COMPUTATIONAL METHODS IN APPLIED MATHEMATICS, Vol. 10 (2010), No. 3, pp. 326–342

A PROBLEM-INDEPENDENT SLOPE LIMITING ALGORITHM FOR THE RUNGE-KUTTA DISCONTINUOUS GALERKIN METHOD

S.A. TOKAREVA[1]

Abstract — This paper deals with the new algorithm of slope limiting in the Runge-Kutta discontinuous Galerkin (RKDG) method. The slope limiting is applied at each intermediate step of the Runge-Kutta process to guarantee the monotonicity of the resulting RKDG scheme. The standard formulation of the RKDG method assumes a manual prescription of the special parameter used in the limiting procedure. Such definition of the limiter makes the method problem-dependent, which is disadvantageous for practical computations. A new problem-independent way of estimating the limiting parameter is proposed and its performance in the second- and third-order RKDG methods is studied in this paper.

2000 Mathematics Subject Classification: 65M08,65M60,65Z05,76G25,76j20.

Keywords: discontinuous Galerkin method, gas dynamics, hyperbolic equations, slope limiting.

Introduction

In this paper, we present a modification of the slope limiting procedure which is an integral part of the Runge-Kutta discontinuous Galerkin (RKDG) method [1–4]. The application of the slope limiter is necessary at each intermediate Runge-Kutta time step to guarantee the monotonicity of the resulting RKDG method. The formulation of the TVB (total variation bounded) slope limiter proposed in [1] and used in the RKDG method assumes a manual prescription of the special M parameter used in the limiting procedure and defining the maximum possible slope of the solution in the case of the piecewise-linear solution space approximation for which no restrictive TVD (total variation diminishing) limiter is applied. Thus, theoretically, for every computation it is possible to find such a value for this parameter that will preserve the RKDG method high order and maintain its monotonicity by suppressing the spurious oscillations of the solution. In practice, a proper value for M must be defined before the computation start since the existing theoretical ways of its real-time estimation are hardly applicable for multidimentional problems, especially for those ivolving discontinuous solutions. Such definition of the limiter makes the method problem-dependent, which is disadvantageous for practical computations. Attempts have been made to construct a problem-independent limiter (see, for example, [5]).

In this paper, we propose a new problem-independent method for estimating the limiting M parameter. A simple but efficient algorithm is based on the approximation of the solution gradient which is then used to estimate the maximum possible solution slope (which is

[1] *Bauman Moscow State Technical University, Applied Mathematics Department,* 2-nd Baumanskaya 5, 105005 Moscow, Russia. E-mail: s.a.tokareva@gmail.com

exactly the M value) on every edge of each mesh triangle independently. This is done at every intermediate Runge-Kutta time step. As a result, the slope limiter is continuously adjusted during the computations both in space and time. We study the performance of the proposed algorithm in the second- and third-order RKDG methods. The new algorithm proved to be robust for a large variety of gas-dynamical problems, which makes the RKDG method more flexible for practical simulations.

1. General scheme of the RKDG method for convection-diffusion systems

Consider the scheme of the RKDG method [1–4] for the following model convection-diffusion system:

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{s=1}^{2} \frac{\partial \mathbf{F}_s(\mathbf{u})}{\partial x_s} = \sum_{s=1}^{2} \frac{\partial \mathbf{R}_s(\mathbf{u}, \nabla \mathbf{u})}{\partial x_s} \tag{1.1}$$

with appropriate initial and boundary conditions, where $\mathbf{u}$ is the vector of unknown variables, $\mathbf{F}_s(\mathbf{u})$, $s = 1, 2$ are the inviscid fluxes, and $\mathbf{R}_s$, $s = 1, 2$ are viscid fluxes.

System (1.1) is first transformed into a system of 1st order PDEs by introducing additional variables

$$\mathbf{q}_1 = \frac{\partial \mathbf{u}}{\partial x_1},\ \mathbf{q}_2 = \frac{\partial \mathbf{u}}{\partial x_2}$$

so that the solution gradient can be considered independently.

Then system (1.1) can be rewritten in the form

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \operatorname{div} \mathbf{F}(\mathbf{u}) &= \operatorname{div} \mathbf{R}(\mathbf{u}, \mathbf{q}_1, \mathbf{q}_2); \\ \mathbf{q}_1 &= \operatorname{div} \mathbf{Q}_1(\mathbf{u}); \\ \mathbf{q}_2 &= \operatorname{div} \mathbf{Q}_2(\mathbf{u}), \end{aligned} \tag{1.2}$$

where $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2]$, $\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2]$, $\mathbf{Q}_1 = [\mathbf{u}, \mathbf{0}]$, $\mathbf{Q}_2 = [\mathbf{0}, \mathbf{u}]$.

Consider an arbitrary triangulation T_h of the computational domain Ω. The approximate solution $\mathbf{u}_h$ and $\mathbf{q}_h$ inside each element $K \in T_h$ can be written as a decomposition over the system of basis functions of maximum degree k $\{\varphi_i\}, i = 1, \dots, n$

$$\begin{aligned} \mathbf{u}_h(x_1, x_2, t) &= \sum_{i=1}^{n} \mathbf{u}_i(t) \varphi_i(x_1, x_2); \\ \mathbf{q}_{1h}(x_1, x_2, t) &= \sum_{i=1}^{n} \mathbf{q}_1^i(t) \varphi_i(x_1, x_2); \\ \mathbf{q}_{2h}(x_1, x_2, t) &= \sum_{i=1}^{n} \mathbf{q}_2^i(t) \varphi_i(x_1, x_2), \end{aligned}$$

where $\mathbf{u}_i(t)$, $\mathbf{q}_1^i(t)$ and $\mathbf{q}_2^i(t)$ are unknown time-dependent coefficients (degrees of freedom) which need to be defined.

Multiplying Eqs. (1.2) sequentially by the basis functions $\varphi_i, i = 1, \dots, n$ and integrating the results with respect to the element K, after integration by parts we obtain the system of ODEs for the degrees of freedom $\mathbf{u}_i(t)$ and the explicit expressions for $\mathbf{q}_1^i(t)$ and $\mathbf{q}_2^i(t)$

$$\sum_{k=1}^{n} \frac{d\mathbf{u}_k(t)}{dt}(\varphi_i, \varphi_k) = \int\limits_K \mathbf{F}(\mathbf{u}_h)\,\mathrm{grad}\,\varphi_i\,dV - \int\limits_{\partial K} \tilde{\mathbf{f}}(\mathbf{u}_h)\varphi_i\,d\Gamma$$

$$- \int\limits_K \mathbf{R}(\mathbf{u}_h, \mathbf{q}_{1h}, \mathbf{q}_{2h})\nabla\varphi_i\,dx + \int\limits_{\partial K} \tilde{\mathbf{r}}(\mathbf{u}_h, \mathbf{q}_{1h}, \mathbf{q}_{2h})\varphi_i\,d\Gamma, \quad (1.3)$$

$$\sum_{k=1}^{n} \mathbf{q}_1^k(t)(\varphi_i, \varphi_k) = \int\limits_{\partial K} \tilde{\mathbf{q}}_1(\mathbf{u}_h)\varphi_i\,d\Gamma - \int\limits_K \mathbf{Q}_1(\mathbf{u}_h)\nabla\varphi_i\,dx;$$

$$\sum_{k=1}^{n} \mathbf{q}_2^k(t)(\varphi_i, \varphi_k) = \int\limits_{\partial K} \tilde{\mathbf{q}}_2(\mathbf{u}_h)\varphi_i\,d\Gamma - \int\limits_K \mathbf{Q}_2(\mathbf{u}_h)\nabla\varphi_i\,dx.$$

Here $\tilde{\mathbf{f}}(\mathbf{u}_h)$, $\tilde{\mathbf{r}}(\mathbf{u}_h, \mathbf{q}_{1h}, \mathbf{q}_{2h})$, $\tilde{\mathbf{q}}_1(\mathbf{u}_h)$, $\tilde{\mathbf{q}}_2(\mathbf{u}_h)$ are numerical fluxes approximating the physical inviscid and viscid fluxes through the cell interface K and depending on the boundary extrapolated solution values inside and outside of K. As a numerical flux for the inviscid term $\tilde{\mathbf{f}}(\mathbf{u}_h)$ one can use any monotonic flux consistent with $\mathbf{F}(\mathbf{u}_h)$ (see [6] for examples).

For the approximation of the viscid fluxes $\tilde{\mathbf{r}}(\mathbf{u}_h, \mathbf{q}_{1h}, \mathbf{q}_{2h})$, $\tilde{\mathbf{q}}_1(\mathbf{u}_h)$ and $\tilde{\mathbf{q}}_2(\mathbf{u}_h)$ the centered fluxes can be used in the form

$$\mathbf{H}(\mathbf{a}, \mathbf{b}) = \frac{1}{2}\Big(\mathbf{H}(\mathbf{a}) + \mathbf{H}(\mathbf{b})\Big).$$

An explicit Runge-Kutta method of order $(k + 1)$ is then applied to solve the system of ODEs (1.3), resulting in the RKDG method of accuracy order $(k + 1)$ in both space and time. A specially designed limiting procedure must be incorporated into each intermediate step of the Runge-Kutta method to provide scheme monotonicity.

In this paper, we use either piecewise-linear or piecewise-quadratic solution space approximations, leading to the methods of the second and third order in both time and space, respectively.

2. Standard slope limiter

As mentioned above, the RKDG method includes the implementation of the special slope limiter at each intermediate Runge-Kutta step to guarantee monotonicity [1].

The slope limiter proposed in [1] is formulated for piecewise-linear functions and in the case of a piecewise-polynomial approximation of higher order the limiter is applied to the projection of the solution onto the space of piecewise-linear functions. This limiter can be used in two versions: TVD (total variation diminishing) and TVB (total variation bounded). The first variant (TVD) is based on the well-known “minmod” limiting of the slopes of the solution at certain points of the mesh elements, e.g., centers of the edges in the case of triangular meshes. This limiter is problem-independent, it preserves the TVD property of the numerical solution, however, it can decrease the order of accuracy in the vicinity of local extrema due to the restrictions of the “minmod” procedure. This difficulty can be overcome by considering the second version of the limiter, namely the TVB limiter, which is a modification of the TVD limiter allowing the numerical solution slope to be restricted by a certain manually defined value M. As a result, there is no decrease in the accuracy at local extrema, but the limiter becomes problem-dependent: the M parameter needs to be redefined for every new computation with changed mesh and/or flow data, which makes such a limiting procedure inefficient in practical computations.

In [1], it is stated that the gas-dynamical computations presented there were implemented using $M = 50\Delta x^2$, where Δx is the mesh size, and that the influence of the value of M on the solution quality decreases as the degree of the basis polynomials is increased. However, our experience suggests that the quantitative characteristics of the solution of Euler and Navier-Stokes equations depend dramatically on M. To illustrate this fact, we present the computational results for the flow around a semicircular cylinder at zero angle of attack, Mach number $\mathrm{M} = 0.05$, and Reynolds number $\mathrm{Re} = 90\,000$. We compute the aerodynamic coefficients C_x and C_y, which are the drag and lift coefficients, respectively, and are defined by the following relations:

$$C_x(t) = \frac{F_x(t)}{1/2\rho V_\infty^2 S}, \; C_y(t) = \frac{F_y(t)}{1/2\rho_\infty V_\infty^2 S},$$

where ρ_∞ and V_∞ are the freestream density and velocity, S is the reference area (or length in the case of two-dimensional profiles), $F_x(t)$ and $F_y(t)$ are the projections of the aerodynamic force $\vec{F}(t)$ on the coordinate axes, and $\vec{F}(t)$ is defined by the integral of the flow pressure $p(t)$ over the profile surface

$$\vec{F}(t) = -\int\limits_{\Omega} p(t)\vec{n}\, d\Omega,$$

with $\vec{n}$ being the unit normal to the surface element $d\Omega$.

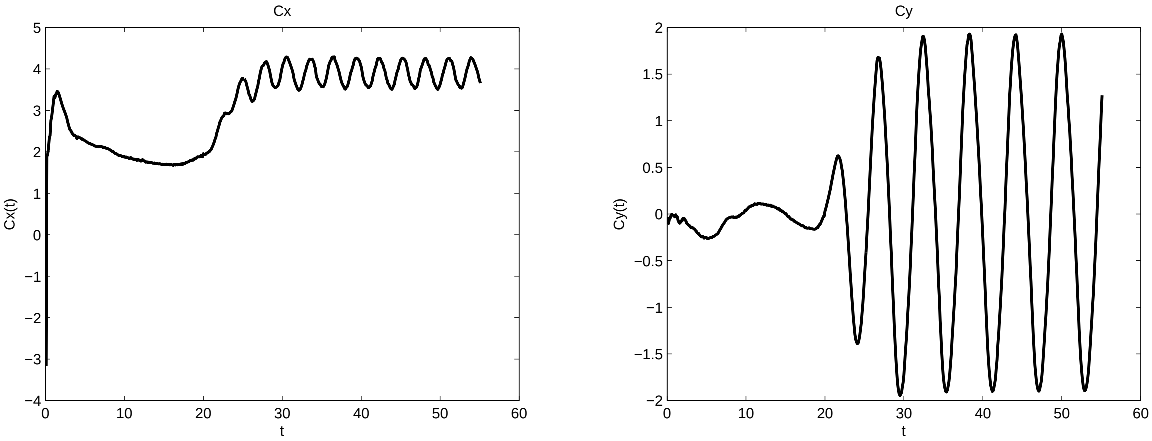


Fig. 2.1. Drag and lift coefficients at $M = 0.5(\Delta x)^2$: average $C_x = 3.7$

Figures 2.1–2.3 show the time dependence for the drag and lift coefficients computed with different values of M.

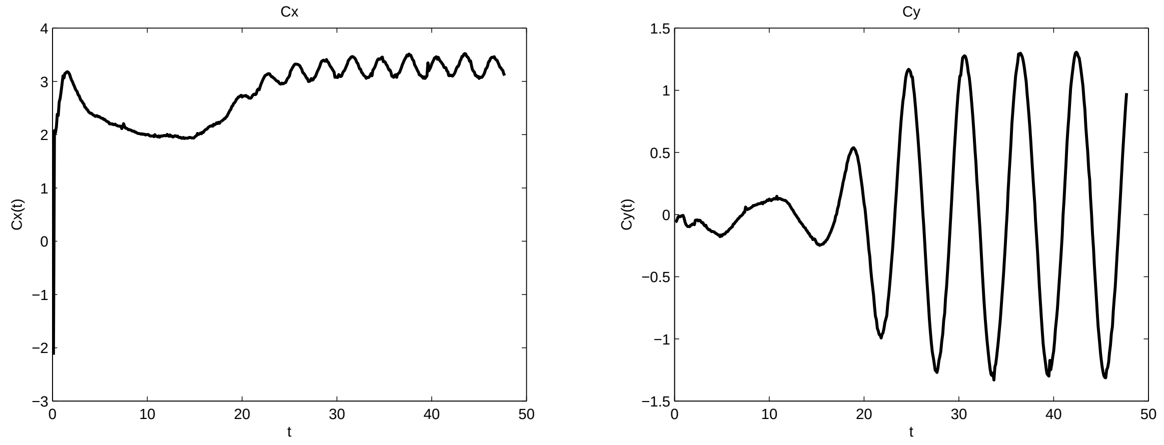


Fig. 2.2. Drag and lift coefficients at $M = 0.05(\Delta x)^2$: average $C_x = 3.2$

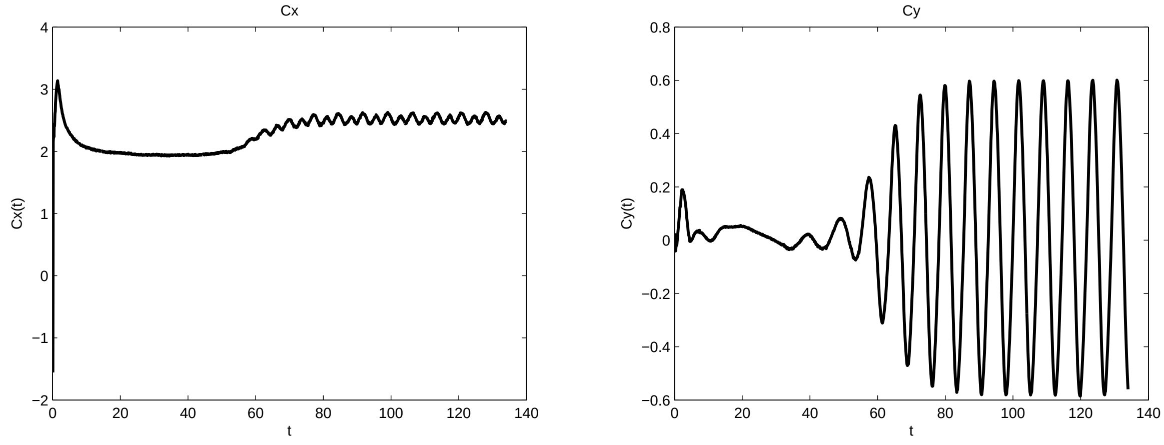


Fig. 2.3. Drag and lift coefficients at $M = 0.01(\Delta x)^2$: average $C_x = 2.5$

As seen from the figures above, the aerodynamic coefficients of the profile strongly depend on the value of the limiting parameter M. Hence, an inaccurate definition of this parameter can lead to sufficient errors in the quantities of the numerical solution. Obviously, it is impossible to predict an optimal estimation of M which would be suitable for the solution of all possible gas-dynamical problems even of similar types. Therefore, a new estimate is needed not only for every new arising problem, but also in the case of a change in the space-time mesh characterictics. The qualitative solution properties may, in general, not be too sensitive to the numerical value of M, as it is in the following testcase of the flow around a semicircle (see Figs. 2.4–2.5).

Consequently, the development of an automatic M estimation algorithm with the possibility of its correction during the computation process has recently become one of the most important and topical questions in the designing of RKDG-based solvers for industrial applications.

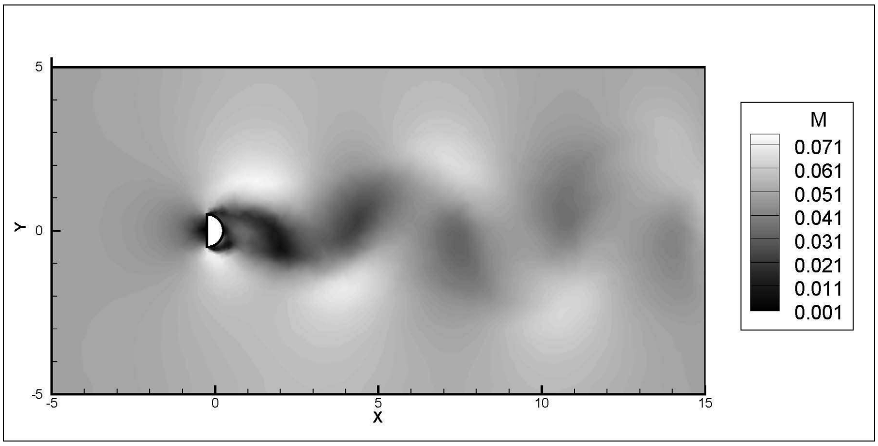


Fig. 2.4. Distribution of Mach numbers

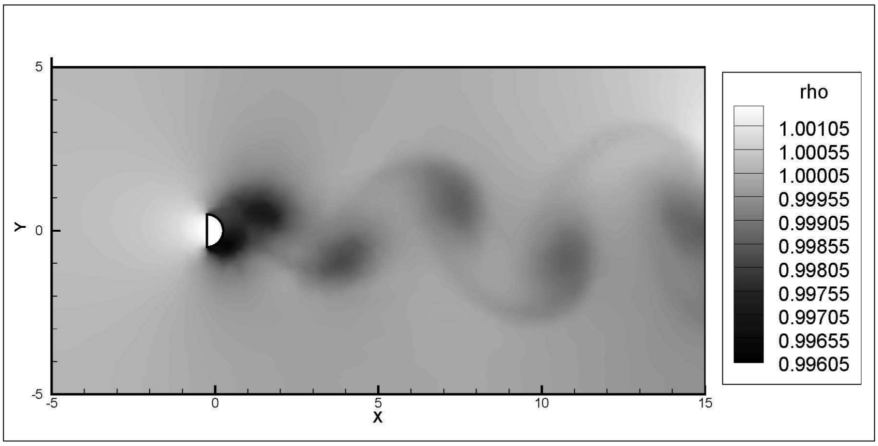


Fig. 2.5. Density distribution

3. Problem-independent limiter

In this paper, we propose a new algorithm for the automatic choice of the M limiting parameter. As in the classical RKDG methods, the limiter for the piecewise-linear functions is based on the computation of the feasible slopes of the numerical solution component u_h at the centers of the edges m_i, $i = 1, 2, 3$ of the triangle K_0 with the center of mass b_0. These slopes are

$$\Delta_i = \bar{m}\big(\tilde{u}_h(m_i, K_0), \nu\Delta\bar{u}(m_i, K_0), M\big), \tag{3.1}$$

where

$$\tilde{u}_h(m_i, K_0) = u_h(m_i) - \bar{u}_{K_0}$$

is the jump of the solution at the center of the i-th edge of the triangle K_0,

$$\bar{u}_{K_0} = \frac{1}{|K_0|}\int\limits_{K_0} u_h\, dV = u_h(b_0)$$

is the average value of the numerical solution over the triangle K_0, $\nu > 1$ (usually $\nu = 1.5$) and $\Delta\bar{u}(m_i, K_0)$ is defined as follows:

$$\Delta\bar{u}(m_i, K_0) = \alpha_1(\bar{u}_{K_i} - \bar{u}_{K_0}) + \alpha_2(\bar{u}_{K_j} - \bar{u}_{K_0}),$$

where K_i and K_j are mesh triangles adjacent to K_0 and $\alpha_{1,2}$ are positive numbers given by the decomposition of the vector $m_i - b_0$ over the two vectors $b_i - b_0$ and $b_j - b_0$, where b_i and b_j are the centers of mass of the triangles K_i and K_j, respectively,

$$m_i - b_0 = \alpha_1(b_i - b_0) + \alpha_2(b_j - b_0).$$

The function $\bar{m}$ in (3.1) stands for the TVB-modified “minmod” function

$$\bar{m} = \begin{cases} \tilde{u}_h(m_i, K_0), & \text{if } |\tilde{u}_h(m_i, K_0)| < M; \\ m\big(\tilde{u}_h(m_i, K_0), \nu\Delta\bar{u}(m_i, K_0)\big), & \text{else,} \end{cases} \tag{3.2}$$

and the function m is the standard “minmod” function

$$m(a_1, a_2) = \begin{cases} s \min\big(|a_1|, |a_2|\big), & \text{if } s = \operatorname{sign}(a_1) = \operatorname{sign}(a_2), \\ 0, & \text{if } \operatorname{sign}(a_1) \neq \operatorname{sign}(a_2). \end{cases}$$

The limited numerical solution component v_h is then expressed as

$$v_h = \bar{u}_{K_0} + \sum_{i=1}^{3} \Delta_i \varphi_i(x_1, x_2).$$

The value of M explicitly defines the maximum possible slope of the numerical solution at the edge center up to which the solution is not monotonized. In the initial RKDG method of [1], this value is taken constant for all mesh cells and, consequently, for all the degrees of freedom of the numerical solution. The automatic M estimation algorithm proposed in the present paper allows to achieve more flexibility in the computations by choosing M for each triangle edge independently and automatically.

The automatic estimation algorithm is based on the decomposition of the numerical solution function component u_h into the Taylor series at the point m_i in the neighbourhood of the center of mass b_0 of the triangle K_0

$$u_h(m_i) = u_h(b_0) + \nabla u_h(b_0) \cdot (m_i - b_0) + (m_i - b_0)^T \mathbf{H}(b_0)(m_i - b_0) + \ldots, \tag{3.3}$$

where $\nabla u_h(b_0)$ is the gradient of u_h at the point b_0, and $\mathbf{H}(b_0)$ is the Hessian matrix at the same point.

Keeping only the first term in the Taylor series (3.3), we obtain

$$u_h(m_i) - \bar{u}_{K_0} \approx \nabla u_h(b_0) \cdot (m_i - b_0),$$

hence, the norm of the u_h jump on the i-th edge of K_0 can be estimated as

$$\|u_h(m_i) - \bar{u}_{K_0}\| \approx \|\nabla u_h(b_0) \cdot (m_i - b_0)\| = M_i, \tag{3.4}$$

where M_i is the sought value for the M parameter on the i-th edge of the triangle K_0.

Therefore, for the computation of M_i it remains to determine the gradient of the numerical solution

$$g_h = \nabla u_h \tag{3.5}$$

at the point b_0.

The solution gradient in the cell K_0 is approximated by a constant function $\bar{g}_h$ as follows:

$$\bar{g}_h = \frac{1}{|K_0|}\int\limits_{K_0} g_h\, dV = \frac{1}{|K_0|}\int\limits_{K_0} \nabla u_h\, dV = \frac{1}{|K_0|}\int\limits_{\partial K_0} \tilde{u}\mathbf{n}\, dS \approx \frac{1}{|K_0|}\sum_{k=1}^{3} \tilde{u}_k \mathbf{n}_k L_k, \qquad (3.6)$$

where $\mathbf{n}_k$ is the outward unit normal to the k-th edge of the triangle K_0 and L_k is the length of this edge. The value of $\tilde{u}_k$ corresponds to the approximation of the numerical solution on the edge, namely the approximation of the Riemann problem solution with initial data $\bar{u}_L^k$, $\bar{u}_R^k$, where

$$\bar{u}_L^k = \bar{u}_{K_0},\ \bar{u}_R^k = \bar{u}_{K_p}$$

are the average values of the solution over K_0 and over the neighbouring cell K_p sharing the same edge k.

We use the HLLC method to approximate the solution of the Riemann problem for Euler equations [6], giving

$$\bar{\mathbf{U}}(0) = \begin{cases} \mathbf{U}_L, & \text{if } 0 \leqslant S_L; \\ \mathbf{U}_{*L}, & \text{if } S_L \leqslant 0 \leqslant S_*; \\ \mathbf{U}_{*R}, & \text{if } S_* \leqslant 0 \leqslant S_R; \\ \mathbf{U}_R, & \text{if } 0 \geqslant S_R, \end{cases}$$

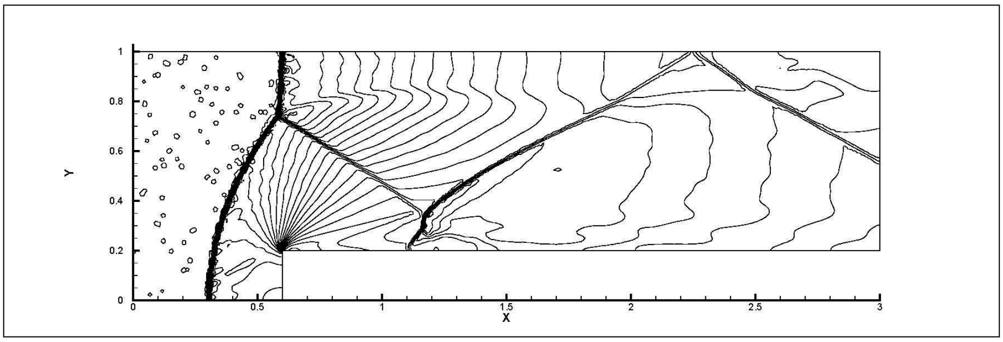
where for $K = L, R$

$$\mathbf{U}_{*K} = \rho_K \left(\frac{S_K - u_K}{S_K - S_*}\right) = \begin{bmatrix} 1 \\ S_* \\ v_K \\ \dfrac{E_K}{\rho_K} + (S_* - u_K)\left(S_* + \dfrac{p_K}{\rho_K(S_K - u_K)}\right) \end{bmatrix},$$

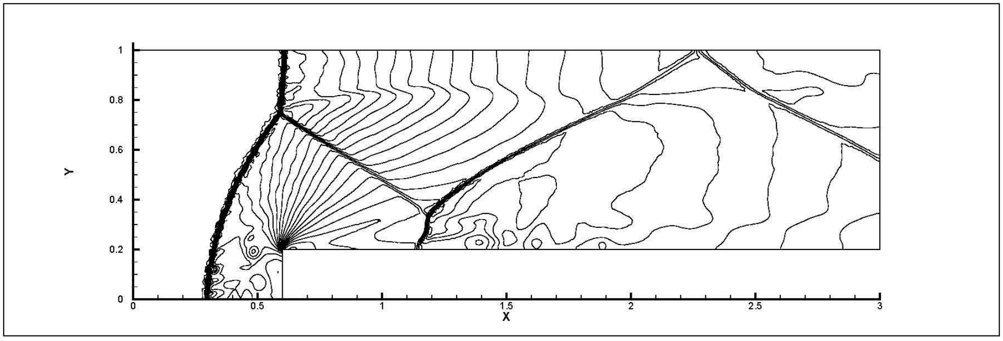
and S_L, S_* and S_R are known estimations of the wave velocities present in the structure of the Riemann problem solution.

3.1. Results: Euler equations

We first study the performance of the proposed automatic limiting parameter M estimation algorithm for the numerical solution of Euler equations. The computational results for the forward-facing step problem [1,4,7] are shown in Figs. 3.1–3.2. Figure 3.1 illustrates 20 density contour lines for two computations: with a manual (upper figure) and an automatic (lower figure) estimation of M. In both computations the piecewise-linear basis functions are used to approximate the solution on the same grids with a typical element size of 1/80. It is evident that the second-order RKDG-method can generate spurious oscillations when setting the M value manually, however, these oscillations are effectively suppressed by the automatic M estimation algorithm.



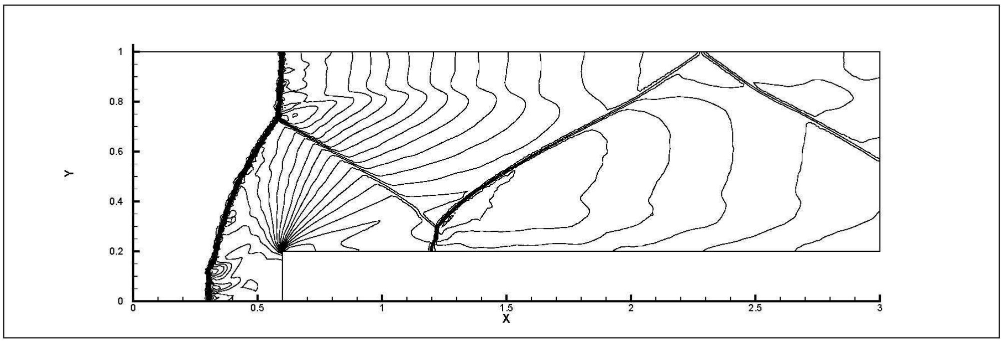
$M = 50(\Delta x)^2$



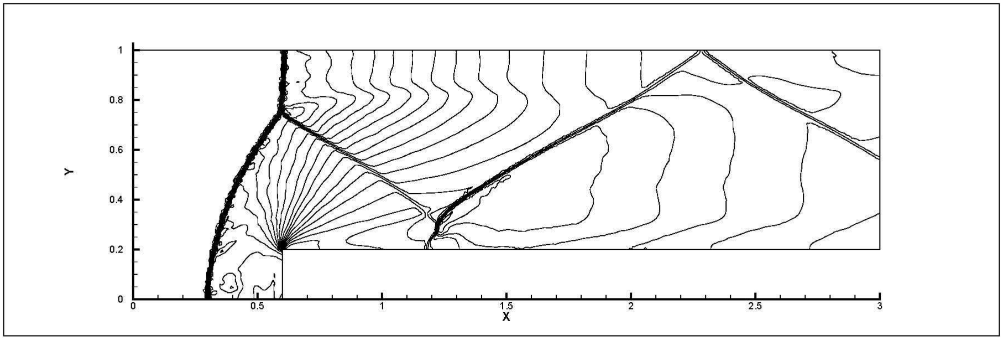
automatic estimation of M

Fig. 3.1. Density contour lines, piecewise-linear solution approximation

The results of analogous computations using piecewise-quadratic basis functions are shown in Fig. 3.2.



$M = 50(\Delta x)^2$

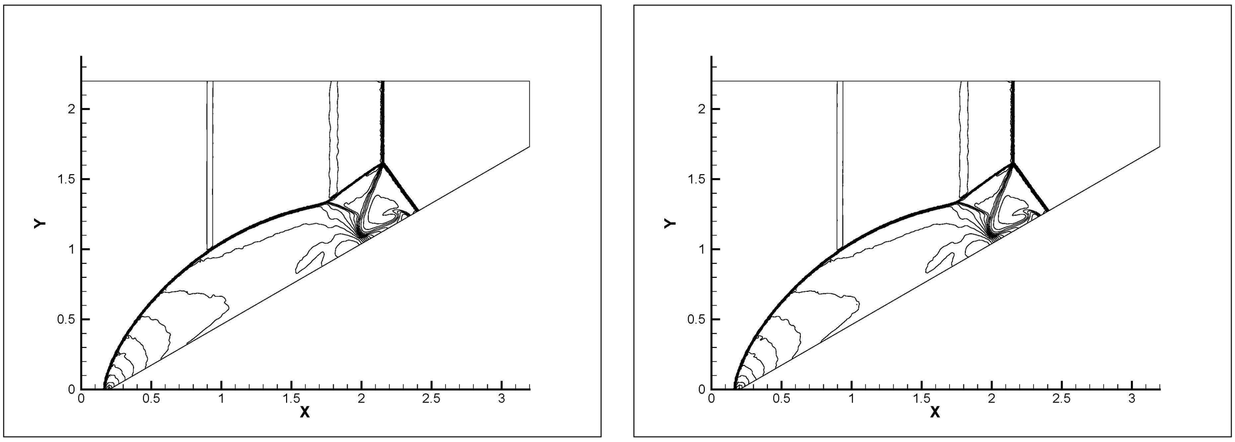


automatic estimation of M

Fig. 3.2. Density contour lines, piecewise-linear solution approximation

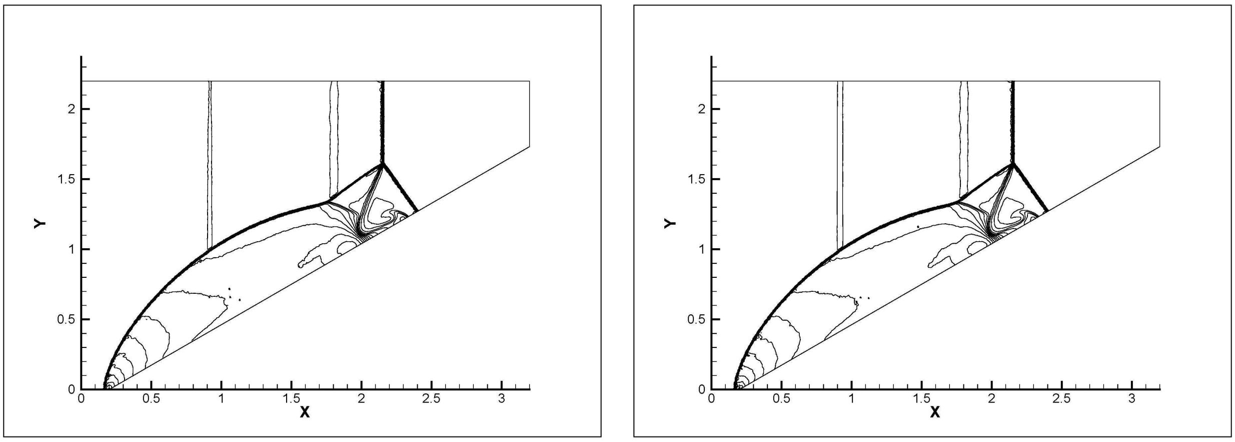
Figures 3.3–3.4 show the computational results for the double Mach reflection problem

[1, 4, 7]. Figure 3.3 corresponds to the piecewise-linear approximation of the solution and Figure 3.4 - to the piecewise-quadratic one.



(a) $M = 50(\Delta x)^2$ (b) automatic estimation of M

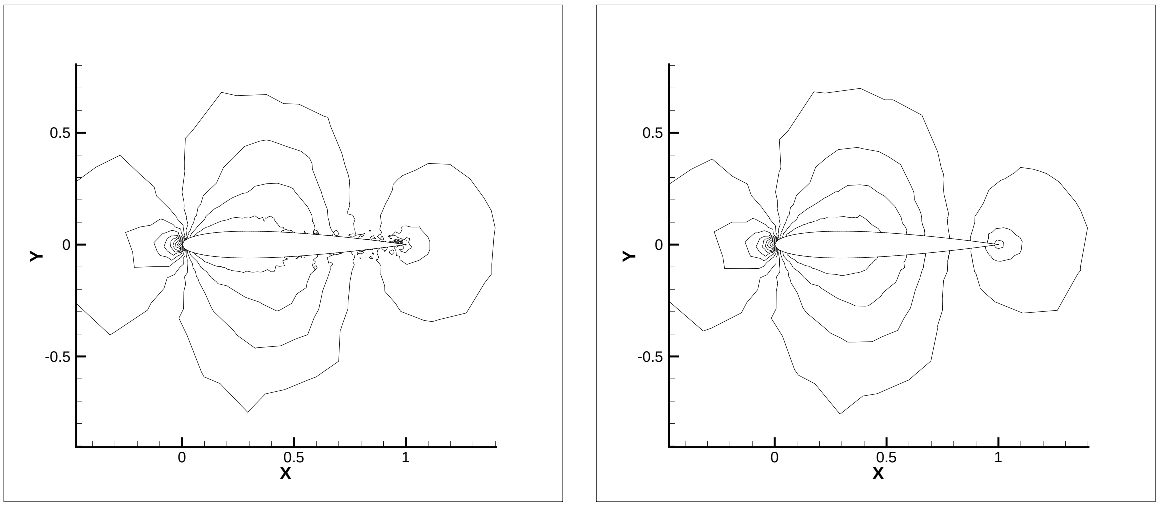
Fig. 3.3. Density contour lines, piecewise-linear solution approximation



(a) $M = 50(\Delta x)^2$ (b) automatic estimation of M

Fig. 3.4. Density contour lines, piecewise-quadratic solution approximation

For the next example of the incorporation of the proposed automatic M estimation algorithm into the RKDG scheme for Euler equations, we consider the computation of the flow around the NACA0012 airfoil at a very low Mach number M = 0.001 (see [3]). For this computation we use the quadratic basis inside each mesh cell. Figure 3.5 illustrates the results obtained using the manually prescribed $M = 50(\Delta x)^2$ (left figure) and the automatically determined M (right figure).



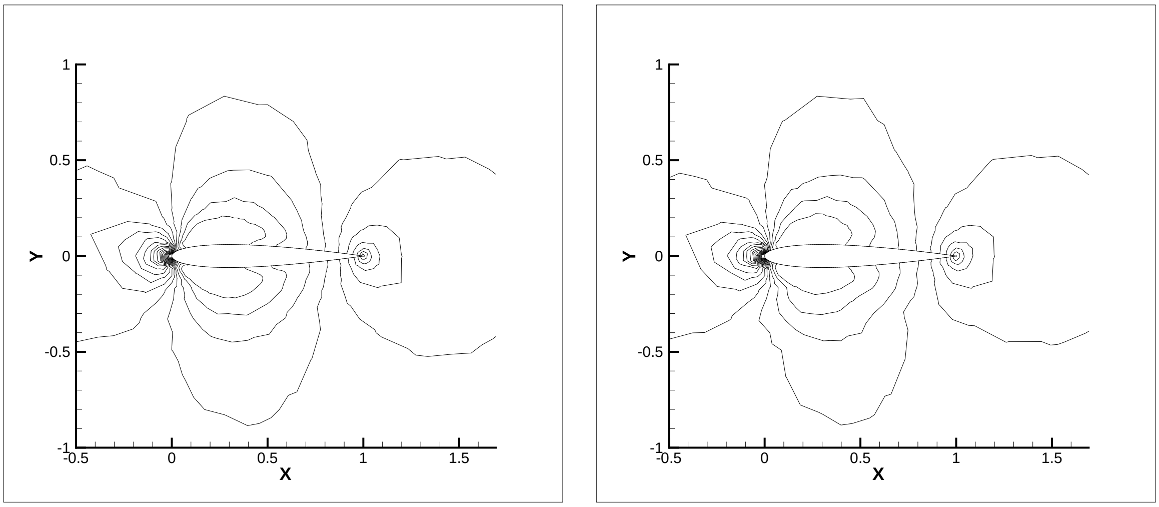
(a) $M = 50(\Delta x)^2$

(b) automatic estimation of M

Fig. 3.5. Mach number contour lines

As seen from the figures above, the automatic choice of M with its correstion after every time step allows to suppress the spurious oscillations inevitably arising in the high-order methods.

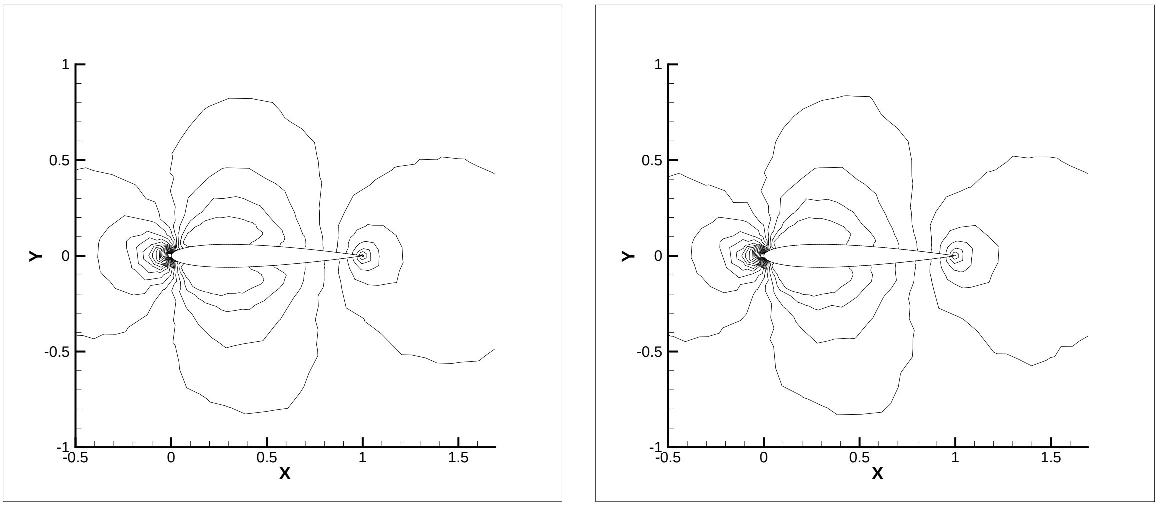
Compare the performance of the second- and third-order RKDG methods with the automatic M estimation algorithm. Figures 3.6–3.8 show 22 Mach number contour lines for the flow around the NACA0012 airfoil at the freestream Mach number M = 0.2. The computational domain is a rectangle with sizes 15Lx10L, where L is the airfoil chord. The mesh size for each computation is indicated in the figure legend. The airfoil is approximated by 200 linear panels.



(a) second-order RKDG

(b) third-order RKDG

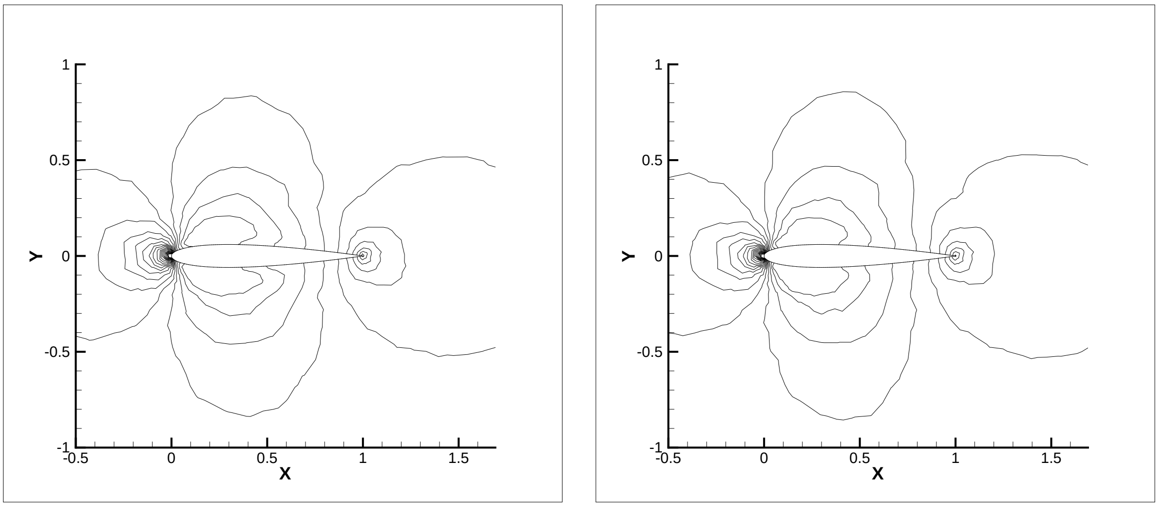
Fig. 3.6. Mach number contour lines (12054 mesh triangles)



(a) second-order RKDG

(b) third-order RKDG

Fig. 3.7. Mach number contour lines (≈ 26000 mesh triangles)



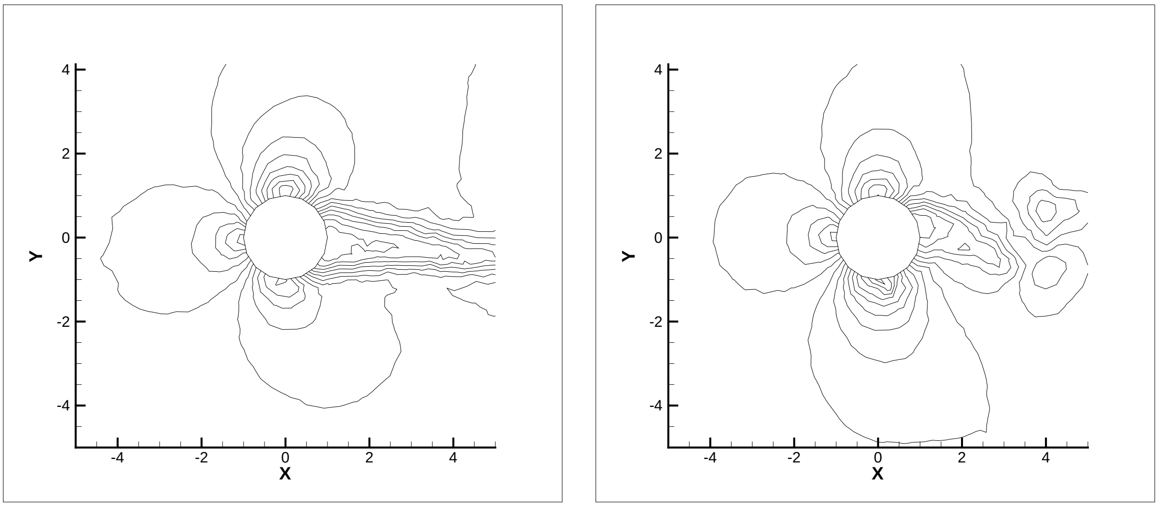
(a) second-order RKDG

(b) third-order RKDG

Fig. 3.8. Mach number contour lines (≈ 50000 mesh triangles)

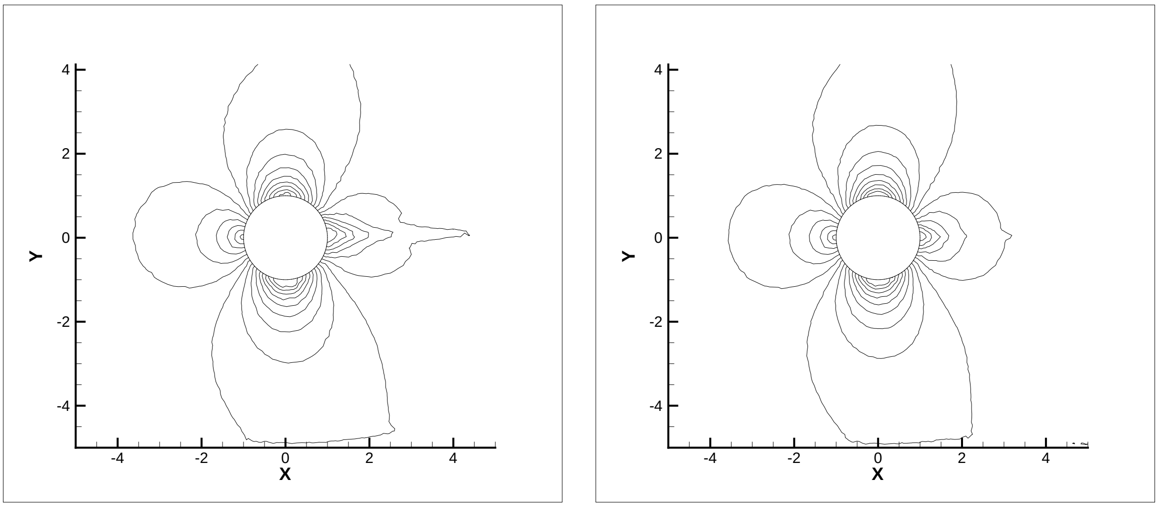
As the flow in this simulation is assumed to be inviscid and hence governed by Euler equations, there should be no boundary layer near the airfoil surface. However, a spurious boundary layer might be generated by a lower-order method unless the airfoil surface is approximated more accurately using more linear edges or turning to curved edges, as observed in the second-order RKDG, regardless of the total number of mesh triangles used. Nevertheless, increasing order of the basis polynomials to second and, consequently, to the third order of approximation helps to minimise the spurious boundary layer while keeping the same linear airfoil surface approximation.

Another useful example demonstrating the difference between the second-order and the third-order RKDG is the typical test case of the 2D subsonic flow around a circle at a Mach number $\mathrm{M} = 0.38$ (see [1, 2]). The exact solution to this problem leads to symmetric contour lines around the cylinder, hence the quality of the numerical solution can easily be assessed by its proximity to symmetry. We have performed computations for the second and third-order RKDG methods on two meshes using the piecewise-linear circle approximation. Figure 3.9 illustrates the results for the second and third-order RKDG methods obtained by the 16 point circle approximation and Figure 3.10 presents the results computed on a finer mesh with 128 points on the circle.



(a) second-order RKDG (b) third-order RKDG

Fig. 3.9. Mach number contour lines (16 points on the circle)



(a) second-order RKDG (b) third-order RKDG

Fig. 3.10. Mach number contour lines (128 points on the circle)

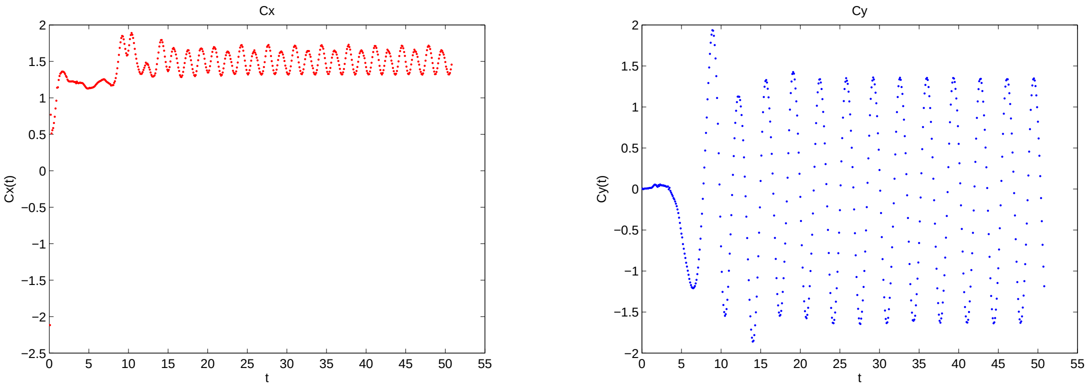
Figures 3.9–3.10 demonstrate the better accuracy of the third-order RKDG method that minimises the non-physical wake in the downstream region behind the cylinder even on a coarse grid.

3.2. Results: Navier-Stokes equations

The algorithm of automatic M estimation makes it possible to obtain a good qualitative and quantitative agreement between computational and experimental data in the numerical simulation of flows governed by the Navier-Stokes equations.

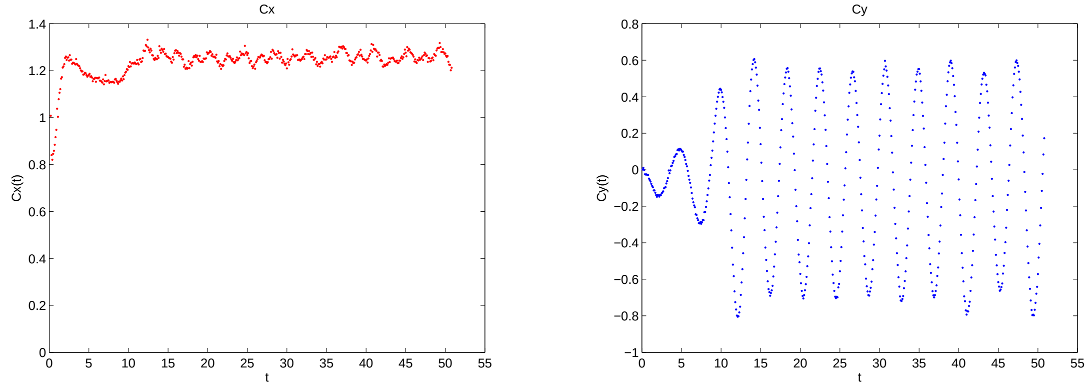
Consider as the first example the flow around a circular cylinder with the following freestream parameters: the Mach number $\mathrm{M} = 0.05$ and the Reynolds number $\mathrm{Re} = 2000$. A flow around a cylinder at such Mach and Reynolds numbers generates the so-called Karman vortex street behind it. It has been established experimentally that, at Reynolds numbers in the range $\mathrm{Re} = 10^3 \ldots 10^5$ the dimensionless vortex wake frequency is virtually constant and is characterized by the Strouhal number $\mathrm{Sh} \approx 0.2$; moreover, the stationary aerodynamic drag and lift coefficients resulting from the time-averaging of the corresponding non-stationary coefficients under steady flow conditions, are also virtually constant: $C_x \approx 0.2$ and $C_y \approx 0$. In all computations the second-order RKDG method was used.

Figures 3.11–3.12 show the time-dependent aerodynamic coefficients, Figure 3.11 corresponds to the algorithm with a manual determination of M, and Figure 3.12 — to the automatic M estimation algorithm.



(a) C_x (b) C_y

Fig. 3.11. Aerodynamic coefficients, $M = 50(\Delta x)^2$



(a) C_x (b) C_y

Fig. 3.12. Aerodynamic coefficients, automatic estimation of M

The version of the RKDG method with manual measurement of M gives inaccurate results for the coefficients: $C_x \approx 1.46$, $C_y \approx -0.12$, Sh ≈ 0.25, while the RKDG method combined with the algorithm of automatic M estimation produces coefficient values that are close to the experimental data: $C_x \approx 1.24$, $C_y \approx -0.07$ and Sh ≈ 0.2.

The typical Karman vortex street is illustrated in Fig. 3.13.

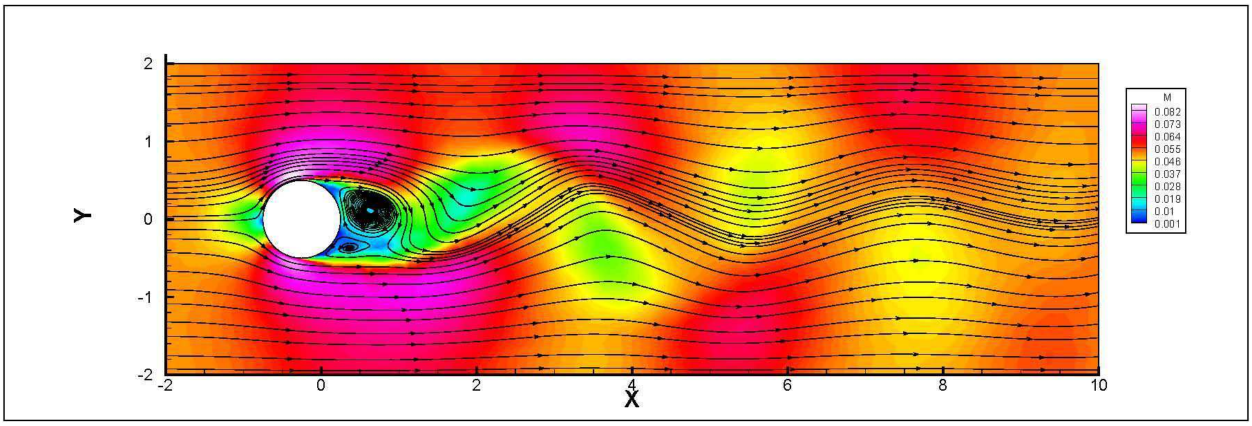


Fig. 3.13. Mach number distribution and streamlines

To estimate the performance of the proposed automatic M limiting parameter choice algorithm we simulated the gas flow around a semicircular cylinder at various angles of attack and compared the computed stationary aerodynamic coefficients with those given by the experiment [8]. Computations were performed for angles of attack from 0° to 180° with a 30° step, the corresponding results are presented in Figs. 3.14–3.15.

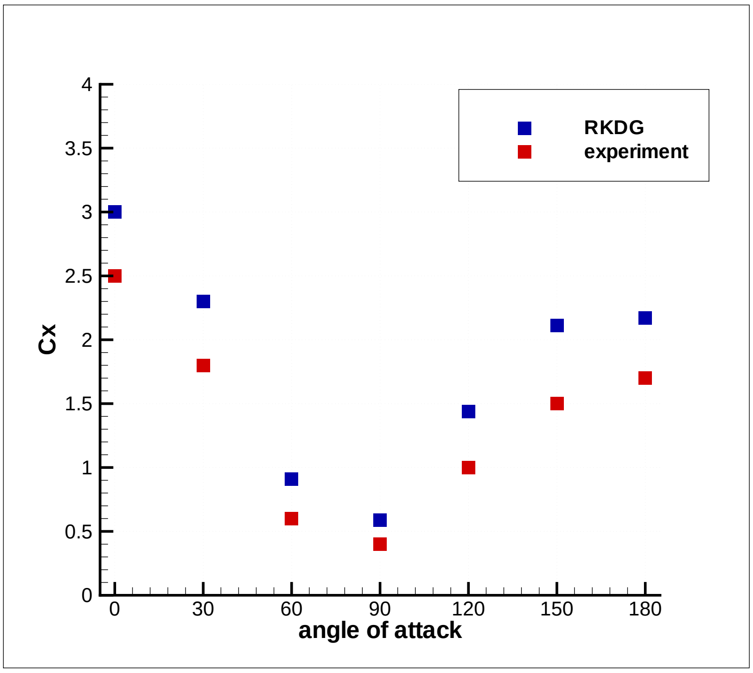


Fig. 3.14. Drag coefficient dependence on the angle of attack

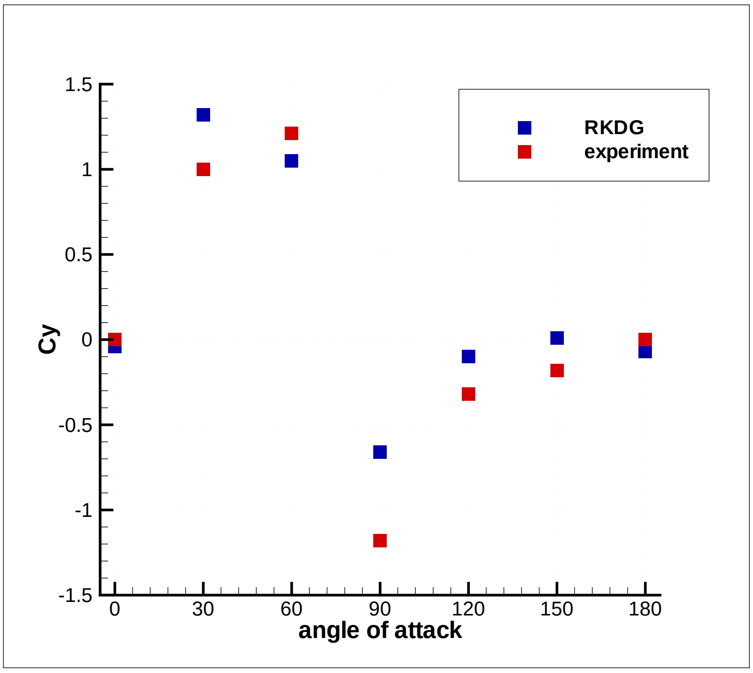


Fig. 3.15. Lift coefficient dependence on the angle of attack

The typical vortex streets generated behind the semicircle at different angles of attack are shown in Figs. 3.16–3.18.

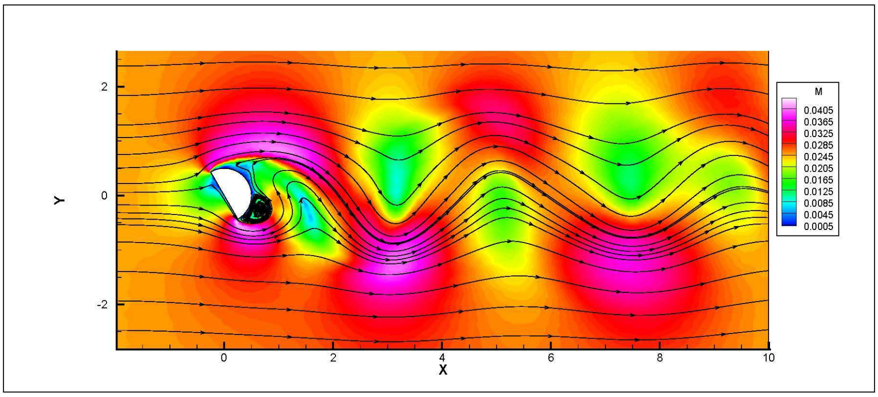


Fig. 3.16. Mach number distribution and streamlines at $\alpha = 30°$

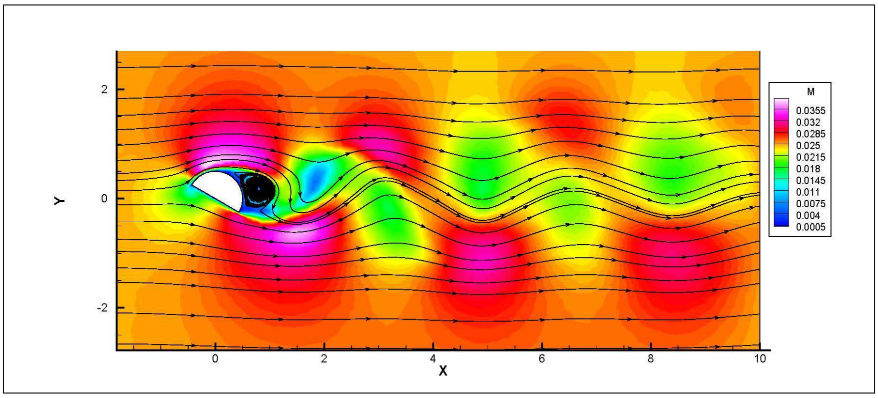


Fig. 3.17. Mach number distribution and streamlines at $\alpha = 60°$

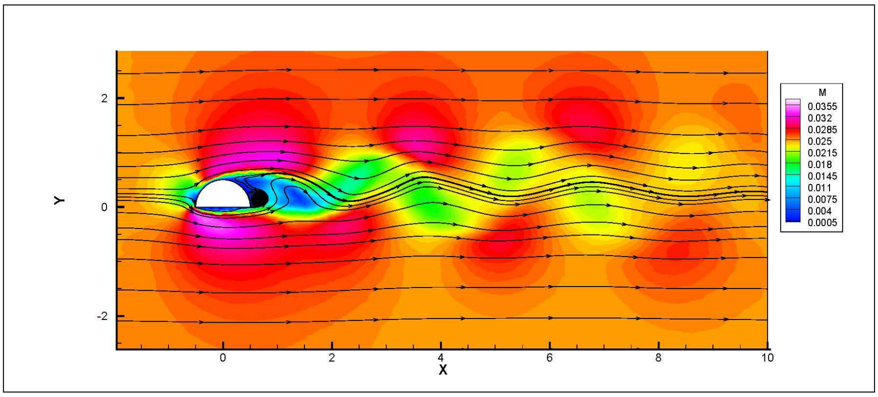


Fig. 3.18. Mach number distribution and streamlines at $\alpha = 90°$

Conclusions

The results of the computations presented in this paper point to a high efficiency and robustness of the proposed algorithm of the automatic limiting parameter estimation when implemented in the RKDG method and justify its utilization for solving a wide class of gas-dynamical problems, including the simulation of ideal and viscid heat conducting gases. This algorithm can be used in both second- and third-order RKDG methods and can easily be incorporated into any higher-order RKDG algorithm. As shown in this paper, the utilization of the higher-order discontinuous Galerkin approximations can be beneficial since it permits attaining the desired accuracy of the solution while holding the geometry approximations relatively crude.

References

1. B. Cockburn and C.-W. Shu, *Runge-Kutta discontinuous Galerkin methods for convection-dominated problems*, J. Sci. Comp., **16** (2001), No. 3, pp. 173–261.

2. F. Bassi and S. Rebay, *High-order accurate discontinuous finite element solution of the 2D Euler equations*, J. Comp. Phys., **138** (1997), pp. 251–285.

3. F. Bassi, C. De Bartolo, R. Hartmann, and A. Nigro, *A discontinuous Galerkin method for inviscid low Mach number flows*, J. Com. Phys., **228** (2009), pp. 3996–4001.

4. M.P. Galanin, E.B. Savenkov, and S.A. Tokareva. *Solving gas dynamics problems with shock waves using the Runge-Kutta discontinuous Galerkin method*, Mathematical Models and Computer Simulations, **1** (2009), No. 5, pp. 635–645.

5. A. Burbeau, P. Sagaut, and Ch.-H. Bruneau, *A problem independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*, **169** (2001), pp. 111-150.

6. E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*, Springer-Verlag, 2009.

7. P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comp. Phys., **54** (1984), pp. 115–173.

8. Z.P. Sluchanovskaya, *Pressure distribution on the surface of the rectangular, triangular and semi-circular cylinders and their aerodynamical coefficients [in Russian]*, Instinute of Mechanics MSU, **24** (1973), pp. 52–60.