2.12)$$

$$w(\cdot, 0) = u_0(\cdot), \quad (2.13)$$

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 (2.12) by $U'(w-c)$, where $U(\cdot)$ denotes the absolute value function and c denotes an arbitrary real number. Since

$$\begin{aligned} U'(w-c) w_t &= U(w-c)_t, \\ U'(w-c) f(w)_x &= (U'(w-c) (f(w) - f(c)))_x \\ &\equiv F(w, c)_x, \end{aligned}$$

and since

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

we obtain

$$U(w-c)_t + F(w, c)_x - \Phi(w, c)_x \leq 0,$$

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 (2.12) by $-(U'(w_x))_x$ and integrate on x from 0 to 1. Since

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

and since

$$\begin{aligned} \int_0^1 -(U'(w_x))_x (\nu(w) w_x)_x &= - \int_0^1 U''(w_x) w_{xx} (\nu'(w) (w_x)^2 + \nu(w) w_{xx}) \\ &= - \int_0^1 U''(w_x) \nu(w) (w_{xx})^2 \\ &\leq 0, \end{aligned}$$

we immediately get that

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

and so,

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

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 4. (L¹-error estimate) *We have*

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

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:

$$\begin{aligned} \forall j = 1, \dots, N : \\ \partial_t u_j + \{h(u_j, u_{j+1}) - h(u_{j-1}, u_j)\} / \Delta_j &= 0, \\ u_j(0) &= \frac{1}{\Delta_j} \int_{I_j} u_0(x) dx, \end{aligned}$$

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:

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

Using the simple identity

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

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

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

where

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

We thus get

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

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

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

and we obtain our entropy inequality:

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

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

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

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. This is indeed the case since the expression

$$v_j^0 \{h(u_j, u_{j+1}) - h(u_{j-1}, u_j)\}$$

is equal to the quantity

$$v_j^0 \{f^+(u_j) - f^+(u_{j-1})\} + v_j^0 \{f^-(u_{j+1}) - f^-(u_j)\},$$

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

$$|u_h(t)|_{TV(0,1)} \leq |u_h(0)|_{TV(0,1)} \leq |u_0|_{TV(0,1)}. \quad (2.14)$$

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

Theorem 5. (L¹-error estimate) *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. Error estimates for the case of arbitrary k have not been obtained, yet. However, Jiang and Shu [51] found a very interesting result in the case in which the nonlinear flux f is strictly convex or concave. In such a situation, the existence of a discrete, local entropy inequality for the scheme for only a *single* entropy is enough to guarantee that the limit of the scheme, if it exists, is the entropy solution. Jiang and Shu [51] found such a discrete, local entropy inequality for the DG-method.

To describe the main idea of their result, let us first consider the model equation

$$u_t + f(u)_x = (\nu u_x)_x.$$

If we multiply the equation by u we obtain, after very simple manipulations,

$$\frac{1}{2} (u)_t^2 + (F(u) - \frac{\nu}{2} (u)_x^2)_x + \Theta = 0,$$

where

$$F(u) = u f(u) - \int^u f(s) ds,$$

and

$$\Theta = \nu (u_x)^2.$$

Since $\Theta \geq 0$, we immediately obtain the following entropy inequality:

$$\frac{1}{2} (u)_t^2 + (F(u) - \frac{\nu}{2} (u)_x^2)_x \leq 0,$$

Now, we only need to mimic the above procedure using the numerical scheme (2.7) instead of the above parabolic equation and obtain a discrete version of the above entropy inequality. To do that, we simply take $v_h = u_h$ in (2.7) and rearrange terms in a suitable way. If we use the following notation:

$$\begin{aligned} \bar{u}_{j+1/2} &= (u_{j+1/2}^+ + u_{j+1/2}^-)/2, \\ [u]_{j+1/2} &= (u_{j+1/2}^+ - u_{j+1/2}^-), \end{aligned}$$

the result can be expressed as follows.

Proposition 6. *We have, for $j = 1, \dots, N$,*

$$\frac{1}{2} \frac{d}{dt} \int_{I_j} u_h^2(x, \cdot) dx + \hat{F}_{j+1/2} - \hat{F}_{j-1/2} + \Theta_j = 0,$$

where

$$\hat{F}_{j+1/2} = \bar{u}_{j+1/2} h(u_h)_{j+1/2} - \int_{\bar{u}_{j-1/2}}^{\bar{u}_{j+1/2}} f(s) ds,$$

and

$$\Theta_j = \int_{u_{j+1/2}^-}^{\bar{u}_{j+1/2}} (f(s) - h(u_h)_{j+1/2}) ds + \int_{\bar{u}_{j-1/2}}^{u_{j-1/2}^+} (f(s) - h(u_h)_{j-1/2}) ds.$$

Since the quantity Θ_j is nonnegative (because the numerical flux is non-decreasing in its first argument and nonincreasing in its second argument), we immediately obtain the following discrete, local entropy inequality:

$$\frac{1}{2} \frac{d}{dt} \int_{I_j} u_h^2(x, \cdot) dx + \hat{F}_{j+1/2} - \hat{F}_{j-1/2} \leq 0.$$

As a consequence, we have the following result.

Theorem 7. *Let f be a strictly convex or concave function. Then, for any $k \geq 0$, if the numerical solution given by the DG method converges, it converges to the entropy solution.*

There is no other formally high-order accurate numerical scheme that has the above property. See Jiang and Shu [51] for further developments of the above result.

2.3 The TVD-Runge-Kutta time discretization

To discretize our ODE system in time, we use the TVD Runge Kutta time discretization introduced in [83]; see also [80] and [81].

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

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

Runge-Kutta discretization parameters			
order	α_{il}	β_{il}	$\max\{\beta_{il}/\alpha_{il}\}$
2	$\begin{matrix} 1 \\ \frac{1}{2} & \frac{1}{2} \end{matrix}$	$\begin{matrix} 1 \\ 0 & \frac{1}{2} \end{matrix}$	1
3	$\begin{matrix} 1 \\ \frac{3}{4} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{matrix}$	$\begin{matrix} 1 \\ 0 & \frac{1}{4} \\ 0 & 0 & \frac{2}{3} \end{matrix}$	1

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

$$|w| \leq |v|, \tag{2.15}$$

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

$$w = v + \delta L_h(v), \tag{2.16}$$

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

For example, the second-order Runge-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 (2.15), (2.16) is satisfied for

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

we have

$$|u_h^{(1)}| \leq |u_h^n|, \quad |w_h| \leq |u_h^{(1)}|,$$

and so,

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

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 [81].

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

Theorem 8. (Stability of the Runge-Kutta discretization) *Assume that the stability property for the single ‘Euler forward’ step (2.15), (2.16) is satisfied for*

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

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

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

Then

$$|u_h^n| \leq |u_h^0|, \quad \forall n \geq 0.$$

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 8. We start by rewriting our time discretization as follows:

- Set $u_h^0 = u_{0h}$;
- For $n = 0, \dots, N-1$ compute u_h^{n+1} from u_h^n as follows:
 1. set $u_h^{(0)} = u_h^n$;

2. for $i = 1, \dots, k + 1$ 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

$$\begin{aligned} |u_h^{(i)}| &\leq \sum_{l=0}^{i-1} \alpha_{il} |w_h^{(il)}|, \quad \text{since } \alpha_{il} \geq 0, \\ &\leq \sum_{l=0}^{i-1} \alpha_{il} |u_h^{(l)}|, \end{aligned}$$

by the stability property (2.15), (2.16), and finally,

$$|u_h^{(i)}| \leq \max_{0 < l < i-1} |u_h^{(l)}|,$$

since

$$\sum_{l=0}^{i-1} \alpha_{il} = 1.$$

It is clear now that that Theorem 8 follows from the above inequality by a simple induction argument. This concludes the proof.

Remarks about the stability in the linear case For the linear case $f(u) = cu$, Chavent and Cockburn [12] 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 [26] 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 8 and we must consider the complete method at once.

When polynomial of degree k are used, a Runge-Kutta of order $(k + 1)$ must be used. If this is the case, for $k = 2$, the $L^\infty(0, T; L^2(0, 1))$ -stability condition can be proven to be the following:

$$c \frac{\Delta t}{\Delta x} \leq \frac{1}{5}.$$

The stability condition for a general value of k is still not known.

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.

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,

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

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 [18] for details.

Theorem 9. (Discrete entropy inequality) *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

$$\max_{1 \leq j \leq N} \frac{\Delta t}{\Delta_j} |f'| \leq 1,$$

then $\Theta_{diss,j}^n \geq 0$, and the following entropy inequality holds:

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

Note that $\Theta_{diss,j}^n \geq 0$ 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 10. (TVD property) *We have*

$$|u_h^{n+1}|_{TV(0,1)} - |u_h^n|_{TV(0,1)} + \Theta_{TV}^n = 0,$$

where

$$\begin{aligned}\Theta_{TV}^n &= \sum_{1 \leq j \leq N} \left(U_{j+1/2}^m - U_{j+1/2}^{m+1} \right) (p_{j+1/2}(u_{j+1}^n) - p_{j+1/2}(u_j^n)) \\ &+ \sum_{1 \leq j \leq N} \frac{\Delta t}{\Delta_j} \left(U_{j-1/2}^m - U_{j+1/2}^{m+1} \right) (f^+(u_j^n) - f^+(u_{j-1}^n)) \\ &- \sum_{1 \leq j \leq N} \frac{\Delta t}{\Delta_j} \left(U_{j+1/2}^m - U_{j-1/2}^{m+1} \right) (f^-(u_{j+1}^n) - f^-(u_j^n))\end{aligned}$$

where

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

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

$$\max_{1 \leq j \leq N} \frac{\Delta t}{\Delta_j} |f'| \leq 1,$$

then $\Theta_{TV}^n \geq 0$, and we have

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

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

Theorem 11. (L¹-error estimate for monotone schemes) *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 \bar{u}_j the mean of u_h on the interval I_j , by setting $v_h = 1$ in the equation (2.7), we obtain,

$$\begin{aligned} \forall j = 1, \dots, N : \\ (\bar{u}_j)_t + \{h(u_{j+1/2}^-, u_{j+1/2}^+) - h(u_{j-1/2}^-, u_{j-1/2}^+)\} / \Delta_j = 0, \end{aligned}$$

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

$$\begin{aligned} \forall j = 1, \dots, N : \\ (\bar{w}_j - \bar{u}_j) / \delta + \{h(u_{j+1/2}^-, u_{j+1/2}^+) - h(u_{j-1/2}^-, u_{j-1/2}^+)\} / \Delta_j = 0. \end{aligned}$$

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

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

Theorem 12. (The TVDM property) *We have*

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

where

$$\begin{aligned} \Theta_{TVM} &= \sum_{1 \leq j \leq N} \left(U'(u)_{j+1/2} - U'(w)_{j+1/2} \right) (p_{j+1/2}(u_h|_{I_{j+1}}) - p_{j+1/2}(u_h|_{I_j})) \\ &+ \sum_{1 \leq j \leq N} \frac{\delta}{\Delta_j} \left(U'(u)_{j-1/2} - U'(w)_{j+1/2} \right) (f^+(u_{j+1/2}^-) - f^+(u_{j-1/2}^-)) \\ &- \sum_{1 \leq j \leq N} \frac{\delta}{\Delta_j} \left(U'(u)_{j+1/2} - U'(w)_{j-1/2} \right) (f^-(u_{j+1/2}^+) - f^-(u_{j-1/2}^+)), \end{aligned}$$

where

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

and

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

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

$$\text{sgn}(\bar{u}_{j+1} - \bar{u}_j) = \text{sgn}(p_{j+1/2}(u_h|_{I_{j+1}}) - p_{j+1/2}(u_h|_{I_j})), \quad (2.17)$$

$$\text{sgn}(\bar{u}_j - \bar{u}_{j-1}) = \text{sgn}(u_{j+1/2}^{n,-} - u_{j-1/2}^{n,-}), \quad (2.18)$$

$$\text{sgn}(\bar{u}_{j+1} - \bar{u}_j) = \text{sgn}(u_{j+1/2}^{n,+} - u_{j-1/2}^{n,+}). \quad (2.19)$$

Note that if the *sign* conditions (2.17) and (2.18) are satisfied, then the *sign* condition (2.19) 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, $\Lambda \Pi_h$.

2.4 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,

- Satisfies the *sign* properties (2.17), (2.18), and (2.19),
- 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 $u_h = \mathcal{M}I_h(v_h)$,

$$\bar{u}_j = \bar{v}_j, \quad j = 1, \dots, N.$$

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

The second requirement, states that if $u_h = \mathcal{M}I_h(v_h)$ and $w_h = u_h + \delta L_h(u_h)$ then

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

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 neighborhood 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 straight line, it trivially satisfies conditions (2.17) and (2.18). However, if u_h is a parabola, conditions (2.17) and (2.18) are not always satisfied. This shows that *it is impossible* to construct the above ideal generalized ‘slope 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!

Fortunately, 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 *sign* properties (2.17), (2.18), and (2.19). These conditions will be stated in terms of the *minmod* function m defined as follows:

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

Proposition 13. Sufficient conditions for the sign properties *Suppose the following CFL condition is satisfied:*

$$\text{For all } j = 1, \dots, N : \quad |\delta| \left(\frac{|f^+|_{Lip}}{\Delta_{j+1}} + \frac{|f^-|_{Lip}}{\Delta_j} \right) \leq 1/2. \quad (2.20)$$

Then, conditions (2.17), (2.18), and (2.19) are satisfied if, for all $j = 1, \dots, N$, we have that

$$\begin{aligned} u_{j+1/2}^- &= \bar{u}_j + m(u_{j+1/2}^- - \bar{u}_j, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j) \\ u_{j-1/2}^+ &= \bar{u}_j - m(\bar{u}_j - u_{j-1/2}^+, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j). \end{aligned} \quad (2.21)$$

Proof. Let us start by showing that the property (2.18) is satisfied. We have:

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

where

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

by conditions (2.21) and (2.22). This implies that the property (2.18) is satisfied. Properties (2.19) and (2.17) 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,

$$v_h|_{I_j} = \bar{v}_j + (x - x_j) v_{x,j}, \quad j = 1, \dots, N,$$

the following generalized slope limiter does satisfy the conditions (2.21) and (2.22):

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

This is the well-known slope limiter of the MUSCL schemes of van Leer [88,89].

b. The less restrictive limiter $\mathcal{M}I_h^1$. The following less restrictive slope limiter also satisfies the conditions (2.21) and (2.22):

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

Moreover, it can be rewritten as follows:

$$\begin{aligned} u_{j+1/2}^- &= \bar{v}_j + m(v_{j+1/2}^- - \bar{v}_j, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j) \\ u_{j-1/2}^+ &= \bar{v}_j - m(\bar{v}_j - v_{j-1/2}^+, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j). \end{aligned} \quad (2.22)$$

We denote this limiter by $\Lambda\Pi_h^1$.

Note that we have that

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

See Theorem 16 below.

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

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

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 to define what could be called the P^1 -part of v_h :

$$v_h^1(x, t) = \sum_{\ell=0}^1 v_j^\ell \varphi_\ell(x),$$

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

- For $j = 1, \dots, N$ compute $u_h|_{I_j}$ as follows:
 1. Compute $u_{j+1/2}^-$ and $u_{j-1/2}^+$ by using (2.22) and (2.23),
 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 $\Lambda\Pi_h^1(v_h^1)$.

d. The limiter $\Lambda\Pi_{h,\alpha}^k$. When instead of (2.22) and (2.23), we use

$$\begin{aligned} u_{j+1/2}^- &= \bar{v}_j + m(v_{j+1/2}^- - \bar{v}_j, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j, C(\Delta x)^\alpha) \\ u_{j-1/2}^+ &= \bar{v}_j - m(\bar{v}_j - v_{j-1/2}^+, \bar{v}_j - \bar{v}_{j-1}, \bar{v}_{j+1} - \bar{v}_j, C(\Delta x)^\alpha), \end{aligned} \quad (2.23)$$

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

This generalized slope limiter is never used in practice, but we consider it here because it is used for theoretical purposes; see Theorem 16 below.

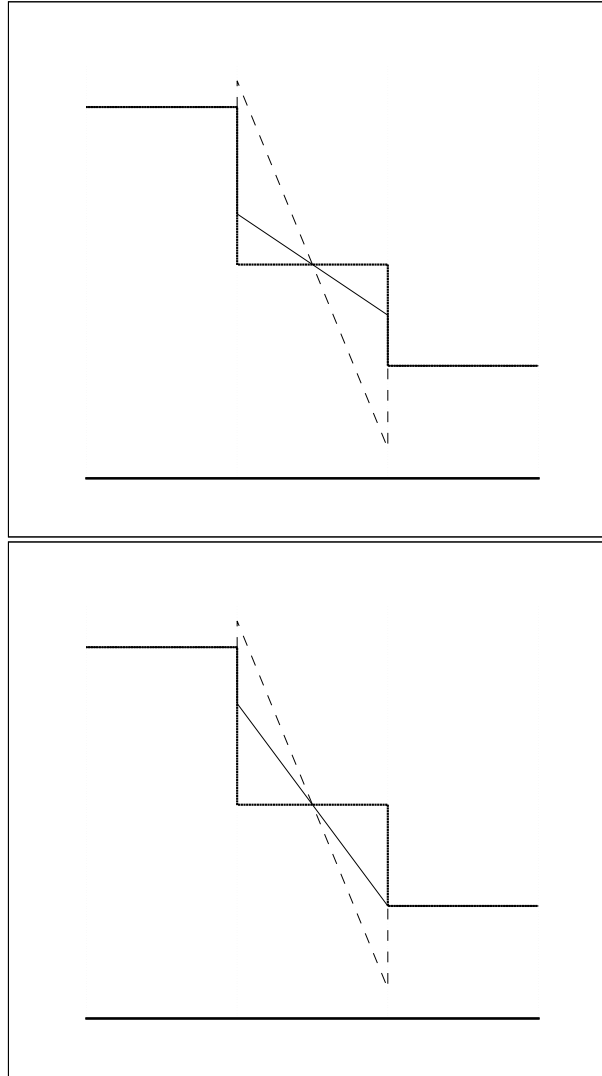


Fig. 2.1. Example of slope limiters: The MUSCL limiter (top) and the less restrictive AII_h^1 limiter (bottom). Displayed are the local means of u_h (thick line), the linear function u_h in the element of the middle before limiting (dotted line) and the resulting function after limiting (solid line).

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 algorithm:

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

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

then we have that

$$|\bar{u}_h|_{TV(0,1)} \leq |\bar{v}_h|_{TV(0,1)}, \quad (2.24)$$

$$|\bar{w}_h|_{TV(0,1)} \leq |\bar{u}_h|_{TV(0,1)}, \quad \forall |\delta| \leq \delta_0, \quad (2.25)$$

where

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

by Proposition 13. By using the above two properties of the generalized slope limiter, it is possible to show that the RKDG method is TVDM.

Theorem 14. (Stability induced by the generalized slope limiter) *Assume that the generalized slope limiter $\Lambda \Pi_h$ satisfies the properties (2.24) and (2.25). Assume also that all the coefficients α_{il} are nonnegative and satisfy the following condition:*

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

Then

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

Proof. The proof of this result is very similar to that of Theorem 8. Thus, we start by rewriting our time discretization as follows:

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

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

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,

$$\begin{aligned} |\bar{u}_h^{(i)}|_{TV(0,1)} &\leq \left| \sum_{l=0}^{i-1} \alpha_{il} \bar{w}_h^{(il)} \right|_{TV(0,1)}, \quad \text{by (2.24),} \\ &\leq \sum_{l=0}^{i-1} \alpha_{il} |\bar{w}_h^{(il)}|_{TV(0,1)}, \quad \text{since } \alpha_{il} \geq 0, \\ &\leq \left| \sum_{l=0}^{i-1} \alpha_{il} \bar{u}_h^{(l)} \right|_{TV(0,1)}, \quad \text{by (2.25),} \\ &\leq \max_{0 \leq l \leq i-1} |\bar{u}_h^{(l)}|_{TV(0,1)}, \end{aligned}$$

since

$$\sum_{l=0}^{i-1} \alpha_{il} = 1.$$

It is clear now that that the inequality

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

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

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

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 [82] 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:

$$\bar{m}(a_1, \dots, a_m) = \begin{cases} a_1 & \text{if } |a_1| \leq M(\Delta x)^2, \\ m(a_1, \dots, a_m) & \text{otherwise,} \end{cases} \quad (2.26)$$

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

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

See [24] for other choices of M .

Thus, if the constant M 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 15. (The TVBM property) *Assume that the generalized slope limiter Π_h is a TVBM slope limiter. Assume also that all the coefficients α_{il} are nonnegative and satisfy the following condition:*

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

Then

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

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 16. (Convergence to the entropy solution) *Assume that the generalized slope limiter Π_h is a TVDM or a TVBM slope limiter. Assume also that all the coefficients α_{il} are nonnegative and satisfy the following condition:*

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

Then there is a subsequence $\{\bar{u}_{h'}\}_{h'>0}$ of the sequence $\{\bar{u}_h\}_{h>0}$ generated by the RKDG scheme that converges in $L^\infty(0, T; L^1(0, 1))$ to a weak solution of the problem (2.1), (2.2).

Moreover, if the TVBM version of the slope limiter $\Lambda\Pi_{h,\alpha}^k$ 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

$$\|\bar{v}_h - \Lambda\Pi_h(v_h)\|_{L^1(0,1)} \leq C \Delta x |\bar{v}_h|_{TV(0,1)},$$

then the above results hold not only to the sequence of the means $\{\bar{u}_h\}_{h>0}$ but to the sequence of the functions $\{u_h\}_{h>0}$.

Error estimates for an *implicit* version of the discontinuous Galerkin method (with the so-called shock-capturing terms) have been obtained by Cockburn and Gremaud [19].

2.5 Computational results

In this section, we display the performance of the RKDG schemes in two simple but typical test problems. We use piecewise linear ($k = 1$) and piecewise quadratic ($k = 2$) elements; the $\Lambda\Pi_h^k$ generalized slope limiter is used.

The first test problem. We consider the simple transport equation with periodic boundary conditions:

$$\begin{aligned} u_t + u_x &= 0, \\ u(x, 0) &= \begin{cases} 1 & .4 < x < .6, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

We use this test problem to show that the use of high-order polynomial approximation does improve the approximation of the discontinuities (or, in this case, ‘contacts’). To amplify the effect of the dissipation of the method, we take $T = 100$, that is, we let the solution travel 100 times across the domain. We run the scheme with $CFL = 0.9 \cdot 1 = 0.9$ for $k = 0$, $CFL = 0.9 \cdot 1/3 = 0.3$ for $k = 1$, and $CFL = 0.9 \cdot 1/5 = 0.18$ for $k = 2$. In Figure 2.2, we can see that the dissipation effect decreases as the degree of the polynomial k increases; we also see that the dissipation effect for a given k decreases as the Δx decreases, as expected. Other experiments in this direction have been performed by Atkins and Shu [1]. For example, they show that when polynomials of degree $k = 11$ are used, there is no detectable decay of the approximate solution.

To assess if the use of high degree polynomials is advantageous, we must compare the efficiencies of the schemes; we only compare the efficiencies of the method for $k = 1$ and $k = 2$. 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 $3 \times 3/2 \times 2$ times more operations per element, its number of operations is $81/16$ times bigger

than the one of the RKDG 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

$$eff.ratio = \frac{16}{81} \frac{error(RKDG(k=1))}{error(RKDG(k=2))}.$$

In Table 2, we see that the use of a higher degree does result in a more efficient resolution of the contact discontinuities. This fact remains true for systems as we can see from the numerical experiments for the double Mach reflection problem in the next chapter.

The second test problem. We consider the standard Burgers equation with periodic boundary conditions:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0,$$

$$u(x, 0) = u_0(x) = \frac{1}{4} + \frac{1}{2} \sin(\pi(2x - 1)).$$

Our purpose is to show that (i) when the constant M is properly chosen, the RKDG method using polynomials of degree k 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

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 3,4, and 5, the history of convergence of the RKDG method using piecewise linear elements is displayed and in Tables 6,7, and 8, 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 $T = 0.05$. The results are displayed in Table 9. 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 efficiency ratio 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 shocks can be seen in Figures 2.3, 2.4, and 2.5. In the Figure 2.3, it can be seen that the shock is captured in essentially two elements. Details of these figures are shown in Figures 2.4 and 2.5, where the approximations right in front of the shock are shown. It is clear that the approximation using quadratic elements is superior

to the approximation using linear elements. Finally, we illustrate in Figure 2.6 how the schemes follow a shock when it goes through a single element.

2.6 Concluding remarks

In this section, 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 discretization is 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 in a very simple way, see [24]. In what follows, we extend the RKDG methods to multidimensional systems.

Table 2

Comparison of the efficiencies of RKDG schemes for $k = 1$ and $k = 2$
Transport equation with $M = 0$, and $T = 100$.

	L ¹ -norm	
Δx	<i>eff.ratio</i>	order
1/10	0.88	-
1/20	0.93	-0.08
1/40	1.81	-0.96
1/80	2.57	-0.50
1/160	3.24	-0.33

Table 3
 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 4
 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

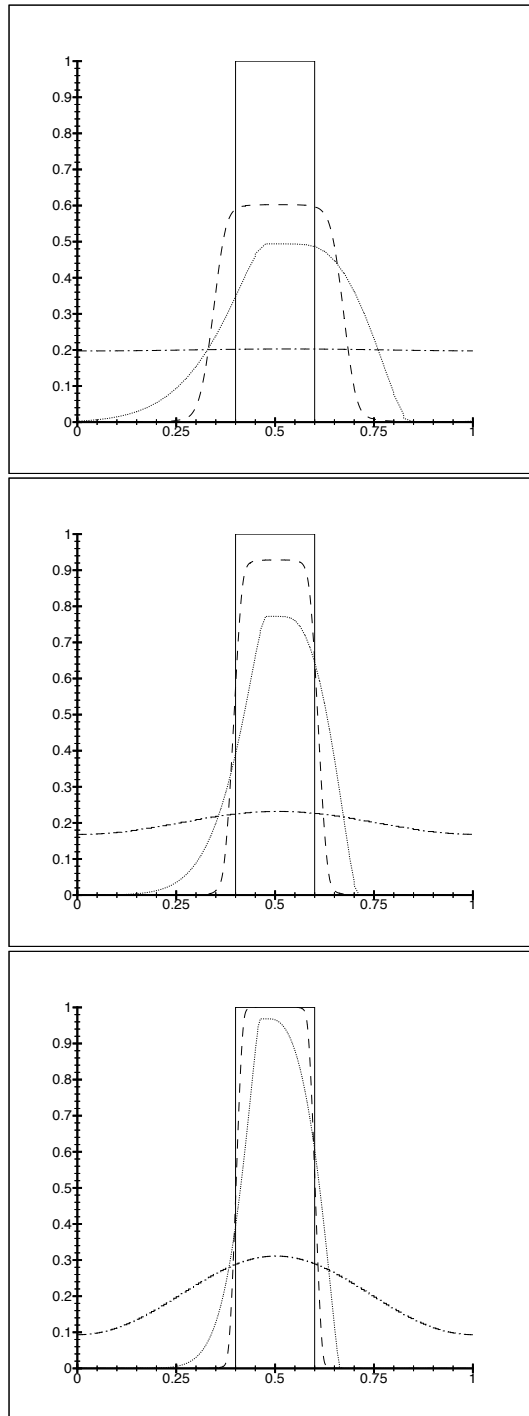


Fig. 2.2. Comparison of the exact and the approximate solutions for the linear case $f(u) = u$. Top: $\Delta x = 1/40$, middle: $\Delta x = 1/80$, bottom: $\Delta x = 1/160$. Exact solution (solid line), piecewise-constant elements (dash/dotted line), piecewise-linear elements (dotted line) and piecewise-quadratic elements (dashed line).

Table 5

Errors in smooth region $\Omega = \{x : |x - shock| \geq 0.1\}$.
 P^1 , $M = 20$, CFL=0.3, $T = 0.4$.

Δx	$L^1(\Omega) - error$		$L^\infty(\Omega) - error$	
	$10^5 \cdot error$	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 6

P^2 , $M = 0$, CFL=0.2, $T = 0.05$.

Δx	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
	$10^5 \cdot error$	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 7
 P^2 , $M = 20$, CFL= 0.2, $T = 0.05$.

Δx	$L^1(0,1) - error$		$L^\infty(0,1) - error$	
	$10^5 \cdot error$	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 8
 Errors in smooth region $\Omega = \{x : |x - shock| \geq 0.1\}$.
 P^2 , $M = 20$, CFL= 0.2, $T = 0.4$.

Δx	$L^1(\Omega) - error$		$L^\infty(\Omega) - error$	
	$10^5 \cdot error$	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 9

Comparison of the efficiencies of RKDG schemes for $k = 1$ and $k = 2$
Burgers equation with $M = 20$, and $T = 0.05$.

Δx	L ¹ -norm		L [∞] -norm	
	<i>eff.ratio</i>	order	<i>eff.ratio</i>	order
1/10	5.68	-	4.69	-
1/20	11.96	-1.07	31.02	-2.73
1/40	25.83	-1.11	70.90	-1.19
1/80	52.97	-1.04	148.42	-1.07

2.7 Appendix: Proof of the L²-error estimates

Proof of the L²-stability In this section, we prove the the stability result of Proposition 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 1.

The continuous case as a model. We start by rewriting the equations (2.4) in *compact form*. If in the equations (2.4) 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

$$\begin{aligned} \forall v : v(t) \text{ is smooth} \quad \forall t \in (0, T) : \\ B(u, v) = 0, \end{aligned} \tag{2.27}$$

where

$$B(u, v) = \int_0^T \int_0^1 \{ \partial_t u(x, t) v(x, t) - c u(x, t) \partial_x v(x, t) \} dx dt.$$

Taking $v = u$, we easily see that we see that

$$B(u, u) = \frac{1}{2} \|u(T)\|_{L^2(0,1)}^2 - \frac{1}{2} \|u_0\|_{L^2(0,1)}^2,$$

and since

$$B(u, u) = 0,$$

by (2.27), we immediately obtain the following L^2 -stability result:

$$\frac{1}{2} \|u(T)\|_{L^2(0,1)}^2 = \frac{1}{2} \|u_0\|_{L^2(0,1)}^2.$$

This is the argument we have to mimic in order to prove Proposition 1.

The discrete case. Thus, we start by finding the discrete version of the form $B(\cdot, \cdot)$. If we replace $v(x)$ by $v_h(x, t)$ in the equation (2.7), sum on j from 1 to N , and integrate in time from 0 to T , we obtain

$$\begin{aligned} \forall v_h : v_h(t) \in V_h^k \quad \forall t \in (0, T) : \\ B_h(u_h, v_h) = 0, \end{aligned} \quad (2.28)$$

where

$$\begin{aligned} B_h(u_h, v_h) = & \int_0^T \int_0^1 \partial_t u_h(x, t) v_h(x, t) dx dt \\ & - \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} c u_h(x, t) \partial_x v_h(x, t) dx dt \\ & - \int_0^T \sum_{1 \leq j \leq N} h(u_h)_{j+1/2}(t) [v_h(t)]_{j+1/2} dt. \end{aligned} \quad (2.29)$$

Following the model provided by the continuous case, we next obtain an expression for $B_h(w_h, w_h)$. It is contained in the following result which will be proved later.

Lemma 17. *We have*

$$B_h(w_h, w_h) = \frac{1}{2} \|w_h(T)\|_{L^2(0,1)}^2 + \Theta_T(w_h) - \frac{1}{2} \|w_h(0)\|_{L^2(0,1)}^2,$$

where

$$\Theta_T(w_h) = \frac{|c|}{2} \int_0^T \sum_{1 \leq j \leq N} [w_h(t)]_{j+1/2}^2 dt.$$

Taking $w_h = u_h$ in the above result and noting that by (2.28),

$$B_h(u_h, u_h) = 0,$$

we get the equality

$$\frac{1}{2} \|u_h(T)\|_{L^2(0,1)}^2 + \Theta_T(u_h) = \frac{1}{2} \|u_h(0)\|_{L^2(0,1)}^2,$$

from which Proposition 1 easily follows, since

$$\frac{1}{2} \|u_h(T)\|_{L^2(0,1)}^2 \leq \frac{1}{2} \|u_0\|_{L^2(0,1)}^2,$$

by (2.8). It only remains to prove Lemma 17.

Proof of Lemma 17. After setting $u_h = v_h = w_h$ in the definition of B_h , (2.29), we get

$$B_h(w_h, w_h) = \frac{1}{2} \|w_h(T)\|_{L^2(0,1)}^2 + \int_0^T \Theta_{diss}(t) dt - \frac{1}{2} \|w_h(0)\|_{L^2(0,1)}^2,$$

where

$$\Theta_{diss}(t) = - \sum_{1 \leq j \leq N} \left\{ h(w_h)_{j+1/2}(t) [w_h(t)]_{j+1/2} + \int_{I_j} c w_h(x, t) \partial_x w_h(x, t) dx \right\}.$$

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

$$\bar{w}_h(x_{j+1/2}) = \frac{1}{2} (w_h(x_{j+1/2}^-) + w_h(x_{j+1/2}^+)),$$

we have, by the definition of the flux h , (2.11),

$$- \sum_{1 \leq j \leq N} \int_{I_j} h(w_h)_{j+1/2} [w_h]_{j+1/2} = - \sum_{1 \leq j \leq N} \left\{ c \bar{w}_h [w_h] - \frac{|c|}{2} [w_h]^2 \right\}_{j+1/2},$$

and

$$\begin{aligned} - \sum_{1 \leq j \leq N} \int_{I_j} c w_h(x) \partial_x w_h(x) dx &= \frac{c}{2} \sum_{1 \leq j \leq N} [w_h^2]_{j+1/2} \\ &= c \sum_{1 \leq j \leq N} \{ \bar{w}_h [w_h] \}_{j+1/2}. \end{aligned}$$

Hence

$$\Theta_{diss}(t) = \frac{|c|}{2} \sum_{1 \leq j \leq N} [u_h(t)]_{j+1/2}^2,$$

and the result follows. This completes the proof of Lemma 17.

This completes the proof of Proposition 1.

Proof of Theorem 2 In this section, we prove the error estimate of Theorem 2 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 2.

The continuous case as a model. By the definition of the form $B(\cdot, \cdot)$, (2.7), we have, for $\nu = 1, 2$,

$$B(w_\nu, v) = 0, \quad \forall v : v(t) \text{ is smooth} \quad \forall t \in (0, T).$$

Since the form $B(\cdot, \cdot)$ is bilinear, from the above equation we obtain the so-called *error equation*:

$$\begin{aligned} \forall v : v(t) \text{ is smooth} \quad \forall t \in (0, T) : \\ B(e, v) = 0, \end{aligned} \tag{2.30}$$

where $e = w_1 - w_2$. Now, since

$$B(e, e) = \frac{1}{2} \|e(T)\|_{L^2(0,1)}^2 - \frac{1}{2} \|e(0)\|_{L^2(0,1)}^2,$$

and

$$B(e, e) = 0,$$

by the error equation (2.30), we immediately obtain the error estimate we sought:

$$\frac{1}{2} \|e(T)\|_{L^2(0,1)}^2 = \frac{1}{2} \|u_{0,1} - u_{0,2}\|_{L^2(0,1)}^2.$$

To prove Theorem 2, we only need to obtain a discrete version of this argument.

The discrete case. Since,

$$\begin{aligned} B_h(u_h, v_h) = 0, \quad \forall v_h : v(t) \in V_h \quad \forall t \in (0, T), \\ B_h(u, v_h) = 0, \quad \forall v_h : v_h(t) \in V_h \quad \forall t \in (0, T), \end{aligned}$$

by (2.7) and by equations (2.4), respectively, we easily obtain our *error equation*:

$$\begin{aligned} \forall v_h : v_h(t) \in V_h \quad \forall t \in (0, T) : \\ B_h(e, v_h) = 0, \end{aligned} \tag{2.31}$$

where $e = w - w_h$.

Now, according to the continuous case argument, we should consider next the quantity $B_h(e, e)$; however, since $e(t)$ is not in the finite element space V_h , it is more convenient to consider $B_h(P_h(e), P_h(e))$, where $P_h(e(t))$ 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 $p \in L^2(0, 1)$ into V_h , $P_h(p)$, is defined as the only element of the finite element space V_h such that

$$\begin{aligned} \forall v_h \in V_h : \\ \int_0^1 (P_h(p)(x) - p(x)) v_h(x) dx = 0. \end{aligned} \tag{2.32}$$

Note that in fact $u_h(t=0) = P_h(u_0)$, by (2.8).

Thus, by Lemma 17, we have

$$B_h(P_h(e), P_h(e)) = \frac{1}{2} \|P_h(e(T))\|_{L^2(0,1)}^2 + \Theta_T(P_h(e)) - \frac{1}{2} \|P_h(e(0))\|_{L^2(0,1)}^2,$$

and since

$$P_h(e(0)) = P_h(u_0 - u_h(0)) = P_h(u_0) - u_h(0) = 0,$$

and

$$B_h(P_h(e), P_h(e)) = B_h(P_h(e) - e, P_h(e)) = B_h(P_h(u) - u, P_h(e)),$$

by the *error equation* (2.31), we get

$$\frac{1}{2} \|P_h(e(T))\|_{L^2(0,1)}^2 + \Theta_T(P_h(e)) = B_h(P_h(u) - u, P_h(e)). \quad (2.33)$$

It only remains to estimate the right-hand side

$$B(P_h(u) - u, P_h(e)),$$

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 $B(P_h(w) - w, P_h(e))$. We start with the following remarkable result.

Lemma 18. *We have*

$$B_h(P_h(u) - u, P_h(e)) = - \int_0^T \sum_{1 \leq j \leq N} h(P_h(u) - u)_{j+1/2}(t) [P_h(e)(t)]_{j+1/2} dt.$$

Proof Setting $p = P_h(u) - u$ and $v_h = P_h(e)$ and recalling the definition of $B_h(\cdot, \cdot)$, (2.29), we have

$$\begin{aligned} B_h(p, v_h) &= \int_0^T \int_0^1 \partial_t p(x, t) v_h(x, t) dx dt \\ &\quad - \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} c p(x, t) \partial_x v_h(x, t) dx dt \\ &\quad - \int_0^T \sum_{1 \leq j \leq N} h(p)_{j+1/2}(t) [v_h(t)]_{j+1/2} dt \\ &= - \int_0^T \sum_{1 \leq j \leq N} h(p)_{j+1/2}(t) [v_h(t)]_{j+1/2} dt, \end{aligned}$$

by the definition of the L^2 -projection (2.32). 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 19. *If $w \in H^{k+1}(I_j \cup I_{j+1})$, then*

$$|h(P_h(w) - w)(x_{j+1/2})| \leq c_k (\Delta x)^{k+1/2} \frac{|c|}{2} |w|_{H^{k+1}(I_j \cup I_{j+1})},$$

where the constant c_k depends solely on k .

Proof. Dropping the argument $x_{j+1/2}$ we have, by the definition (2.11) of the flux h ,

$$\begin{aligned} |h(P_h(w) - w)| &= \left| \frac{c}{2}(P_h(w)^+ + P_h(w)^-) - \frac{|c|}{2}(P_h(w)^+ - P_h(w)^-) - cw \right| \\ &= \left| \frac{c - |c|}{2}(P_h(w)^+ - w) + \frac{c + |c|}{2}(P_h(w)^- - w) \right| \\ &\leq |c| \max\{|P_h(w)^+ - w|, |P_h(w)^- - w|\} \end{aligned}$$

and the result follows from the properties of P_h after a simple application of the Bramble-Hilbert lemma; see [16]. This completes the proof.

An immediate consequence of this result is the estimate we wanted.

Lemma 20. *We have*

$$B_h(P_h(u) - u, P_h(e)) \leq c_k^2 (\Delta x)^{2k+1} \frac{|c|}{2} T |u_0|_{H^{k+1}(0,1)}^2 + \frac{1}{2} \Theta_T(P_h(e)),$$

where the constant c_k depends solely on k .

Proof. After using Young's inequality in the right-hand side of Lemma 18, we get

$$\begin{aligned} B_h(P_h(u) - u, P_h(e)) &\leq \int_0^T \sum_{1 \leq j \leq N} \frac{1}{|c|} |h(P_h(u) - u)_{j+1/2}(t)|^2 \\ &\quad + \int_0^T \sum_{1 \leq j \leq N} \frac{|c|}{4} [P_h(e)(t)]_{j+1/2}^2 dt. \end{aligned}$$

By Lemma 19 and the definition of the form Θ_T , we get

$$\begin{aligned} B_h(P_h(u) - u, P_h(e)) &\leq c_k^2 (\Delta x)^{2k+1} \frac{|c|}{4} \int_0^T \sum_{1 \leq j \leq N} |u|_{H^{k+1}(I_j \cup I_{j+1})}^2 + \frac{1}{2} \Theta_T(P_h(e)) \\ &\leq c_k^2 (\Delta x)^{2k+1} \frac{|c|}{2} T |u_0|_{H^{k+1}(0,1)}^2 \frac{1}{2} \Theta_T(P_h(e)). \end{aligned}$$

This completes the proof.

Conclusion. Finally, inserting in the equation (2.33) the estimate of its right hand side obtained in Lemma 20, we get

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

Theorem 2 now follows from the above estimate and from the following inequality:

$$\begin{aligned} \|e(T)\|_{L^2(0,1)} &\leq \|u(T) - P_h(u(T))\|_{L^2(0,1)} + \|P_h(e(T))\|_{L^2(0,1)} \\ &\leq c'_k (\Delta x)^{k+1} |u_0|_{H^{k+1}(0,1)} + \|P_h(e(T))\|_{L^2(0,1)}. \end{aligned}$$

Proof of Theorem 3 To prove Theorem 3, we only have to suitably modify the proof of Theorem 2. The modification consists in *replacing* the L^2 -projection of the error, $P_h(e)$, by *another* projection that we denote by $R_h(e)$.

Given a function $p \in L^\infty(0,1)$ that is continuous on each element I_j , we define $R_h(p)$ as the only element of the finite element space V_h such that

$$\begin{aligned} \forall j = 1, \dots, N : \\ R_h(p)(x_{j,\ell}) - p(x_{j,\ell}) = 0, \quad \ell = 0, \dots, k, \end{aligned} \quad (2.34)$$

where the points $x_{j,\ell}$ are the Gauss-Radau quadrature points of the interval I_j . We take

$$x_{j,k} = \begin{cases} x_{j+1/2} & \text{if } c > 0, \\ x_{j-1/2} & \text{if } c < 0. \end{cases} \quad (2.35)$$

The special nature of the Gauss-Radau quadrature points is captured in the following property:

$$\begin{aligned} \forall \varphi \in P^\ell(I_j), \quad \ell \leq k, \quad \forall p \in P^{2k-\ell}(I_j) : \\ \int_{I_j} (R_h(p)(x) - p(x)) \varphi(x) dx = 0. \end{aligned} \quad (2.36)$$

Compare this equality with (2.32).

The quantity $B_h(R_h(e), R_h(e))$. To prove our error estimate, we start by considering the quantity $B_h(R_h(e), R_h(e))$. By Lemma 17, we have

$$B_h(R_h(e), R_h(e)) = \frac{1}{2} \|R_h(e(T))\|_{L^2(0,1)}^2 + \Theta_T(R_h(e)) - \frac{1}{2} \|R_h(e(0))\|_{L^2(0,1)}^2,$$

and since

$$B_h(R_h(e), R_h(e)) = B_h(R_h(e) - e, R_h(e)) = B_h(R_h(u) - u, R_h(e)),$$

by the *error equation* (2.31), we get

$$\frac{1}{2} \|R_h(e(T))\|_{L^2(0,1)}^2 + \Theta_T(R_h(e)) = \frac{1}{2} \|R_h(e(0))\|_{L^2(0,1)}^2 + B_h(R_h(u) - u, R_h(e)).$$

Next, we estimate the term $B(R_h(u) - u, R_h(e))$.

Estimating $B(R_h(u) - u, R_h(e))$. The following result corresponds to Lemma 18.

Lemma 21. *We have*

$$\begin{aligned} B_h(R_h(u) - u, v_h) &= \int_0^T \int_0^1 (R_h(\partial_t u)(x, t) - \partial_t u(x, t)) v_h(x, t) dx dt \\ &\quad - \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} c(R_h(u)(x, t) - u(x, t)) \partial_x v_h(x, t) dx dt. \end{aligned}$$

Proof Setting $p = R_h(u) - u$ and $v_h = R_h(e)$ and recalling the definition of $B_h(\cdot, \cdot)$, (2.29), we have

$$\begin{aligned} B_h(p, v_h) &= \int_0^T \int_0^1 \partial_t p(x, t) v_h(x, t) dx dt \\ &\quad - \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} c p(x, t) \partial_x v_h(x, t) dx dt \\ &\quad - \int_0^T \sum_{1 \leq j \leq N} h(p)_{j+1/2}(t) [v_h(t)]_{j+1/2} dt. \end{aligned}$$

But, from the definition (2.11) of the flux h , we have

$$\begin{aligned} h(R_h(u) - u) &= \frac{c}{2}(R_h(u)^+ + R_h(u)^-) - \frac{|c|}{2}(R_h(u)^+ - R_h(u)^-) - cu \\ &= \frac{c - |c|}{2}(R_h(u)^+ - u) + \frac{c + |c|}{2}(R_h(u)^- - u) \\ &= 0, \end{aligned}$$

by (2.35) and the result follows.

Next, we need some approximation results.

Lemma 22. *If $w \in H^{k+2}(I_j)$, and $v_h \in P^k(I_j)$, then*

$$\left| \int_{I_j} (R_h(w) - w)(x) v_h(x) dx \right| \leq c_k (\Delta x)^{k+1} \|w\|_{H^{k+1}(I_j)} \|v_h\|_{L^2(I_j)},$$

and

$$\left| \int_{I_j} (R_h(w) - w)(x) \partial_x v_h(x) dx \right| \leq c_k (\Delta x)^{k+1} \|w\|_{H^{k+2}(I_j)} \|v_h\|_{L^2(I_j)},$$

where the constant c_k depends solely on k .

Proof. The first inequality follows from the property (2.36) with $\ell = k$ and from standard approximation results. The second follows in a similar way from the property (2.36) 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 23. *We have*

$$B_h(R_h(u) - u, R_h(e)) \leq c_k (\Delta x)^{k+1} |u_0|_{H^{k+2}(0,1)} \int_0^T \|R_h(e(t))\|_{L^2(0,1)} dt,$$

where the constant c_k depends solely on k and $|c|$.

Conclusion. Finally, inserting in the equation (2.33) the estimate of its right hand side obtained in Lemma 23, we get

$$\begin{aligned} \|R_h(e(T))\|_{L^2(0,1)}^2 + \Theta_T(R_h(e)) &\leq \|R_h(e(0))\|_{L^2(0,1)}^2 \\ &\quad + c_k (\Delta x)^{k+1} |u_0|_{H^{k+2}(0,1)} \int_0^T \|R_h(e(t))\|_{L^2(0,1)} dt. \end{aligned}$$

After applying a simple variation of the Gronwall lemma, we obtain

$$\begin{aligned} \|R_h(e(T))\|_{L^2(0,1)} &\leq \|R_h(e(0))(x)\|_{L^2(0,1)} + c_k (\Delta x)^{k+1} T |u_0|_{H^{k+2}(0,1)} \\ &\leq c'_k (\Delta x)^{k+1} |u_0|_{H^{k+2}(0,1)}. \end{aligned}$$

Theorem 3 now follows from the above estimate and from the following inequality:

$$\begin{aligned} \|e(T)\|_{L^2(0,1)} &\leq \|u(T) - R_h(u(T))\|_{L^2(0,1)} + \|R_h(e(T))\|_{L^2(0,1)} \\ &\leq c'_k (\Delta x)^{k+1} |u_0|_{H^{k+1}(0,1)} + \|R_h(e(T))\|_{L^2(0,1)}. \end{aligned}$$

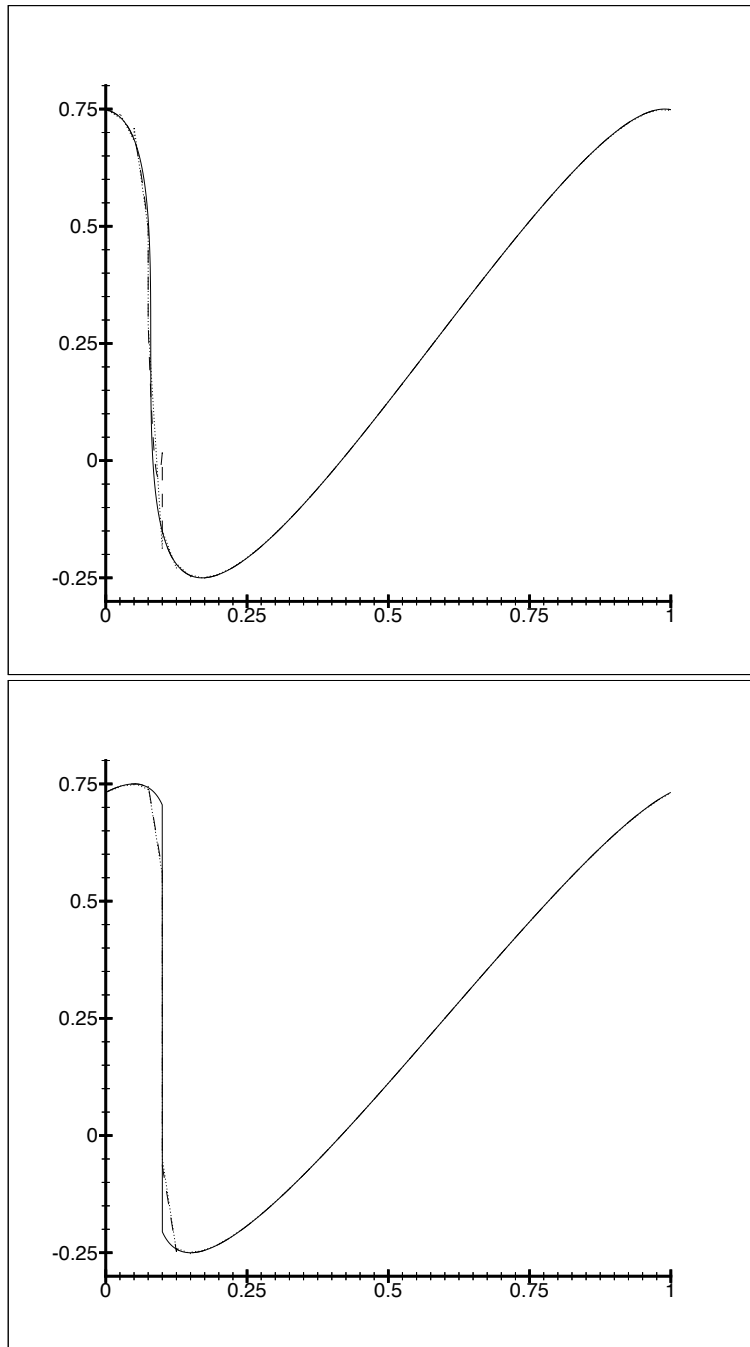


Fig. 2.3. Comparison of the exact and the approximate solutions obtained with $M = 20$, $\Delta x = 1/40$ at $T = 1/\pi$ (top) and at $T = 0.40$ (bottom): Exact solution (solid line), piecewise linear solution (dotted line), and piecewise quadratic solution (dashed line).

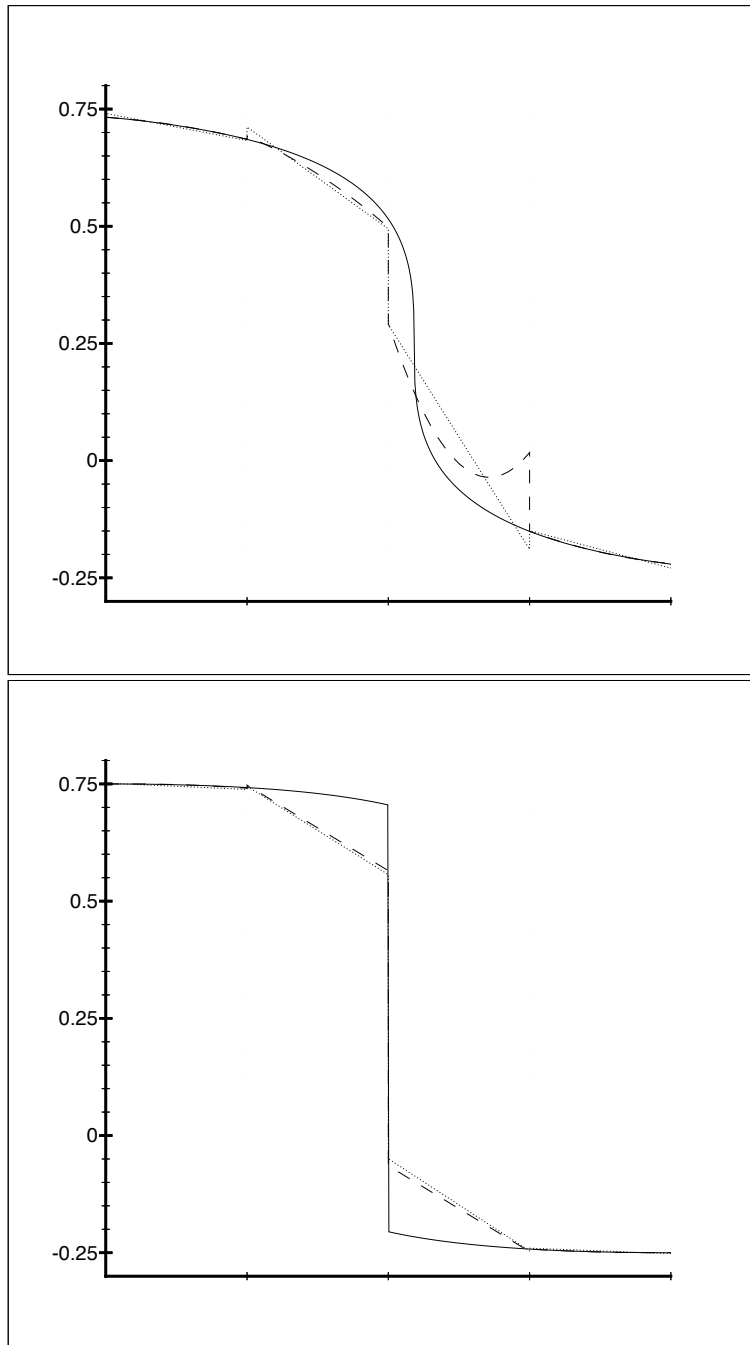


Fig. 2.4. Detail of previous figures. Behavior of the approximate solutions four elements around the shock at $T = 1/\pi$ (top) and at $T = 0.40$ (bottom): Exact solution (solid line), piecewise linear solution (dotted line), and piecewise quadratic solution (dashed line).

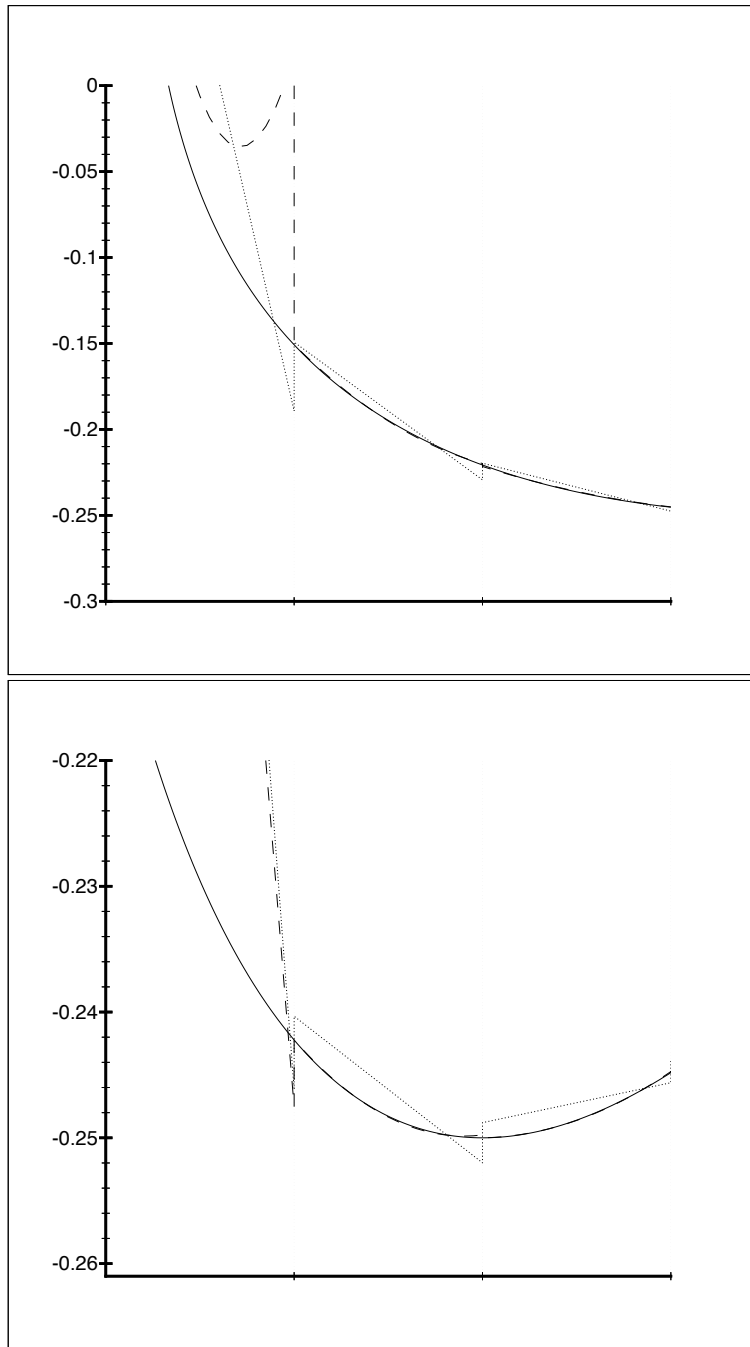


Fig. 2.5. Detail of previous figures. Behavior of the approximate solutions two elements in front of the shock at $T = 1/\pi$ (top) and at $T = 0.40$ (bottom): Exact solution (solid line), piecewise linear solution (dotted line), and piecewise quadratic solution (dashed line).

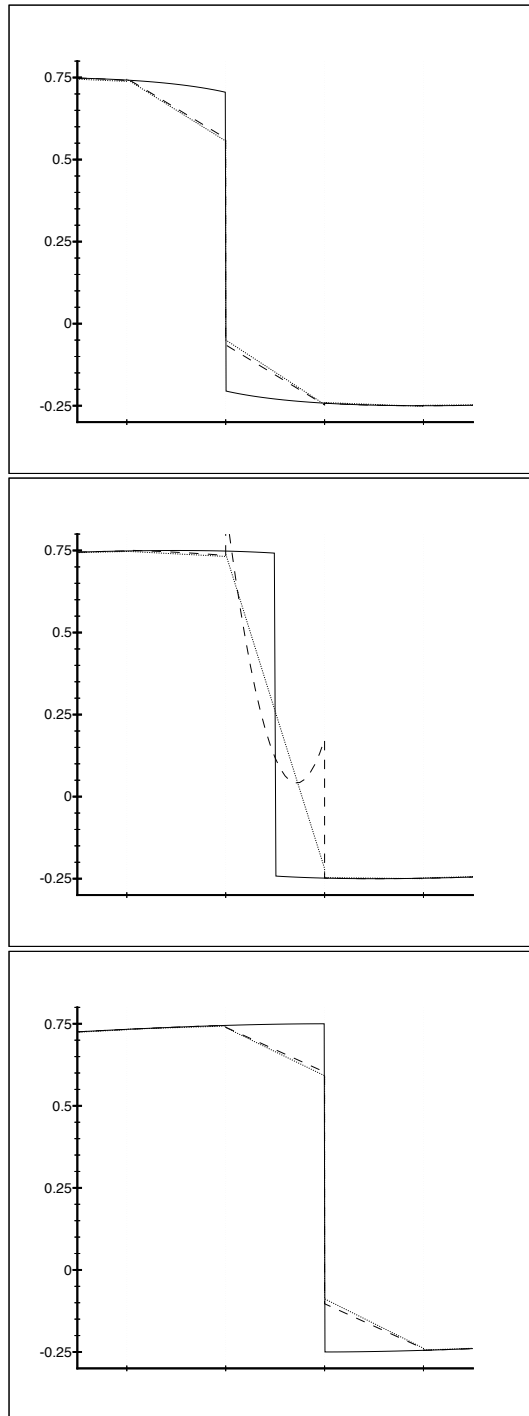


Fig. 2.6. Comparison of the exact and the approximate solutions obtained with $M = 20$, $\Delta x = 1/40$ as the shock passes through one element. Exact solution (solid line), piecewise linear elements (dotted line) and piecewise quadratic elements (dashed line). Top: $T = 0.40$, middle: $T = 0.45$, and bottom: $T = 0.50$.

3 The Hamilton-Jacobi equations in one space dimension

3.1 Introduction

In this chapter, we extend the RKDG method to the following simple problem for the Hamilton-Jacobi equation

$$\varphi_t + H(\varphi_x) = 0, \quad \text{in } (0, 1) \times (0, T), \quad (3.1)$$

$$\varphi(x, 0) = \varphi_0(x), \quad \forall x \in (0, 1), \quad (3.2)$$

where we take periodic boundary conditions. The material in this section is based in the work of Hu and Shu [43].

3.2 The RKDG method

The main idea to extend the RKDG method to this case, is to realize that $u = \varphi_x$ satisfies the following problem:

$$u_t + H(u)_x = 0, \quad \text{in } (0, 1) \times (0, T), \quad (3.3)$$

$$u(x, 0) = (\varphi_0)_x(x), \quad \forall x \in (0, 1), \quad (3.4)$$

and that φ can be computed from u by solving the following problem:

$$\varphi_t = -H(u), \quad \text{in } (0, 1) \times (0, T), \quad (3.5)$$

$$\varphi(x, 0) = \varphi_0(x), \quad \forall x \in (0, 1). \quad (3.6)$$

A straightforward application of the RKDG method to the equations (3.3), (3.4) produces a piecewise polynomial approximation u_h to $u = \varphi_x$. If the approximating polynomials are taken to be of degree $(k - 1)$, it is reasonable to seek an approximation φ_h to φ that is piecewise a polynomial of degree k . To obtain it, we can discretize (3.5), (3.6) in one of the following ways:

(i) Take $\varphi_h(\cdot, t)$ in V_h^k such that

$$\begin{aligned} \forall j = 1, \dots, N, \quad v_h \in P^k(I_j) : \\ \int_{I_j} \partial_t \varphi_h(x, t) v_h(x) dx = - \int_{I_j} H(u_h(x, t)) v_h(x) dx, \\ \int_{I_j} \varphi(x, 0) v_h(x) dx = \int_{I_j} \varphi_0(x) v_h(x) dx. \end{aligned}$$

(ii) Take $\varphi_h(\cdot, t)$ in V_h^k such that

$$\forall j = 1, \dots, N : \quad \partial_x \varphi_h(x, t) = u(x, t) \quad \forall x \in I_j.$$

This determines φ_h up to a constant. To find this constant, we impose the following condition:

$$\begin{aligned} \forall j = 1, \dots, N : \\ \frac{d}{dt} \int_{I_j} \varphi_h(x, t) dx &= - \int_{I_j} H(u_h(x, t)) dx, \\ \int_{I_j} \varphi(x, 0) dx &= \int_{I_j} \varphi_0(x) dx. \end{aligned}$$

(iii) Pick one element, say I_J , and determine the values of φ_h on it by requiring the following conditions:

$$\begin{aligned} \partial_x \varphi_h(x, t) &= u(x, t) \quad \forall x \in I_J, \\ \frac{d}{dt} \int_{I_J} \varphi_h(x, t) dx &= - \int_{I_J} H(u_h(x, t)) dx, \\ \int_{I_J} \varphi(x, 0) dx &= \int_{I_J} \varphi_0(x) dx. \end{aligned}$$

Then, compute φ_h as follows:

$$\varphi_h(x, t) = \varphi_h(x_J, t) + \int_{x_J}^x u(s, t) ds.$$

Note that, unlike the previous approaches, the approximate solution φ_h is now continuous.

The advantage of the first two approaches is that they can be carried out in parallel. On the other hand, in the third approach, only a single ODE has to be solved; moreover, the integration in space takes place just at the very end of the whole computation. This approach is much more efficient.

It could be argued that in the third approach, the recovered values of φ depend upon the choice of the starting point x_1 . However, this difference is on the level of truncation errors and does not affect the order of accuracy. Hu and Shu [43] used both the second and third approaches; they report that their numerical experience is that, when there are singularities in the derivatives, the second approach will often produce dents and bumps when the integral path in time passes through the singularities. This can be avoided in the third approach. Indeed, the main idea of using the third approach is to choose the element I_J so that the time integral paths *do not cross* derivative singularities. This cannot be always be done with a single element I_J , but it is always possible to switch to *another* element *before* the singularity in the derivative hits the current element I_J . If the number of discontinuities in the derivative is finite, this needs to be done only a finite number of times. This maintains the efficiency of the method.

Note that all the properties fo the RKDG method obtained in the previous section apply to the approximate solution u_h . In particular, a consequence of

the work of Jiang and Shu [51], is the following result for the approximation to the derivative φ_x , u_h ; see also Proposition 6 and Theorem 7.

Theorem 24. *For any of the above methods and any polynomial degree $k \geq 0$, we have*

$$\frac{d}{dt} \int_0^1 u_h^2(x, t) dx \leq 0. \quad (3.7)$$

Moreover, if the Hamiltonian H is a strictly convex or concave function, for any $k \geq 0$, if the numerical solution given by the DG method converges, it converges to the viscosity solution.

Note that the above result trivially implies the TVB (total variation bounded) property for the numerical solution φ_h . Indeed

$$TV(\varphi_h(t)) = \int_a^b |u_h(x, t)| dx \leq \sqrt{b-a} \|\varphi'_0\|_{L^2(a,b)}.$$

This is a rather strong stability result, considering that it holds independently of the degree of the polynomial approximation even when the derivative of the solution φ_x develops discontinuities and without the application of the generalized slope limiter!

3.3 Computational results

In this section, we present the numerical experiments of Hu and Shu [43] showing the performance of the RKDG method. Our main purpose is to assess the accuracy of the method and see if the generalized slope limiter needs to be used. We display the results obtained with the third approach to compute φ_h .

The first test problem. One dimensional Burgers' equation:

$$\begin{aligned} \varphi_t + \frac{(\varphi_x + 1)^2}{2} &= 0, & \text{in } (-1, 1) \times (0, T), \\ \varphi(x, 0) &= -\cos(\pi x), & \forall x \in (-1, 1), \end{aligned}$$

with periodic boundary conditions.

The local Lax-Friedrichs flux is used. At $T = 0.5/\pi^2$, the solution is still smooth. We list the errors and the numerical orders of accuracy in Table 3.1. We observe that, except for the P^1 case which seems to be only first order, P^k for $k > 1$ seems to provide close to $(k+1)$ -th order accuracy. The meshes used are all uniform, and errors are computed at the middle point of each interval.

To investigate the accuracy problem further, we use non-uniform meshes obtained by randomly shifting the cell boundaries in a uniform mesh in the

Table 3.1. Accuracy for 1D Burgers equation (uniform mesh), $T = 0.5/\pi^2$.

		P^1		P^2		P^3		P^4	
N	L^1 error	order	L^1 error	order	L^1 error	order	L^1 error	order	
10	0.17E+00	—	0.14E-02	—	0.21E-03	—	0.57E-05	—	
20	0.78E-01	1.12	0.18E-03	2.92	0.13E-04	3.94	0.73E-06	2.97	
40	0.35E-01	1.16	0.24E-04	2.97	0.75E-06	4.17	0.32E-07	4.52	
80	0.16E-01	1.12	0.28E-05	3.08	0.43E-07	4.12	0.12E-08	4.79	
160	0.76E-02	1.02	0.31E-06	3.19	0.25E-08	4.10	0.48E-10	4.59	

		P^1		P^2		P^3		P^4	
N	L^∞ error	order	L^∞ error	order	L^∞ error	order	L^∞ error	order	
10	0.29E+00	—	0.24E-02	—	0.69E-03	—	0.13E-04	—	
20	0.13E+00	1.13	0.33E-03	2.88	0.61E-04	3.51	0.16E-05	2.99	
40	0.58E-01	1.15	0.37E-04	3.15	0.58E-05	3.39	0.13E-06	3.64	
80	0.27E-01	1.11	0.48E-05	2.97	0.38E-06	3.93	0.59E-08	4.44	
160	0.13E-01	1.07	0.59E-06	3.00	0.23E-07	4.07	0.25E-09	4.57	

Table 3.2. Accuracy for 1D Burgers equation (non-uniform mesh), $T = 0.5/\pi^2$.

		P^1		P^2		P^3		P^4	
N	L^2 error	order	L^2 error	order	L^2 error	order	L^2 error	order	
10	0.74E+00	—	0.34E-02	—	0.32E-03	—	0.53E-04	—	
20	0.34E+00	1.11	0.51E-03	2.76	0.24E-04	3.72	0.20E-05	4.71	
40	0.15E+00	1.19	0.65E-04	2.96	0.17E-05	3.82	0.71E-07	4.84	
80	0.67E-01	1.17	0.90E-05	2.86	0.13E-06	3.72	0.20E-08	5.15	
160	0.31E-01	1.13	0.11E-05	3.02	0.81E-08	4.02	0.74E-10	4.76	

		P^1		P^2		P^3		P^4	
N	L^1 error	order	L^1 error	order	L^1 error	order	L^1 error	order	
10	0.53E+00	—	0.17E-02	—	0.23E-03	—	0.30E-05	—	
20	0.24E+00	1.13	0.21E-03	3.05	0.14E-04	4.01	0.40E-06	2.89	
40	0.11E+00	1.19	0.26E-04	2.99	0.78E-06	4.20	0.16E-07	4.65	
80	0.47E-01	1.17	0.37E-05	2.82	0.47E-07	4.05	0.61E-09	4.70	
160	0.21E-01	1.13	0.41E-06	3.16	0.27E-08	4.15	0.26E-10	4.56	

		P^1		P^2		P^3		P^4	
N	L^∞ error	order	L^∞ error	order	L^∞ error	order	L^∞ error	order	
10	0.62E+00	—	0.36E-02	—	0.69E-03	—	0.11E-04	—	
20	0.29E+00	1.11	0.47E-03	2.94	0.61E-04	3.52	0.16E-05	2.81	
40	0.13E+00	1.16	0.67E-04	2.80	0.47E-05	3.70	0.13E-06	3.64	
80	0.58E-01	1.14	0.17E-04	2.01	0.62E-06	2.91	0.59E-08	4.45	
160	0.27E-01	1.11	0.19E-05	3.11	0.31E-07	4.32	0.33E-09	4.17	

range $[-0.1h, 0.1h]$. In order to avoid possible superconvergence at cell centers, we also give the “real” L^2 error (computed by a 6-point Gaussian quadrature in each cell). The results are shown in Table 3.2.

At $T = 3.5/\pi^2$, the solution has developed a discontinuous derivative. In Fig. 3.1, we show the sharp corner-like numerical solution with 41 elements obtained with P^k for $k = 1, 2, 3, 4$ with a uniform mesh. Here and below, the solid line is the exact solution, the circles are numerical solutions (only one point per element is drawn).

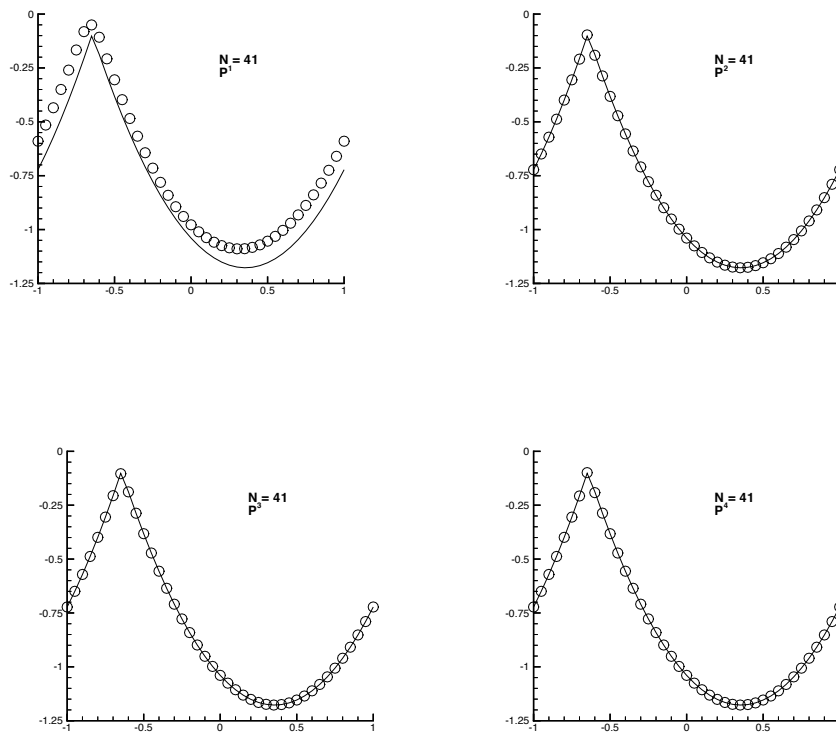


Fig. 3.1. One-dimensional Burgers' equation, $T = 3.5/\pi^2$.

The second test problem. One dimensional equation with a non-convex flux:

$$\begin{aligned} \varphi_t - \cos(\varphi_x + 1) &= 0, & \text{in } (-1, 1) \times (0, T), \\ \varphi(x, 0) &= -\cos(\pi x), & \forall x \in (-1, 1), \end{aligned}$$

with periodic boundary conditions.

The local Lax-Friedrichs flux and uniform meshes are used. At $T = 0.5/\pi^2$, the solution is still smooth. The accuracy of the numerical solution is listed in Table 3.3. We observe similar accuracy as in the previous example.

Table 3.3. Accuracy for 1D non-convex, $H(u) = -\cos(u + 1)$, $T = 0.5/\pi^2$.

	P^1		P^2		P^3		P^4	
N	L^1 error	order	L^1 error	order	L^1 error	order	L^1 error	order
10	0.84E-01	—	0.10E-02	—	0.34E-03	—	0.24E-04	—
20	0.36E-01	1.23	0.15E-03	2.75	0.30E-04	3.49	0.13E-05	4.28
40	0.15E-01	1.26	0.21E-04	2.84	0.15E-05	4.33	0.59E-07	4.42
80	0.68E-02	1.14	0.27E-05	2.97	0.94E-07	4.00	0.21E-08	4.78

	P^1		P^2		P^3		P^4	
N	L^∞ error	order	L^∞ error	order	L^∞ error	order	L^∞ error	order
10	0.18E+00	—	0.15E-02	—	0.11E-02	—	0.99E-04	—
20	0.73E-01	1.31	0.27E-03	2.43	0.22E-03	2.35	0.13E-04	2.95
40	0.31E-01	1.24	0.47E-04	2.54	0.18E-04	3.63	0.59E-06	4.44
80	0.14E-01	1.16	0.85E-05	2.47	0.14E-05	3.75	0.26E-07	4.49

At $T = 1.5/\pi^2$, the solution has developed corner-like discontinuity in the derivative. The numerical result with 41 elements is shown in Fig. 3.2.

The third test problem. Riemann problem for the one dimensional equation with a non-convex flux:

$$\begin{aligned} \varphi_t + \frac{1}{4}(\varphi_x^2 - 1)(\varphi_x^2 - 4) &= 0, & \text{in } (-1, 1) \times (0, T), \\ \varphi(x, 0) &= -2|x|, & \forall x \in (-1, 1), \end{aligned}$$

For this test problem, the use of the generalized slope limiter proved to be *essential* since otherwise the approximate solution does not converge to the viscosity solution; this is the only example in which we use the nonlinear limiting. We remark that for the finite difference schemes, such nonlinear limiting or the adaptive stencil in ENO is needed in most cases in order to enforce stability and to obtain non-oscillatory results.

Numerical results at $T = 1$ with 81 elements, using the local Lax-Friedrichs flux, is shown in Fig. 3.3. The results of using the Godunov flux is shown in Fig. 3.4. We can see that while for P^1 , the results of using two different monotone fluxes are significantly different in resolution, this difference is greatly reduced for higher order of accuracy. In most of the high order cases, the simple local Lax-Friedrichs flux is a very good choice.

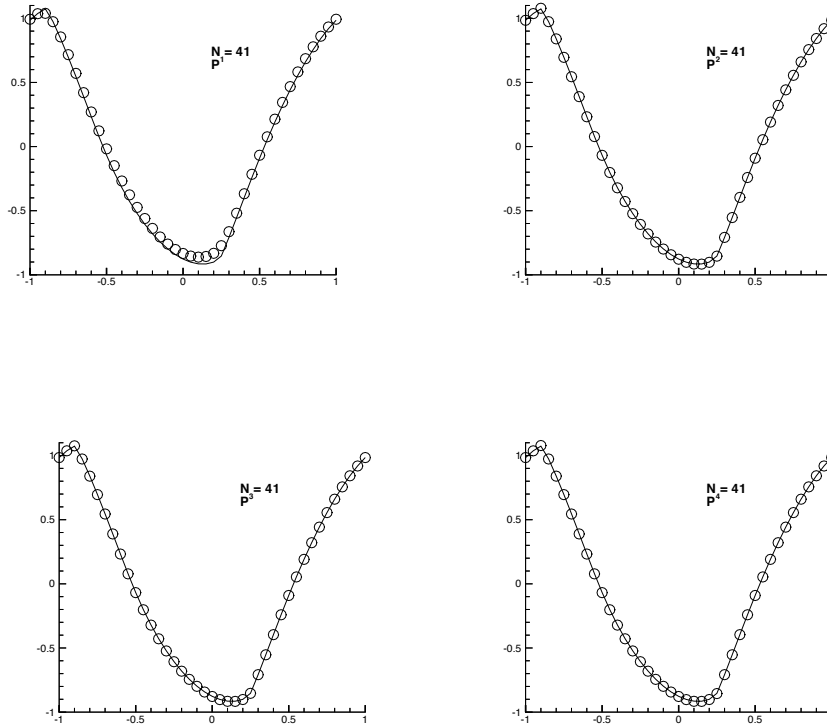


Fig. 3.2. One dimension non-convex, $H(u) = -\cos(u + 1)$, $T = 1.5/\pi^2$.

3.4 Concluding Remarks.

In this section, we have extended the RKDG method, originally devised for nonlinear conservation laws, to the Hamilton-Jacobi equations. The extension was carried out by exploiting the fact that the derivative of the solution of the Hamilton-Jacobi equation satisfies a nonlinear conservation law.

The numerical experiments show that when polynomials of degree k are used, the method is of order $(k+1)$ in L^2 , except when $k = 1$; this phenomenon remains to be explained. Also, we have seen that the use of slope limiters was only needed in the third test problem- otherwise the convergence to the viscosity solution did not take place.

The scheme can be extended to the case of a bounded domain in a very simple way. The extension of the scheme to the multidimensional case is not quite straightforward and will be carried out after we study how to define the RKDG method for multidimensional conservation laws.

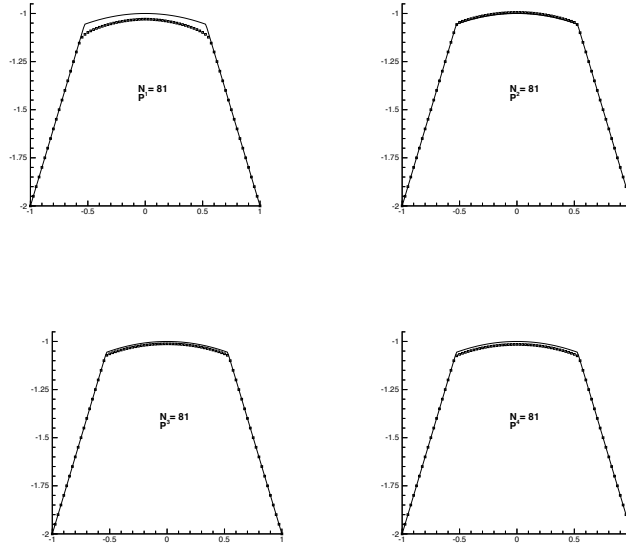


Fig. 3.3. One dimension Riemann problem, local Lax-Friedrichs flux, $H(u) = \frac{1}{4}(u^2 - 1)(u^2 - 4)$, $T = 1$.

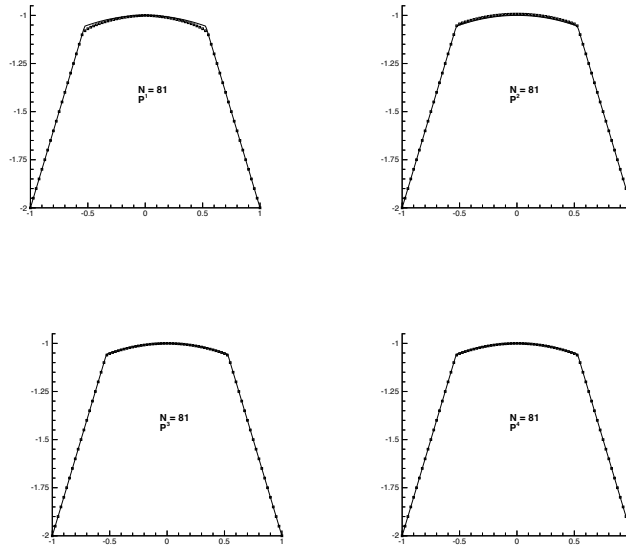


Fig. 3.4. One dimension Riemann problem, Godunov flux, $H(u) = \frac{1}{4}(u^2 - 1)(u^2 - 4)$, $T = 1$.

4 The RKDG method for multi-dimensional systems

4.1 Introduction

In this section, we extend the RKDG methods to multidimensional systems:

$$u_t + \nabla f(u) = 0, \quad \text{in } \Omega \times (0, T), \quad (4.1)$$

$$u(x, 0) = u_0(x), \quad \forall x \in \Omega, \quad (4.2)$$

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 sought.

This section contains material from the papers [21], [20], and [28]. It also contains numerical results from the paper by Bassi and Rebay [4] in two dimensions and from the paper by Warburton, Lomtev, Kirby and Karniadakis [91] 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 $u_h^0 = \mathcal{M}\Pi_h P_{V_h}(u_0)$;
- For $n = 0, \dots, N - 1$ compute u_h^{n+1} as follows:
 1. set $u_h^{(0)} = u_h^n$;

2. for $i = 1, \dots, k + 1$ 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 from 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 \mathcal{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 element space V_h , integrate over the element K of the triangulation \mathcal{T}_h and replace the exact solution u by its approximation $u_h \in V_h$:

$$\frac{d}{dt} \int_K u_h(t, x) v_h(x) dx + \int_K \operatorname{div} f(u_h(t, x)) v_h(x) dx = 0, \quad \forall v_h \in V_h. \quad (4.3)$$

Integrating by parts formally we obtain

$$\begin{aligned} & \frac{d}{dt} \int_K u_h(t, x) v_h(x) dx + \sum_{e \in \partial K} \int_e f(u_h(t, x)) \cdot n_{e,K} v_h(x) d\Gamma \\ & - \int_K f(u_h(t, x)) \cdot \nabla v_h(x) dx = 0, \quad \forall v_h \in V_h, \end{aligned}$$

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 $h_{e,K}(u_h(t, x^{\operatorname{int}(K)}), u_h(t, x^{\operatorname{ext}(K)}))$. 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

$$\begin{aligned} & \frac{d}{dt} \int_K u_h(t, x) v_h(x) dx + \sum_{e \in \partial K} \int_e h_{e,K}(t, x) v_h(x) d\Gamma \\ & - \int_K f(u_h(t, x)) \cdot \nabla v_h(x) dx = 0, \quad \forall v_h \in V_h. \end{aligned}$$

Finally, we replace the integrals by quadrature rules that we shall choose as follows:

$$\int_e h_{e,K}(t, x) v_h(x) d\Gamma \approx \sum_{l=1}^L \omega_l h_{e,K}(t, x_{el}) v(x_{el}) |e|, \quad (4.4)$$

$$\int_K f(u_h(t, x)) \cdot \nabla v_h(x) dx \approx \sum_{j=1}^M \omega_j f(u_h(t, x_{Kj})) \cdot \nabla v_h(x_{Kj}) |K|. \quad (4.5)$$

Thus, we finally obtain, for each element $K \in \mathcal{T}_h$, the weak formulation:

$$\begin{aligned} & \frac{d}{dt} \int_K u_h(t, x) v_h(x) dx + \sum_{e \in \partial K} \sum_{l=1}^L \omega_l h_{e,K}(t, x_{el}) v(x_{el}) |e| \\ & - \sum_{j=1}^M \omega_j f(u_h(t, x_{K_j})) \cdot \nabla v_h(x_{K_j}) |K| = 0, \quad \forall v_h \in V_h. \end{aligned}$$

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 $-\operatorname{div} f(u)$. The following result gives an indication of the quality of this approximation.

Proposition 25. *Let $f(u) \in W^{k+2, \infty}(\Omega)$, and set $\gamma = \operatorname{trace}(u)$. 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 $\mathcal{F} = \{\mathcal{T}_h\}_{h>0}$ is regular, i.e., that there is a constant σ such that:*

$$\frac{h_K}{\rho_K} \geq \sigma, \quad \forall K \in \mathcal{T}_h, \quad \forall \mathcal{T}_h \in \mathcal{F}, \quad (4.6)$$

where h_K is the diameter of K , and ρ_K is the diameter of the biggest ball included in K . Then, if $V(K) \supset P^k(K)$, $\forall K \in \mathcal{T}_h$:

$$\|L_h(u, \gamma) + \operatorname{div} f(u)\|_{L^\infty(\Omega)} \leq C h^{k+1} |f(u)|_{W^{k+2, \infty}(\Omega)}.$$

For a proof, see [20].

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 [20].

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:

$$h_{e,K}(a,b) = \frac{1}{2} [\mathbf{f}(a) \cdot n_{e,K} + \mathbf{f}(b) \cdot n_{e,K} - \alpha_{e,K} (b - a)].$$

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 [20], the quadrature rules for the edges of the elements, (4.4), must be exact for polynomials of degree $2k+1$, and the quadrature rules for the interior of the elements, (4.5), 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

$$\int_{-1}^1 g(x)dx \approx g\left(-\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}\right), \quad (4.1)$$

for the P^1 case, and the following three point Gaussian rule

$$\int_{-1}^1 g(x)dx \approx \frac{5}{9} \left[g\left(-\frac{3}{5}\right) + g\left(\frac{3}{5}\right) \right] + \frac{8}{9} g(0), \quad (4.2)$$

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:

$$\begin{aligned} &\approx \frac{1}{4} \left[g\left(-1, \frac{1}{\sqrt{3}}\right) + g\left(-1, -\frac{1}{\sqrt{3}}\right) \right. \\ &\quad + g\left(-\frac{1}{\sqrt{3}}, -1\right) + g\left(\frac{1}{\sqrt{3}}, -1\right) \\ &\quad + g\left(1, -\frac{1}{\sqrt{3}}\right) + g\left(1, \frac{1}{\sqrt{3}}\right) \\ &\quad \left. + g\left(\frac{1}{\sqrt{3}}, 1\right) + g\left(-\frac{1}{\sqrt{3}}, 1\right) \right] \\ &\quad + 2g(0,0). \end{aligned} \quad (4.3)$$

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 (4.5), 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 $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$:

$$u_h(x, y, t) = \bar{u}(t) + u_x(t)\phi_i(x) + u_y(t)\psi_j(y) \quad (4.4)$$

where

$$\phi_i(x) = \frac{x - x_i}{\Delta x_i/2}, \quad \psi_j(y) = \frac{y - y_j}{\Delta y_j/2}, \quad (4.5)$$

and

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \quad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}.$$

The degrees of freedoms, to be evolved in time, are then

$$\bar{u}(t), \quad u_x(t), \quad u_y(t).$$

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

$$1, \quad \phi_i(x), \quad \psi_j(y),$$

are orthogonal, hence the local mass matrix is diagonal:

$$M = \Delta x_i \Delta y_j \text{diag} \left(1, \frac{1}{3}, \frac{1}{3} \right).$$

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:

$$\begin{aligned} u_h(x, y, t) = & \bar{u}(t) + u_x(t)\phi_i(x) + u_y(t)\psi_j(y) \\ & + u_{xy}(t)\phi_i(x)\psi_j(y) \\ & + u_{xx}(t) \left(\phi_i^2(x) - \frac{1}{3} \right) \\ & + u_{yy}(t) \left(\psi_j^2(y) - \frac{1}{3} \right), \end{aligned} \quad (4.6)$$

where $\phi_i(x)$ and $\psi_j(y)$ are defined by (4.5). 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

$$1, \phi_i(x), \psi_j(y), \phi_i(x)\psi_j(y), \phi_i^2(x) - \frac{1}{3}, \psi_j^2(y) - \frac{1}{3},$$

are orthogonal, hence the local mass matrix is diagonal:

$$M = \Delta x_i \Delta y_j \text{diag} \left(1, \frac{1}{3}, \frac{1}{3}, \frac{1}{9}, \frac{4}{45}, \frac{4}{45} \right).$$

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

$$M = |K| \text{diag} \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right).$$

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 \mathcal{T}_h , we have:

$$\int_K \Lambda \Pi_h u_h = \int_K u_h.$$

3. Slope limiting: on each element K of \mathcal{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 [24] and [20].

The rectangular elements The limiting is performed on u_x and u_y in (4.4), using the differences of the means. For a scalar equation, u_x would be limited (replaced) by

$$\bar{m}(u_x, \bar{u}_{i+1,j} - \bar{u}_{ij}, \bar{u}_{ij} - \bar{u}_{i-1,j}) \quad (4.7)$$

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 [24]. 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 (4.7).

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} , $\bar{u}_{i+1,j} - \bar{u}_{ij}$ 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 (4.7) 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 $i = 0, 1, 2, 3$.

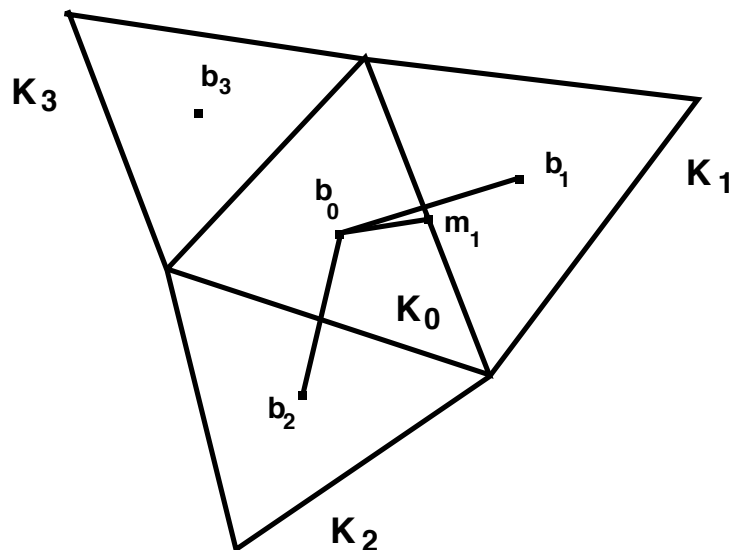


Fig. 4.1. Illustration of limiting.

Since we have that

$$m_1 - b_0 = \alpha_1 (b_1 - b_0) + \alpha_2 (b_2 - b_0),$$

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 ,

$$u_h(m_1) - u_h(b_0) = \alpha_1 (u_h(b_1) - u_h(b_0)) + \alpha_2 (u_h(b_2) - u_h(b_0)),$$

and since

$$\bar{u}_{K_i} = \frac{1}{|K_i|} \int_{K_i} u_h = u_h(b_i), \quad i = 0, 1, 2, 3,$$

we have that

$$\begin{aligned} \tilde{u}_h(m_1, K_0) &\equiv u_h(m_1) - \bar{u}_{K_0} \\ &= \alpha_1 (\bar{u}_{K_1} - \bar{u}_{K_0}) + \alpha_2 (\bar{u}_{K_2} - \bar{u}_{K_0}) \\ &\equiv \Delta \bar{u}(m_1, K_0). \end{aligned}$$

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 $(x, y) \in K_0$,

$$\begin{aligned} u_h(x, y) &= \sum_{i=1}^3 u_h(m_i) \varphi_i(x, y) \\ &= \bar{u}_{K_0} + \sum_{i=1}^3 \tilde{u}_h(m_i, K_0) \varphi_i(x, y). \end{aligned}$$

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

$$pos = \sum_{i=1}^3 \max(0, \Delta_i), \quad neg = \sum_{i=1}^3 \max(0, -\Delta_i),$$

and set

$$\theta^+ = \min\left(1, \frac{neg}{pos}\right), \quad \theta^- = \min\left(1, \frac{pos}{neg}\right).$$

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 [92] and later by many others. We use exactly the same setup as in [92], 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 [92]. 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 [92], 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, we 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 [92] 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 [92] 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 erroneous 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.

To have a better idea of the nature of the singularity at the corner, we display the values of the density and the entropy along the line $y = 0.2$; note that the corner is located on this line at $x = 0.6$. In Figure 4.9, we show the results obtained with P^1 and in Figure 4.10, the results obtained with P^2 . At the corner ($x = 0.6$), we can see that there is a jump both in the entropy and in the density. As the meshsize decreases, the jump in the entropy does not vary significantly; however, the jump in the density does. The sharp decrease in the density right after the corner can be interpreted as a cavitation effect that the scheme seems to be able to better approximate as the meshsize decreases.

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 4.11. In Figure 4.12, 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 4.13, 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 4.14 and 4.15 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 [4] in two dimensions and Warburton, Lomtev, Kirby and Karniadakis [91] in three dimensions.

The purpose of the numerical results of Bassi and Rebay [4] 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 4.16, 4.17, 4.18, and 4.19, 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 4.20, 4.21, 4.22, and 4.23. In these pictures the approximate solutions with piecewise 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 4.20 and 4.21. In [4], Bassi and Rebay show that the RKDG method using polynomials 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 [91], Warburton, Lomtev, Kirby and Karniadakis present the same test problem in a three dimensional setting. In Figure 4.24, 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 exponentially 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.

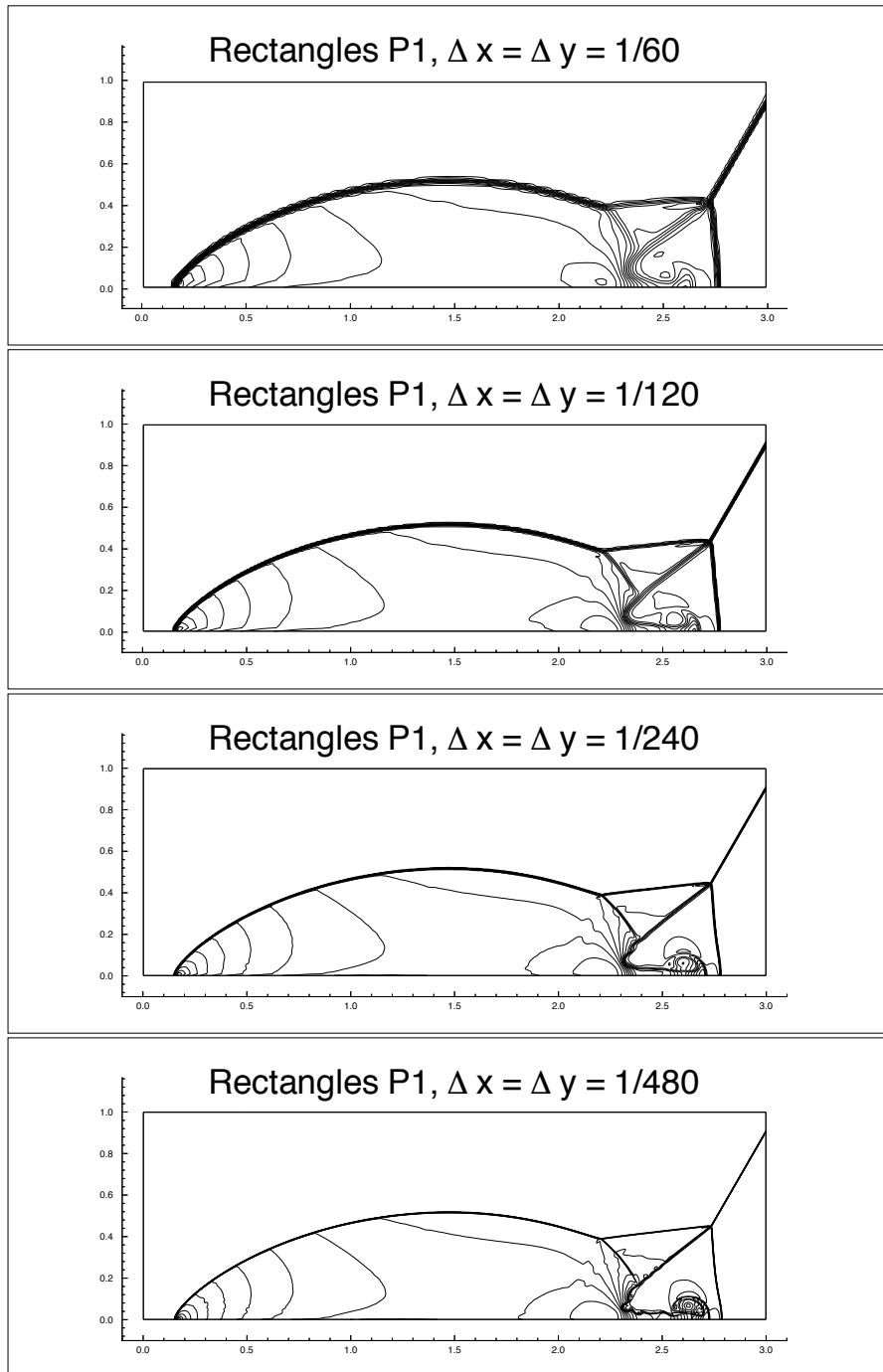


Fig. 4.2. Double Mach reflection problem. Second order P^1 results. Density ρ . 30 equally spaced contour lines from $\rho = 1.3965$ to $\rho = 22.682$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{60}$, $\frac{1}{120}$, $\frac{1}{240}$, and $\frac{1}{480}$.

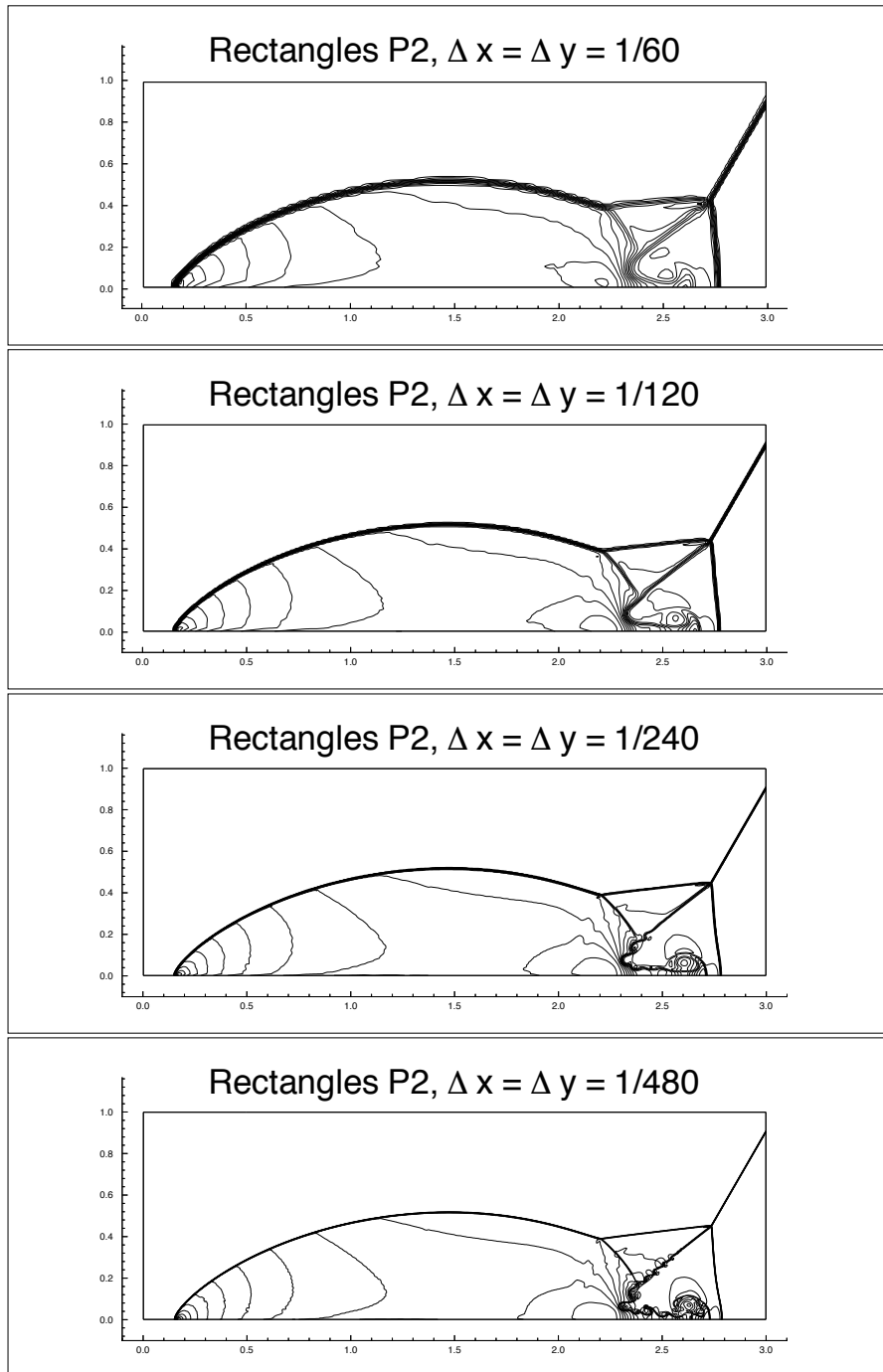


Fig. 4.3. Double Mach reflection problem. Third order P^2 results. Density ρ . 30 equally spaced contour lines from $\rho = 1.3965$ to $\rho = 22.682$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{60}$, $\frac{1}{120}$, $\frac{1}{240}$, and $\frac{1}{480}$.

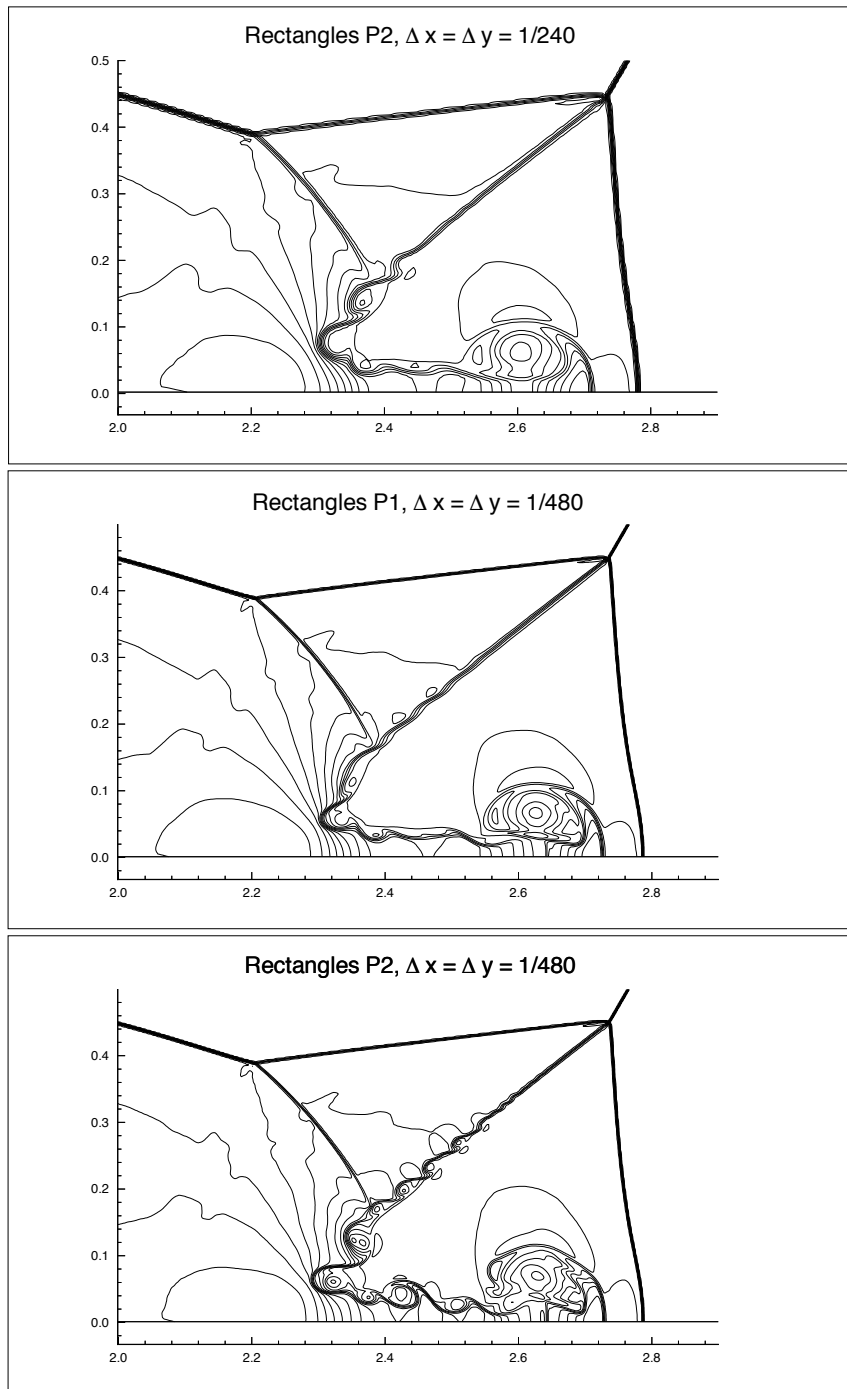


Fig. 4.4. Double Mach reflection problem. Blowed-up region around the double Mach stems. Density ρ . Third order P^2 with $\Delta x = \Delta y = \frac{1}{240}$ (top); second order P^1 with $\Delta x = \Delta y = \frac{1}{480}$ (middle); and third order P^2 with $\Delta x = \Delta y = \frac{1}{480}$ (bottom).

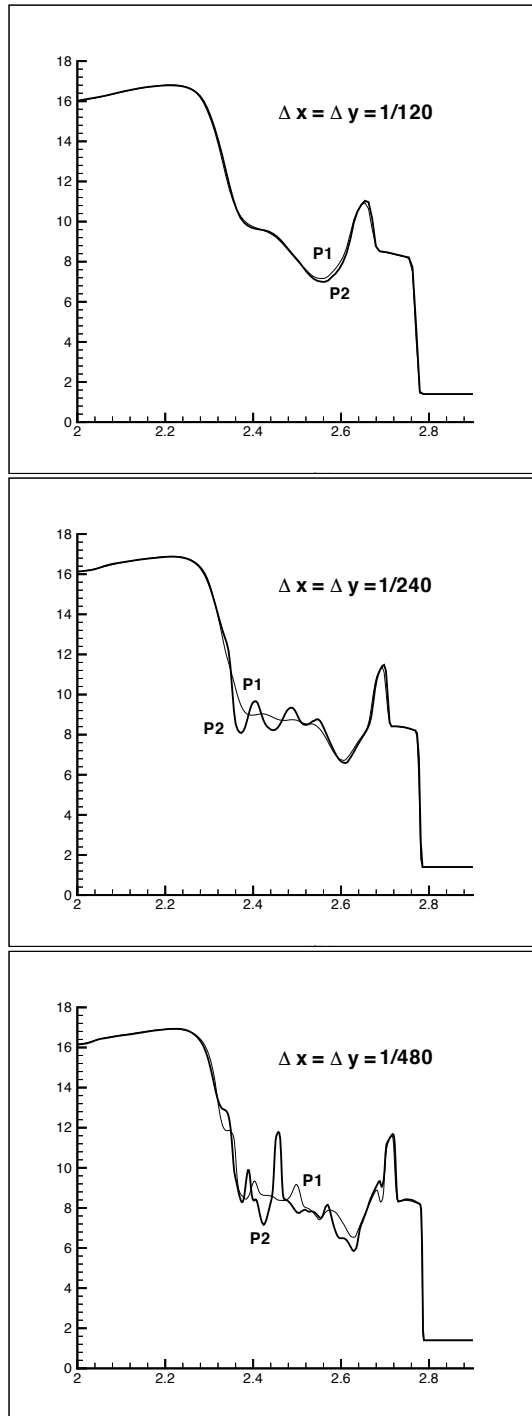


Fig. 4.5. Double Mach reflection problem. Cut at $y = 0.04$ of the blown-up region. Density ρ . Comparison of second order P^1 with third order P^2 on the same mesh

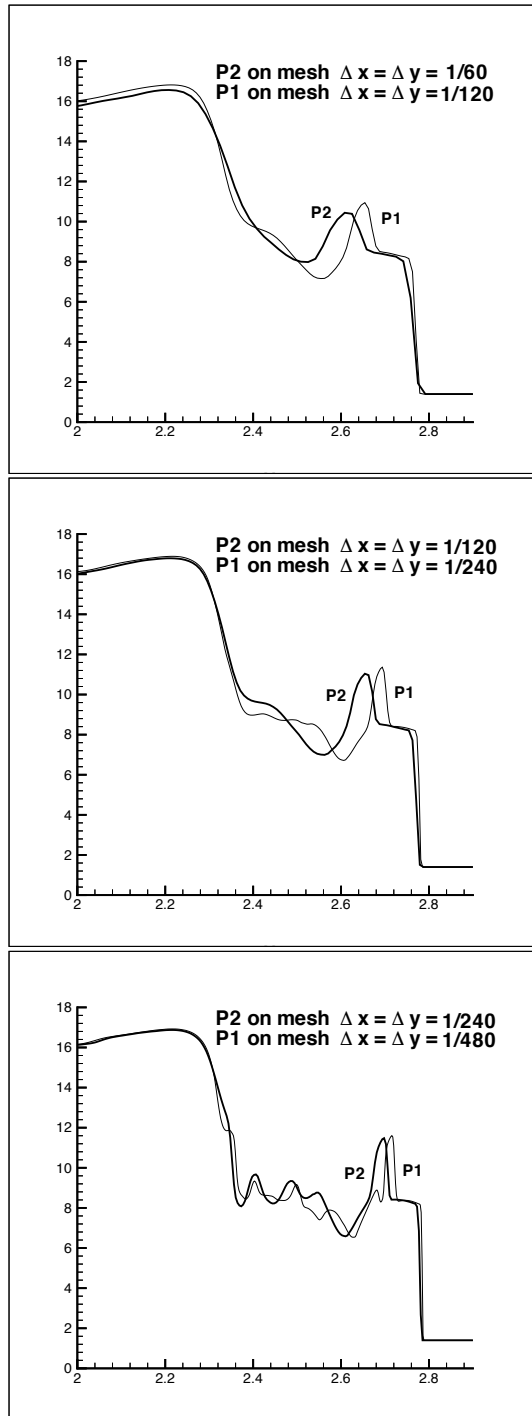


Fig. 4.6. Double Mach reflection problem. Cut at $y = 0.04$ of the blown-up region. Density ρ . Comparison of second order P^1 with third order P^2 on a coarser mesh

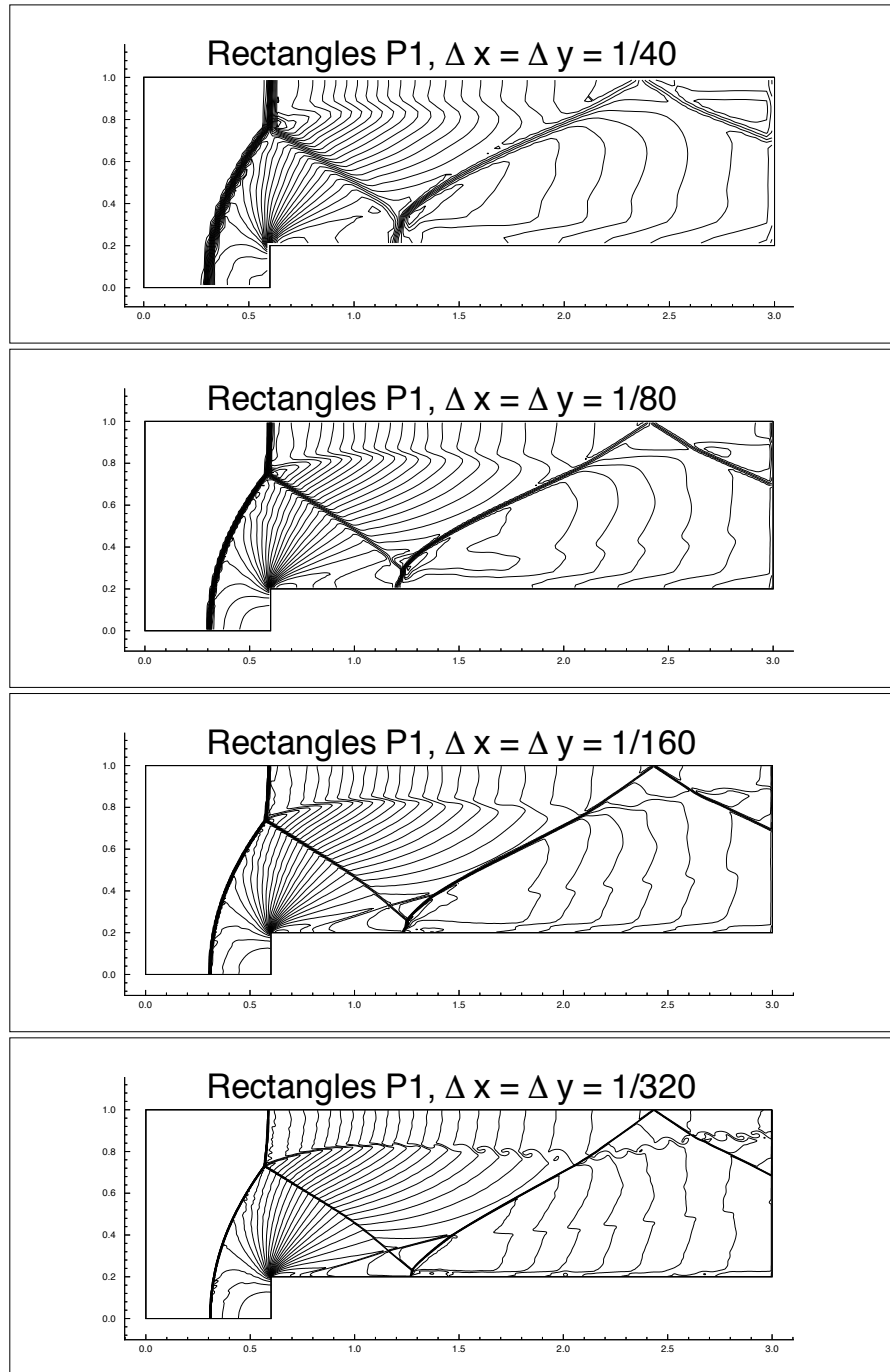


Fig. 4.7. Forward facing step problem. Second order P^1 results. Density ρ . 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}$, $\frac{1}{80}$, $\frac{1}{160}$, and $\frac{1}{320}$.

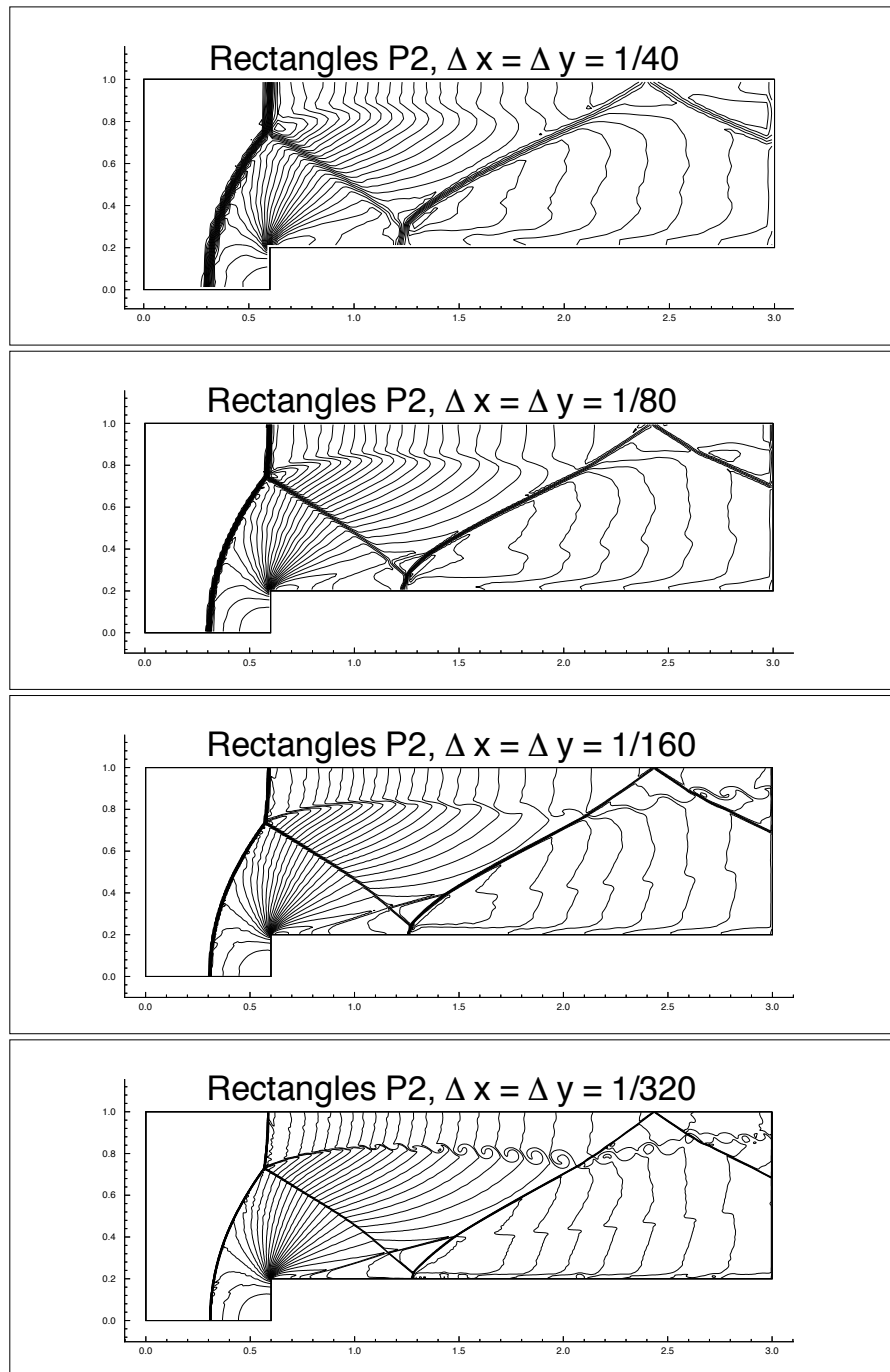


Fig. 4.8. Forward facing step problem. Third order P^2 results. Density ρ . 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}$, $\frac{1}{80}$, $\frac{1}{160}$, and $\frac{1}{320}$.

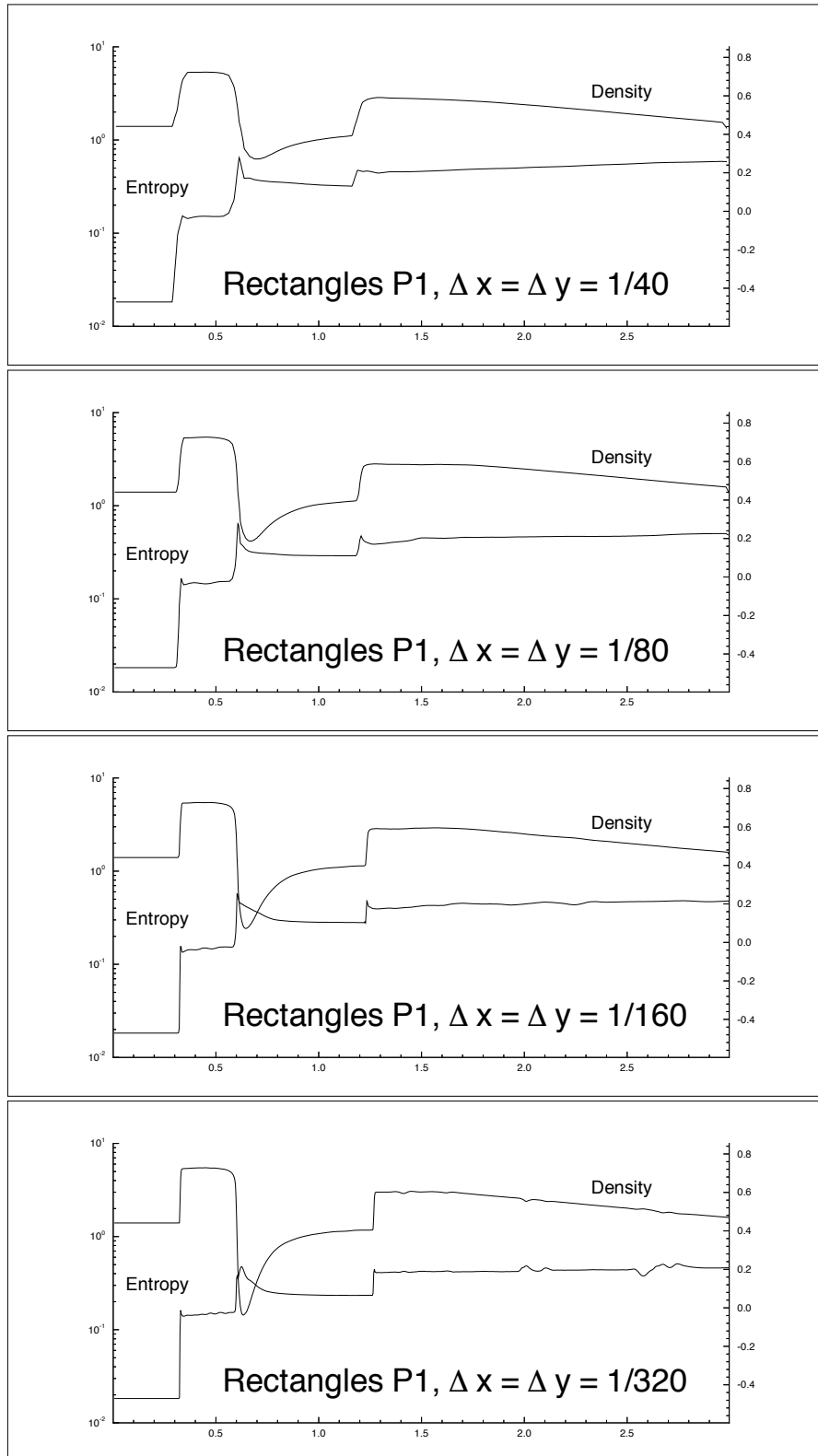


Fig. 4.9. Forward facing step problem. Second order P^1 results. Values of the density and entropy along the line $y = .2$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}$, $\frac{1}{80}$, $\frac{1}{160}$, and $\frac{1}{320}$.

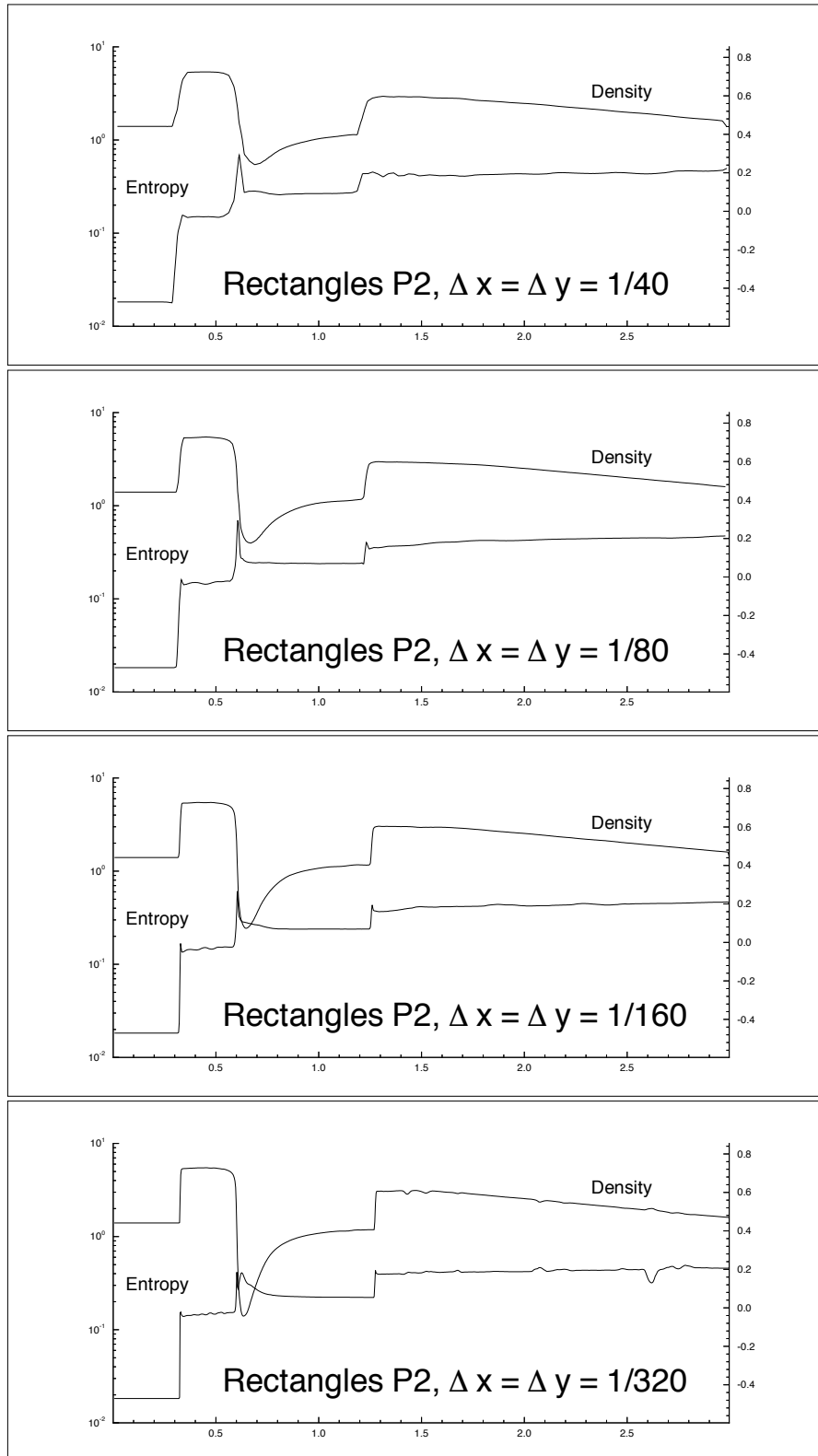


Fig. 4.10. Forward facing step problem. Third order P^2 results. Values of the density and entropy along the line $y = .2$. Mesh refinement study. From top to bottom: $\Delta x = \Delta y = \frac{1}{40}, \frac{1}{80}, \frac{1}{160},$ and $\frac{1}{320}$.

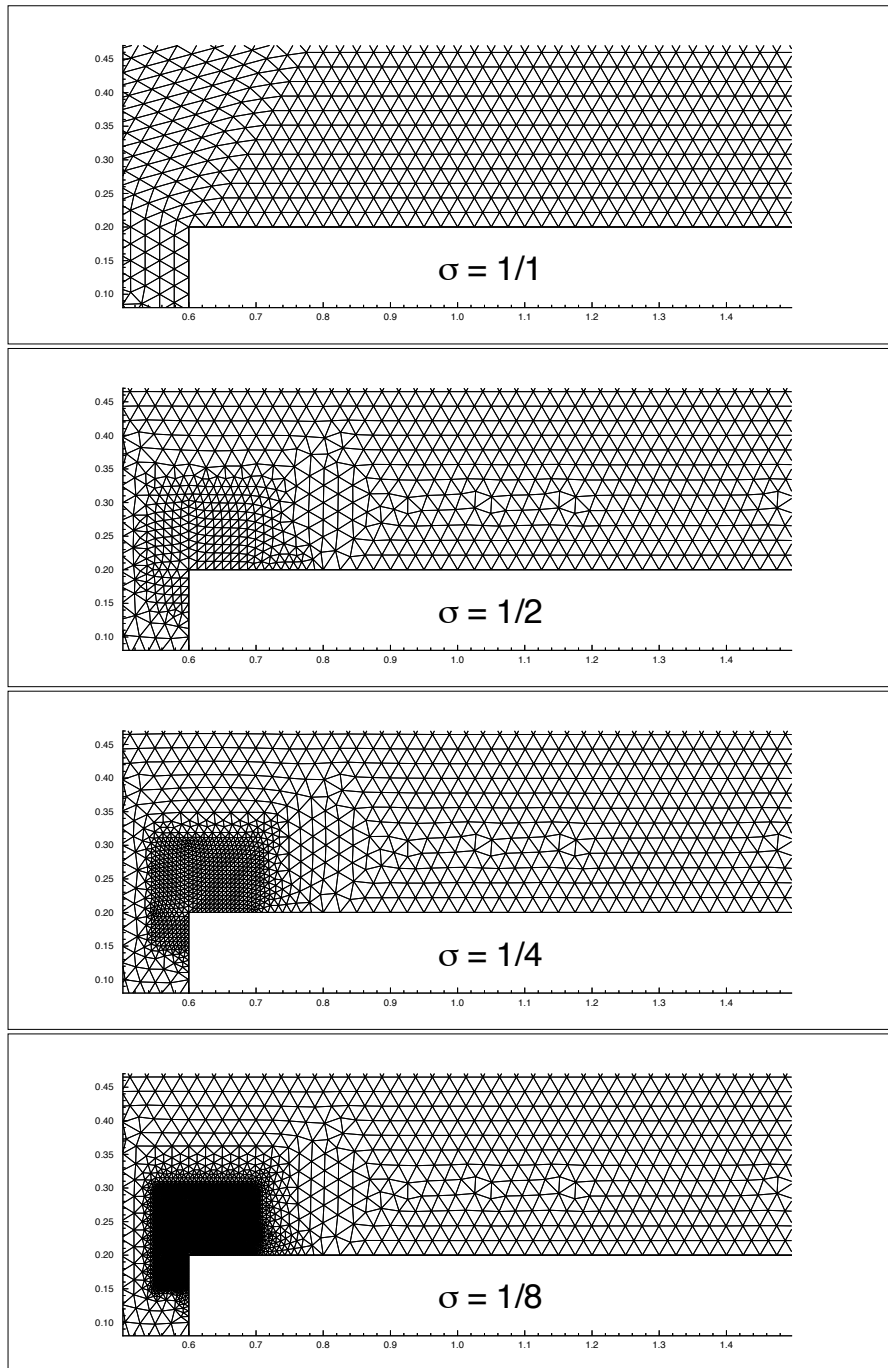


Fig. 4.11. Forward facing step problem. Detail of the triangulations associated with the different values of σ . The parameter σ is the ratio between the typical size of the triangles near the corner and that elsewhere.

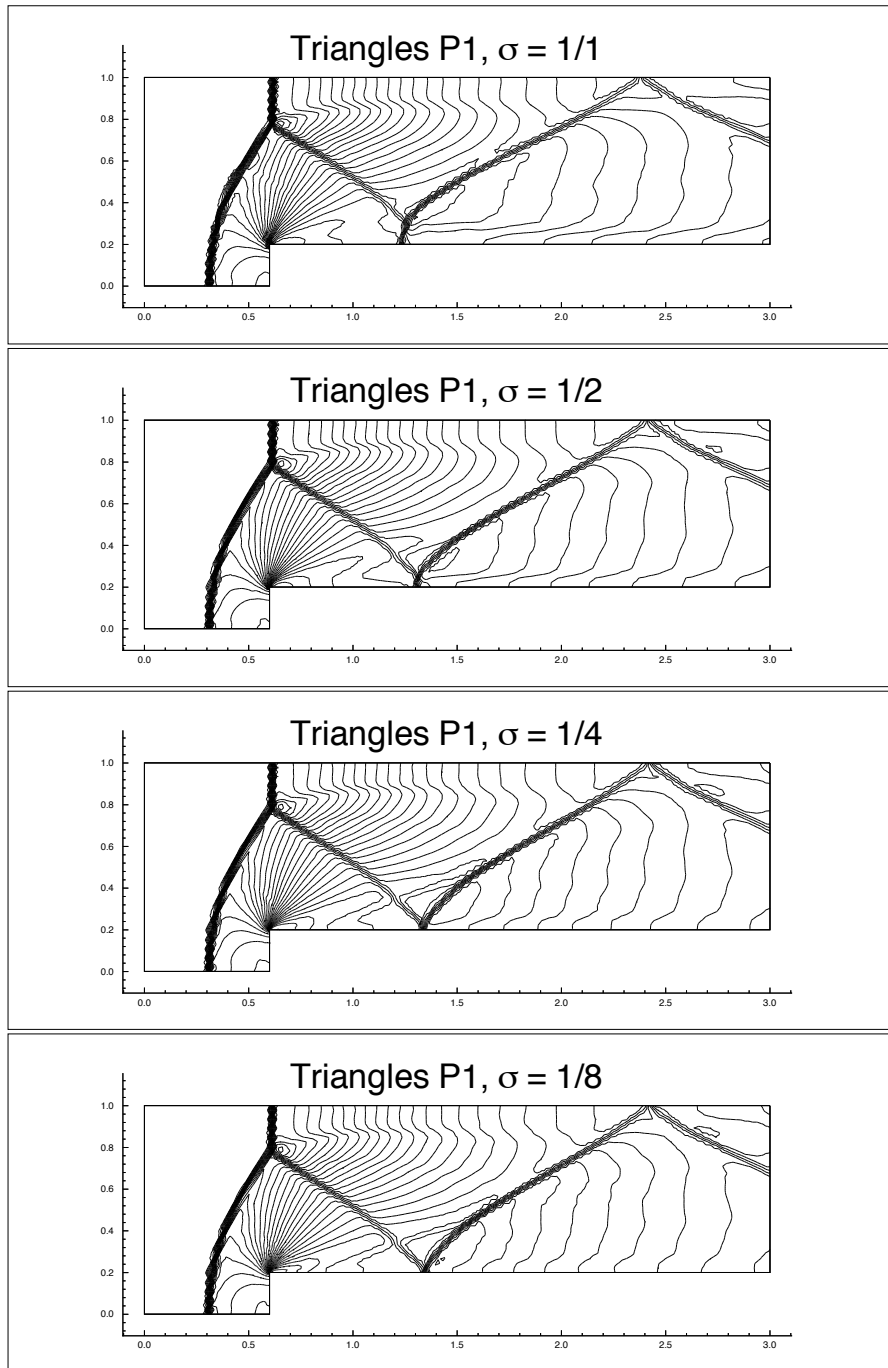


Fig. 4.12. Forward facing step problem. Second order P^1 results. Density ρ . 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Triangle code. Progressive refinement near the corner

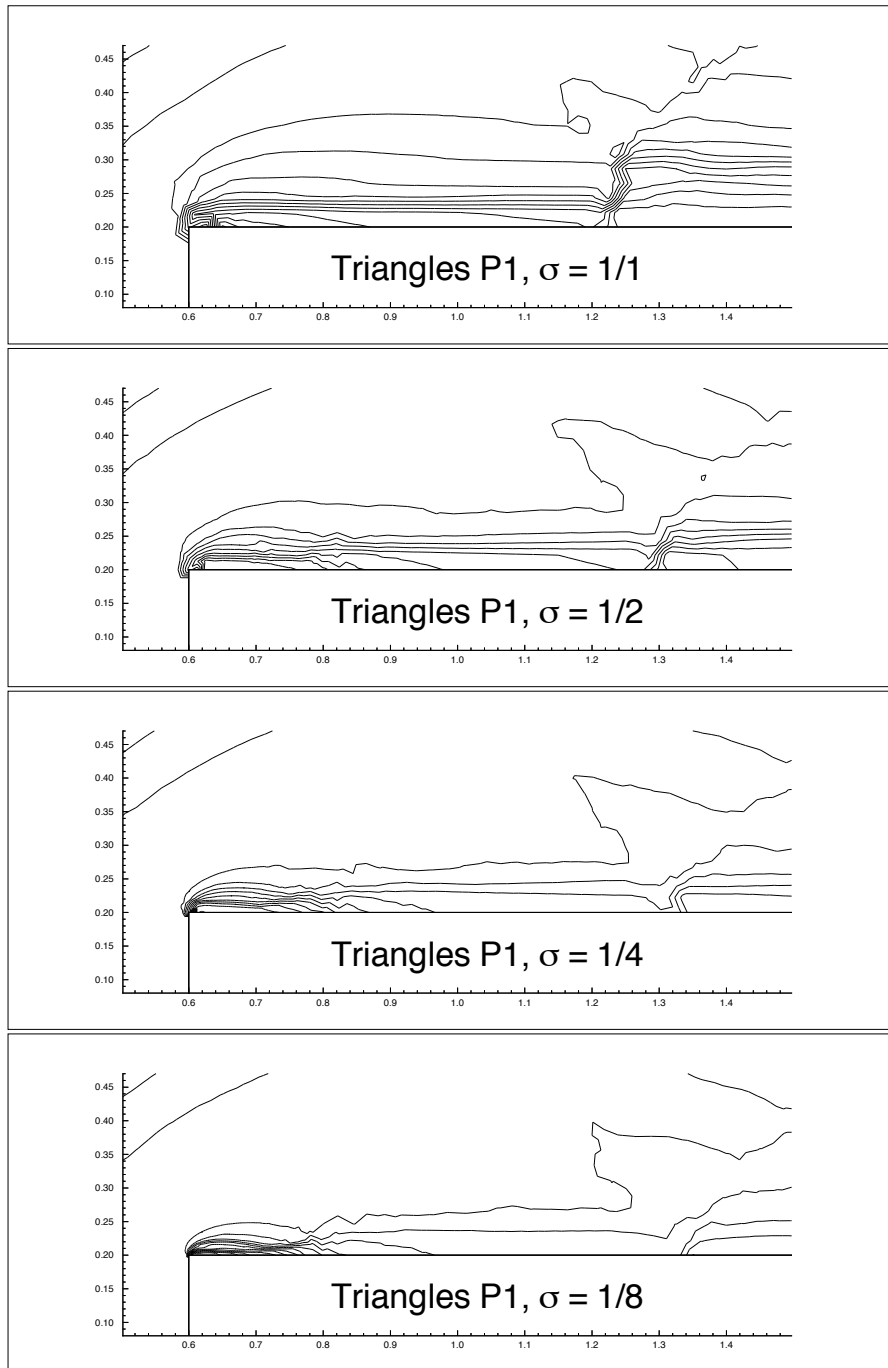


Fig. 4.13. Forward facing step problem. Second order P^1 results. Entropy level curves around the corner. Triangle code. Progressive refinement near the corner

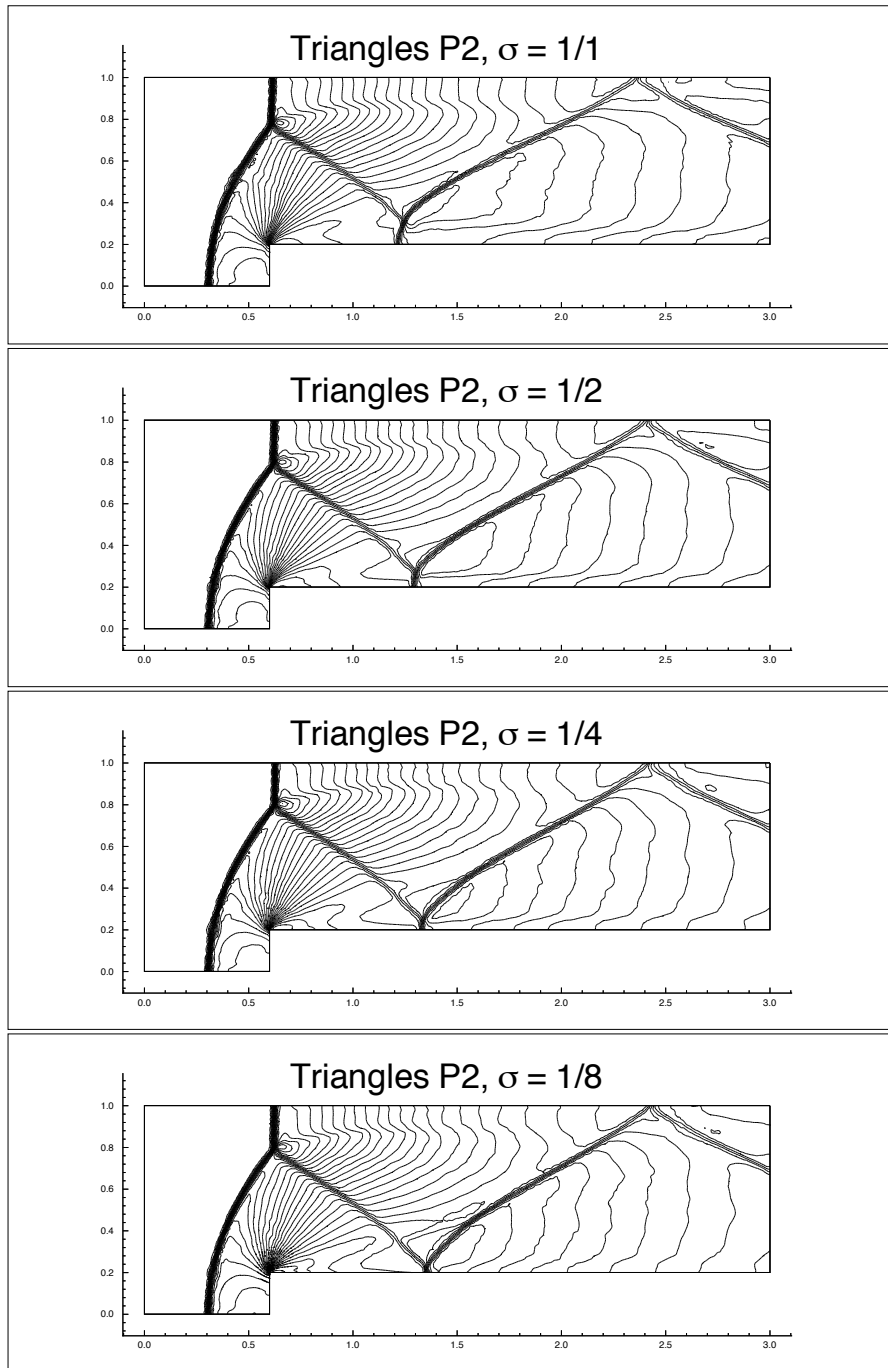


Fig. 4.14. Forward facing step problem. Third order P^2 results. Density ρ . 30 equally spaced contour lines from $\rho = 0.090338$ to $\rho = 6.2365$. Triangle code. Progressive refinement near the corner

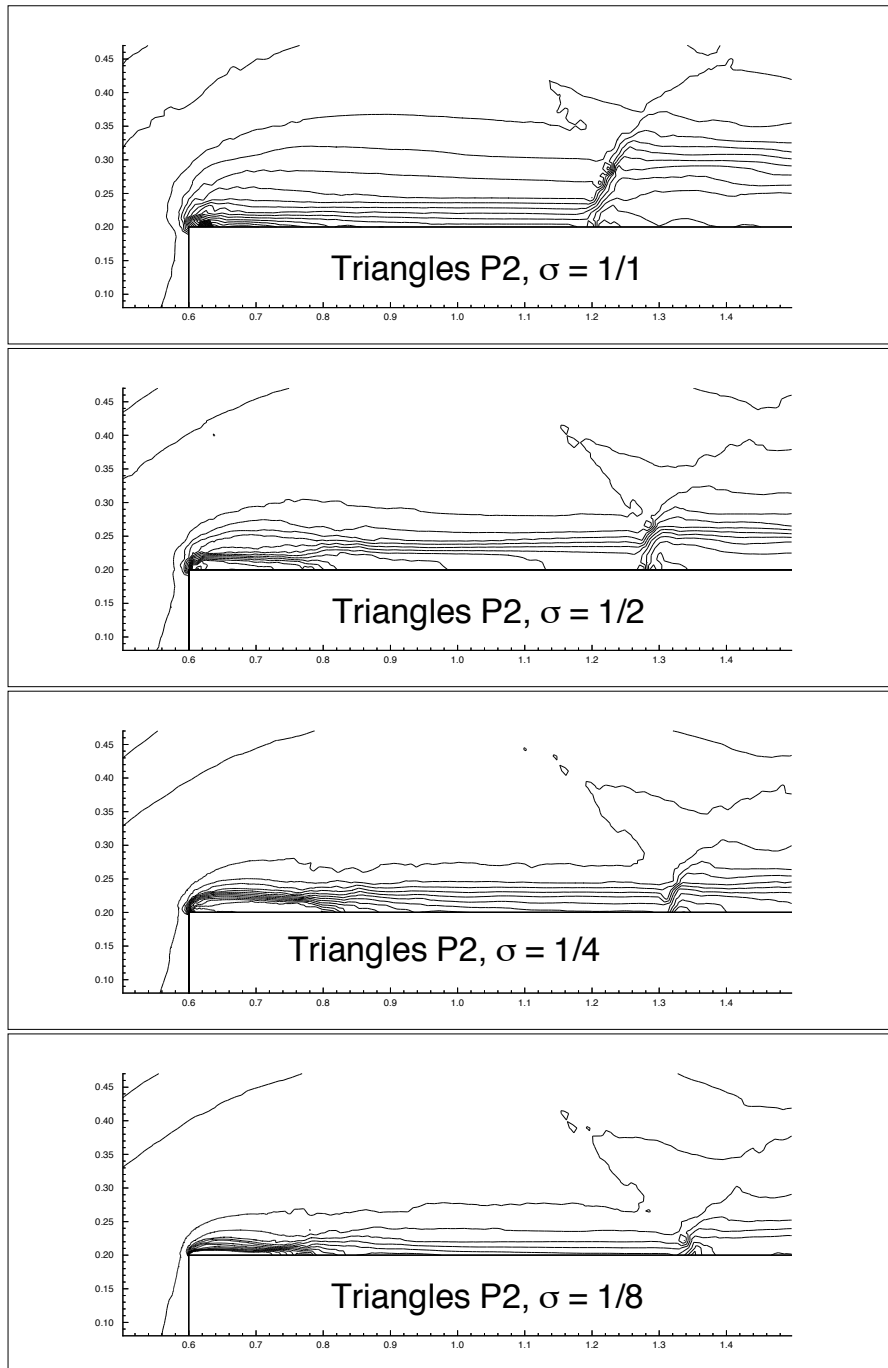


Fig. 4.15. Forward facing step problem. Third order P^1 results. Entropy level curves around the corner. Triangle code. Progressive refinement near the corner

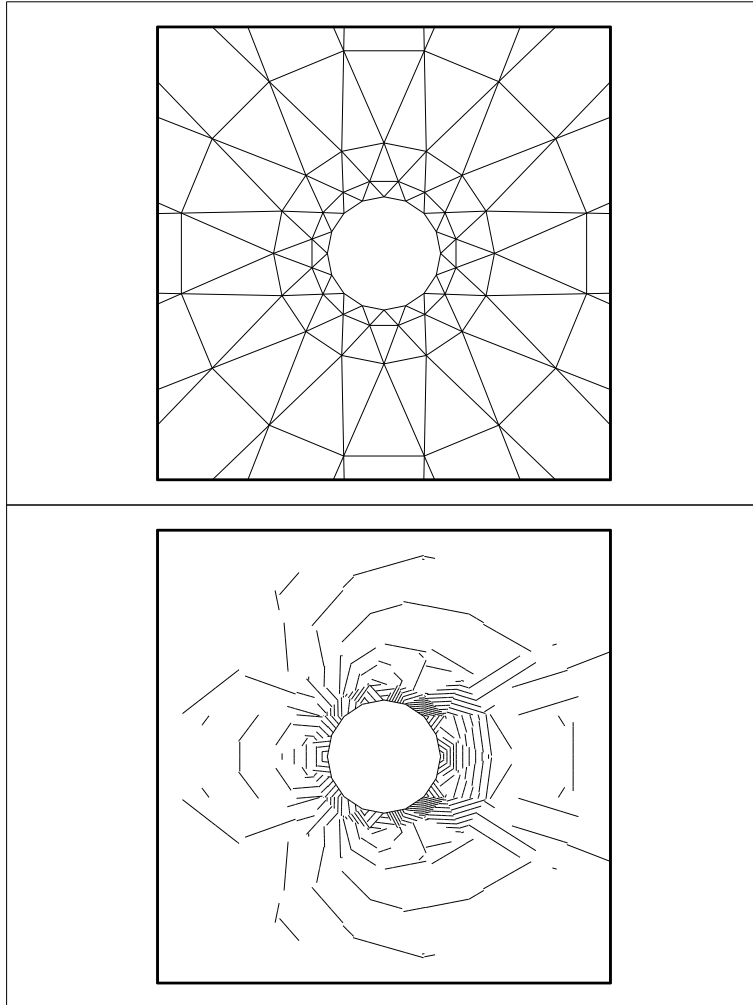


Fig. 4.16. Grid “ 16×8 ” with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

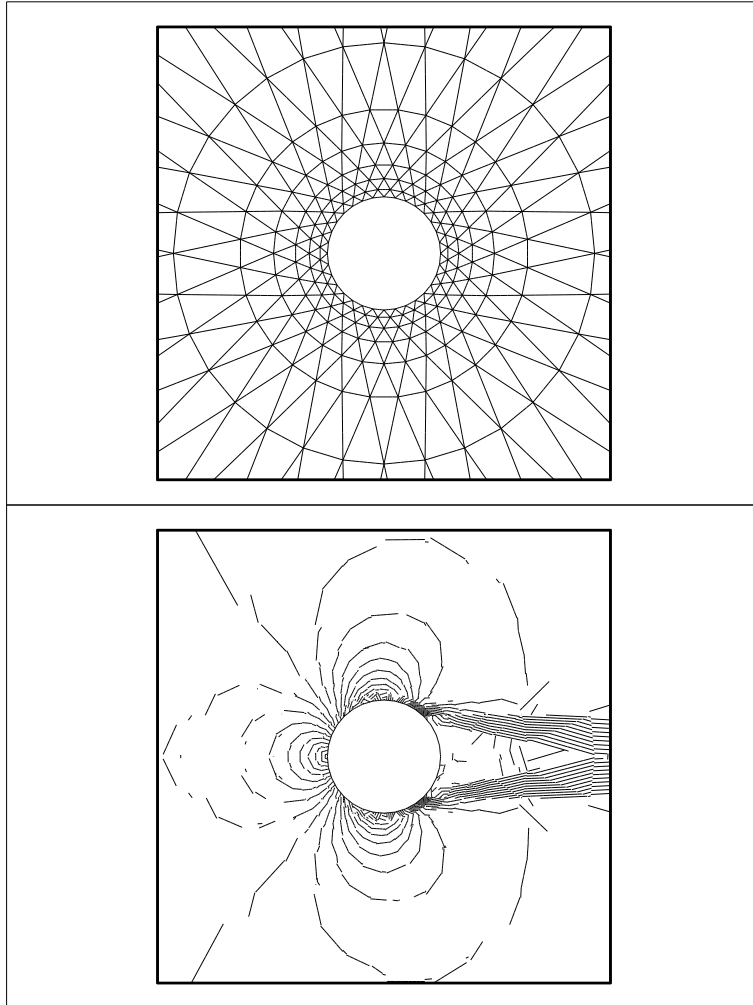


Fig. 4.17. Grid “ 32×8 ” with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

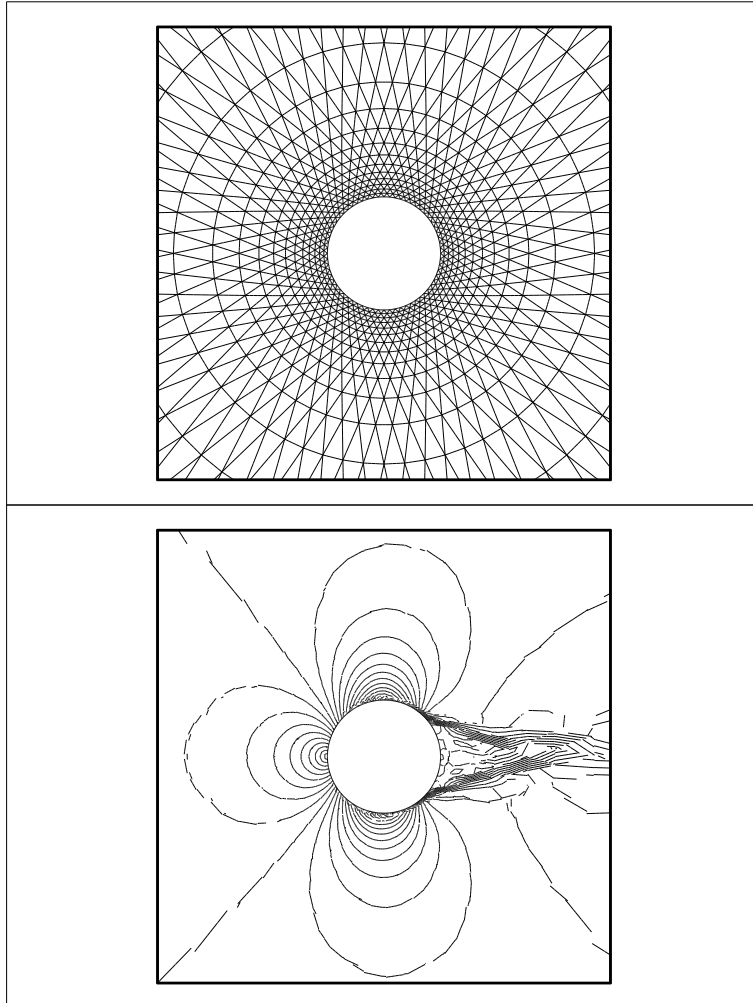


Fig. 4.18. Grid “ 64×16 ” with a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

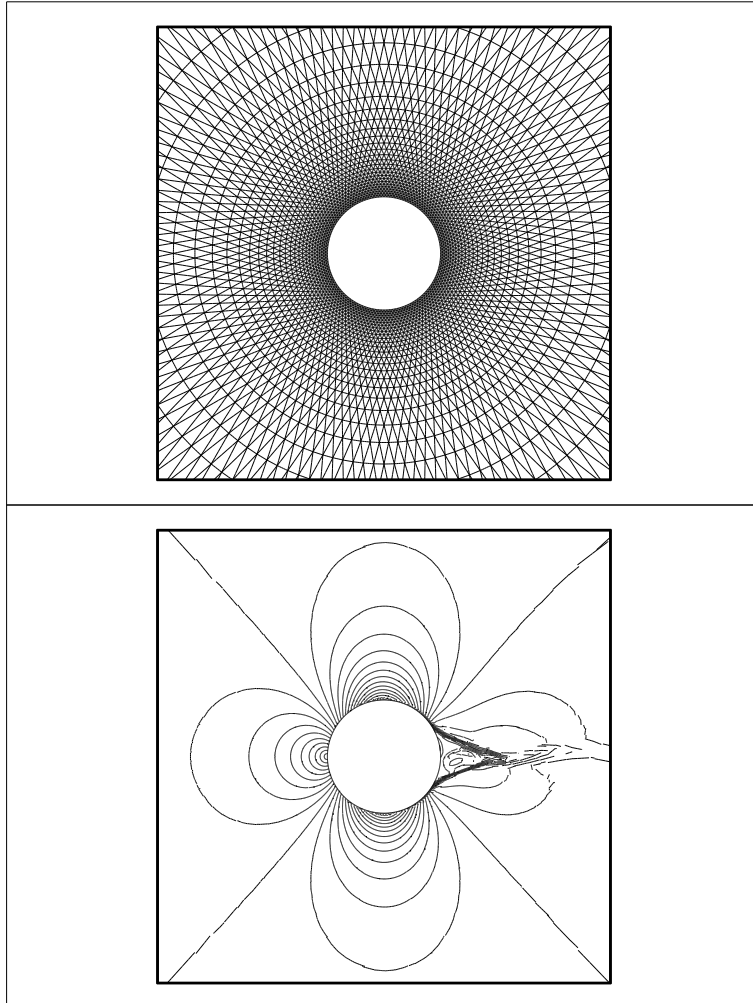


Fig. 4.19. Grid “ 128×32 ” a piecewise linear approximation of the circle (top) and the corresponding solution (Mach isolines) using P^1 elements (bottom).

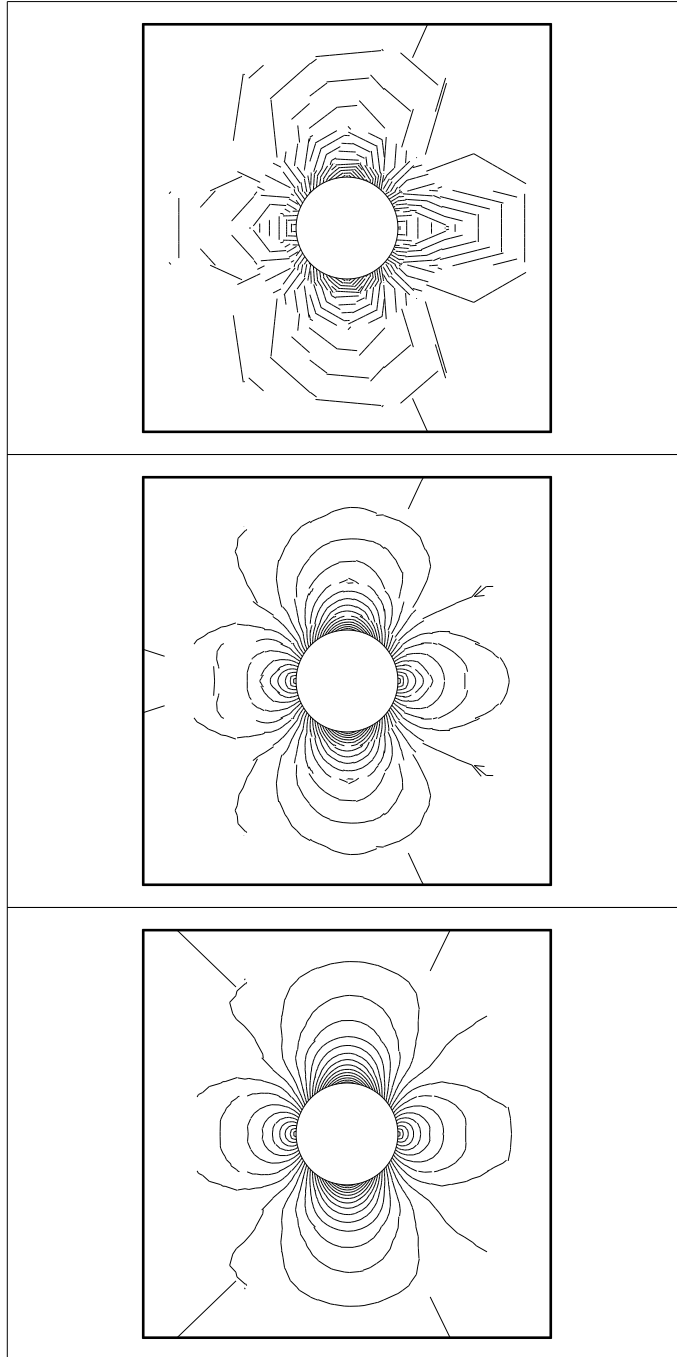


Fig. 4.20. Grid “ 16×4 ” with exact rendering of the circle and the corresponding P^1 (top), P^2 (middle), and P^3 (bottom) approximations (Mach isolines).

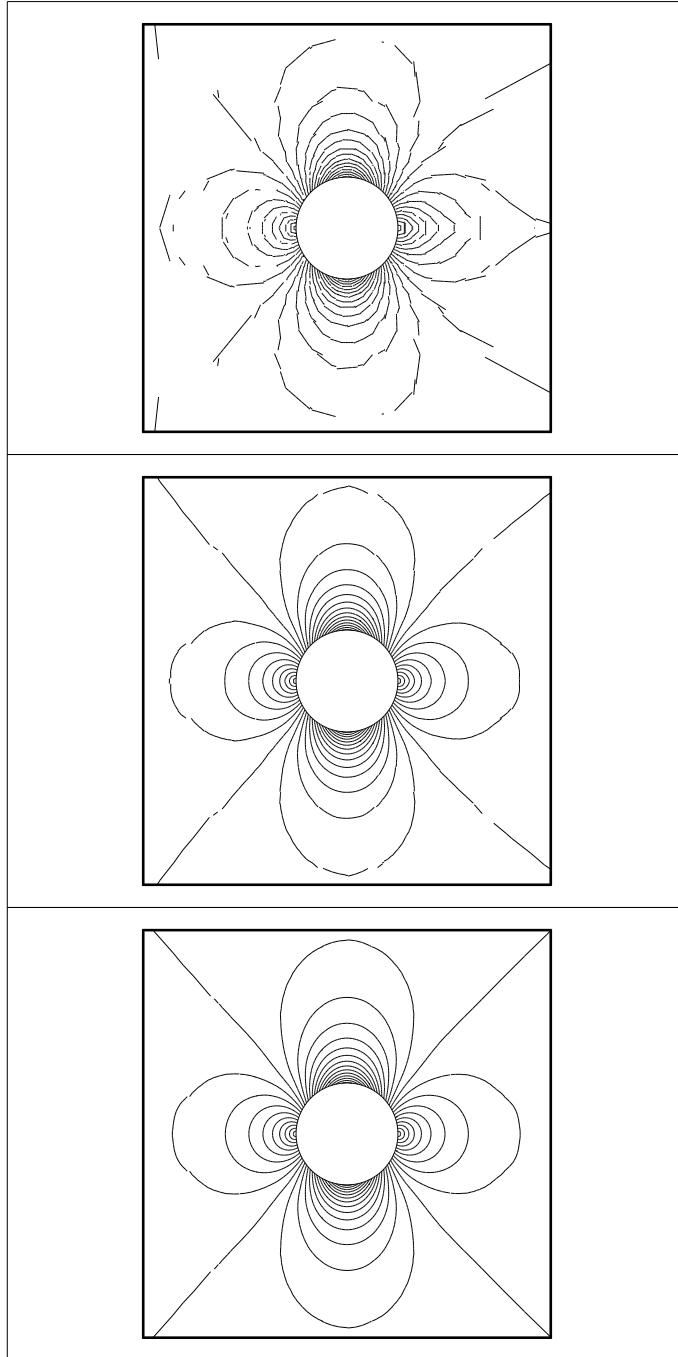


Fig. 4.21. Grid “ 32×8 ” with exact rendering of the circle and the corresponding P^1 (top), P^2 (middle), and P^3 (bottom) approximations (Mach isolines).

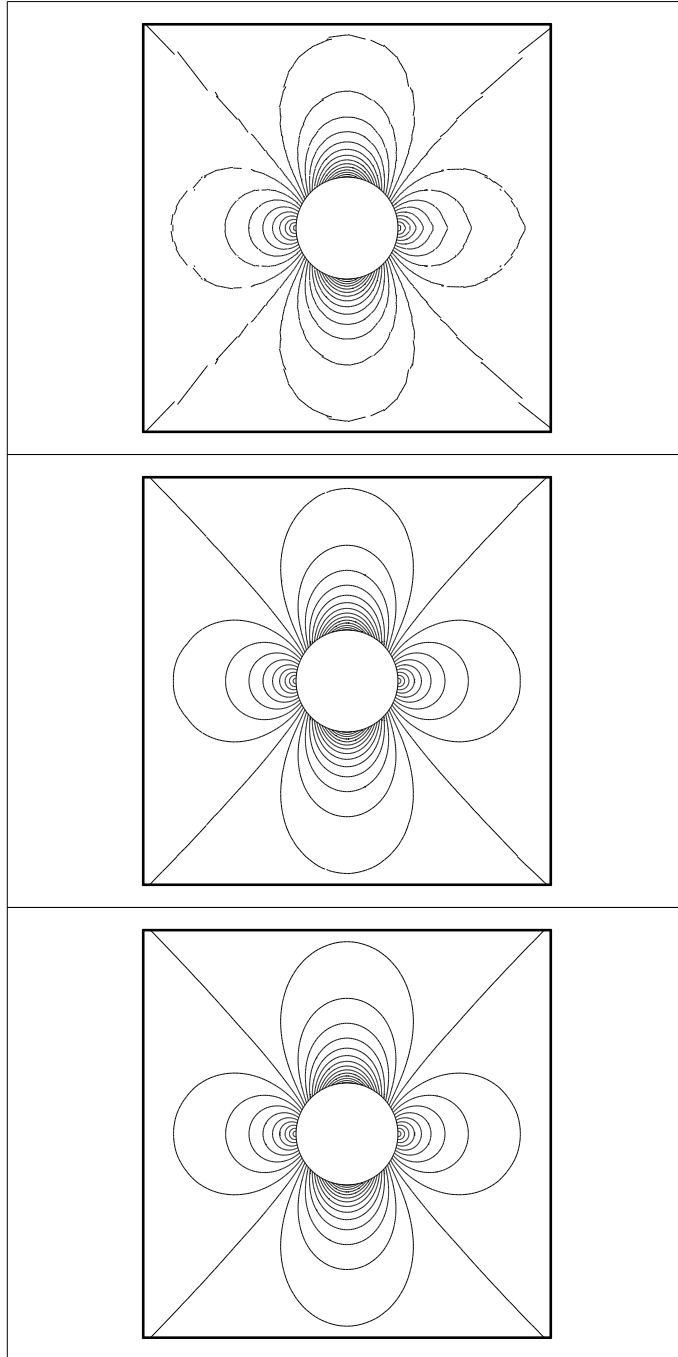


Fig. 4.22. Grid “64 × 16” with exact rendering of the circle and the corresponding P^1 (top), P^2 (middle), and P^3 (bottom) approximations (Mach isolines).

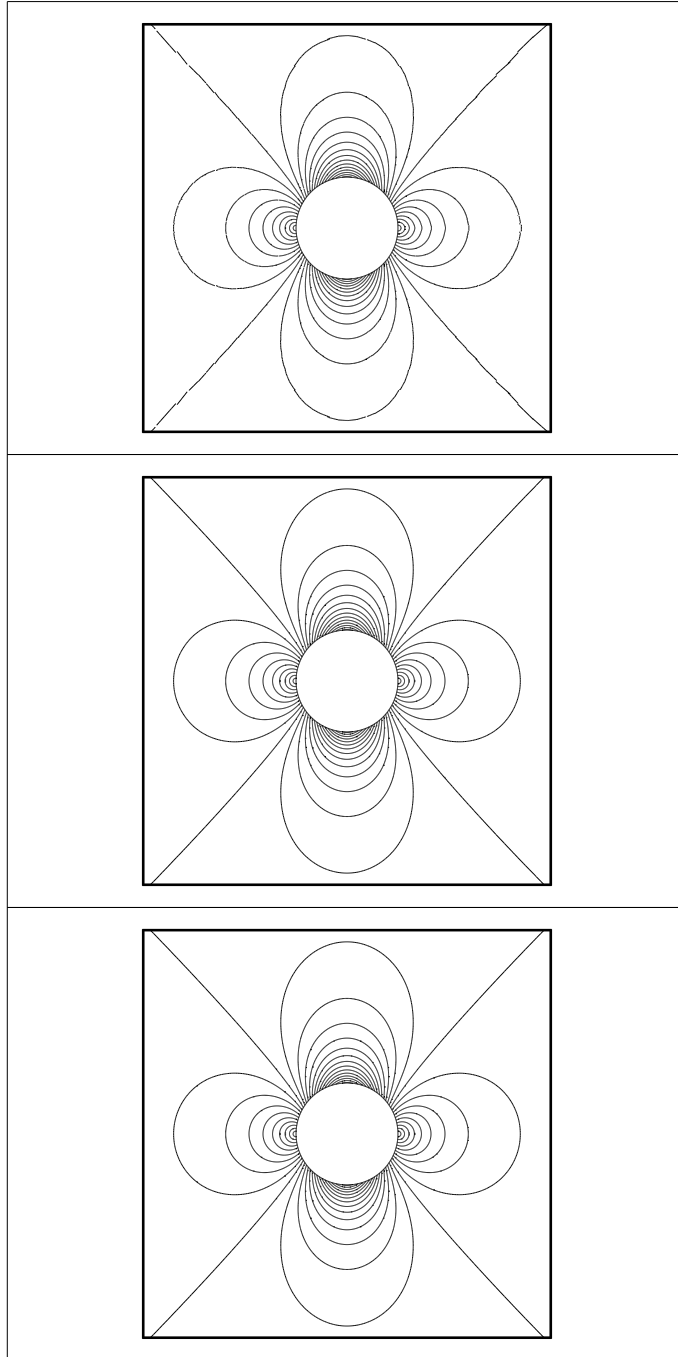


Fig. 4.23. Grid “128 × 32” with exact rendering of the circle and the corresponding P^1 (top), P^2 (middle), and P^3 (bottom) approximations (Mach isolines).

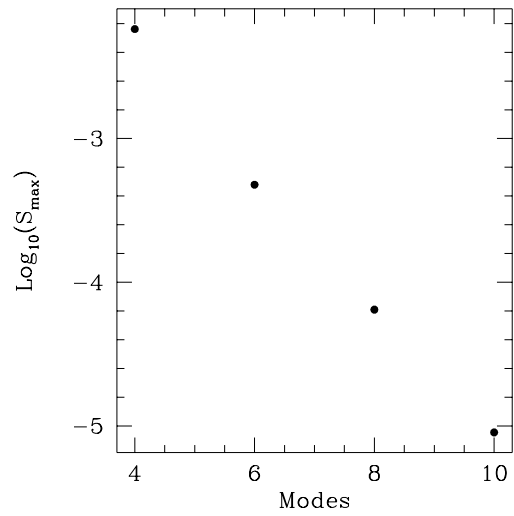
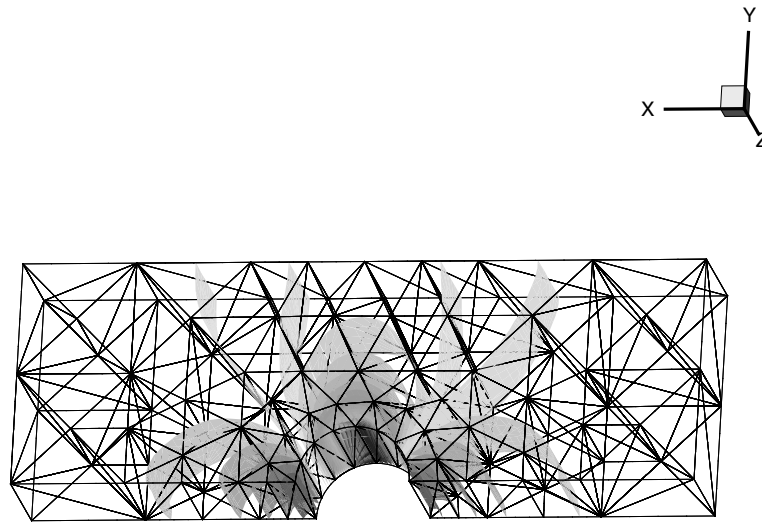


Fig. 4.24. Three-dimensional flow over a semicircular bump. Mesh and density isosurfaces (top) and history of convergence with p -refinement of the maximum entropy generated (bottom). The degree of the polynomial plus one is plotted on the 'modes' axis.

5 The Hamilton-Jacobi equations in several space dimensions

5.1 Introduction

In this chapter, we consider the RKDG method for multidimensional Hamilton-Jacobi equations. The model problem we consider is the following:

$$\varphi_t + H(\varphi_x, \varphi_y) = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (5.1)$$

$$\varphi(x, y, 0) = \varphi_0(x, y), \quad \forall (x, y) \in (0, 1)^2, \quad (5.2)$$

where we take periodic boundary conditions. The material in this section is based in the work of Hu and Shu [43].

5.2 The RKDG method

As in the one-dimensional case, the main idea to extend the RKDG method to this case, is to realize that $u = \varphi_x$ and $v = \varphi_y$ satisfy the following problem:

$$u_t + H(u, v)_x = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (5.3)$$

$$v_t + H(u, v)_y = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (5.4)$$

$$u(x, y, 0) = (\varphi_0)_x(x, y), \quad \forall (x, y) \in (0, 1)^2, \quad (5.5)$$

$$v(x, y, 0) = (\varphi_0)_y(x, y), \quad \forall (x, y) \in (0, 1)^2, \quad (5.6)$$

and that φ can be computed from u and v by solving the following problem:

$$\varphi_t = -H(u, v), \quad \text{in } (0, 1) \times (0, T), \quad (5.7)$$

$$\varphi(x, 0) = \varphi_0(x), \quad \forall x \in (0, 1). \quad (5.8)$$

Again, a straightforward application of the RKDG method to (5.3), (5.5), produces an approximation u_h to $u = \varphi_x$; and a straightforward application of the RKDG method to (5.4), (5.6), produces an approximation v_h to $v = \varphi_y$. Both u_h and v_h are taken to be piecewise polynomials of degree $(k-1)$. Then, φ_h is computed in one of the following ways:

(i) Take $\varphi_h(\cdot, t)$ in V_h^k such that

$$\forall K \in \mathcal{T}_h, \quad w_h \in P^k(K) :$$

$$\int_K \partial_t \varphi_h(x, y, t) w_h(x, y) dx dy = - \int_K H(u_h(x, t), v_h(x, y)) w_h(x, y) dx dy$$

$$\int_K \varphi_h(x, y, 0) w_h(x, y) dx dy = \int_K \varphi_0(x, y) w_h(x, y) dx dy.$$

(ii) Take $\varphi_h(\cdot, t)$ in V_h^k such that, $\forall K \in \mathcal{T}_h$:

$$\|\nabla \varphi_h - (u_h, v_h)\|_{L^2(K)} = \min_{\psi \in P^k(K)} \|\nabla \psi - (u_h, v_h)\|_{L^2(K)}.$$

This determines φ_h up to a constant. To find this constant, we impose the following condition:

$$\begin{aligned} \forall K \in \mathcal{T}_h : \\ \frac{d}{dt} \int_K \varphi_h(x, y, t) dx dy &= - \int_K H(u_h(x, y, t), v_h(x, y, t)) dx dy, \\ \int_K \varphi(x, y, 0) dx dy &= \int_K \varphi_0(x, y) dx dy. \end{aligned}$$

(iii) Pick the element K_J and determine the values of φ_h on it in such a way that

$$\|\nabla \varphi_h - (u_h, v_h)\|_{L^2(K_J)} = \min_{\psi \in P^k(K)} \|\nabla \psi - (u_h, v_h)\|_{L^2(K_J)}.$$

and that

$$\begin{aligned} \frac{d}{dt} \int_{K_J} \varphi_h(x, y, t) dx dy &= - \int_{K_J} H(u_h(x, y, t)) dx dy, \\ \int_{K_J} \varphi(x, y, 0) dx dy &= \int_{K_J} \varphi_0(x, y) dx dy. \end{aligned} \tag{5.9}$$

Then, compute φ_h as follows:

$$\varphi(B, t) = \varphi(A, t) + \int_A^B (\varphi_x dx + \varphi_y dy). \tag{5.10}$$

to determine the missing constant. The path should be taken to avoid crossing a derivative discontinuity, if possible.

We remark again that, in the third approach, the recovered values of φ_h depend on the choice of the starting point A as well as the integration path. However this difference is on the level of truncation errors and does not affect the order of accuracy as is shown in the computational results we show next.

5.3 Computational results

The purpose of the numerical experiments we report in this section is to assess the accuracy of the method, to see if the generalized slope limiter is actually needed, and to evaluate the effect of changing the integration path. The third approach is used.

First test problem. Two dimensional Burgers' equation:

$$\begin{aligned} \varphi_t + \frac{(\varphi_x + \varphi_y + 1)^2}{2} &= 0, & \text{in } (-2, 2)^2 \times (0, T), \\ \varphi(x, y, 0) &= -\cos\left(\frac{\pi(x+y)}{2}\right), & \forall (x, y) \in (-2, 2)^2, \end{aligned}$$

with periodic boundary conditions.

We first use uniform rectangular meshes and the local Lax-Friedrichs flux. At $T = 0.5/\pi^2$, the solution is still smooth. The errors (computed at the center of the cells) and orders of accuracy are listed in Table 5.1. It seems that only k -th order of accuracy is achieved when φ is a piecewise polynomial of degree k . Next, as in the one dimensional case, we use non-uniform rectangular meshes obtained from the tensor product of one dimensional nonuniform meshes (the meshes in two directions are independent). Again, we give the “real” L^2 -errors computed by a 6×6 point Gaussian quadrature as well. The results are shown in Table 5.2.

The results in Tables 5.1 and 5.1 are obtained by updating the element at the left-lower corner with time, and then taking an integration path consisting of line segments starting from the corner and parallel to the x -axis first, then vertically to the point. To further address the issue of the dependency of the computed values of the solution φ on the integration path and starting point, we use another path which starts vertically, then parallelly with the x -axis to reach the point. In Table 5.3, we list the *difference* of two recovered solutions φ from these two different integration paths, for the non-uniform mesh cases. We can see that these differences are at the levels of local truncation errors and decay in the same order as the errors. Thus the choice of integration path in recovering φ does not affect accuracy.

At $T = 1.5/\pi^2$, the solution has discontinuous derivatives. Fig. 5.1 is the graph of the numerical solution with 40×40 elements (uniform mesh).

Finally we use triangle based triangulation, the mesh with $h = \frac{1}{4}$ is shown in Fig. 5.2. The accuracy at $T = 0.5/\pi^2$ is shown in Table 5.4. Similar accuracy pattern is observed as in the rectangular case. The result at $T = 1.5/\pi^2$, when the derivative is discontinuous, is shown in Fig. 5.3.

Second test problem. We consider the following problem:

$$\begin{aligned} \varphi_t - \cos(\varphi_x + \varphi_y + 1) &= 0, & \text{in } (-2, 2)^2 \times (0, T), \\ \varphi(x, y, 0) &= -\cos\left(\frac{\pi(x+y)}{2}\right), & \forall (x, y) \in (-2, 2)^2, \end{aligned}$$

with periodic boundary conditions.

For this example we use uniform rectangular meshes. The local Lax-Friedrichs flux is used. The solution is smooth at $T = 0.5/\pi^2$. The accuracy of the numerical solution is shown in Table 5.5.

The solution has developed a discontinuous derivative at $T = 1.5/\pi^2$. Results with 40×40 elements are shown in Fig. 5.4.

Third test problem. The level set equation in a domain with a hole:

$$\begin{aligned} \varphi_t + \text{sign}(\varphi_0)(\sqrt{\varphi_x^2 + \varphi_y^2} - 1) &= 0, & \text{in } \Omega \times (0, T), \\ \varphi(x, y, 0) &= -\cos\left(\frac{\pi(x+y)}{2}\right), & \forall (x, y) \in \Omega, \end{aligned}$$

Table 5.1. Accuracy for 2D Burgers equation, uniform rectangular mesh, $T = 0.5/\pi^2$.

$N \times N$	P^1		P^2		P^3	
	L^1 error	order	L^1 error	order	L^1 error	order
10×10	8.09E-02	—	8.62E-03	—	3.19E-03	—
20×20	3.36E-02	1.268	1.72E-03	2.325	3.49E-04	3.192
40×40	1.48E-02	1.183	3.93E-04	2.130	6.64E-05	2.394
80×80	6.88E-03	1.105	9.74E-05	2.013	1.14E-05	2.542
160×160	3.31E-03	1.056	2.45E-05	1.991	1.68E-06	2.763

$N \times N$	P^1		P^2		P^3	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10×10	2.62E-01	—	3.56E-02	—	8.65E-03	—
20×20	1.14E-01	1.201	8.40E-03	2.083	1.16E-03	2.899
40×40	5.00E-02	1.189	2.02E-03	2.056	1.98E-04	2.551
80×80	2.39E-02	1.065	4.92E-04	2.038	3.13E-05	2.661
160×160	1.16E-02	1.043	1.21E-04	2.024	4.41E-06	2.827

Table 5.2. Accuracy for 2D Burgers equation, non-uniform rectangular mesh, $T = 0.5/\pi^2$.

$N \times N$	P^1		P^2		P^3	
	L^2 error	order	L^2 error	order	L^2 error	order
10×10	4.47E-01	—	6.28E-02	—	1.61E-02	—
20×20	1.83E-01	1.288	1.50E-02	2.066	2.06E-03	2.966
40×40	8.01E-02	1.192	3.63E-03	2.047	3.48E-04	2.565
80×80	3.82E-02	1.068	9.17E-04	1.985	6.03E-05	2.529
160×160	1.87E-02	1.031	2.34E-04	1.970	8.58E-06	2.813

$N \times N$	P^1		P^2		P^3	
	L^1 error	order	L^1 error	order	L^1 error	order
10×10	8.16E-02	—	9.16E-03	—	3.39E-03	—
20×20	3.41E-02	1.259	2.09E-03	2.132	4.12E-04	3.041
40×40	1.50E-02	1.185	5.21E-04	2.004	7.03E-05	2.551
80×80	7.16E-03	1.067	1.42E-04	1.875	1.24E-05	2.503
160×160	3.50E-03	1.033	3.85E-05	1.883	1.76E-06	2.817

$N \times N$	P^1		P^2		P^3	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10×10	2.83E-01	—	4.68E-02	—	1.00E-02	—
20×20	1.25E-01	1.179	1.23E-02	1.928	1.39E-03	2.847
40×40	5.74E-02	1.123	3.54E-03	1.797	2.29E-04	2.602
80×80	2.78E-02	1.046	1.15E-03	1.622	5.11E-05	2.164
160×160	1.42E-02	0.969	2.72E-04	2.080	7.16E-06	2.835

Table 5.3. Differences of the solution φ recovered by two different integration paths, non-uniform mesh, Burgers equation.

$N \times N$	P^1		P^2		P^3	
	L^1 error	order	L^1 error	order	L^1 error	order
10×10	8.61E-03	—	2.90E-03	—	1.15E-03	—
20×20	4.64E-03	0.892	1.28E-03	1.180	2.44E-04	2.237
40×40	2.54E-03	0.869	4.12E-04	1.635	3.76E-05	2.698
80×80	1.81E-03	0.489	1.39E-04	1.568	6.71E-06	2.486
160×160	1.09E-03	0.732	3.66E-05	1.925	8.79E-07	2.932

Table 5.4. Accuracy for 2D Burgers equation, triangular mesh, $T = 0.5/\pi^2$.

h	P^2				P^3			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
1	5.48E-02	—	1.52E-01	—	1.17E-02	—	2.25E-02	—
1/2	1.35E-02	2.02	6.26E-02	1.28	1.35E-03	3.12	4.12E-03	2.45
1/4	2.94E-03	2.20	1.55E-02	2.01	1.45E-04	3.22	4.31E-04	3.26
1/8	6.68E-04	2.14	3.44E-03	2.17	1.71E-05	3.08	7.53E-05	2.52

Table 5.5. Accuracy, 2D, $H(u, v) = -\cos(u + v + 1)$, $T = 0.5/\pi^2$.

$N \times N$	P^1		P^2		P^3	
	L^1 error	order	L^1 error	order	L^1 error	order
10×10	6.47E-02	—	8.31E-03	—	1.35E-02	—
20×20	2.54E-02	1.349	1.93E-03	2.106	1.57E-03	3.104
40×40	1.05E-02	1.274	4.58E-04	2.075	2.39E-04	2.716
80×80	4.74E-03	1.147	1.13E-04	2.019	2.89E-05	3.048
160×160	2.23E-03	1.088	2.83E-05	1.997	4.38E-06	2.722

$N \times N$	P^1		P^2		P^3	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10×10	1.47E-01	—	1.88E-02	—	2.36E-02	—
20×20	6.75E-02	1.123	7.34E-03	1.357	3.44E-03	2.778
40×40	2.65E-02	1.349	1.83E-03	2.004	4.59E-04	2.906
80×80	1.18E-02	1.167	4.55E-04	2.008	5.78E-05	2.989
160×160	2.23E-03	1.088	1.13E-04	2.010	8.54E-06	2.759

where $\Omega = \{(x, y) : 1/2 < \sqrt{x^2 + y^2} < 1\}$.

This problem was introduced in [84]. Its exact solution φ has the same zero level set as φ_0 , and the steady state solution is the distance function to that zero level curve. We use this problem to test the effect on the accuracy of the approximation of using various integration paths (5.10) when there is a hole in the region. Notice that the exact steady state solution is the distance function to the inner boundary of domain when boundary condition is adequately prescribed. We compute the time dependent problem to reach a steady state solution, using the exact solution for the boundary conditions of φ_x and φ_y . Four symmetric elements near the outer boundary are updated by (5.9), all other elements are recovered from (5.10) by the shortest path to the nearest one of above four elements. The results are shown in Table 5.6. Also shown in Table 5.6 is the error (difference) between the numerical solution φ thus recovered, and the value of φ after another integration along a circular path (starting and ending at the same point in (5.10)). We can see that the difference is small with the correct order of accuracy, further indicating that the dependency of the recovered solution φ on the integration path is on the order of the truncation errors even for such problems with holes. Finally, the mesh with 1432 triangles and the solution with 5608 triangles are shown in Fig. 5.5.

Table 5.6. Errors for the level set equation, triangular mesh with P^2 .

N	Errors for the Solution				Errors by Integration Path			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
403	1.02E-03	—	1.32E-03	—	1.61E-04	—	5.71E-04	—
1432	1.23E-04	3.05	2.73E-04	2.27	5.84E-05	1.46	1.68E-04	1.78
5608	1.71E-05	2.85	3.18E-05	3.10	9.32E-06	2.65	4.36E-05	1.95
22238	2.09E-06	3.03	5.01E-06	2.67	1.43E-06	2.70	6.63E-06	2.72

Fourth test problem. Two dimensional Riemann problem:

$$\begin{aligned} \varphi_t + \sin(\varphi_x + \varphi_y) &= 0, & \text{in } (-1, 1)^1 \times (0, T), \\ \varphi(x, y, 0) &= \pi(|y| - |x|), & \forall (x, y) \in (-1, 1)^2, \end{aligned}$$

For this example we use a uniform rectangular mesh with 40×40 elements. The local Lax-Friedrichs flux is used. As was mentioned in Example 4.3, we have found out that a nonlinear limiting is needed, for convergence towards an viscosity solution. We show the numerical solution at $T = 1$ in Fig. 5.6.

Fifth test problem. A problem from optimal control [73]:

$$\begin{aligned}\varphi_t + (\sin y)\varphi_x + (\sin x + \text{sign}(\varphi_y))\varphi_y &= \frac{1}{2} \sin^2 y + (1 - \cos x), \\ \varphi(x, y, 0) &= 0,\end{aligned}$$

where the space domain is $(-\pi, \pi)^2$ and the boundary conditions are periodic. We use a uniform rectangular mesh of 40×40 elements and the local Lax-Friedrichs flux. The solution at $T = 1$ is shown in Fig. 5.7, while the optimal control $w = \text{sign}(\varphi_y)$ is shown in Fig. 5.8.

Notice that our method computes $\nabla\varphi$ as an independent variable. It is very desirable for those problems in which the most interesting features are contained in the first derivatives of φ , as in this optimal control problem.

Sixth test problem. A problem from computer vision [78]:

$$\begin{aligned}\varphi_t + I(x, y)\sqrt{1 + \varphi_x^2 + \varphi_y^2} - 1 &= 0, & \text{in } (-1, 1)^2 \times (0, T), \\ \varphi(x, y, 0) &= 0, & \forall (x, y) \in (-1, 1)^2,\end{aligned}$$

with $\varphi = 0$ as the boundary condition. The steady state solution of this problem is the shape lighted by a source located at infinity with vertical direction. The solution is not unique if there are points at which $I(x, y) = 1$. Conditions must be prescribed at those points where $I(x, y) = 1$. Since our method is a finite element method, we need to prescribe suitable conditions at the correspondent elements. We take

$$I(x, y) = 1/\sqrt{1 + (1 - |x|)^2 + (1 - |y|)^2} \quad (5.1)$$

The exact steady solution is $\varphi(x, y, \infty) = (1 - |x|)(1 - |y|)$. We use a uniform rectangular mesh of 40×40 elements and the local Lax-Friedrichs flux. We impose the exact boundary conditions for $u = \varphi_x, v = \varphi_y$ from the above exact steady solution, and take the exact value at one point (the lower left corner) to recover φ . The results for P^2 and P^3 are presented in Fig. 5.3, while Fig. 5.9 contains the history of iterations to the steady state.

Next we take

$$I(x, y) = 1/\sqrt{1 + 4y^2(1 - x^2)^2 + 4x^2(1 - y^2)^2} \quad (5.2)$$

The exact steady solution is $\varphi(x, y, \infty) = (1 - x^2)(1 - y^2)$. We again use a uniform rectangular mesh of 40×40 elements, the local Lax-Friedrichs flux, impose the exact boundary conditions for $u = \varphi_x, v = \varphi_y$ from the above exact steady solution, and take the exact value at one point (the lower left corner) to recover φ . A continuation method is used, with the steady solution using

$$I_\varepsilon(x, y) = 1/\sqrt{1 + 4y^2(1 - x^2)^2 + 4x^2(1 - y^2)^2 + \varepsilon} \quad (5.3)$$

for bigger ε as the initial condition for smaller ε . The sequence of ε used are $\varepsilon = 0.2, 0.05, 0$. The results for P^2 and P^3 are presented in Fig. 5.10.

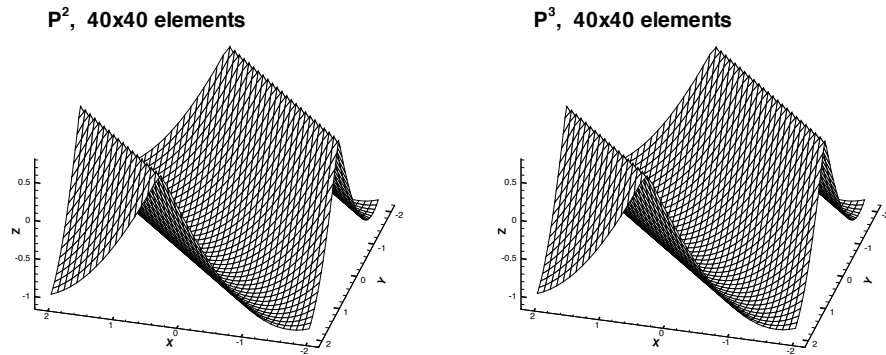


Fig. 5.1. Two dimension Burgers' equation, rectangular mesh, $T=1.5/\pi^2$.

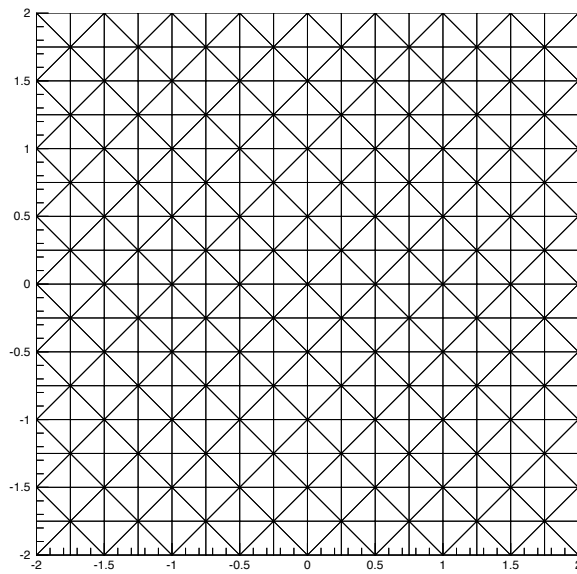


Fig. 5.2. Triangulation for two dimensional Burgers equation, $h = \frac{1}{4}$.

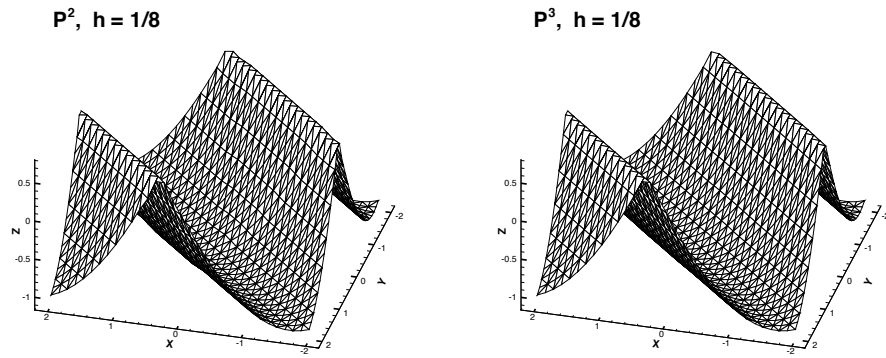


Fig. 5.3. Two dimension Burgers' equation, triangular mesh, $T=1.5/\pi^2$.

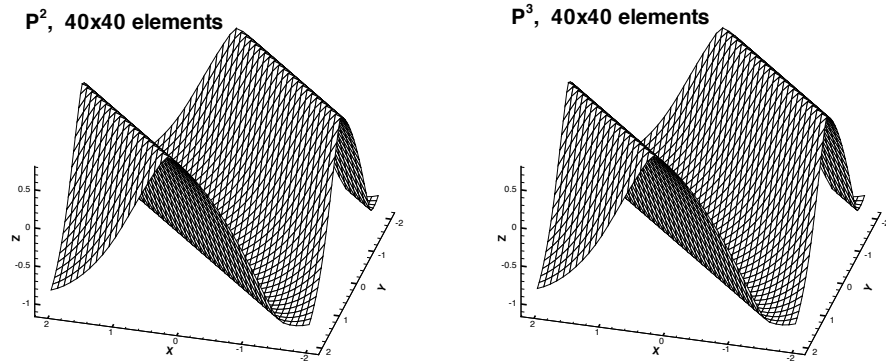
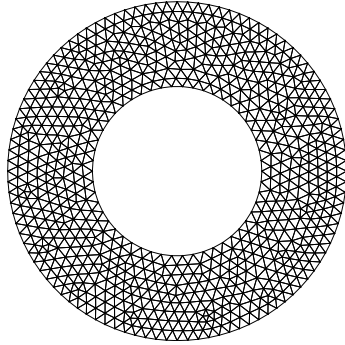


Fig. 5.4. Two dimensional, $H(u, v) = -\cos(u + v + 1)$, $T = 1.5/\pi^2$.

Mesh: 1432 triangles



Solution: 5608 triangles

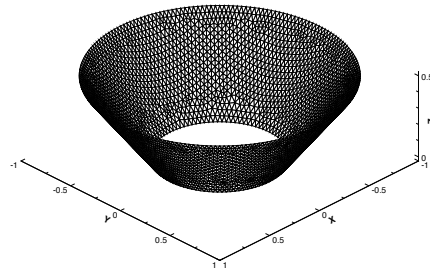
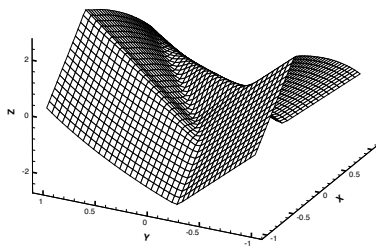


Fig. 5.5. The level set equation, P^2 .

P^2 , 40x40 elements



P^3 , 40x40 elements

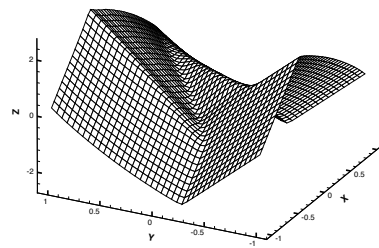
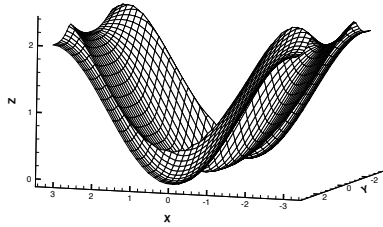


Fig. 5.6. Two dimensional Riemann problem, $H(u, v) = \sin(u + v)$, $T = 1$.

P^2 , 40x40 elements



P^3 , 40x40 elements

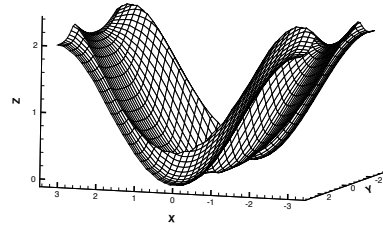
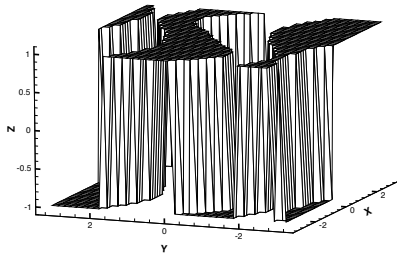


Fig. 5.7. Control problem, $T = 1$.

P^2 , 40x40 elements



P^3 , 40x40 elements

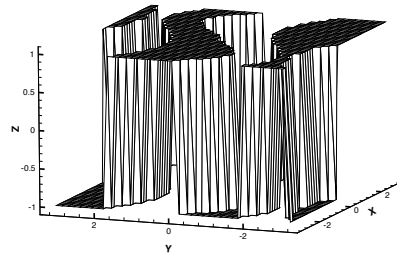
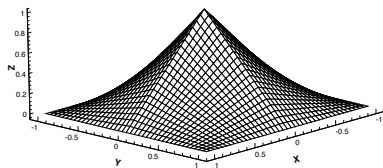
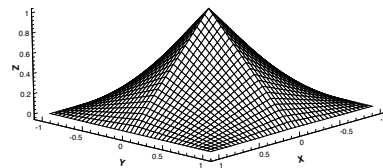


Fig. 5.8. Control problem, $T = 1$, $w = \text{sign}(\varphi_y)$.

P^2 , 40x40 elements



P^3 , 40x40 elements



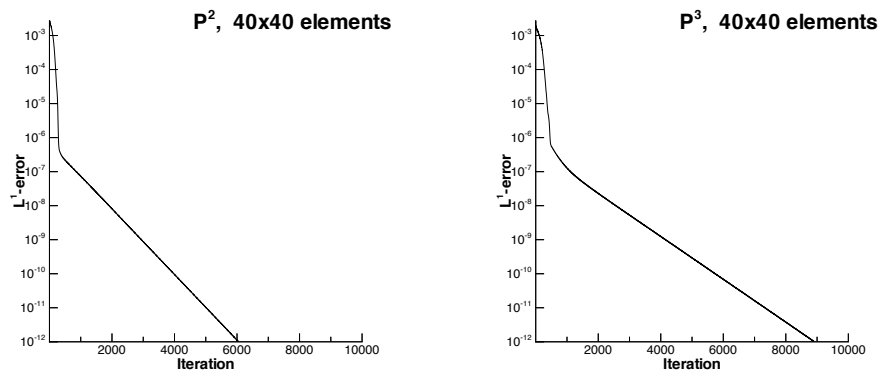


Fig. 5.9. Computer vision problem, history of iterations.

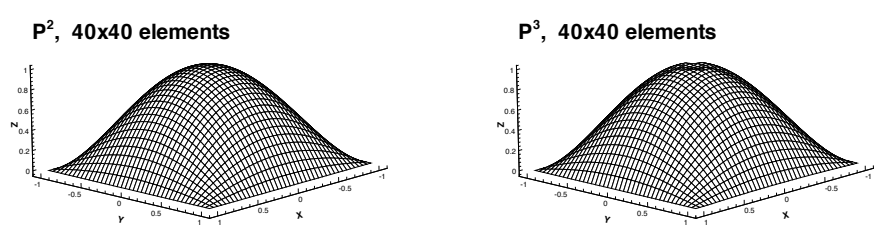


Fig. 5.10. Computer vision problem, $\varphi(x, y, \infty) = (1 - x^2)(1 - y^2)$.

6 Convection diffusion: The LDG method

6.1 Introduction

In this chapter, which follows the work by Cockburn and Shu [27], 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, in §3, some preliminary numerical results displaying the performance of the method. In this simple setting, the main ideas of how to devise 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 $L^\infty(0, T; L^2)$ -norm for polynomials of degree $k \geq 0$ in the linear case is obtained; this estimate is sharp. In §4, 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 $L^\infty(0, T; L^2)$ -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 §5, the extension of the LDG method to multidimensional systems is briefly described and in §6, 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 [63] are presented.

6.2 The LDG methods for the one-dimensional case

In this section, we present and analyze the LDG methods for the following simple model problem:

$$\partial_t u + \partial_x (f(u) - a(u) \partial_x u) = 0 \quad \text{in } Q, \quad (6.1)$$

$$u(t=0) = u_0 \quad \text{on } (0, 1), \quad (6.2)$$

where $Q = (0, T) \times (0, 1)$, 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 (6.1), (6.2) as follows:

$$\partial_t u + \partial_x (f(u) - \sqrt{a(u)} q) = 0 \quad \text{in } Q, \quad (6.3)$$

$$q - \partial_x g(u) = 0 \quad \text{in } Q, \quad (6.4)$$

$$u(t=0) = u_0, \quad \text{on } (0, 1), \quad (6.5)$$

where $g(u) = \int^u \sqrt{a(s)} ds$. The LDG method for (6.1), (6.2) is now obtained by simply discretizing the above system with the Discontinuous Galerkin method.

To do that, we follow [24] and [21]. We define the flux $\mathbf{h} = (h_u, h_q)^t$ as follows:

$$\mathbf{h}(u, q) = (f(u) - \sqrt{a(u)} q, -g(u))^t. \quad (6.6)$$

For each partition of the interval $(0, 1)$, $\{x_{j+1/2}\}_{j=0}^N$, we set, for $j = 1, \dots, N$:

$$I_j = (x_{j-1/2}, x_{j+1/2}), \quad \Delta x_j = x_{j+1/2} - x_{j-1/2}, \quad (6.7)$$

and

$$\Delta x = \max_{1 \leq j \leq N} \Delta x_j. \quad (6.8)$$

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

$$V_h = V_h^k = \{v \in L^1(0, 1) : v|_{I_j} \in P^k(I_j), j = 1, \dots, N\}, \quad (6.9)$$

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 (6.3), (6.4), and (6.5) 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 (6.3) and (6.4),

$$\begin{aligned} & \int_{I_j} \partial_t u(x, t) v_u(x) dx - \int_{I_j} h_u(\mathbf{w}(x, t)) \partial_x v_u(x) dx \\ & + h_u(\mathbf{w}(x_{j+1/2}, t)) v_u(x_{j+1/2}^-) - h_u(\mathbf{w}(x_{j-1/2}, t)) v_u(x_{j-1/2}^+) = 0, \end{aligned} \quad (6.10)$$

$$\begin{aligned} & \int_{I_j} q(x, t) v_q(x) dx - \int_{I_j} h_q(\mathbf{w}(x, t)) \partial_x v_q(x) dx \\ & + h_q(\mathbf{w}(x_{j+1/2}, t)) v_q(x_{j+1/2}^-) - h_q(\mathbf{w}(x_{j-1/2}, t)) v_q(x_{j-1/2}^+) = 0, \end{aligned} \quad (6.11)$$

$$\int_{I_j} u(x, 0) v_i(x) dx = \int_{I_j} u_0(x) v_i(x) dx. \quad (6.12)$$

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:

$$\begin{aligned} \forall v_{h,u} \in P^k(I_j) : \\ \int_{I_j} \partial_t u_h(x, t) v_{h,u}(x) dx - \int_{I_j} h_u(\mathbf{w}_h(x, t)) \partial_x v_{h,u}(x) dx \\ + \hat{h}_u(\mathbf{w}_h)_{j+1/2}(t) v_{h,u}(x_{j+1/2}^-) - \hat{h}_u(\mathbf{w}_h)_{j-1/2}(t) v_{h,u}(x_{j-1/2}^+) = 0 \end{aligned} \quad (6.13)$$

$$\begin{aligned} \forall v_{h,q} \in P^k(I_j) : \\ \int_{I_j} q_h(x, t) v_{h,q}(x) dx - \int_{I_j} h_q(\mathbf{w}_h(x, t)) \partial_x v_{h,q}(x) dx \\ + \hat{h}_q(\mathbf{w}_h)_{j+1/2}(t) v_{h,q}(x_{j+1/2}^-) - \hat{h}_q(\mathbf{w}_h)_{j-1/2}(t) v_{h,q}(x_{j-1/2}^+) = 0, \end{aligned} \quad (6.14)$$

$$\begin{aligned} \forall v_{h,i} \in P^k(I_j) : \\ \int_{I_j} (u_h(x, 0) - u_0(x)) v_{h,i}(x) dx = 0 \end{aligned} \quad (6.15)$$

It only remains to choose the numerical flux $\hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}(t)$. We use the notation:

$$[p] = p^+ - p^-, \quad \bar{p} = \frac{1}{2}(p^+ + p^-), \quad p_{j+1/2}^\pm = p(x_{j+1/2}^\pm).$$

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 [71]) 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:

$$\hat{\mathbf{h}}(\mathbf{w}^-, \mathbf{w}^+) = \hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+) + \hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+). \quad (6.16)$$

The convective flux is given by

$$\hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+) = (\hat{f}(u^-, u^+), 0)^t, \quad (6.17)$$

where $\hat{f}(u^-, u^+)$ is any locally Lipschitz E-flux consistent with the nonlinearity f , and the diffusive flux is given by

$$\hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+) = \left(-\frac{[g(u)]}{[u]} \bar{q}, -\overline{g(u)} \right)^t - \mathcal{C}_{diff}[\mathbf{w}], \quad (6.18)$$

where

$$\mathcal{C}_{diff} = \begin{pmatrix} 0 & c_{12} \\ -c_{12} & 0 \end{pmatrix}, \quad (6.19)$$

$$c_{12} = c_{12}(\mathbf{w}^-, \mathbf{w}^+) \quad \text{is locally Lipschitz,} \quad (6.20)$$

$$c_{12} \equiv 0 \quad \text{when } a(\cdot) \equiv 0. \quad (6.21)$$

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 (6.19); 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 (6.14) follows from the fact that, by (6.18), 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 (6.21) 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:

$$\hat{\mathbf{h}}(\mathbf{w}^-, \mathbf{w}^+) = \left(\frac{[\phi(u)]}{[u]} - \frac{[g(u)]}{[u]} \bar{q}, -\overline{g(u)} \right)^t - \mathcal{C}[\mathbf{w}],$$

where

$$\mathcal{C} = \begin{pmatrix} c_{11} & c_{12} \\ -c_{12} & 0 \end{pmatrix}, \quad c_{11} = \frac{1}{[u]} \left(\frac{[\phi(u)]}{[u]} - \hat{f}(u^-, u^+) \right). \quad (6.22)$$

with $\phi(u)$ defined by $\phi(u) = \int^{u^-} f(s) ds$. Since $\hat{f}(\cdot, \cdot)$ is an E-flux,

$$c_{11} = \frac{1}{[u]^2} \int_{u^-}^{u^+} (f(s) - \hat{f}(u^-, u^+)) ds \geq 0,$$

and so, by (6.19), the matrix \mathcal{C} is semipositive definite. The property (iv) follows from this fact and from the following result.

Proposition 26. (Stability) *We have,*

$$\frac{1}{2} \int_0^1 u_h^2(x, T) dx + \int_0^T \int_0^1 q_h^2(x, t) dx dt + \Theta_{T, \mathcal{C}}([\mathbf{w}_h]) \leq \frac{1}{2} \int_0^1 u_0^2(x) dx,$$

where $\Theta_{T, \mathcal{C}}([\mathbf{w}_h])$ is the following expression:

$$\int_0^T \sum_{1 \leq j \leq N} \left\{ [\mathbf{w}_h(t)]^t \mathcal{C} [\mathbf{w}_h(t)] \right\}_{j+1/2} dt.$$

For a proof, see the appendix. 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 approximation properties of the finite element space V_h , we can prove the following error estimate. We denote the $L^2(0, 1)$ -norm of the ℓ -th derivative of u by $|u|_\ell$.

Theorem 27. (Error estimate) *Let \mathbf{e} be the approximation error $\mathbf{w} - \mathbf{w}_h$. Then we have,*

$$\left\{ \int_0^1 |e_u(x, T)|^2 dx + \int_0^T \int_0^1 |e_q(x, t)|^2 dx dt + \Theta_{T, \mathcal{C}}([\mathbf{e}]) \right\}^{1/2} \leq C (\Delta x)^k,$$

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 C identically zero.

For a proof, see the appendix. The above error estimate gives a suboptimal order of convergence, but it is sharp for the LDG methods. Indeed, Bassi *et al* [5] 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 \mathcal{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 \mathcal{C} , which enhances the stability properties of the LDG method. Indeed, in the purely hyperbolic case, the quantity

$$\int_0^T \sum_{1 \leq j \leq N} \left\{ [u_h(t)]^t c_{11} [u_h(t)] \right\}_{j+1/2} dt,$$

is uniformly bounded. This additional control on the jumps of the variable u_h 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.* [5] 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 \mathcal{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 \mathcal{C} . In fact, the numerical results of section §2.b in uniform grids indicate that with yet another choice of the matrix \mathcal{C} , see (6.23), 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.

6.3 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

$$\begin{aligned} \partial_t u + c \partial_x u - a \partial_x^2 u &= 0 \quad \text{in } (0, T) \times (0, 2\pi), \\ u(t = 0, x) &= \sin(x), \quad \text{on } (0, 2\pi), \end{aligned}$$

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 \mathcal{C} defined by

$$\mathcal{C} = \begin{pmatrix} \frac{|c|}{2} & -\frac{\sqrt{a}}{2} \\ \frac{\sqrt{a}}{2} & 0 \end{pmatrix}. \quad (6.23)$$

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 [81], 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 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.

Finally, to show that the order of accuracy could really degenerate to k for P^k , as was already observed in [5], we rerun the heat equation case $a = 1, c = 0$ with the central flux

$$c = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

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 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

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

6.4 The LDG methods for the multidimensional case

In this section, we consider the LDG methods for the following convection-diffusion model problem

$$\partial_t u + \sum_{1 \leq i \leq d} \partial_{x_i} (f_i(u)) - \sum_{1 \leq j \leq d} a_{ij}(u) \partial_{x_j} u = 0 \quad \text{in } Q, \quad (6.24)$$

$$u(t=0) = u_0 \quad \text{on } (0,1)^d, \quad (6.25)$$

where $Q = (0, T) \times (0, 1)^d$, 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.

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

$$a_{ij}(u) = \sum_{1 \leq \ell \leq d} b_{i\ell}(u) b_{\ell j}(u). \quad (6.26)$$

Then we define the new scalar variables $q_\ell = \sum_{1 \leq j \leq d} b_{\ell j}(u) \partial_{x_j} u$ and rewrite the problem (6.24), (6.25) as follows:

$$\partial_t u + \sum_{1 \leq i \leq d} \partial_{x_i} (f_i(u)) - \sum_{1 \leq \ell \leq d} b_{i\ell}(u) q_\ell = 0 \quad \text{in } Q, \quad (6.27)$$

$$q_\ell - \sum_{1 \leq j \leq d} \partial_{x_j} g_{\ell j}(u) = 0 \quad \ell = 1, \dots, d, \quad \text{in } Q, \quad (6.28)$$

$$u(t=0) = u_0 \quad \text{on } (0,1)^d, \quad (6.29)$$

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 $\mathbf{w} = (u, \mathbf{q})^t = (u, q_1, \dots, q_d)^t$ and, for each $i = 1, \dots, d$, introduce the flux

$$\mathbf{h}_i(\mathbf{w}) = (f_i(u) - \sum_{1 \leq \ell \leq d} b_{i\ell}(u) q_\ell, -g_{1i}(u), \dots, -g_{di}(u))^t. \quad (6.30)$$

We consider triangulations of $(0,1)^d$, $\mathcal{T}_{\Delta x} = \{K\}$, 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 $\mathcal{E}_{\Delta x}$ the set of all faces e of the border of K for all $K \in \mathcal{T}_{\Delta x}$. The diameter of K is denoted by Δx_K and the maximum Δx_K , for $K \in \mathcal{T}_{\Delta x}$ is denoted by Δx . We require, for the sake of simplicity, that the triangulations $\mathcal{T}_{\Delta x}$ be regular, that is, there is a constant independent of Δx such that

$$\frac{\Delta x_K}{\rho_K} \leq \sigma \quad \forall K \in \mathcal{T}_{\Delta x},$$

where ρ_K denotes the diameter of the maximum ball included in K .

We seek an approximation $\mathbf{w}_h = (u_h, \mathbf{q}_h)^t = (u_h, q_{h1}, \dots, q_{hd})^t$ 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

$$V_h = V_h^k = \{v \in L^1((0,1)^d) : v|_K \in P^k(K) \forall K \in \mathcal{T}_{\Delta x}\}, \quad (6.31)$$

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 $\mathcal{T}_{\Delta x}$. We obtain the following weak formulation on each element K of the triangulation $\mathcal{T}_{\Delta x}$:

$$\begin{aligned} & \forall v_{h,u} \in P^k(K) : \\ & \int_K \partial_t u_h(x, t) v_{h,u}(x) dx - \sum_{1 \leq i \leq d} \int_K h_{i,u}(\mathbf{w}_h(x, t)) \partial_{x_i} v_{h,u}(x) dx \\ & + \int_{\partial K} \hat{h}_u(\mathbf{w}_h, \mathbf{n}_{\partial K})(x, t) v_{h,u}(x) d\Gamma(x) = 0, \end{aligned} \quad (6.32)$$

for $\ell = 1, \dots, d$:

$$\begin{aligned} & \forall v_{h,q_\ell} \in P^k(K) : \\ & \int_K q_{h\ell}(x, t) v_{h,q_\ell}(x) dx - \sum_{1 \leq j \leq d} \int_K h_{j,q_\ell}(\mathbf{w}_h(x, t)) \partial_{x_j} v_{h,q_\ell}(x) dx \\ & + \int_{\partial K} \hat{h}_{q_\ell}(\mathbf{w}_h, \mathbf{n}_{\partial K})(x, t) v_{h,q_\ell}(x) d\Gamma(x) = 0, \end{aligned} \quad (6.33)$$

$$\begin{aligned} & \forall v_{h,u} \in P^k(K) : \\ & \int_K u_h(x, 0) v_{h,i}(x) dx = \int_K u_0(x) v_{h,i}(x) dx, \end{aligned} \quad (6.34)$$

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 $(\hat{h}_u, \hat{h}_{q_1}, \dots, \hat{h}_{q_d})^t \equiv \hat{\mathbf{h}} \equiv \hat{\mathbf{h}}(\mathbf{w}_h, \mathbf{n}_{\partial K})(x, t)$.

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

$$\sum_{1 \leq i \leq d} \mathbf{h}_i n_{\partial K, i},$$

- (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 $a(\cdot) \equiv 0$,
- (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

$$\hat{\mathbf{h}}_{conv}(\mathbf{w}^-, \mathbf{w}^+; \mathbf{n}) = (\hat{f}(u^-, u^+; \mathbf{n}), 0)^t,$$

where $\hat{f}(u^-, u^+; \mathbf{n})$ is any locally Lipschitz E-flux which is conservative and consistent with the nonlinearity

$$\sum_{1 \leq i \leq d} f_i(u) n_i,$$

and the diffusive flux $\hat{\mathbf{h}}_{diff}(\mathbf{w}^-, \mathbf{w}^+; \mathbf{n})$ is given by

$$\left(- \sum_{1 \leq i, \ell \leq d} \frac{[g_{i\ell}(u)]}{[u]} \bar{q}_\ell n_i, - \sum_{1 \leq i \leq d} \overline{g_{i1}(u)} n_i, \dots, - \sum_{1 \leq i \leq d} \overline{g_{id}(u)} n_i \right)^t - \mathcal{C}_{diff}[\mathbf{w}],$$

where

$$\mathcal{C}_{diff} = \begin{pmatrix} 0 & c_{12} & c_{13} & \cdots & c_{1d} \\ -c_{12} & 0 & 0 & \cdots & 0 \\ -c_{13} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_{1d} & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$c_{1j} = c_{1j}(\mathbf{w}^-, \mathbf{w}^+) \quad \text{is locally Lipschitz for } j = 1, \dots, d,$$

$$c_{1j} \equiv 0 \quad \text{when } a(\cdot) \equiv 0 \quad \text{for } j = 1, \dots, d.$$

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:

$$\left(- \sum_{1 \leq i, \ell \leq d} \frac{[g_{i\ell}(u)]}{[u]} \bar{q}_\ell n_i, - \sum_{1 \leq i \leq d} \overline{g_{i1}(u)} n_i, \dots, - \sum_{1 \leq i \leq d} \overline{g_{id}(u)} n_i \right)^t - \mathcal{C}[\mathbf{w}],$$

where

$$\mathcal{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1d} \\ -c_{12} & 0 & 0 & \cdots & 0 \\ -c_{13} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_{1d} & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$c_{11} = \frac{1}{[u]} \left(\sum_{1 \leq i \leq d} \frac{[\phi_i(u)]}{[u]} n_i - \hat{f}(u^-, u^+; \mathbf{n}) \right),$$

where $\phi_i(u) = \int^u f_i(s) ds$. Since $\hat{f}(\cdot, \cdot; \mathbf{n})$ is an E-flux,

$$\begin{aligned} c_{11} &= \frac{1}{[u]^2} \int_{u^-}^{u^+} \left(\sum_{1 \leq i \leq d} f_i(s) n_i - \hat{f}(u^-, u^+; \mathbf{n}) \right) ds \\ &\geq 0, \end{aligned}$$

and so the matrix \mathcal{C} is semipositive definite. The property (iv) follows from this fact and from the following result.

Proposition 28. (Stability) *We have,*

$$\begin{aligned} \frac{1}{2} \int_{(0,1)^d} u_h^2(x, T) dx + \int_0^T \int_{(0,1)^d} |\mathbf{q}_h(x, t)|^2 dx dt + \Theta_{T, \mathcal{C}}([\mathbf{w}_h]) \\ \leq \frac{1}{2} \int_{(0,1)^d} u_0^2(x) dx, \end{aligned}$$

where the quantity $\Theta_{T, \mathcal{C}}([\mathbf{w}_h])$ is given by

$$\int_0^T \sum_{e \in \mathcal{E}_{\Delta x}} \int_e [\mathbf{w}_h(x, t)]^t \mathcal{C} [\mathbf{w}_h(x, t)] d\Gamma(x) dt.$$

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 $|u|_{k+1}^2$.

Theorem 29. (Error estimate) *Let \mathbf{e} be the approximation error $\mathbf{w} - \mathbf{w}_h$. Then we have, for arbitrary, regular grids,*

$$\begin{aligned} \left\{ \int_{(0,1)^d} |e_u(x, T)|^2 dx + \int_0^T \int_{(0,1)^d} |\mathbf{e}_q(x, t)|^2 dx dt + \Theta_{T, \mathcal{C}}([\mathbf{e}]) \right\}^{1/2} \\ \leq C (\Delta x)^k, \end{aligned}$$

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 \mathcal{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 .

6.5 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 $\mathbf{q} = (\partial_{x_1} u, \dots, \partial_{x_d} u)$ as the auxiliary variables. Although with this choice, the L^2 -stability result will not be available theoretically, this would not cause any problem in practical implementations.

6.6 Some numerical results

Next, we present some numerical results from the papers by Bassi and Rebay [3] and Lomtev and Karniadakis [63].

- **Smooth, steady state solutions.** We start by displaying the convergence of the method for a p -refinement done by Lomtev and Karniadakis [63]. In Figure 6.1, 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 [63].

Now, let us consider the laminar, transonic flow around the NACA0012 airfoil at an angle of attack of ten degrees, free stream Mach number $M = 0.8$, and Reynolds number (based on the free stream velocity and the airfoil chord) equal to 73; the wall temperature is set equal to the free stream 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 [63] 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 6.3, taken from [63], 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 [63] computed with polynomials of degree 6. We can see good agreement of both computations. In Figure 6.2, taken from [63], 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, free stream Mach number $M = 0.5$, and Reynolds number equal to 5000. In figure 6.4, we can see the Mach isolines corresponding to linear, quadratic, and cubic elements. In the figures 6.5, 6.6, and 6.7 details of the results with cubic elements are shown. Note how the boundary layer is captured within 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 unstructured 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 6.8, 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 accommodate the exact boundaries. In Figure 6.9, 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 branch) 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 [79], or in terms of velocity-vorticity (IVVA), by Trujillo [87], for the *incompressible* Navier Stokes equations; results produced by the code called PRISM are also included, see Newmann [69]. The experimental data was taken from Macagno and Hung [67]. 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 [63]. 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 6.10, 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 6.11, 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 [63].

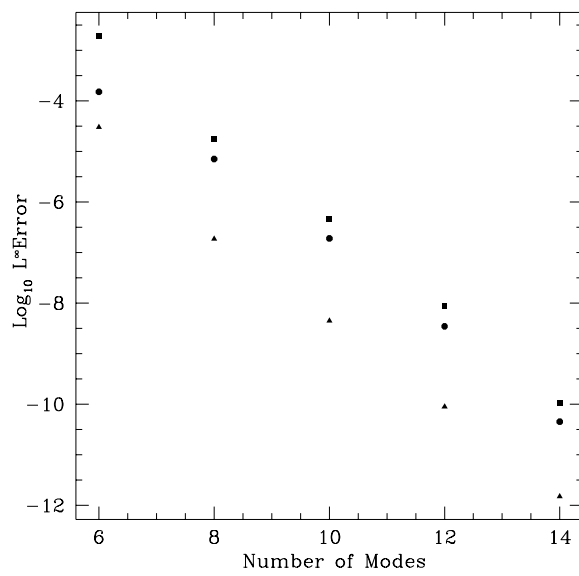


Fig. 6.1. Maximum errors of the density (triangles), momentum (circles) and energy (squares) as a function of the degree of the approximating polynomial plus one (called "number of modes" in the picture).

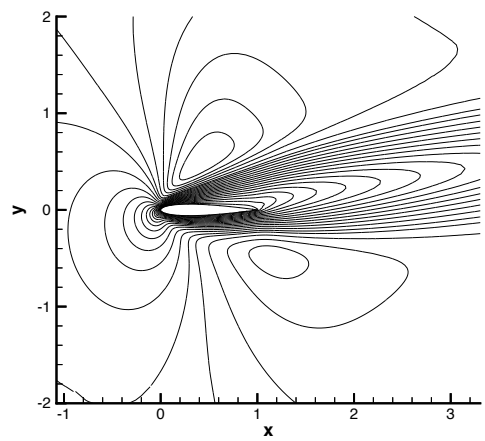
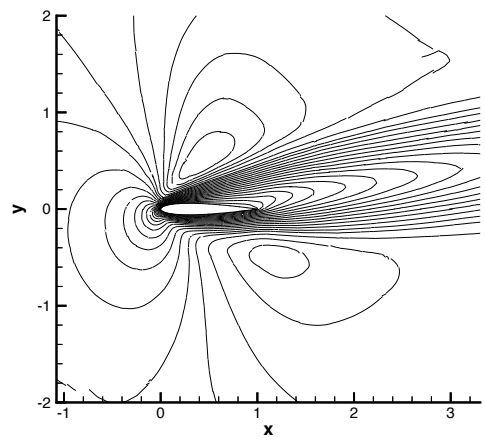
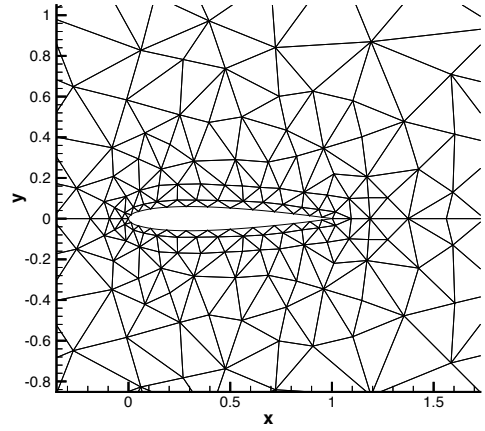


Fig. 6.2. Mesh (top) and Mach isolines around the NACA0012 airfoil, ($Re = 73$, $M = 0.8$, angle of attack of ten degrees) for quadratic (middle) and quartic (bottom) elements.

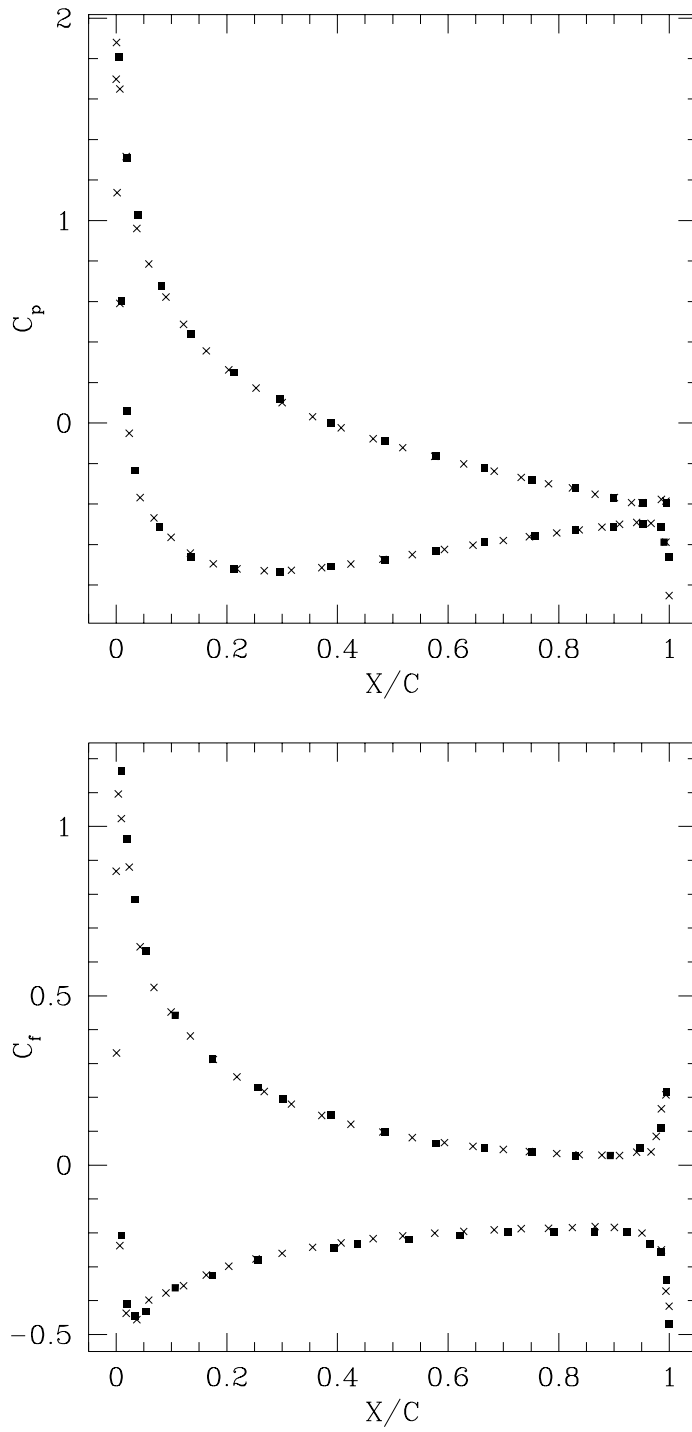


Fig. 6.3. Pressure (top) and drag(bottom) coefficient distributions. The squares were obtained by Bassi and Rebay [3] with cubics and the crosses by Lomtev and Karniadakis [63] with polynomials of degree 6.

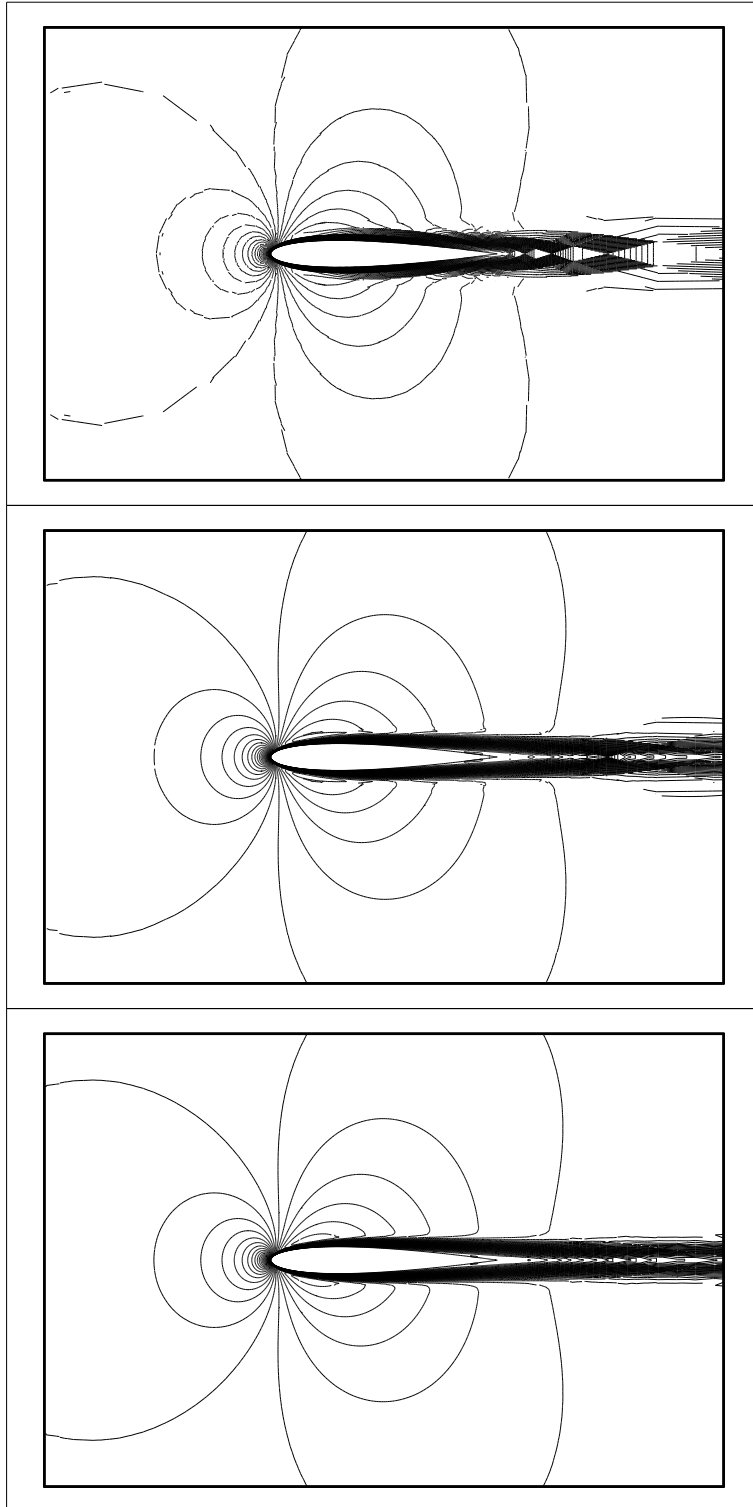


Fig. 6.4. Mach isolines around the NACA0012 airfoil, ($Re = 5000$, $M = 0.5$, zero angle of attack) for the linear (top), quadratic (middle), and cubic (bottom) elements.

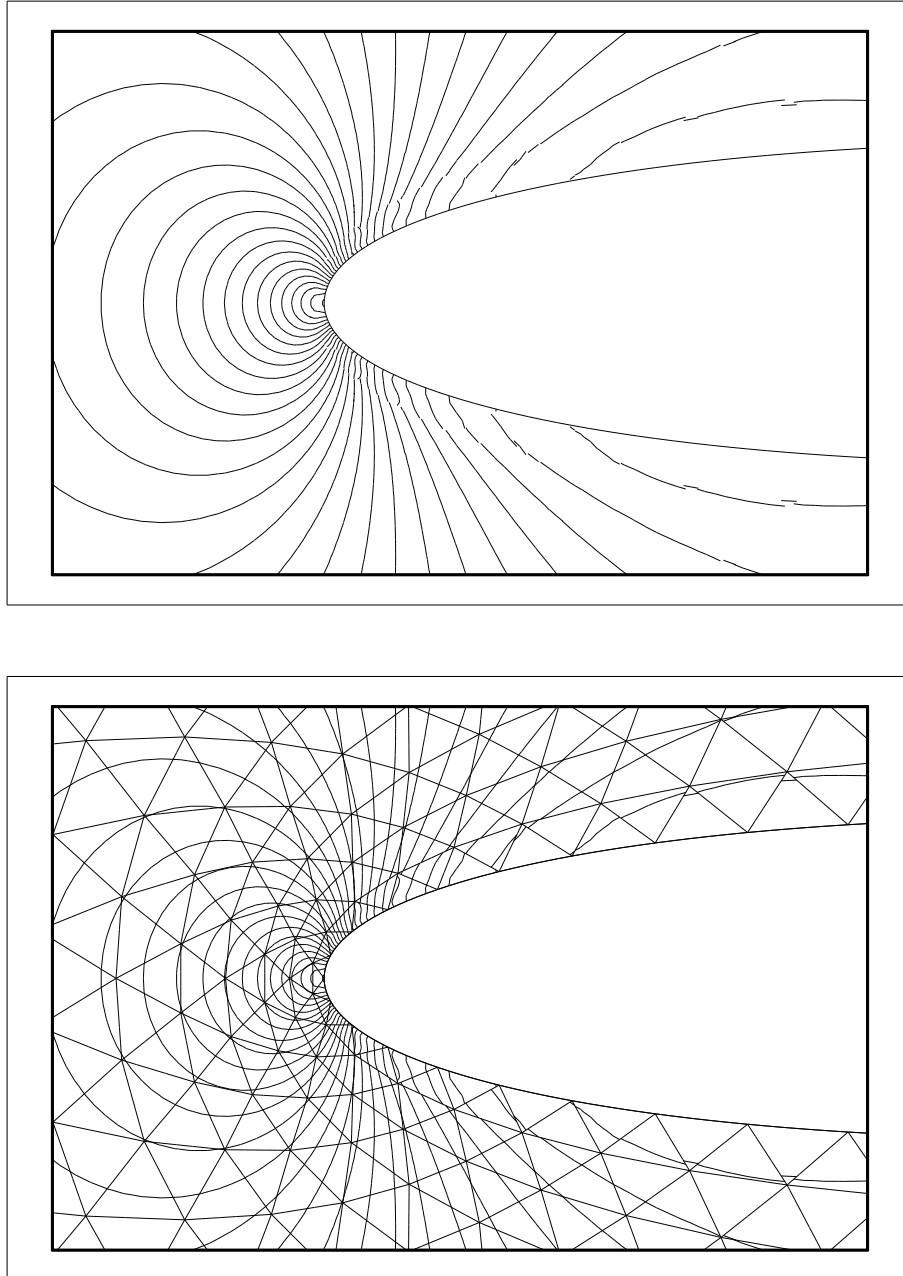


Fig. 6.5. Pressure isolines around the NACA0012 airfoil, ($Re = 5000, M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

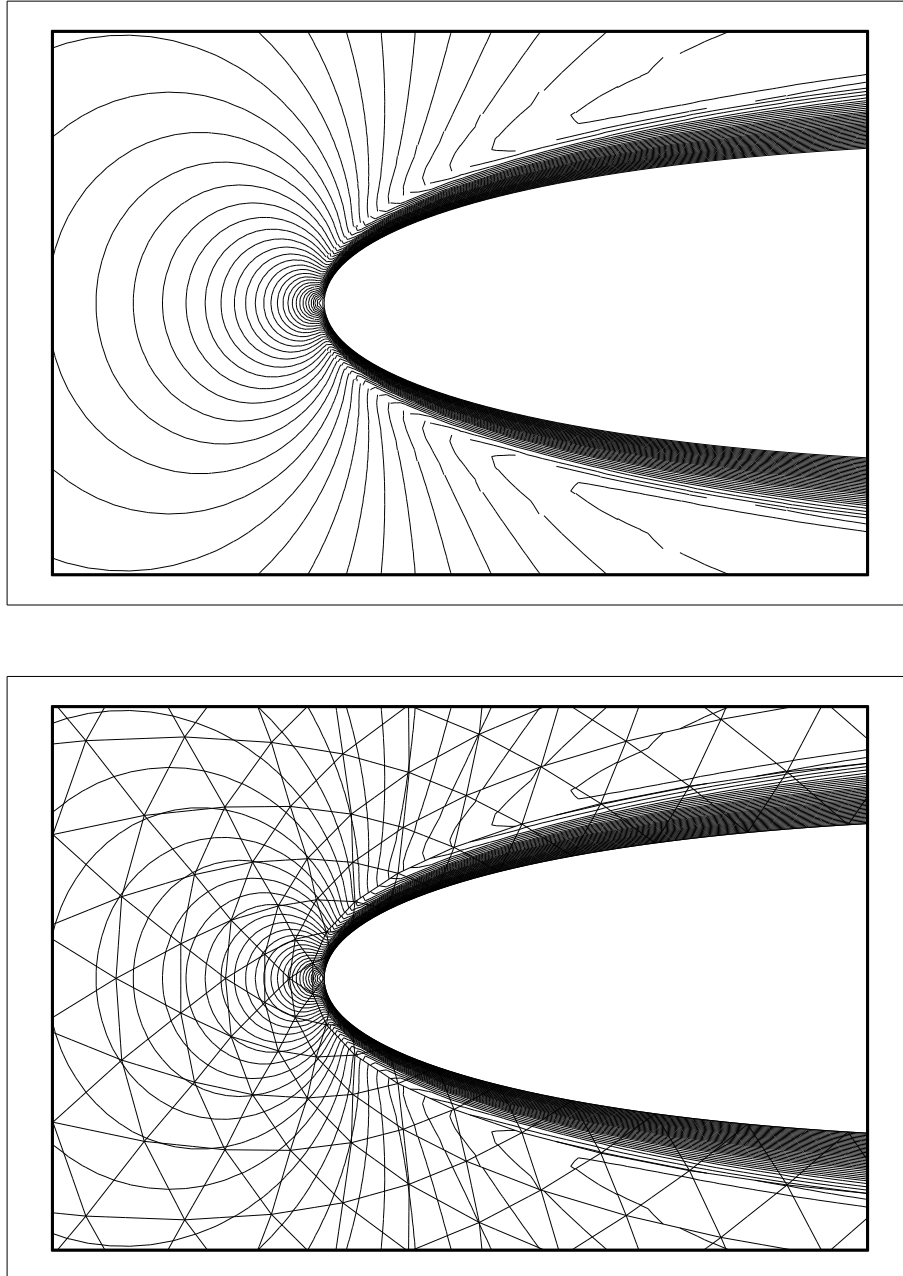


Fig. 6.6. Mach isolines around the leading edge of the NACA0012 airfoil, ($Re = 5000$, $M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

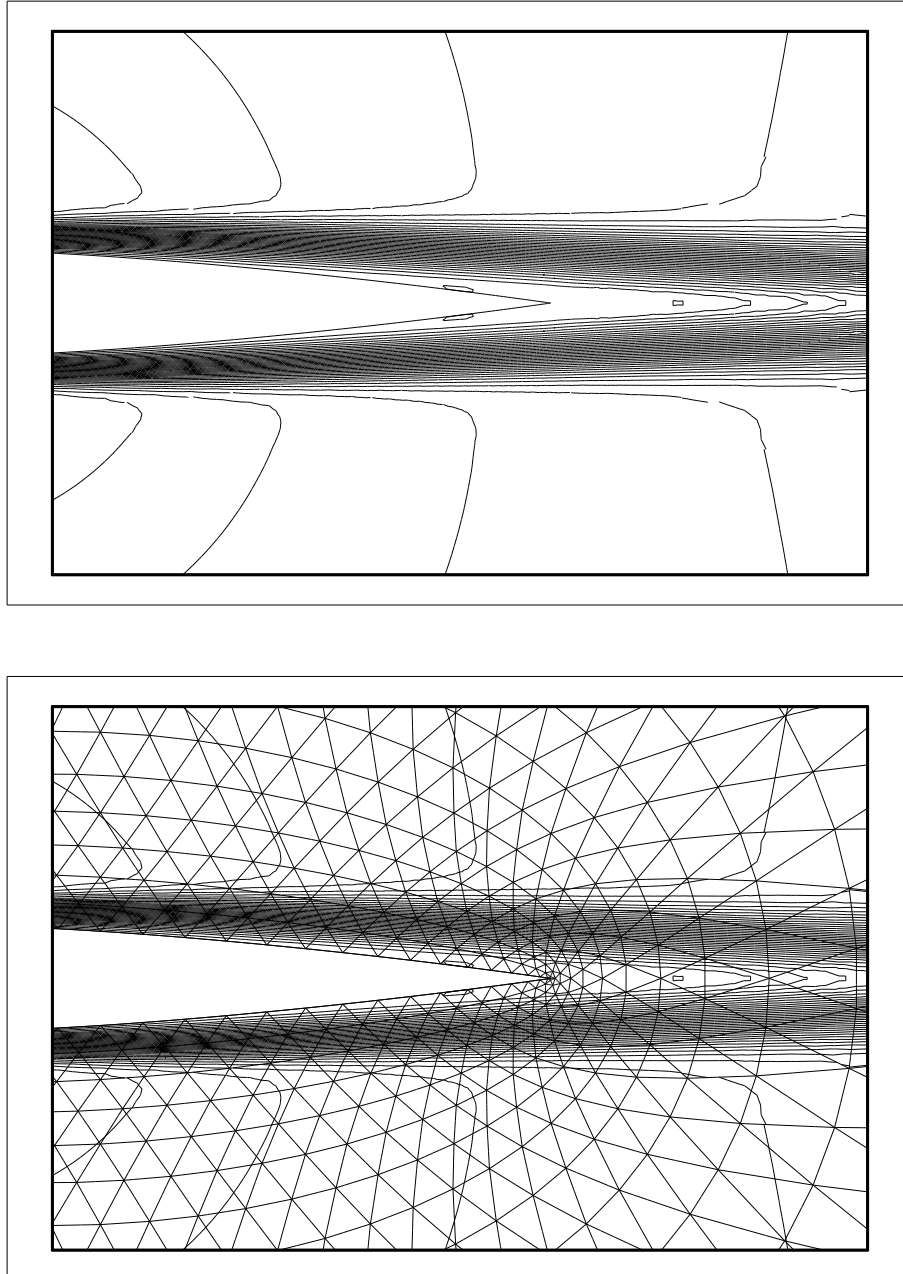


Fig. 6.7. Mach isolines around the trailing edge of the NACA0012 airfoil, ($Re = 5000$, $M = 0.5$, zero angle of attack) for the for cubic elements without (top) and with (bottom) the corresponding grid.

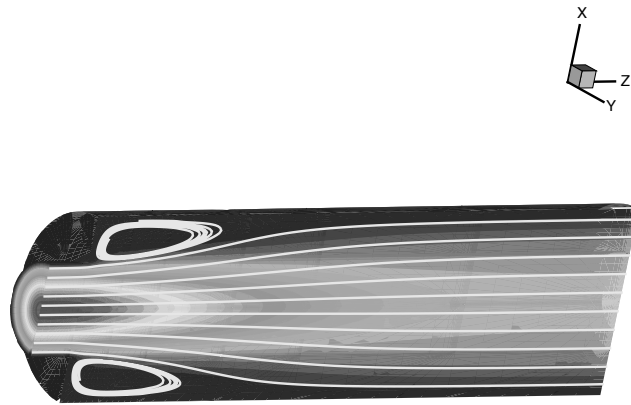


Fig. 6.8. Expansion pipe flow at Reynolds number 50 and Mach number 0.2. Velocity profile and streamlines computed with a mesh of 600 elements and polynomials of degree 5.

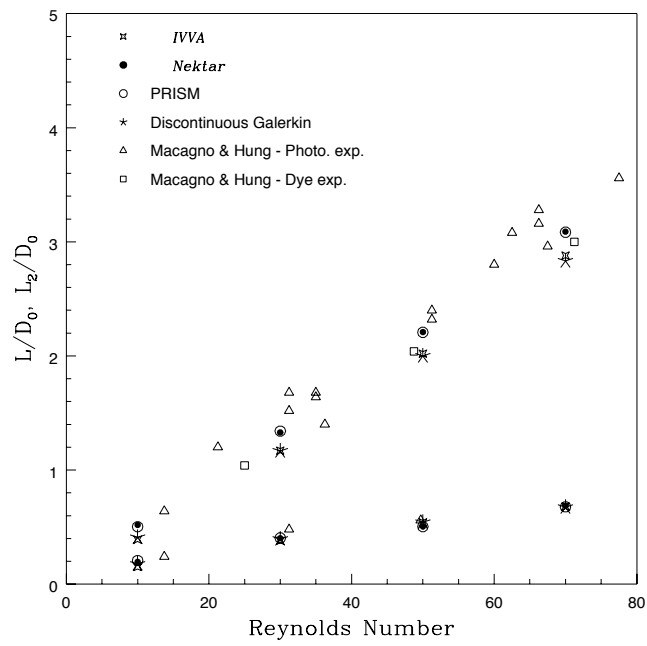


Fig. 6.9. Expansion pipe flow: Comparison between computational and experimental results.

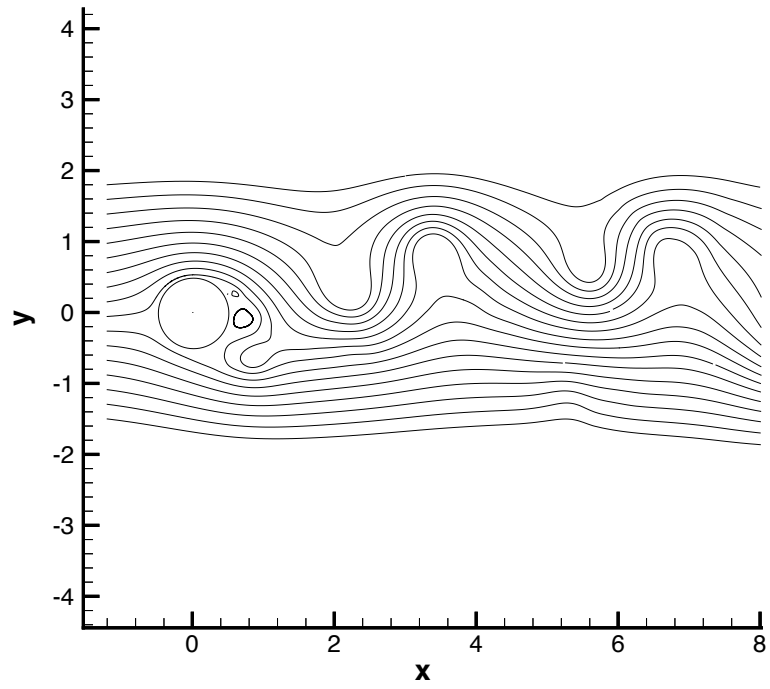


Fig. 6.10. Flow around a cylinder with Reynolds number 10,000 and Mach number 0.2. Streamlines. A mesh of 680 elements was used with polynomials that could change degree from element to element; the maximum degree was 5.

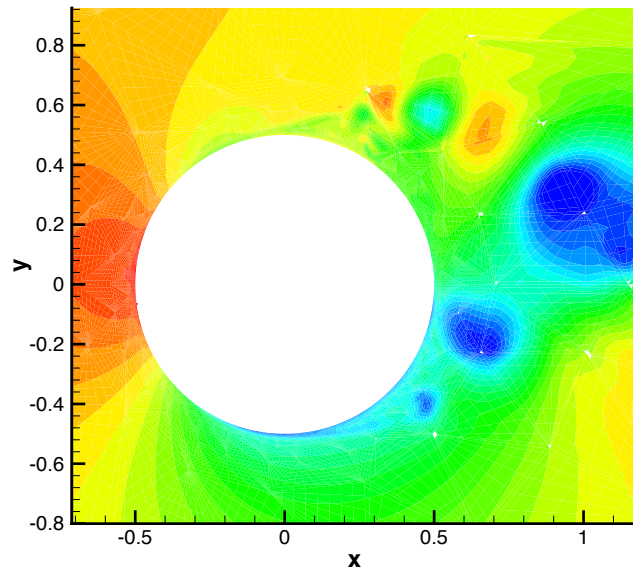
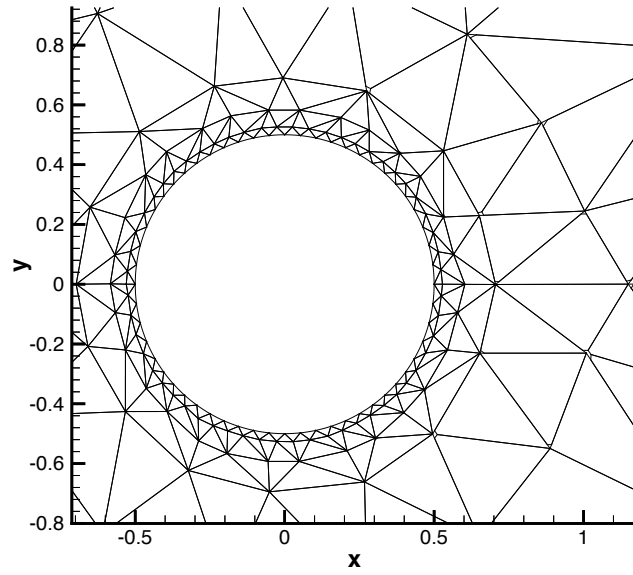


Fig. 6.11. Flow around a cylinder with Reynolds number 10,000 and Mach number 0.2. Detail of the mesh (top) and density (bottom) around the cylinder.

6.7 Appendix: Proof of the L²-error estimates

Proof of Proposition 26 In this section, we prove the nonlinear stability result of Proposition 26. 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 26.

The continuous case as a model. We start by rewriting the equations (6.10) and (6.11), in *compact form*. If in equations (6.10) and (6.11) we replace $v_u(x)$ and $v_q(x)$ by $v_u(x, t)$ and $v_q(x, t)$, respectively, add the resulting equations, sum on j from 1 to N , and integrate in time from 0 to T , we obtain that

$$B(\mathbf{w}, \mathbf{v}) = 0, \quad \forall \text{ smooth } \mathbf{v}, \quad (6.35)$$

where

$$\begin{aligned} B(\mathbf{w}, \mathbf{v}) &= \int_0^T \int_0^1 \partial_t u(x, t) v_u(x, t) dx dt \\ &\quad + \int_0^T \int_0^1 q(x, t) v_q(x, t) dx dt \\ &\quad - \int_0^T \int_0^1 \mathbf{h}(\mathbf{w}(x, t))^t \partial_x \mathbf{v}(x, t) dx dt. \end{aligned} \quad (6.36)$$

Note that if we use the fact that

$$\mathbf{h}(\mathbf{w}(x, t))^t \partial_x \mathbf{w}(x, t) = \partial_x (\phi(u) - g(u) q)$$

is a complete derivative, we see that

$$\begin{aligned} B(\mathbf{w}, \mathbf{w}) &= \frac{1}{2} \int_0^1 u^2(x, T) dx + \int_0^T \int_0^1 q^2(x, t) dx dt \\ &\quad - \frac{1}{2} \int_0^1 u_0^2(x) dx, \end{aligned} \quad (6.37)$$

and that $B(\mathbf{w}, \mathbf{w}) = 0$, by (6.35). As a consequence, we immediately obtain the following L²-stability result:

$$\frac{1}{2} \int_0^1 u^2(x, T) dx + \int_0^T \int_0^1 q^2(x, t) dx dt = \frac{1}{2} \int_0^1 u_0^2(x) dx.$$

This is the argument we have to mimic in order to prove Proposition 26.

The discrete case. Thus, we start by finding a *compact form* of equations (6.13) and (6.14). If we replace $v_{h,u}(x)$ and $v_{h,q}(x)$ by $v_{h,u}(x, t)$ and $v_{h,q}(x, t)$ in the equations (6.13) and (6.14), add them up, sum on j from 1 to N and integrate in time from 0 to T , we obtain

$$\begin{aligned} B_h(\mathbf{w}_h, \mathbf{v}_h) &= 0, \\ \forall \mathbf{v}_h(t) \in V_h^k \times V_h^k, \quad \forall t \in (0, T). \end{aligned} \quad (6.38)$$

where

$$\begin{aligned}
B_h(\mathbf{w}_h, \mathbf{v}_h) &= \int_0^T \int_0^1 \partial_t u_h(x, t) v_{h,u}(x, t) dx dt \\
&+ \int_0^T \int_0^1 q_h(x, t) v_{h,q}(x, t) dx dt \\
&- \int_0^T \sum_{1 \leq j \leq N} \hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}^t(t) [\mathbf{v}_h(t)]_{j+1/2} dt \\
&- \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} \mathbf{h}(\mathbf{w}_h(x, t))^t \partial_x \mathbf{v}_h(x, t) dx dt.
\end{aligned} \tag{6.39}$$

Next, we obtain an expression for $B_h(\mathbf{w}_h, \mathbf{w}_h)$. It is contained in the following result.

Lemma 30. *We have*

$$\begin{aligned}
B_h(\mathbf{w}_h, \mathbf{w}_h) &= \frac{1}{2} \int_0^1 u_h^2(x, T) dx \\
&+ \int_0^T \int_0^1 q_h^2(x, t) dx dt + \Theta_{T,C}([\mathbf{w}_h]) \\
&- \frac{1}{2} \int_0^1 u_h^2(x, 0) dx,
\end{aligned}$$

where $\Theta_{T,C}([\mathbf{w}_h])$ is defined in Proposition 26.

Next, since $B_h(\mathbf{w}_h, \mathbf{w}_h) = 0$, by (6.38), we get the equality

$$\frac{1}{2} \int_0^1 u_h^2(x, T) dx + \int_0^T \int_0^1 q_h^2(x, t) dx dt + \Theta_{T,C}([\mathbf{w}_h]) = \frac{1}{2} \int_0^1 u_h^2(x, 0) dx$$

from which Proposition 26 easily follows since

$$\frac{1}{2} \int_0^1 u_h^2(x, 0) dx \leq \frac{1}{2} \int_0^1 u_0^2(x) dx,$$

by (6.12). It remains to prove Lemma 30.

Proof of Lemma (30). After setting $\mathbf{v}_h = \mathbf{w}_h$ in (6.39), we get

$$\begin{aligned}
B(\mathbf{w}_h, \mathbf{w}_h) &= \frac{1}{2} \int_0^1 u_h^2(x, T) dx + \int_0^T \int_0^1 q_h^2(x, t) dx dt \\
&+ \int_0^T \Theta_{diss}(t) dt - \frac{1}{2} \int_0^1 u_h^2(x, 0) dx,
\end{aligned}$$

where $\Theta_{diss}(t)$ is given by

$$- \sum_{1 \leq j \leq N} \left\{ \hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}^t(t) [\mathbf{w}_h(t)]_{j+1/2} + \int_{I_j} \mathbf{h}(\mathbf{w}_h(x, t))^t \partial_x \mathbf{w}_h(x, t) dx \right\}.$$

It only remains to show that

$$\int_0^T \Theta_{diss}(t) dt = \Theta_{T,C}([\mathbf{w}_h]).$$

To do that, we proceed as follows. Since

$$\begin{aligned} \mathbf{h}(\mathbf{w}_h(x,t))^t \partial_x \mathbf{w}_h(x,t) b &= (f(u_h) - \sqrt{a(u_h)} q_h) \partial_x u_h - g(u_h) \partial_x q_h \\ &= \partial_x \left(\int^{u_h} f(s) ds - g(u_h) q_h \right) \\ &= \partial_x (\phi(u_h) - g(u_h) q_h) \\ &\equiv \partial_x H(\mathbf{w}_h(x,t)), \end{aligned}$$

we get

$$\begin{aligned} \Theta_{diss}(t) &= \sum_{1 \leq j \leq N} \left\{ [H(\mathbf{w}_h(t))]_{j+1/2} - \hat{\mathbf{h}}(\mathbf{w}_h)_{j+1/2}^t(t) [\mathbf{w}_h(t)]_{j+1/2} \right\} \\ &\equiv \sum_{1 \leq j \leq N} \left\{ [H(\mathbf{w}_h(t))] - \hat{\mathbf{h}}(\mathbf{w}_h)^t(t) [\mathbf{w}_h(t)] \right\}_{j+1/2} \end{aligned}$$

Since, by the definition of H ,

$$\begin{aligned} [H(\mathbf{w}_h(t))] &= [\phi(u_h(t))] - [g(u_h(t)) q_h(t)] \\ &= [\phi(u_h(t))] - [g(u_h(t)) \bar{q}_h(t) - [q_h(t)] \overline{g(u_h(t))}], \end{aligned}$$

and since $(\hat{h}_u, \hat{h}_q)^t = \hat{\mathbf{h}}$, we get

$$\begin{aligned} \Theta_{diss}(t) &= \sum_{1 \leq j \leq N} \left\{ [\phi(u_h(t))] - [g(u_h(t)) \bar{q}_h(t) - [u_h(t)] \hat{h}_u] \right\}_{j+1/2} \\ &\quad + \sum_{1 \leq j \leq N} \left\{ -[q_h(t)] \overline{g(u_h(t))} - [q_h(t)] \hat{h}_q \right\}_{j+1/2}, \end{aligned}$$

This is the crucial step to obtain the L^2 -stability of the LDG methods, since the above expression gives us key information about the form that the flux $\hat{\mathbf{h}}$ should have in order to make $\Theta_{diss}(t)$ a nonnegative quantity and hence enforce the L^2 -stability of the LDG methods. Thus, by taking $\hat{\mathbf{h}}$ as in (6.16), we get

$$\Theta_{diss}(t) = \sum_{1 \leq j \leq N} \left\{ [\mathbf{w}_h(t)]^t C [\mathbf{w}_h(t)] \right\}_{j+1/2},$$

and the result follows. This completes the proof.

This completes the proof of Proposition 26.

Proof of Theorem 27 In this section, we prove the error estimate of Theorem 27 which holds for the linear case $f'(\cdot) \equiv c$ and $a(\cdot) \equiv a$. To do that, we first show how to estimate the error between the solutions $\mathbf{w}_\nu = (u_\nu, q_\nu)^t$, $\nu = 1, 2$, of

$$\begin{aligned} \partial_t u_\nu + \partial_x (f(u_\nu) - \sqrt{a(u_\nu)} q_\nu) &= 0 \quad \text{in } (0, T) \times (0, 1), \\ q_\nu - \partial_x g(u_\nu) &= 0 \quad \text{in } (0, T) \times (0, 1), \\ u_\nu(t=0) &= u_{0,\nu}, \quad \text{on } (0, 1). \end{aligned}$$

Then, we mimic the argument in order to prove Theorem 27.

The continuous case as a model. By the definition of the form $B(\cdot, \cdot)$, (6.36), we have, for $\nu = 1, 2$,

$$B(\mathbf{w}_\nu, \mathbf{v}) = 0, \quad \forall \text{ smooth } \mathbf{v}(t), \quad \forall t \in (0, T).$$

Since in this case, the form $B(\cdot, \cdot)$ is bilinear, from the above equation we obtain the so-called *error equation*:

$$B(\mathbf{e}, \mathbf{v}) = 0, \quad \forall \text{ smooth } \mathbf{v}(t), \quad \forall t \in (0, T),$$

where $\mathbf{e} = \mathbf{w}_1 - \mathbf{w}_2$. Now, from (6.37), we get that

$$B(\mathbf{e}, \mathbf{e}) = \frac{1}{2} \int_0^1 e_u^2(x, T) dx + \int_0^T \int_0^1 e_q^2(x, t) dx dt - \frac{1}{2} \int_0^1 e_u^2(x, 0) dx,$$

and since $e_u(x, 0) = u_{0,1}(x) - u_{0,2}(x)$ and $B(\mathbf{e}, \mathbf{e}) = 0$, by the *error equation*, we immediately obtain the error estimate we sought:

$$\frac{1}{2} \int_0^1 e_u^2(x, T) dx + \int_0^T \int_0^1 e_q^2(x, t) dx dt = \frac{1}{2} \int_0^1 (u_{0,1}(x) - u_{0,2}(x))^2 dx \quad (6.40)$$

To prove Theorem 27, we only need to obtain a discrete version of this argument.

The discrete case. Since,

$$\begin{aligned} B_h(\mathbf{w}_h, \mathbf{v}_h) &= 0, \quad \forall \mathbf{v}_h(t) \in V_h \times V_h, \quad \forall t \in (0, T), \\ B_h(\mathbf{w}, \mathbf{v}_h) &= 0, \quad \forall \mathbf{v}_h(t) \in V_h \times V_h, \quad \forall t \in (0, T), \end{aligned}$$

by (6.38) and by equations (6.10) and (6.11), respectively, we immediately obtain our *error equation*:

$$B_h(\mathbf{e}, \mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h(t) \in V_h \times V_h, \quad \forall t \in (0, T),$$

where $\mathbf{e} = \mathbf{w} - \mathbf{w}_h$. Now, according to the continuous case argument, we should consider next the quantity $B_h(\mathbf{e}, \mathbf{e})$; however, since \mathbf{e} is not in the finite element space, it is more convenient to consider $B_h(P_h(\mathbf{e}), P_h(\mathbf{e}))$, where

$$P_h(\mathbf{e}(t)) = (P_h(e_u(t)), P_h(e_q(t)))$$

is the so-called L^2 -projection of $\mathbf{e}(t)$ into the finite element space $V_h^k \times V_h^k$. The L^2 -projection of the function p into V_h , $P_h(p)$, is defined as the only element of the finite element space V_h such that

$$\forall v_h \in V_h : \quad \int_0^1 (P_h(p)(x) - p(x)) v_h(x) dx = 0. \quad (6.41)$$

Note that, in fact $u_h(t=0) = P_h(u_0)$, by (6.15).

Thus, by Lemma 30, we have

$$\begin{aligned} B_h(P_h(\mathbf{e}), P_h(\mathbf{e})) &= \frac{1}{2} \int_0^1 |P_h(e_u(T))(x)|^2 dx \\ &\quad + \int_0^T \int_0^1 |P_h(e_q(t))(x)|^2 dx dt \\ &\quad + \Theta_{T,C}([P_h(\mathbf{e})]) \\ &\quad - \frac{1}{2} \int_0^1 |P_h(e_u(0))(x)|^2 dx, \end{aligned}$$

and since

$$P_h(e_u(0)) = P_h(u_0 - u_h(0)) = P_h(u_0) - u_h(0) = 0,$$

by (6.15) and (6.41), and

$$B_h(P_h(\mathbf{e}), P_h(\mathbf{e})) = B_h(P_h(\mathbf{e}) - \mathbf{e}, P_h(\mathbf{e})) = B_h(P_h(\mathbf{w}) - \mathbf{w}, P_h(\mathbf{e})),$$

by the *error equation*, we get

$$\begin{aligned} &\frac{1}{2} \int_0^1 |P_h(e_u(T))(x)|^2 dx + \int_0^T \int_0^1 |P_h(e_q(t))(x)|^2 dx dt + \Theta_{T,C}([P_h(\mathbf{e})]) \\ &= B_h(P_h(\mathbf{w}) - \mathbf{w}, P_h(\mathbf{e})). \end{aligned} \quad (6.42)$$

Note that since in our continuous model, the right-hand side is zero, we expect the term $B(P_h(\mathbf{w}) - \mathbf{w}, P_h(\mathbf{e}))$ to be small.

Estimating the right-hand side. To show that this is so, we must suitably treat the term $B(P_h(\mathbf{w}) - \mathbf{w}, P_h(\mathbf{e}))$.

Lemma 31. *For $\mathbf{p} = P_h(\mathbf{w}) - \mathbf{w}$, we have*

$$\begin{aligned} B_h(\mathbf{p}, P_h(\mathbf{e})) &= \frac{1}{2} \Theta_{T,C}(\bar{\mathbf{p}}) + \frac{1}{2} \int_0^T \int_0^1 |P_h(e_q(t))(x)|^2 dx dt \\ &\quad + \frac{1}{2} (\Delta x)^{2k} \int_0^T C_1(t) dt \\ &\quad + (\Delta x)^k \int_0^T C_2(t) \left\{ \int_0^1 |P_h(e_u(t))(x)|^2 dx \right\}^{1/2} dt, \end{aligned}$$

where

$$C_1(t) = 2c_k^2 \left\{ \left(\frac{(|c| + c_{11})^2}{c_{11}} \Delta x + 4|c_{12}|^2 d_k^2 \right) |u(t)|_{\hat{k}+1}^2 + 4a d_k^2 (\Delta x)^{2(\hat{k}-k)} |u(t)|_{\hat{k}+1}^2 \right\},$$

$$C_2(t) = \sqrt{8} c_k d_k \left\{ \sqrt{a} |c_{12}| |u(t)|_{\hat{k}+2} + a (\Delta x)^{(\hat{k}-k)} |u(t)|_{\hat{k}+2} \right\}.$$

where the constants c_k and d_k depend solely on k , and $\hat{k} = k$ except when the grids are uniform and k is even, in which case $\hat{k} = k + 1$.

Note how c_{11} appears in the denominator of $C_1(t)$. However, $C_1(t)$ remains bounded as c_{11} goes to zero since the convective numerical flux is an E-flux.

To prove this result, we will need the following auxiliary lemmas. We denote by $|u|_{H^{(k+1)}(J)}^2$ the integral over J of the square of the $(k + 1)$ -th derivative of u .

Lemma 32. For $\mathbf{p} = P_h(\mathbf{w}) - \mathbf{w}$, we have

$$\begin{aligned} |\overline{p_u}_{j+1/2}| &\leq c_k (\Delta x)^{\hat{k}+1/2} |u|_{H^{(k+1)}(J_{j+1/2})}, \\ [p_u]_{j+1/2} &\leq c_k (\Delta x)^{\hat{k}+1/2} |u|_{H^{(k+1)}(J_{j+1/2})}, \\ |\overline{p_q}_{j+1/2}| &\leq c_k \sqrt{a} (\Delta x)^{\hat{k}+1/2} |u|_{H^{(k+2)}(J_{j+1/2})}, \\ [p_q]_{j+1/2} &\leq c_k \sqrt{a} (\Delta x)^{\hat{k}+1/2} |u|_{H^{(k+2)}(J_{j+1/2})}, \end{aligned}$$

where $J_{j+1/2} = I_j \cup I_{j+1}$, the constant c_k depends solely on k , and $\hat{k} = k$ except when the grids are uniform and k is even, in which case $\hat{k} = k + 1$.

Proof. The two last inequalities follow from the first two and from the fact that $q = \sqrt{a} \partial_x u$. The two first inequalities with $\hat{k} = k$ follow from the definitions of $\overline{p_u}$ and $[p_u]$ and from the following estimate:

$$|P_h(u)(x_{j+1/2}^\pm) - u_{j+1/2}| \leq \frac{1}{2} c_k (\Delta x)^{\hat{k}+1/2} |u|_{H^{(k+1)}(J_{j+1/2})},$$

where the constant c_k depends solely on k . This inequality follows from the fact that

$$P_h(u)(x_{j+1/2}^\pm) - u_{j+1/2} = 0$$

when u is a polynomial of degree k and from a simple application of the Bramble-Hilbert lemma.

To prove the inequalities in the case in which $\hat{k} = k + 1$, we only need to show that if u is a polynomial of degree $k + 1$ for k even, then $\overline{p_u} = 0$. It is clear that it is enough to show this equality for the particular choice

$$u(x) = ((x - x_{j+1/2})/(\Delta x/2))^{k+1}.$$

To prove this, we recall that if P_ℓ denotes the Legendre polynomials of order ℓ :

- (i) $\int_{-1}^1 P_\ell(s) P_m(s) ds = \frac{2}{2\ell+1} \delta_{\ell m}$,
- (ii) $P_\ell(\pm 1) = (\pm 1)^\ell$, and
- (iii) $P_\ell(s)$ is a linear combination of odd (even) powers of s for odd (even) values of ℓ .

Since we are assuming that the grid is uniform, $\Delta x_j = \Delta x_{j+1} = \Delta x$, we can write, by (i), that

$$P_h(u)(x) = \sum_{0 \leq \ell \leq k} \frac{2\ell+1}{2} \left\{ \int_{-1}^1 P_\ell(s) u(x_j + \frac{1}{2}\Delta x s) ds \right\} P_\ell\left(\frac{x - x_j}{\Delta x/2}\right),$$

for $x \in I_j$. Hence, for our particular choice of u , we have that the value of $\overline{p_{u_{j+1/2}}}$ is given by

$$\begin{aligned} & \frac{1}{2} \sum_{0 \leq \ell \leq k} \frac{2\ell+1}{2} \int_{-1}^1 P_\ell(s) \cdot \{(s-1)^{k+1} P_\ell(1) + (s+1)^{k+1} P_\ell(-1)\} ds \\ &= \frac{1}{2} \sum_{0 \leq \ell, i \leq k} \frac{2\ell+1}{2} k+1(i) \int_{-1}^1 P_\ell(s) s^i \{(-1)^{k+1-i} P_\ell(1) + P_\ell(-1)\} ds \\ &= \frac{1}{2} \sum_{0 \leq \ell, i \leq k} \frac{2\ell+1}{2} k+1(i) \int_{-1}^1 P_\ell(s) s^i \{(-1)^{k+1-i} + (-1)^\ell\} ds, \end{aligned}$$

by (ii). When the factor $\{(-1)^{k+1-i} + (-1)^\ell\}$ is different from zero, $|k+1-i+\ell|$ is even and since k is also even, $|i-\ell|$ is odd. In this case, by (iii),

$$\int_{-1}^1 P_\ell(s) s^i ds = 0,$$

and so $\overline{p_{u_{j+1/2}}} = 0$. This completes the proof.

We will also need the following result that follows from a simple scaling argument.

Lemma 33. *We have*

$$|[P_h(p)]_{j+1/2}| \leq d_k (\Delta x)^{-1/2} \|P_h(p)\|_{L^2(J_{j+1/2})},$$

where $J_{j+1/2} = I_j \cup I_{j+1}$ and the constant d_k depends solely on k .

We are now ready to prove Lemma 31.

Proof of Lemma 31. To simplify the notation, let us set $\mathbf{v}_h = P_h \mathbf{e}$. By the definition of $B_h(\cdot, \cdot)$, we have

$$\begin{aligned}
B_h(\mathbf{p}, \mathbf{v}_h) &= \int_0^T \int_0^1 \partial_t p_u(x, t) v_{h,u}(x, t) dx dt \\
&\quad + \int_0^T \int_0^1 p_q(x, t) v_{h,q}(x, t) dx dt \\
&\quad - \int_0^T \sum_{1 \leq j \leq N} \hat{\mathbf{h}}(\mathbf{p})_{j+1/2}^t(\mathbf{v}_h(t))_{j+1/2} dt \\
&\quad - \int_0^T \sum_{1 \leq j \leq N} \int_{I_j} \mathbf{h}(\mathbf{p}(x, t))^t \partial_x \mathbf{v}_h(x, t) dx dt \\
&= - \int_0^T \sum_{1 \leq j \leq N} \hat{\mathbf{h}}(\mathbf{p})_{j+1/2}^t(\mathbf{v}_h(t))_{j+1/2} dt,
\end{aligned}$$

by the definition of the L^2 -projection (6.41).

Now, recalling that $\mathbf{p} = (p_u, p_q)^t$ and that $\mathbf{v}_h = (v_u, v_q)^t$, we have

$$\begin{aligned}
\hat{\mathbf{h}}(\mathbf{p})^t [\mathbf{v}_h(t)] &= (c \overline{p_u} - c_{11} [p_u]) [v_u] \\
&\quad + (-\sqrt{a} \overline{p_q} - c_{12} [p_q]) [v_u] \\
&\quad + (-\sqrt{a} \overline{p_u} + c_{12} [p_u]) [v_q] \\
&\equiv \theta_1 + \theta_2 + \theta_3.
\end{aligned}$$

By Lemmas 32 and 33, and writing J instead $J_{j+1/2}$, we get

$$\begin{aligned}
|\theta_1| &\leq c_k (\Delta x)^{k+1/2} |u|_{H^{k+1}(J)} (|c| + c_{11}) |[v_u]|, \\
|\theta_2| &\leq c_k d_k (\Delta x)^k (a |u|_{H^{\hat{k}+2}(J)} (\Delta x)^{\hat{k}-k} \\
&\quad + \sqrt{a} |c_{12}| |u|_{H^{\hat{k}+2}(J)}) \|v_u\|_{L^2(J)}, \\
|\theta_3| &\leq c_k d_k (\Delta x)^k (\sqrt{a} |u|_{H^{\hat{k}+1}(J)} (\Delta x)^{\hat{k}-k} \\
&\quad + |c_{12}| |u|_{H^{\hat{k}+1}(J)}) \|v_q\|_{L^2(J)}.
\end{aligned}$$

This is the crucial step for obtaining our error estimates. Note that the treatment of θ_1 is very different than the treatment of θ_2 and θ_3 . The reason for this difference is that the upper bound for θ_1 can be controlled by the form $\Theta_{T,C}([\mathbf{v}_h])$ - we recall that $\mathbf{v}_h = P_h(\mathbf{e})$. This is not the case for the upper bound for θ_2 because $\Theta_{T,C}[\mathbf{v}_h] \equiv 0$ if $c = 0$ nor it is the case for the upper bound for θ_3 because $\Theta_{T,C}[\mathbf{v}_h]$ does not involve the jumps $[v_q]$!

Thus, after a suitable application of Young's inequality and simple algebraic manipulations, we get

$$\begin{aligned} \hat{\mathbf{h}}(\mathbf{p})^t [\mathbf{v}_h(t)] &\leq \frac{1}{2} c_{11} [v_u]^2 + \frac{1}{4} \|v_q\|_{L^2(J)}^2 \\ &\quad + \frac{1}{4} C_{1,J}(t) (\Delta x)^{2k} + C_{2,J}(t) (\Delta x)^k \|v_u\|_{L^2(J)}, \end{aligned}$$

where

$$\begin{aligned} C_{1,J}(t) &= c_k^2 \left(\frac{(|c| + c_{11})^2}{c_{11}} \Delta x + 4 |c_{12}|^2 d_k^2 \right) |u(t)|_{H^{k+1}(J)}^2 \\ &\quad + 4a c_k^2 d_k^2 (\Delta x)^{2(k-k)} |u(t)|_{H^{k+1}(J)}^2, \end{aligned}$$

and

$$C_{2,J}(t) = c_k d_k \left\{ \sqrt{a} |c_{12}| |u(t)|_{H^{k+2}(J)} + a (\Delta x)^{(k-k)} |u(t)|_{H^{k+2}(J)} \right\}.$$

Since

$$B_h(\mathbf{p}, \mathbf{v}_h) \leq \int_0^T \sum_{1 \leq j \leq N} |\hat{\mathbf{h}}(\mathbf{p})_{j+1/2}^t(t) [\mathbf{v}_h(t)]_{j+1/2}| dt,$$

and since $J_{j+1/2} = I_j \cup I_{j+1}$, the result follows after simple applications of the Cauchy-Schwartz inequality. This completes the proof.

Conclusion. Combining the equation (6.42) with the estimate of Lemma 31, we easily obtain, after a simple application of Gronwall's lemma,

$$\begin{aligned} &\left\{ \int_0^1 |P_h(e_u(T))(x)|^2 dx + \int_0^T \int_0^1 |P_h(e_q(t))(x)|^2 dx dt + \Theta_{T,C}([P_h(\mathbf{e}))] \right\}^{1/2} \\ &\leq (\Delta x)^k \left\{ \sqrt{\int_0^T C_1(t) dt} + \int_0^T C_2(t) dt \right\}. \end{aligned}$$

Theorem 27 follows easily from this inequality, Lemma 33, and from the following simple approximation result:

$$\|p - P_h(p)\|_{L^2(0,1)} \leq g_k (\Delta x)^{k+1} |p|_{H^{(k+1)}(0,1)}$$

where g_k depends solely on k .

7 The LDG method for other nonlinear parabolic problems: Propagating surfaces

7.1 Introduction

In this chapter, we briefly show how to extend the LDG method to nonlinear second-order parabolic equations. We consider the following model problem:

$$\begin{aligned}\varphi_t + F(D\varphi, D^2\varphi) &= 0, & \text{in } (0, 1)^d \times (0, T), \\ \varphi(x, 0) &= \varphi_0(x), & \forall (x) \in (0, 1)^d,\end{aligned}$$

where we take periodic boundary conditions and assume that F is nonincreasing in the second variable. For the definition and properties of the *viscosity* solution of this and more general problems of this type, see the work by Crandall, Ishii, and Lions [29].

For simplicity, we only consider the two-dimensional case, $d = 2$:

$$\varphi_t + F(\varphi_x, \varphi_y, \varphi_{xx}, \varphi_{xy}, \varphi_{yy}) = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.1)$$

$$\varphi(x, 0) = \varphi_0(x), \quad \forall (x) \in (0, 1)^2, \quad (7.2)$$

with periodic boundary conditions. The material presented in this section is based in the work of Hu and Shu [43].

7.2 The method

The idea to extend the LDG method to this case, is to rewrite the problem (7.1), (7.2) for φ as follows:

$$\varphi_t = -F(u, v, p, q, r), \quad \text{in } (0, 1) \times (0, T), \quad (7.3)$$

$$\varphi(x, 0) = \varphi_0(x), \quad \forall x \in (0, 1). \quad (7.4)$$

where (u, v, p, q, r) solves the following problem:

$$u_t + F(u, v, p, q, r)_x = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.5)$$

$$v_t + H(u, v, p, q, r)_y = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.6)$$

$$p - u_x = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.7)$$

$$q - u_y = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.8)$$

$$r - v_x = 0, \quad \text{in } (0, 1)^2 \times (0, T), \quad (7.9)$$

$$u(x, y, 0) = (\varphi_0)_x(x, y), \quad \forall (x, y) \in (0, 1)^2, \quad (7.10)$$

$$v(x, y, 0) = (\varphi_0)_y(x, y), \quad \forall (x, y) \in (0, 1)^2. \quad (7.11)$$

Again, a straightforward application of the LDG method to the above problem produces an approximation $(u_h, v_h, p_h, q_h, r_h)$ to (u, v, p, q, r) . We can take each of the approximate solutions to be piecewise a polynomial of degree $k - 1$. Then, we define the approximation φ_h to φ by solving the problem (7.3), (7.4) in the manner described in the chapter on RKDG methods for multidimensional Hamilton-Jacobi equations.

7.3 Computational results

We present a couple of numerical results that display the good performance of the method. Our main purpose is to show that the method works well if both quadrangles and triangles are used.

First test problem. We consider the problem of a propagating surface:

$$\begin{cases} \varphi_t - (1 - \varepsilon K) \sqrt{1 + \varphi_x^2 + \varphi_y^2} = 0, & 0 < x < 1, 0 < y < 1 \\ \varphi(x, y, 0) = 1 - \frac{1}{4}(\cos(2\pi x - 1))(\cos(2\pi y - 1)) \end{cases} \quad (7.12)$$

where K is the mean curvature defined by

$$K = -\frac{\varphi_{xx}(1 + \varphi_y^2) - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}(1 + \varphi_x^2)}{(1 + \varphi_x^2 + \varphi_y^2)^{\frac{3}{2}}}, \quad (7.13)$$

and ε is a small constant. Periodic boundary condition is used.

This problem was studied in [72] by using the finite difference ENO schemes.

We first use a uniform rectangular mesh of 50×50 elements and the local Lax-Friedrichs flux. The results of $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$ are presented in Fig. 7.1 and Fig. 7.2, respectively. Notice that the surface at $T = 0$ is shifted downward by 0.35 in order to show the detail of the solution at $T = 0.3$.

Next we use a triangulation shown in Fig. 7.3. We refine the mesh around the center of domain where the solution develops discontinuous derivatives (for the $\varepsilon = 0$ case). There are 2146 triangles and 1128 nodes in this triangulation. The solutions are displayed in Fig. 7.4 and Fig. 7.5, respectively, for $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$. Notice that we again shift the solution at $T = 0.0$ downward by 0.35 to show the detail of the solutions at later time.

Second test problem. The problem of a propagating surface on a unit disk. The equation is the same as (7.12) in the previous example, but it is solved on a unit disk $x^2 + y^2 < 1$ with an initial condition

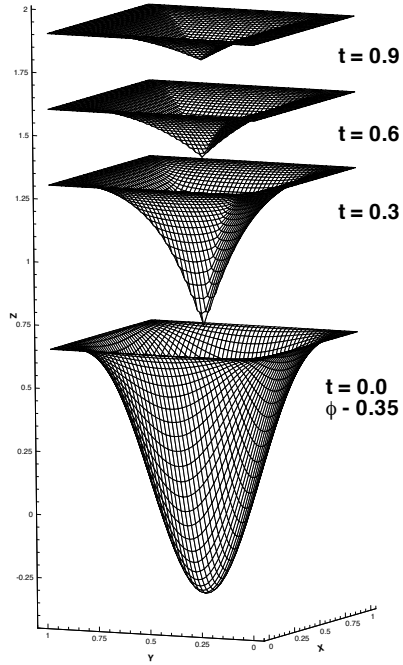
$$\varphi(x, y, 0) = \sin\left(\frac{\pi(x^2 + y^2)}{2}\right)$$

and a Neumann type boundary condition $\nabla\varphi = 0$.

It is difficult to use rectangular meshes for this problem. Instead we use the triangulation shown in Fig. 7.6. Notice that we have again refined the mesh near the center of the domain where the solution develops discontinuous derivatives. There are 1792 triangles and 922 nodes in this triangulation. The solutions with $\varepsilon = 0$ are displayed in Fig. 7.7. Notice that the solution at $t = 0$ is shifted downward by 0.2 to show the detail of the solution at later time.

The solution with $\varepsilon = 0.1$ are displayed in Fig. 7.8. Notice that the solution at $t = 0$ is again shifted downward by 0.2 to show the detail of the solution at later time.

P², 50x50 elements



P³, 50x50 elements

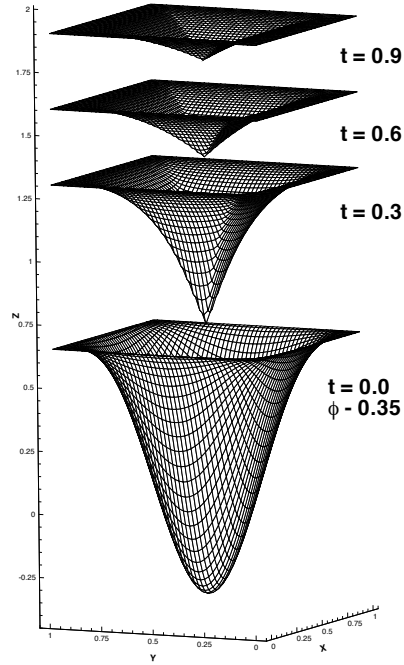
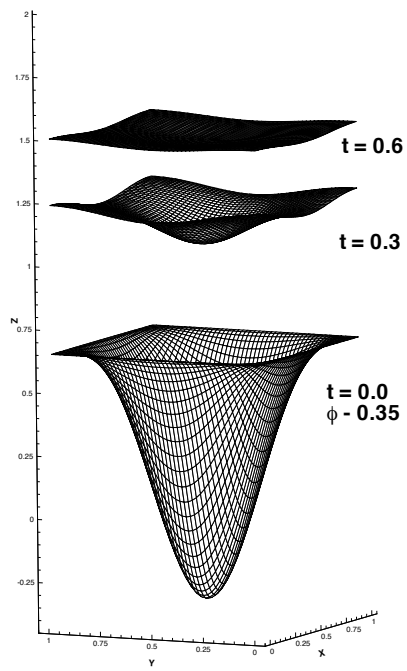


Fig. 7.1. Propagating surfaces, rectangular mesh, $\varepsilon = 0$.

7.4 Concluding remarks

We have shown, briefly, how to extend the LDG method originally devised for nonlinear convection-diffusion equations to second-order parabolic equations that have a viscosity solution. We have shown that the method works well without slope limiting and that it works well in both quadrangles and triangles.

P^2 , 50x50 elements



P^3 , 50x50 elements

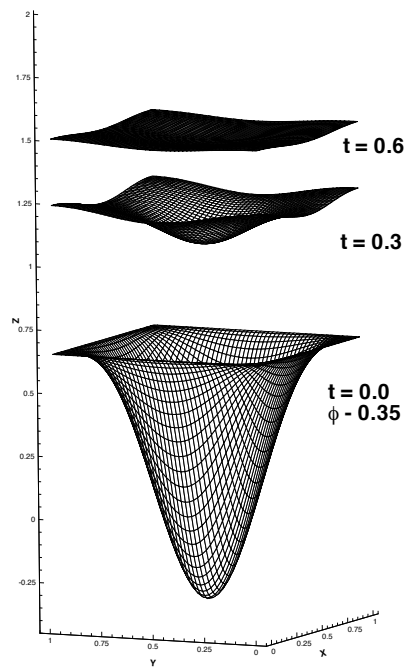


Fig. 7.2. Propagating surfaces, rectangular mesh, $\varepsilon = 0.1$.

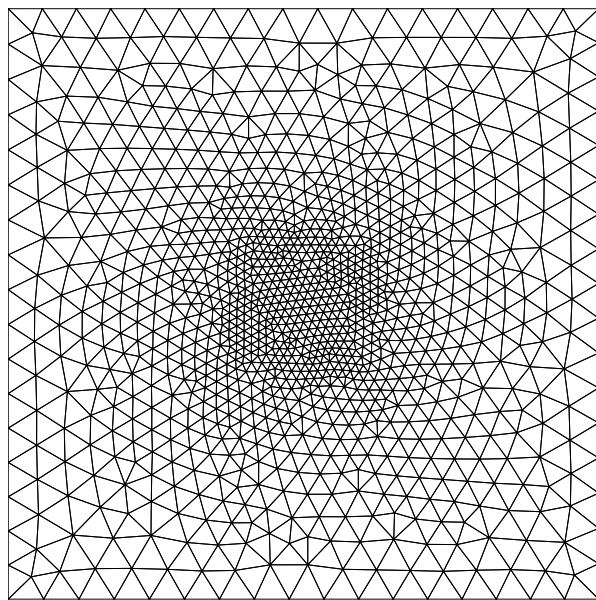
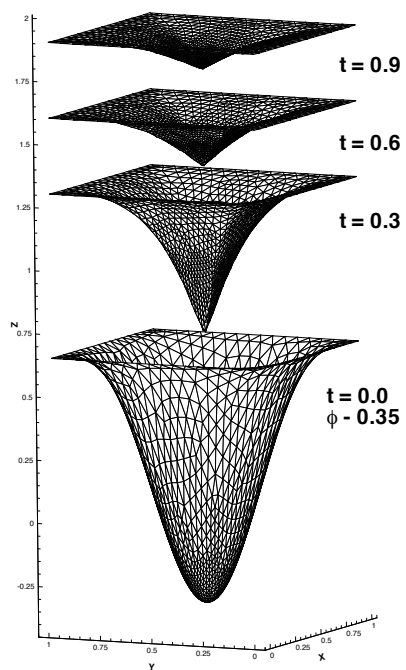


Fig. 7.3. Triangulation used for the propagating surfaces.

P^2 , triangles



P^3 , triangles

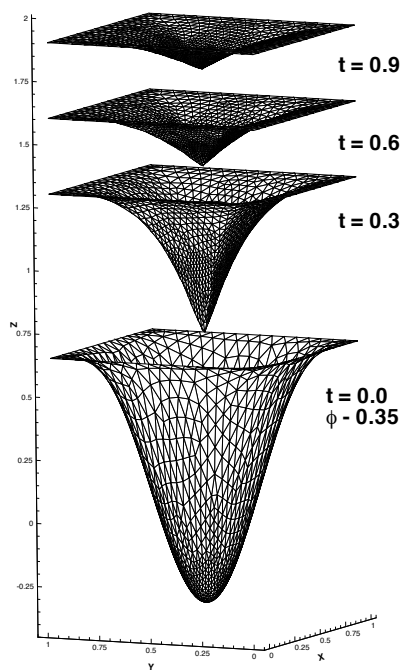
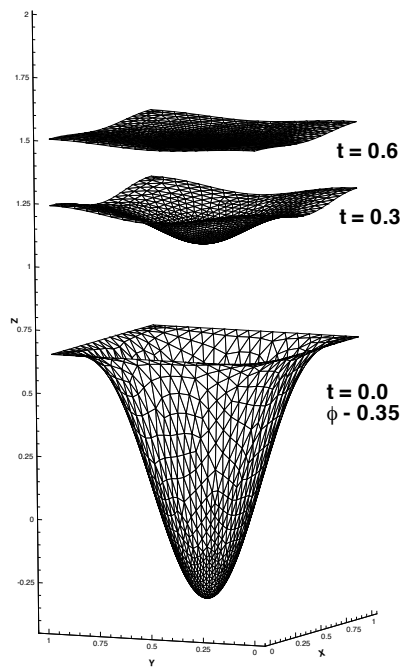


Fig. 7.4. Propagating surfaces, triangular mesh, $\varepsilon = 0$.

P^2 , triangles



P^3 , triangles

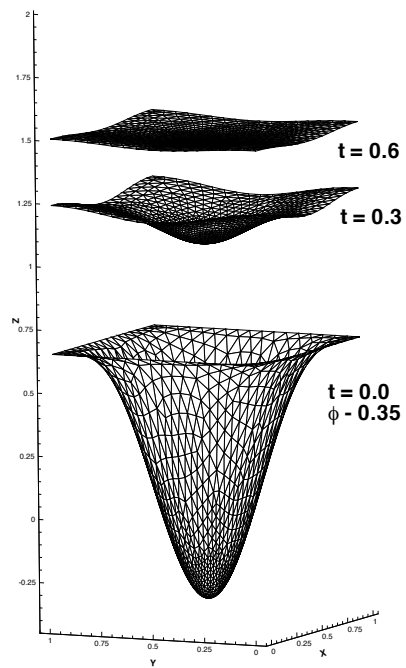


Fig. 7.5. Propagating surfaces, triangular mesh, $\varepsilon = 0.1$.

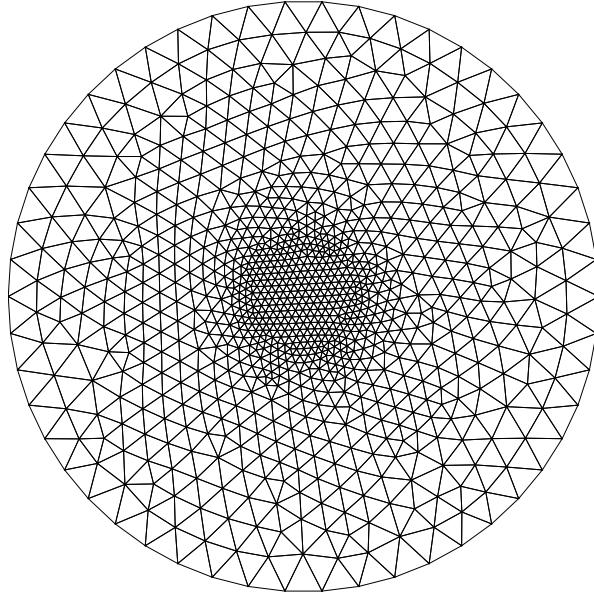


Fig. 7.6. Triangulation for the propagating surfaces on a disk.

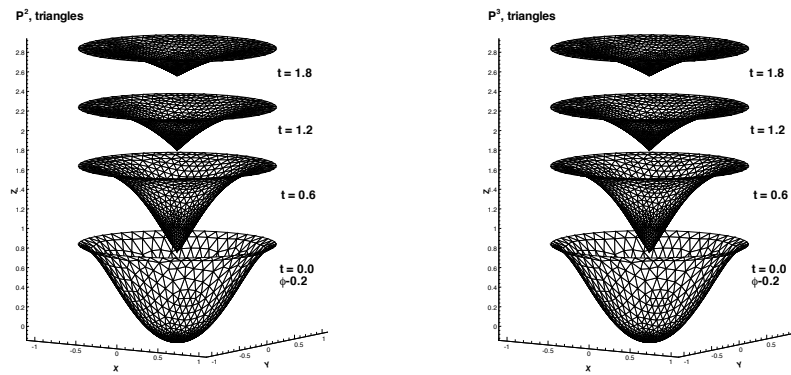


Fig. 7.7. Propagating surfaces on a disk, triangular mesh, $\varepsilon = 0$.

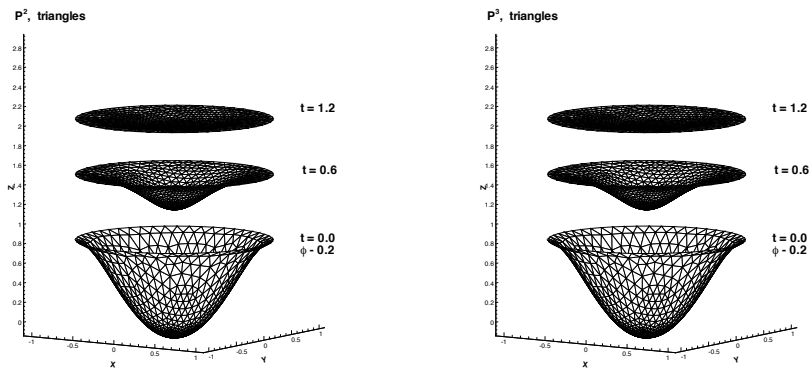


Fig. 7.8. Propagating surfaces on a disk, triangular mesh, $\varepsilon = 0.1$.

References

1. H.L. Atkins and C.-W. Shu. Quadrature-free implementation of discontinuous Galerkin methods for hyperbolic equations. Technical Report 96-51, ICASE, 1996. To appear in AIAA J.
2. I. Babuška, C.E. Baumann, and J.T. Oden. A discontinuous *hp* finite element method for diffusion problems: 1-D analysis. Technical Report 22, TICAM, 1997.
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 and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138:251-285, 1997.
5. 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. In *Proceedings of the Second European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, 1997. Antwerpen, Belgium.
6. C.E. Baumann. *An hp-adaptive discontinuous Galerkin method for computational fluid dynamics*. PhD thesis, The University of Texas at Austin, 1997.
7. C.E. Baumann and J.T. Oden. A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.* To appear.
8. C.E. Baumann and J.T. Oden. A discontinuous *hp* finite element method for the Navier-Stokes equations. In *10th. International Conference on Finite Element in Fluids*, 1998.
9. C.E. Baumann and J.T. Oden. A discontinuous *hp* finite element method for the solution of the Euler equation of gas dynamics. In *10th. International Conference on Finite Element in Fluids*, 1998.
10. K.S. Bey and J.T. Oden. A Runge-Kutta discontinuous Galerkin finite element method for high speed flows. *10th. AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24-27*, 1991.
11. R. Biswas, K.D. Devine, and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255-283, 1994.
12. G. Chavent and B. Cockburn. The local projection P^0 P^1 -discontinuous-Galerkin finite element method for scalar conservation laws. *M²AN*, 23:565-592, 1989.
13. G. Chavent and G. Salzano. A finite element method for the 1D water flooding problem with gravity. *J. Comput. Phys.*, 45:307-344, 1982.
14. 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.
15. 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.
16. P. Ciarlet. *The finite element method for elliptic problems*. North Holland, 1975.
17. B. Cockburn. An introduction to the discontinuous Galerkin method for convection-dominated problems. In *Advanced numerical approximation of nonlinear hyperbolic equations*, A. Quarteroni, editor, Lecture Notes in Mathematics, CIME subseries. Springer Verlag. To appear.

18. B. Cockburn and P.-A. Gresho. A priori error estimates for numerical methods for scalar conservation laws. part I: The general approach. *Math. Comp.*, 65:533–573, 1996.
19. B. Cockburn and P.A. Gresho. Error estimates for finite element methods for nonlinear conservation laws. *SIAM J. Numer. Anal.*, 33:522–554, 1996.
20. 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.
21. 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.
22. B. Cockburn, M. Luskin, C.-W. Shu, and E. Süli. A priori error estimates for the discontinuous Galerkin method. in preparation.
23. B. Cockburn and C. Schwab. *hp*-error analysis for the local discontinuous Galerkin method. In preparation.
24. 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.
25. B. Cockburn and C.W. Shu. The P^1 -RKDG method for two-dimensional Euler equations of gas dynamics. Technical Report 91-32, ICASE, 1991.
26. B. Cockburn and C.W. Shu. The Runge-Kutta local projection P^1 -discontinuous Galerkin method for scalar conservation laws. *M²AN*, 25:337–361, 1991.
27. B. Cockburn and C.W. Shu. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM J. Numer. Anal.*, 35:2440–2463, 1998.
28. B. Cockburn and C.W. Shu. The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.
29. M.G. Crandall and H. Ishii and P.L. Lions. User's guide to viscosity solutions of second-order partial differential equations. *Bull. Amer. Math. Soc.*, 27:1–67, 1992.
30. H.L. deCougny, K.D. Devine, J.E. Flaherty, R.M. Loy, C. Ozturan, and M.S. Shephard. Load balancing for the parallel adaptive solution of partial differential equations. *Appl. Numer. Math.*, 16:157–182, 1994.
31. K.D. Devine and J.E. Flaherty. Parallel adaptive *hp*-refinement techniques for conservation laws. *Appl. Numer. Math.*, 20:367–386, 1996.
32. K.D. Devine, J.E. Flaherty, R.M. Loy, and S.R. Wheat. Parallel partitioning strategies for the adaptive solution of conservation laws. In I Babuška, W.D. Henshaw, J.E. Hopcroft, J.E. Olinger, and T. Tezduyar, editors, *Modeling, mesh generation, and adaptive numerical methods for partial differential equations*, volume 75, pages 215–242, 1995.
33. K.D. Devine, J.E. Flaherty, S.R. Wheat, and A.B. Maccabe. A massively parallel adaptive finite element method with dynamic load balancing. In *Proceedings Supercomputing'93*, pages 2–11, 1993.
34. 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.
35. 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.

36. 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.
37. K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems V: Long time integration. *SIAM J. Numer. Anal.*, 32:1750–1762, 1995.
38. K. Eriksson, C. Johnson, and V. Thomée. Time discretization of parabolic problems by the discontinuous Galerkin method. *RAIRO, Anal. Numér.*, 19:611–643, 1985.
39. R.S. Falk and G.R. Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM J. Numer. Anal.* To appear.
40. J.E. Flaherty, R.M. Loy, M.S. Shephard, B.K. Szymanski, J.D. Teresco, and L.H. Ziantz. Adaptive refinement with octree load-balancing for the parallel solution of three-dimensional conservation laws. Technical report, IMA Preprint Series # 1483, 1997.
41. J. Goodman and R. LeVeque. On the accuracy of stable schemes for 2D scalar conservation laws. *Math. Comp.*, 45:15–21, 1985.
42. P. Houston, C. Schwab, and E. Süli. Stabilized *hp*-finite element methods for hyperbolic problems. *SIAM J. Numer. Anal.* To appear.
43. C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.* To appear.
44. T. Hughes and A. Brook. Streamline upwind-Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.
45. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, I. *Comput. Methods Appl. Mech. Engrg.*, 54:223–234, 1986.
46. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, II. *Comput. Methods Appl. Mech. Engrg.*, 54:341–355, 1986.
47. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, III. *Comput. Methods Appl. Mech. Engrg.*, 58:305–328, 1986.
48. T. Hughes, L.P. Franca, M. Mallet, and A. Misukami. A new finite element formulation for computational fluid dynamics, IV. *Comput. Methods Appl. Mech. Engrg.*, 58:329–336, 1986.
49. T. Hughes and M. Mallet. A high-precision finite element method for shock-tube calculations. *Finite Element in Fluids*, 6:339–, 1985.
50. 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.
51. G. Jiang and C.-W. Shu. On cell entropy inequality for discontinuous Galerkin methods. *Math. Comp.*, 62:531–538, 1994.
52. C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46:1–26, 1986.
53. C. Johnson and J. Saranen. Streamline diffusion methods for problems in fluid mechanics. *Math. Comp.*, 47:1–18, 1986.
54. 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.

55. 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.
56. D.S. Kershaw, M.K. Prasad, and M.J. Shaw and J.L. Milovich. 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous Galerkin method. *Comput. Methods Appl. Mech. Engrg.*, 158:81–116, 1998.
57. D.A. Kopriva. A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations. Technical Report 97-66, Florida State University-SCRI, 1997.
58. P. LeSaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical aspects of finite elements in partial differential equations*, pages 89–145. Academic Press, 1974.
59. Q. Lin, N. Yan, and A.-H. Zhou. An optimal error estimate of the discontinuous Galerkin method. *Journal of Engineering Mathematics*, 13:101–105, 1996.
60. Q. Lin and A.-H. Zhou. Convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *Acta Math. Sci.*, 13:207–210, 1993.
61. W. B. Lindquist. Construction of solutions for two-dimensional Riemann problems. *Comp. & Maths. with Appls.*, 12:615–630, 1986.
62. 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.
63. I. Lomtev and G.E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations. *Int. J. Num. Meth. Fluids*. in press.
64. I. Lomtev and G.E. Karniadakis. A discontinuous spectral/ *hp* element Galerkin method for the Navier-Stokes equations on unstructured grids. In *Proc. IMACS WC'97*, 1997. Berlin, Germany.
65. I. Lomtev and G.E. Karniadakis. Simulations of viscous supersonic flows on unstructured *hp*-meshes. *AIAA-97-0754*, 1997. 35th. Aerospace Sciences Meeting, Reno.
66. I. Lomtev, C.W. Quillen, and G.E. Karniadakis. Spectral/*hp* methods for viscous compressible flows on unstructured 2D meshes. *J. Comput. Phys.*, 144:325–357, 1998.
67. E.O. Macagno and T. Hung. Computational and experimental study of a captive annular eddy. *J.F.M.*, 28:43–XX, 1967.
68. X. Makridakis and I. Babuška. On the stability of the discontinuous Galerkin method for the heat equation. *SIAM J. Numer. Anal.*, 34:389–401, 1997.
69. Newmann. *A Computational Study of Fluid/Structure Interactions: Flow-Induced Vibrations of a Flexible Cable*. PhD thesis, Princeton University, 1996.
70. J.T. Oden, Ivo Babuška, and C.E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *J. Comput. Phys.*, 146:491–519, 1998.
71. S. Osher. Riemann solvers, the entropy condition and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235, 1984.
72. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation. *J. Comput. Phys.*, 79:12–49, 1988.
73. S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28:907–922, 1991.
74. C. Ozturan, H.L. deCougny, M.S. Shephard, and J.E. Flaherty. Parallel adaptive mesh refinement and redistribution on distributed memory computers. *Comput. Methods Appl. Mech. Engrg.*, 119:123–137, 1994.

75. 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.
76. W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
77. G.R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Math. Comp.*, 50:75–88, 1988.
78. E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29:867–884, 1992.
79. S.J. Sherwin and G. Karniadakis. Tetrahedral *hp*-finite elements: Algorithms and flow simulations. *J. Comput. Phys.*, 124:314–345, 1996.
80. C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
81. C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.*, 83:32–78, 1989.
82. C.W. Shu. TVB uniformly high order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.
83. C.W. Shu. TVD time discretizations. *SIAM J. Sci. Stat. Comput.*, 9:1073–1084, 1988.
84. M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solution to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.
85. C. Tong and G.Q. Chen. Some fundamental concepts about systems of two spatial dimensional conservation laws. *Acta Mathematica Scientia (English Ed.)*, 6:463–474, 1986.
86. C. Tong and Y.-X. Zheng. Two dimensional Riemann problems for a single conservation law. *Trans. Amer. Math. Soc.*, 312:589–619, 1989.
87. J.R. Trujillo. *Effective high-order vorticity-velocity formulation*. PhD thesis, Princeton University, 1997.
88. B. van Leer. Towards the ultimate conservation difference scheme, II. *J. Comput. Phys.*, 14:361–376, 1974.
89. B. van Leer. Towards the ultimate conservation difference scheme, V. *J. Comput. Phys.*, 32:1–136, 1979.
90. D. Wagner. The Riemann problem in two space dimensions for a single conservation law. *SIAM J. Math. Anal.*, 14:534–559, 1983.
91. T.C. Warburton, I. Lomtev, R.M. Kirby, and G.E. Karniadakis. A discontinuous Galerkin method for the Navier-Stokes equations in hybrid grids. In M. Hafez and J.C. Heirich, editors, *10th. International Conference on Finite Elements in Fluids, Tucson, Arizona*, 1998.
92. P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–173, 1984.
93. A.-H. Zhou and Q. Lin. Optimal and superconvergence estimates of the finite element method for a scalar hyperbolic equation. *Acta Math. Sci.*, 14:90–94, 1994.