SIAM J. Sci. Comput.
Vol. 27, No. 4, pp. 1329–1346

ON OSCILLATIONS IN DISCONTINUOUS GALERKIN DISCRETIZATION SCHEMES FOR STEADY STATE PROBLEMS[*]

N. B. PETROVSKAYA[†]

Abstract. High order discontinuous Galerkin discretization schemes are considered for steady state problems. We discuss the issue of oscillations arising when Newton's method is employed to obtain a steady state solution. It will be demonstrated that flux approximation near flux extrema may produce spurious oscillations propagating over the domain of computation. The control over the numerical flux in the problem allows one to obtain nonoscillating convergent solutions.

Key words. discontinuous Galerkin, spurious oscillations, Newton's method

AMS subject classifications. 49M15, 65L20, 65L60

DOI. 10.1137/040603085

1. Introduction.

In recent years a variety of discretization methods have been developed to solve complex problems of physics and engineering. One of them is a discontinuous Galerkin (DG) discretization scheme. Introduced in [11] and further developed by many authors (see [4] for the review of DG schemes), the DG method is a finite element scheme which uses piecewise polynomial approximation in space. The method also involves an approximate Riemann solver, since the approximate solution is discontinuous at grid interfaces.

The hyperbolic systems of conservation laws present a wide class of problems where the DG method can be successfully applied. The DG discretization scheme affords optimal orders of convergence for smooth problems by using high order approximating spaces. For the problems whose solutions have strong gradients and/or discontinuities, solution oscillations may occur when a high order DG scheme is used to discretize a conservation law. Since the nonphysical oscillations have a disastrous impact on the convergence of the approximate solution, a limiting procedure that eliminates the oscillations near discontinuities should be addressed. A number of authors have contributed to the issue of limiters for DG schemes in recent years (e.g., see [5], [6], [9]). It has been demonstrated many times that stabilization of the scheme by means of local limiters allows one to obtain accurate nonoscillating solutions to nonlinear hyperbolic problems.

The local limiters are not always helpful, however, when steady state solutions to conservation laws are considered. In practice, a time-dependent algorithm (e.g., a backward Euler integration) is used to approach a steady state solution. The time step is usually scaled as a function of the norm of the residual, so that the scheme with an infinitely large time step is equivalent to Newton's method. Thus it seems to be a reasonable strategy to solve time-dependent equations only at the early stages of computations. Once the basin of attraction has been approached, the Newton method may be exploited in order to provide a faster convergence rate. Meanwhile, our numerical experience shows that a transient solution may exhibit strong oscillations over

[*] Received by the editors January 12, 2004; accepted for publication (in revised form) June 14, 2005; published electronically January 6, 2006. This work was supported by The Boeing Company under contract 104AE.

http://www.siam.org/journals/sisc/27-4/60308.html

[†] School of Mathematics, The University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom (N.B.Petrovskaya@bham.ac.uk).

the entire domain of computation, if the Newton iteration method is used to solve a system of nonlinear equations obtained as a result of a high order DG discretization in space. Those oscillations may appear for a smooth solution as well as a discontinuous one, and their excitation does not depend on how close the initial guess for the Newton method is to the fixed point considered as a steady state solution for the problem. The spurious oscillations propagating over the domain cannot be eliminated by means of a standard limiting procedure [5], and their nature requires careful study.

In our work, we consider two nonlinear scalar equations in order to examine a high order DG discretization for steady state solutions. Simple enough, they nevertheless demonstrate the difficulties arising in the solution of steady state problems. In our first example the exact solution is smooth, while the solution to the second problem has a discontinuity. It will be shown that in both cases a standard high order DG discretization yields a divergent solution.

Based on our consideration, we conclude that a high order DG scheme is not able to recognize flux extrema that may result in a singular Jacobian when Newton's method is used to solve the problem. Moreover, a transient solution may generate nonphysical flux extrema which lead to a singular matrix as well. Thus, spurious solution oscillations occur in the problem due to incorrect flux approximation, so that a high order DG discretization requires flux control over each grid cell. We present a flux control procedure that allows one to obtain convergent solutions.

2. The problem statement. We consider an ordinary differential equation written for a function $u(x)$ in the conservative form

$$F_x(x,u)=0, \quad x\in\Omega=[0,1], \tag{1}$$

where $F(x,u(x))$ is a flux function. An appropriate boundary condition

$$\mathbf{B}u=0 \tag{2}$$

is provided for (1), where $\mathbf{B}$ denotes a boundary condition operator.

For numerical solution of the boundary-value problem (1), (2) we introduce the element partition G of the region, $G=\bigcup_{i=1}^{N} e_i$, $e_i=[x_i,x_{i+1}]$, $1\le i\le N$, where x_i is a nodal coordinate, and $h_i=x_{i+1}-x_i$ is a grid step size. We also use the notation x_i-0 and x_i+0 for the left and right limits at the point x_i.

Let $u(x)$ be the solution to the problem (1), (2). In order to find the approximate solution $u_h(x)$, a weak formulation of the problem is used. Multiplying (1) by test function $\phi_k(x)$, defined on the cell e_i for $k=0,1,\ldots,K$ as

$$\phi_k(x)=\left(\frac{x-x_i}{h_i}\right)^k, \quad x\in e_i,$$

and integrating by parts over the cell e_i, we obtain

$$F(x_{i+1},u(x_{i+1}))\phi_k(x_{i+1})-F(x_i,u(x_i))\phi_k(x_i)-\int_{x_i}^{x_{i+1}}F(x,u)\frac{d\phi_k(x)}{dx}dx=0, \qquad k=0,1,\ldots,K. \tag{3}$$

We now replace the function $u(x)$ in (3) by the approximate solution $u_h(x)$. The DG discretization seeks for the approximation $u_h(x)$ to the solution $u(x)$ such that

$u_h(x)$ is a piecewise polynomial function over Ω. The approximate solution $u_h(x)$ is expanded on the cell e_i as

$$u_h(x) = \sum_{k=0}^{K} u_k \phi_k(x), \quad k = 0, 1, \ldots, K, \quad x \in e_i. \tag{4}$$

Since $u_h(x)$ is discontinuous at cell interfaces, (3) considered for the solution $u_h(x)$ requires us to define numerical flux $\tilde{F}(u_h)$. Suppose that the flux $\tilde{F}(u_h)$, which generally depends on the two values of the approximate solution at any grid point, is chosen for a given problem. Then the DG discretization scheme reads

$$\tilde{F}(u_h(x_{i+1}))\phi_k(x_{i+1}) - \tilde{F}(u_h(x_i))\phi_k(x_i) - \int_{x_i}^{x_{i+1}} F(x, u_h(x)) \frac{d\phi_k(x)}{dx} dx = 0, \\ k = 0, 1, \ldots, K. \tag{5}$$

For steady state problem (1), (2), the DG space discretization over the grid results in the following system of nonlinear equations:

$$\mathbf{R}(\mathbf{u}) = 0, \tag{6}$$

where the vector $\mathbf{R}(\mathbf{u})$ is the residual of the DG method given by (5) on each grid cell and $\mathbf{u}$ is the solution vector. We use Newton's iteration method to solve the nonlinear equations (6). Let $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ be the solution vector at the nth and $(n+1)$th solution iteration, respectively. Then the linearized system is

$$\mathbf{J}(\mathbf{u}^n)(\mathbf{u}^{n+1} - \mathbf{u}^n) = -\mathbf{R}(\mathbf{u}^n), \tag{7}$$

where the Jacobian matrix $\mathbf{J}(\mathbf{u}) = [\partial \mathbf{R}/\partial \mathbf{u}]$ and residual $\mathbf{R}(\mathbf{u})$ are taken from the nth iteration. The GMRES algorithm [12], [2] is used to solve numerically the algebraic system of linear equations obtained at each Newton iteration.

In the remainder of our paper we discuss oscillations appearing in solution (5), (7). One important observation about the DG scheme is that for the steady state problem (1), (2) the solution on the ith grid cell impacts on the solution on neighboring cells only by means of the numerical flux $\tilde{F}(u_h)$. Hence, the two possible ways of the excitation of oscillations at the nth Newton iteration are as follows.

1. The numerical flux required to define the DG discretization on the cell e_i is correct, but the approximate solution on the cell does not converge in a particular norm. That may happen, for instance, when a discontinuity presented in the cell is approximated by smooth function (4). The solution overshoots arising as a result of such approximation are local and do not affect the solution on other cells.
2. The numerical flux is not correct on the ith cell. The incorrect flux approximation produces solution oscillations that will propagate over the domain at the next Newton iterations and result in a divergent solution.

While local limiters can be successfully used to smooth the local solution overshoots, another approach is required to recognize and eliminate the spurious oscillations propagating over the grid. That approach will be discussed below.

3. The numerical flux in steady state problems.

In this section, we address a numerical flux used in the formulation of the DG discretization. Usually, oscillations arising in the approximate solution are associated with solution discontinuities. Thus, our aim is to verify the definition of the numerical flux and demonstrate that the oscillations may appear for a smooth solution as well as a discontinuous function.

We begin our consideration with a simple example of (1) that illustrates the problem. Let the flux $F(x,u)$ be

$$F(x,u) = p(x)f(u), \quad p(x) = \frac{1}{((x-x_0)(x-x_1))^2}, \quad f(u) = (u-A)^2. \tag{8}$$

The problem parameters x_0, x_1, and A and the boundary condition are chosen to provide a smooth solution to the problem

$$U(x) = A + C(x-x_0)(x-x_1), \quad x \in [0,1],$$

where C is a constant. Namely, we take $A = 1.0$, $C = -1.0$, $x_0 = -0.5$, and $x_1 = 1.5$, so that bifurcation points $x = x_0$ and $x = x_1$ lie outside the domain of computation. The boundary condition is

$$\int_0^1 u(x)dx = B, \tag{9}$$

where the value B is defined by integrating the exact solution with the parameters above.

The model problem (1), (8) is numerically solved by using the DG discretization approach. We choose the Engquist–Osher definition [10] to approximate the flux at grid interfaces. Let u_l and u_r be the left and right states at the interface x_i, respectively. The numerical flux reads

$$\tilde{F}^{EO}(u_l,u_r) = \int_0^{u_r} \min(F'(s),0)ds + \int_0^{u_l} \max(F'(s),0)ds + F(0). \tag{10}$$

For the problem (8), the flux has a single extremum point, $u = A$. Hence, the numerical flux (10) is as follows:

$$\tilde{f}(u_l,u_r) = \begin{cases} f(u_l), & u_l > A,\, u_r > A, \\ f(u_r), & u_l < A,\, u_r < A, \\ f(A), & u_l < A,\, u_r > A, \\ f(u_l) + f(u_r) - f(A), & u_l > A,\, u_r < A. \end{cases} \tag{11}$$

The numerical experience with the problem shows that the convergence of the Newton method depends strongly on the choice of initial guess. Consider a sine wave function

$$u_0(x,s_0) = \sin(2\pi x) + s_0,$$

where s_0 is a parameter. Let us consider $s_0^{\mathrm{I}} = 2.3$ and $s_0^{\mathrm{II}} = 1.8$. For the initial guess $u_0^{\mathrm{I}} = u_0(x,s_0^{\mathrm{I}})$, the flux $f(u)$ is a monotone function over the domain of definition $u_0 \in [u_{min}^{\mathrm{I}}, u_{max}^{\mathrm{I}}]$. For the function $u_0^{\mathrm{II}} = u_0(x,s_0^{\mathrm{II}})$, we have $u_{min}^{\mathrm{II}} < A$, $u_{max}^{\mathrm{II}} > A$, so that the flux approximation is required at the extremum point $u = A$ at the first Newton step.

Although the curves u_0^{I} and u_0^{II} are close to each other, i.e., $||u_0^{\mathrm{II}} - u_0^{\mathrm{I}}||_{L^\infty} = |s_0^{\mathrm{II}} - s_0^{\mathrm{I}}|$, $x \in [0,1]$, the convergence results are quite different for the two functions. Starting with the initial guess u_0^{I}, Newton's method rapidly converges to the approximate solution $u_h(x)$. The convergence results obtained on a sequence of uniform grids confirm the consistency of the approximation (5), (11). In particular, the DG scheme with polynomial degree $K = 2$ provides a precise reconstruction of the quadratic function $U(x)$. Meanwhile, the choice of u_0^{II} as initial guess for the problem results in a divergent solution for any polynomial degree $K > 0$.

Let us look more closely at the numerical flux used in the problem. For a scalar flux function, the definition of the numerical flux is essentially based on the analysis of the flux derivative, the results of which depend on how the solution variation $u_r - u_l$ is determined. In the definition (10), the solution variation is assigned to the grid interface x_i, $u_l = u_h(x_i - 0)$, $u_r = u_h(x_i + 0)$. In other words, the definition (10) implies that the flux variation is due only to the solution variation at grid interfaces, i.e., $F(u) = \text{const}$ within a grid cell. Evidently, this assumption will be correct if the approximate solution is constant over each cell. Meanwhile, for a high order DG discretization scheme the approximate solution varies in the domain $[x_i, x_{i+1}]$. The solution variation $\delta u_h = u_h(x_{i+1} - 0) - u_h(x_i + 0)$ may generate a flux extremum at the interior point of the cell e_i, while the flux remains a monotone function at both interfaces x_i and x_{i+1}. Below we compare flux approximation in a piecewise constant and a higher order DG discretization and demonstrate that for a high order scheme, considering a flux as a monotone function at the interfaces of the cell where the flux extremum is located may lead to incorrect DG approximation.

As an example of a high order DG scheme, let us consider a piecewise linear approximate solution

$$u_h(x) = u_0 + u_1\phi_1(x), \quad x \in e_i.$$

Since a DG discretization with polynomial degree K results in $K+1$ unknown solution values (degrees of freedom) per cell, two DG equations (5) per cell must be employed for a piecewise linear solution. Those equations are a discrete conservation law,

$$p(x_{i+1})\tilde{f}(u_l, u_r) - p(x_i)\tilde{f}(u_l, u_r) = 0, \tag{12}$$

and the momentum equation,

$$p(x_{i+1})\tilde{f}(u_l, u_r) - \frac{1}{h_i}\int_{x_i}^{x_{i+1}} p(x)f(u_h(x))dx = 0. \tag{13}$$

For the purpose of further discussion it is more convenient to define the DG discretization on a given cell in terms of the solution values rather than the solution derivatives in the expansion (4) (see [13] for more details). We now introduce the notation u_i^l and u_i^r for the left and right solution states at the interface x_i, respectively. Then the values u_i^r and u_{i+1}^l are defined by the piecewise linear function $u_h(x)$ on a given cell e_i. The evident transformation

$$u_i^r \equiv u_h(x_i + 0) = u_0, \quad u_{i+1}^l \equiv u_h(x_{i+1} - 0) = u_0 + u_1,$$

allows one to consider left and right solution states as unknown degrees of freedom for a piecewise linear solution.

The flux approximation in the DG equations at the first Newton step is illustrated in Figure 1 for a piecewise constant and a linear approximate solution. The flux function $f(u) = (u - A)^2$ is shown in the $(u, f(u))$-plane in Figure 1(a)–(b). The solution degrees of freedom are stationed at the u-axis, and the solution values required to approximate the flux at cell interfaces are displayed as black dots. The

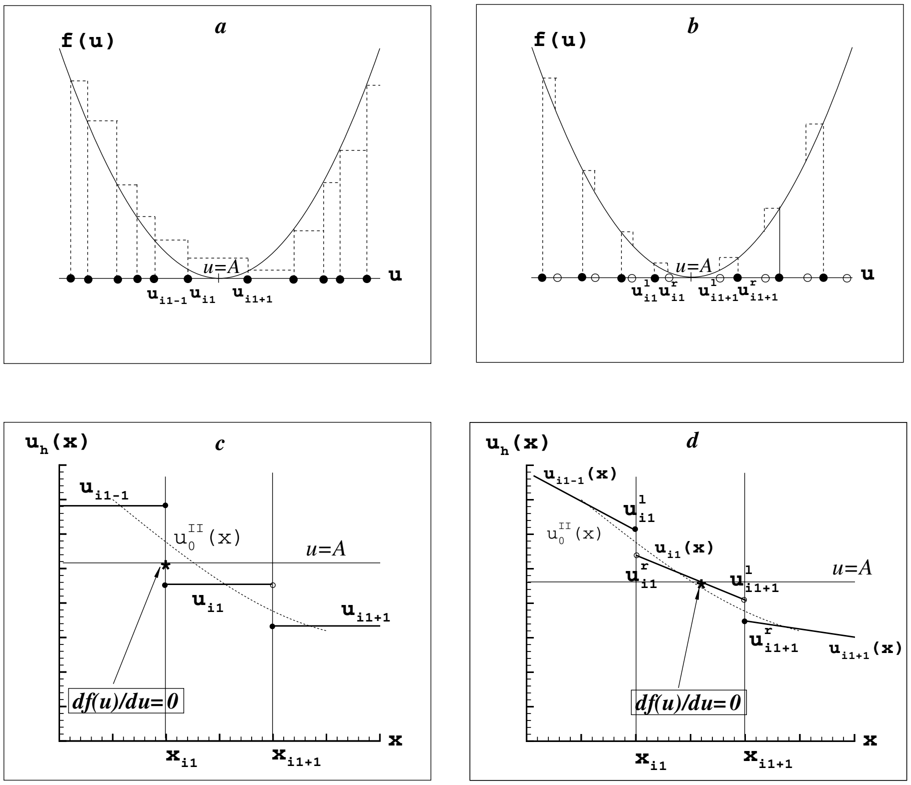

FIG. 1. (a)–(b) *The numerical flux near the extremum point for the problem* (1), (8). (a) *The numerical flux for the piecewise constant approximate solution. All degrees of freedom are involved in the flux approximation.* (b) *The flux approximation for a piecewise linear DG solution. Both degrees of freedom are not involved in the flux approximation on the cell where the flux extremum is located.* (c)–(d) *The DG discretization in the* $(x, u_h(x))$*-plane.* (c) *Piecewise constant solution approximation captures the flux extremum.* (d) *Higher order solution approximation misses the extremum point inside the grid cell.*

corresponding values of the numerical flux (10) at grid interfaces are shown at the curve $f(u)$. In the figure, the flux approximation is presented in the vicinity of the cell $e_{i1} : u^r_{i1} > A,\ u^l_{i1+1} < A$. In other words, the initial guess is a decreasing function near the extremum point, and the solution states at the grid interfaces corresponding to greater values of u are shown at the left from the extremum point $u = A$.

The flux approximation for the piecewise constant approximate solution (one degree of freedom per cell) is presented in Figure 1(a). It can be seen from the figure that, according to the definition (10), a discrete conservation law (12) involves one unknown solution value per cell. In other words, the number of DG equations to be solved over the grid is equal to the number of unknown solution values, as all degrees of freedom are involved in the flux approximation.

Now we look at the flux approximation in a piecewise linear DG scheme shown in Figure 1(b). Let flux be a monotone function over a given cell e_i. Again the DG equation (12) requires one degree of freedom per cell to balance the monotone fluxes at the left and right cell interfaces. That degree of freedom is chosen to provide the upwind discretization; i.e., we take u^r_i for $df(u)/du < 0$ or u^l_{i+1} for $df(u)/du > 0$ to approximate the flux. A second degree of freedom on the cell is not involved in the discrete conservation equation (12), and thus another equation is required to define the second unknown solution value on the cell. This equation appears as the momentum equation (13) in the DG scheme. Hence, there are two DG equations for two unknown solution values on any cell where the flux is a monotone function through the cell.

The situation is different, however, on the cell where the flux extremum is located at the interior point. Now both degrees of freedom on the cell are not involved in the flux approximation, as the definition of the numerical flux concerns only the solution on the adjacent cells. Namely, we have $u^l_{i1} > A$, $u^r_{i1} > A$ at the left cell interface, so that the upwind flux is $f(u^l_{i1})$. Meanwhile, $u^l_{i1+1} < A$, $u^r_{i1+1} < A$, and the numerical flux is $f(u^r_{i1+1})$ at the right cell interface.

The DG equations on the cell e_{i1} are illustrated in the $(x, u_h(x))$-plane in Figure 1(c)–(d) for a piecewise constant and a linear discretization, respectively. Again, the solution values required to define the flux at the interfaces of the cell e_{i1} are shown as black dots. It can be seen from the figure that for the piecewise constant DG discretization, displayed in Figure 1(c), the extremum point at the interface is taken into account in the definition of the numerical flux. The discrete conservation law reads

$$p(x_{i1+1})f(u^r_{i1+1}) - p(x_{i1})(f(u^l_{i1}) + f(u^r_{i1}) - f(A)) = 0.$$

For the linear approximate solution, shown in Figure 1(d), the "phantom" solution on the cell e_{i1} is not involved in the flux definition. The discrete conservation law is

$$p(x_{i1+1})f(u^r_{i1+1}) - p(x_{i1})f(u^l_{i1}) = 0, \tag{14}$$

where $u^l_{i1} = u_h(x_{i1} - 0)$ and $u^r_{i1+1} = u_h(x_{i1+1} + 0)$ are the solution unknowns on the adjacent cells.

Thus, we still need two equations to define the unknowns (u^r_{i1}, u^l_{i1+1}), while the DG discretization provides us with only one more equation that involves the unknowns on the cell e_{i1}. Let us also note that, by the definition of the numerical flux (10), the unknowns (u^r_{i1}, u^l_{i1+1}) on the cell e_{i1} are not required for the discretization on neighboring cells (see Figure 1(b)), and thus the number of DG equations to be solved over the grid is not equal to the number of unknown solution values for a piecewise linear solution. This indicates that the solution should be reduced to piecewise constant approximation on the cell e_{i1} in order to avoid the underdetermined system of equations over the grid. In other words, the number of degrees of freedom is excessive on the cell e_{i1}, and only one of them should be used for the discretization.

The same conclusion can be made in the case of a DG discretization with any polynomial degree $K > 0$. For instance, a piecewise quadratic approximation $K = 2$ results in three unknown values per cell. Instead of coefficients (u_0, u_1, u_2) in the expansion (4), these unknowns can be defined as u^r_i, u^l_{i+1}, and u^m_i, where $u^m_i = u_h(x_m)$, $x_m = 0.5(x_i + x_{i+1})$. Again, the discrete conservation law (12) on the cell e_{i1} does not require any of these degrees of freedom, as only the solution on adjacent cells is concerned, and thus we have two equations per cell to find three unknown values.

Consequently, solving the underdetermined system of equations arising as a result of a high order DG discretization leads to a singular Jacobian in the Newton method. For a piecewise linear solution, let us define a local Jacobian $\mathbf{j}_{k_1k_2} = [\partial R_{k_1}/\partial u_{k_2}]$ by linearizing the two DG equations on the cell e_{i1}. Here the local index $k_1 = 0, 1$ is related to the DG residual on the cell, while the local index $k_2 = 0, 1$ refers to the corresponding degree of freedom from the set (u^r_{i1}, u^l_{i1+1}). Then the first row in the 2×2-matrix $\mathbf{j}_{k_1k_2}$ has only zero entries, as we have $j_{00} \equiv \partial R_0/\partial u^r_{i1} = 0$ and

$j_{01} \equiv \partial R_0/\partial u^l_{i1+1} = 0$ for the residual R_0 defined by (14). Hence, the rank of the local Jacobian is $\mathrm{rank}(\mathbf{j}_{k_1k_2}) = 1 < \dim(\mathbf{j}_{k_1k_2}) = 2$, and by reordering the rows and columns we can obtain a zero column in the matrix $\mathbf{j}_{k_1k_2}$.

We now consider the matrix $\mathbf{j}_{k_1k_2}$ as the block J_{lm} of the Jacobian $\mathbf{J}$ of the linearized system (7). The global indices for the block entries are $l = k_1 + M_1$ and $m = k_2 + M_1$, where $M_1 = 2\,i1 - 1$. Since the definition (10) provides the exact flux splitting for the problem, we have the Jacobian entries $J_{lM1} = 0$, $J_{lM1+1} = 0$, $\forall l$: $l < M_1$ or $l > M_1 + 1$. On the other hand, it follows from the above consideration that the rows and columns of the Jacobian matrix can be reordered to provide

$$J_{lM_1} = 0, \quad l = M_1, M_1 + 1,$$

while we still have $J_{lM1} = 0$, $\forall l$: $l < M_1$ or $l > M_1 + 1$ in the reordered matrix. Hence, a zero column appears in the Jacobian of the system (7). The singular Jacobian leads to an incorrect transient solution (whose appearance depends strongly on the robustness of the GMRES solver used in the problem). That solution, in turn, will impact on the flux at the next Newton iterations, so that the oscillations will rapidly propagate over the domain resulting in the divergence of the method. Let us also notice that a similar analysis can be carried out for a DG discretization with any polynomial degree $K > 0$ to demonstrate that a singular Jacobian will appear in the problem, if the number of unknowns on the cell exceeds the number of equations.

The above results bring us to the conclusion that the nature of oscillations arising in the steady state problem (8), (9) is different from that appearing in approximate solution to hyperbolic conservation laws. In the latter case the oscillations arise near a solution discontinuity, and the approximation implies a well-defined numerical flux over the computational domain. Now the numerical flux $\tilde{f}(u)$ is not a correct approximation to the flux function $f(u)$ at the extremum point, while the solution remains a smooth monotone function near the flux extremum.

We are now going back to the problem (8). The above analysis reveals that a high order approximate solution should be reduced to a piecewise constant approximation near a flux extremum in order to avoid nonphysical oscillations in the problem. If the flux extremum generates a "phantom" solution on a given grid cell e_i (i.e., the number of unknowns exceeds the number of equations on the cell) at the nth Newton iteration, then we compute $\bar{u} = \frac{1}{h_i}\int_{x_i}^{x_{i+1}} u_h(x)dx$ and define the approximate solution on the cell e_i as $u_h(x) = \bar{u}$. The equations

$$\begin{gathered} p(x_{i+1})\tilde{f}(u_h) - p(x_i)\tilde{f}(u_h) = 0, \quad k = 0, \\ u_k = 0, \quad k = 1, \ldots, K, \end{gathered}$$

are then considered on the cell e_i to obtain the solution at the $(n+1)$th Newton iteration. A modified DG discretization allows one to obtain the convergent solution with the polynomial degree $K > 0$ for the function $u_0^{\mathrm{II}}(x)$ considered as the initial guess.

Finally, let us mention that the problem of flux approximation in a high order DG discretization cannot be completely solved by considering another numerical flux in a problem. The discussion in this section demonstrates that the number of degrees of freedom should be equal to the number of equations on each grid cell in a high order DG scheme. Hence, many of the upwind fluxes which provide exact flux splitting at the flux extremum point are hardly appropriate for a high order discretization, as they do not meet the requirement above. For instance, the Godunov flux is defined as

$$\tilde{F}^G(u_l, u_r) = \begin{cases} \min\limits_{u_l \le u \le u_r} F(u) & \text{if } u_l \le u_r, \\ \max\limits_{u_l \le u \le u_r} F(u) & \text{otherwise} \end{cases}$$

for the left state u_l and right state u_r at a given grid interface. It can be seen from the flux definition that the Godunov flux presents us with the same problem as the Engquist–Osher flux (10). Since the discrete conservation law does not involve the solution near the extremum for a high order DG discretization, the number of unknowns exceeds the number of equations on the cell, where the extremum point is located. On the other hand, the space-centered fluxes, such as the local Lax–Friedrichs flux,

$$\tilde{F}^{LLF}(u_l, u_r) = \frac{1}{2}[F(u_l) + F(u_r) - C(u_r - u_l)],$$
$$C = \max_{min(u_l,u_r) \le s \le max(u_l,u_r)} |F'(s)|,$$

use both left and right solution states to approximate the flux, which makes the number of degrees of freedom equal to the number of equations on each grid cell. However, the space-centered fluxes cannot be considered a priori as a preferable choice for a high order DG discretization, as the issues of accuracy also should be taken into account. For instance, it is well known that the Lax–Friedrichs flux is more dissipative than upwind fluxes, and thus it may not be acceptable for a problem under consideration (e.g., see [14]). This makes the choice of a numerical flux in steady state problems a complicated task which requires further study and discussion.

4. The numerical flux for time-dependent problems. Approximate Riemann solvers have been successfully used many times in numerical solution of the hyperbolic systems of conservation laws (e.g., see [7] and the references therein). Thus, it is instructive to compare the results obtained for the steady state problem above with the convergence of a nonlinear solver for a time-dependent problem. The inviscid Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{15}$$

is a well-known example of a nonlinear hyperbolic equation which provides us with a quadratic flux function $f(u) = \frac{u^2}{2}$ similar to that in (8). We solve (15) in the domain $x \in [0, 1]$ due to a periodic boundary condition. The initial condition is taken from [3], where it has been chosen as a sine wave function,

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

The exact solution is smooth for any time $T < 1/\pi$, while the shock appears at later times.

For numerical solution of the conservation law (15), a DG discretization in space is combined with a backward Euler time integration scheme which results in a system of nonlinear equations at each time step. This system is linearized in order to obtain the solution at the upper time level. Notice that the choice of the initial condition requires the flux approximation near the extremum point $u = 0$. Nevertheless, the nonlinear solver provides a convergent solution at any time $T > 0$. An approximate

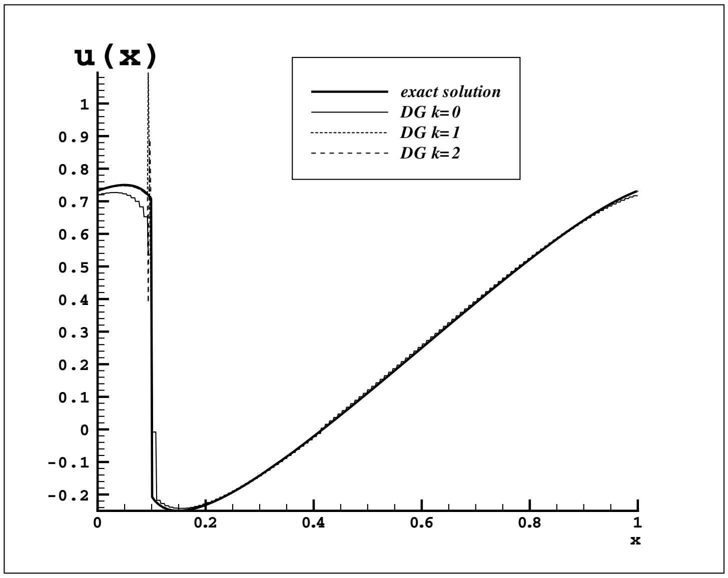

FIG. 2. *Exact and numerical solution to the Burgers equation. Smooth approximation to the solution generates local oscillations at the shock for the DG solution with polynomial degree $K > 0$. The solution overshoots can be eliminated by means of local limiters.*

solution at $T = 0.4$ is shown in Figure 2 for DG discretizations with polynomial degree $K \geq 0$ on a uniform grid of 128 cells. Let us notice that the DG $K = 1$ and $K = 2$ approximate solutions oscillate near the shock. However, those oscillations are local and can be eliminated by means of a limiting procedure [5].

The robustness of the nonlinear solver for the time-dependent problem (15) is readily explained based on the analysis of the Jacobian matrix. Consider the conservation law

$$\frac{\partial u}{\partial t} + F_x(x, u) = 0, \quad x \in \Omega. \tag{16}$$

The semidiscrete formulation of (16) on the cell e_i is

$$\int_{e_i} \frac{\partial u}{\partial t} \phi_k(x) dx + R_k^{DG}(\mathbf{u}) = 0, \quad k = 0, 1, \dots, K,$$

where $R_k^{DG}(\mathbf{u})$ is the DG residual given by (5).

Let $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ be the solution vector over the grid at time t^n and $t^{n+1} = t^n + \Delta t$, respectively. After discretizing in time, the implicit scheme for the hyperbolic equation (16) reads

$$\mathbf{M}(\mathbf{u}^{n+1} - \mathbf{u}^n) = -\Delta t \mathbf{R}^{DG}(\mathbf{u}^{n+1}), \tag{17}$$

where positive diagonal matrix $\mathbf{M}$ is given by

$$M_{kl} = \int_{e_i} \phi_k(x) \phi_l(x) dx, \quad k, l = 0, 1, \dots, K,$$

on each grid cell.

The linearization of the residual vector yields the system of equations

$$\mathbf{J}(\mathbf{u}^{n+1} - \mathbf{u}^n) = -\mathbf{R}(\mathbf{u}^n).$$

The Jacobian matrix is

$$\mathbf{J} = \frac{\mathbf{M}}{\Delta t} + \mathbf{J}^{DG},$$

where $\mathbf{J}^{DG}$ is the Jacobian of steady state problem (1), $\mathbf{J}^{DG} = [\partial \mathbf{R}^{DG}/\partial \mathbf{u}]$.

It can be seen from the above expression that the presence of mass matrix $\mathbf{M}$ in the discretization ensures diagonal entries in the Jacobian, even if matrix $\mathbf{J}^{DG}$ is singular. Hence, the time derivative can be considered as a stabilization term for high order DG discretizations.

5. The flux correction for steady state solutions. Lack of stabilization terms in steady state problems makes it difficult to use Newton's method for their numerical solution. On the one hand, it has been demonstrated in section 3 of the paper that an approximate Riemann solver may result in a singular matrix near the flux extremum. On the other hand, to obtain a convergent solution it is not sufficient to control the flux only near the extremum point. Below we demonstrate that a general case, unlike a simple model problem considered above, requires flux control on any grid cell.

Consider a scalar flux function $F(u)$. Any smooth function $F(u)$ that is not monotone in the domain of definition yields a multivalued solution $u(x)$ to the steady state equation (1). (From a geometric point of view, this means that the solution $F(x, u) = C$ to (1) intersects the curve $F(u)$ more than one time in the $(u, F(u))$-plane.) For the boundary-value problem (1), (2), the uniqueness of the solution is defined by a boundary condition.[1] However, a transient solution may experience jumps from one solution branch to another, until the basin of attraction is approached. Those local bifurcations may change the sign of the derivative dF/du and produce nonphysical flux extrema on the cell. As a result, a "phantom" solution appears on the cell, which, in turn, leads to incorrect flux approximation and a singular Jacobian in the problem.

The flux correction algorithm that we present in this section detects all flux extrema over the grid. It is important to notice here that one should distinguish between physical and nonphysical extrema in the problem. At the physical flux extremum, the solution should be reduced to piecewise constant approximation, as the number of degrees of freedom is not equal to the number of equations for a high order DG discretization. Nonphysical extrema appear in the regions where the flux should be a monotone function. In this case it is sufficient to render the flux monotone on a given cell, as the number of unknowns is equal to the number of equations on the cells where the flux is a monotone function.

The topic of the treatment of physical flux extrema has been covered in section 3. Below we discuss how to eliminate the nonphysical flux extrema in the problem. Let us denote the extremum points of the function $F(u)$ as $u_1, u_2, \ldots, u_{P-1}$. The domain of definition of the variable u can be partitioned as $\mathcal{D}_u = \bigcup_{p=0}^{P-1} [u_p, u_{p+1}]$, where u_0

[1] For a weak solution, additional constraints, such as the entropy condition (e.g., see [8]), are also required to provide the uniqueness of the solution.

and u_P are the boundary points of the domain. Consider the values $u(x_i - 0)$ and $u(x_{i+1} + 0)$, i.e., the approximate solution taken from the adjacent cells at the left and right interfaces of the cell e_i. Each of these values lies between two extremum (or boundary) points, $u(x_i - 0) \in [u_p, u_{p+1}]$, $u(x_{i+1} + 0) \in [u_q, u_{q+1}]$, where $0 \le p, q < P$. We now consider $u(x_i + 0)$ and $u(x_{i+1} - 0)$, which are the boundary values of the solution approximation in the cell e_i. Our goal is to detect nonphysical flux extrema within each grid cell. Instead of limiting the solution variation, we bound the flux variation in the cell in order to eliminate the flux oscillations.

Namely, we require that

$$u(x_i + 0) \in [u_p, u_{p+1}], \qquad u(x_{i+1} - 0) \in [u_q, u_{q+1}].$$

In other words, an approximate Riemann solver used in the problem must give the same choice of the numerical flux for the solution considered at $[u_l = u(x_i + 0), u_r = u(x_{i+1} - 0)]$ as for the interval $[u_l = u(x_i - 0), u_r = u(x_{i+1} + 0)]$.

From an algorithmic point of view, it is convenient to introduce the following formal description of our approach. Let us denote the left and right solution states at the interface x_i as u_{1i} and u_{2i}, respectively. Given numerical flux $\tilde{F}(u_h)$, we define *state vector* $\mathbf{s}_i = (s_{1i}, s_{2i})^T$ at each grid interface x_i, $i = 1, \dots, N+1$, as follows:

$$\mathbf{s}_{li} = \begin{cases} 1 & \text{if } u_{li} \text{ is required to define } \tilde{F}(u_h), \quad l = 1, 2, \\ 0 & \text{otherwise.} \end{cases}$$

Once the state vector has been defined at each grid interface, the cell e_i can be described by *state matrix* $\mathbf{S}_i$,

$$\mathbf{S}_i = \begin{bmatrix} s_{1\,i} & s_{1\,i+1} \\ s_{2\,i} & s_{2\,i+1} \end{bmatrix},$$

where the columns of the matrix $\mathbf{S}_i$ are state vectors taken at the left and right cell interface, respectively.

The values $u(x_i - 0)$ and $u(x_{i+1} + 0)$ define the main diagonal of the matrix $\mathbf{S}_i$, while the $u(x_i + 0)$ and $u(x_{i+1} - 0)$ define the off-diagonal entries. Hence, the flux within the cell can be controlled by means of the matrix $\mathbf{S}_i$. In particular, it can be easily seen that zero off-diagonal entries of the matrix indicate the "phantom" solution, which yields incorrect flux approximation in the cell e_i.

Below we illustrate our approach with a nonlinear boundary-value problem known as the problem of mass flow in a convergent-divergent nozzle [1]. Let A(x) be the area of the nozzle, $A(x) = 1/2 + 2(x - 1/2)^2$, $0 \le x \le 1$, and let $u(x)$ be the velocity deviation.

The conservation law is

$$\frac{dF(x, u(x))}{dx} \equiv \frac{d(A(x)m(u))}{dx} = 0, \quad x \in [0, 1], \tag{18}$$

where the mass flux through the nozzle is given by

$$m(u) = \frac{1}{2}(1 - u^2). \tag{19}$$

The value $u_s = 0$ (sonic point) is a flux extremum point.

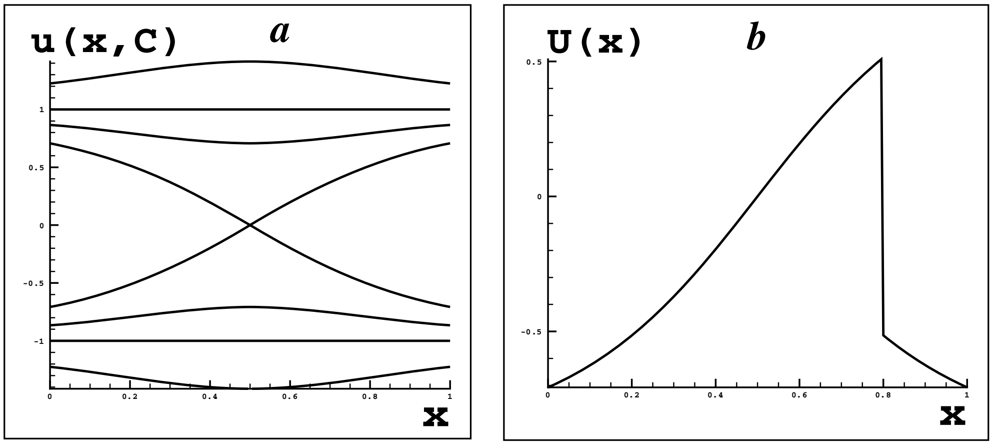

FIG. 3. *The nozzle problem.* (a) *The solution parametric field* $u(x, C)$*.* (b) *A discontinuous solution to the boundary-value problem* (18), (9).

A solution to the problem (18) is multivalued. The solutions are given by

$$u_{1,2}(x) = \pm\sqrt{1 - \frac{2C}{A(x)}},$$

where C is a constant. The solution parametric field $u(x, C)$ is shown in Figure 3(a). The value $x_s = 1/2$ is a solution extremum point for any $C \neq 0$ from the domain of the definition of C.

The value C is a controlling parameter for the problem. Let us choose $C = 1/4$, such that $u(x_s) = u_s$ and the point $P_s = (x_s, u_s)$ becomes the solution bifurcation point. Then the solution may be discontinuous at the point x_{sh}:

$$(20) \qquad U(x) = \begin{cases} -\sqrt{(1 - 1/2A(x))}, & 0 \leq x \leq x_s \text{ or } x_{sh} + 0 \leq x \leq 1, \\ \sqrt{(1 - 1/2A(x))}, & x_s \leq x \leq x_{sh} - 0. \end{cases}$$

Equation (18) is solved due to the boundary condition (9) which determines the shock location x_{sh}. Integrating the solution over the domain $[0, 1]$ yields the following algebraic equation with respect to the variable x_{sh}:

$$I_1 + I_2(x_{sh}) + I_3(x_{sh}) = B,$$

where

$$I_1 = -\int_0^{x_s} \sqrt{1 - 1/2A(x)}dx, \quad I_2(x_{sh}) = \int_{x_s}^{x_{sh}} \sqrt{1 - 1/2A(x)}dx,$$

$$I_3(x_{sh}) = -\int_{x_{sh}}^{1} \sqrt{1 - 1/2A(x)}dx.$$

Solving this equation for a given value of B, the shock location can be defined. If we choose $B = -0.25$, then the shock will be located at $x_{sh} = 0.798074$. The discontinuous solution $U(x)$ is shown in Figure 3(b).

Consider the approximate solution $u_h(x)$ at the interface x_i. Given the left and right states at the interface, the Engquist–Osher numerical flux is similar to that in (11),

$$\text{(21)}\quad \tilde{m}(u_l, u_r) = \begin{cases} m(u_r), \quad u_l < 0,\, u_r < 0 & \text{(subsonic case)},\\ m(u_l), \quad u_l > 0,\, u_r > 0 & \text{(supersonic case)},\\ m(0), \quad u_l < 0,\, u_r > 0 & \text{(sonic case)},\\ m(u_l) + m(u_r) - m(0), \quad u_l > 0,\, u_r < 0 & \text{(shock case)}. \end{cases}$$

First, we use the standard DG approach to solve the boundary problem (18), (9). The problem is solved on a sequence of uniform grids. The initial guess on the first grid of eight nodes is chosen as $u_0(x) = \text{const} = -1.0$. The initial guess for the next finer grid is obtained by linear interpolation of the solution taken from a previous grid. The results with the standard DG discretization are that Newton's method fails to obtain a convergent solution for any polynomial degree $K > 0$. Only a piecewise constant discretization reconstructs the discontinuous solution $U(x)$.

A DG discretization with polynomial degree $K > 0$ yields a singular Jacobian on a shock cell. However, simple reduction to piecewise constant approximation near the shock is not successful in the problem and results in a divergent solution. A more thorough control of the numerical flux is required. For this purpose, we compute the matrix $\mathbf{S}_i$ on each grid cell e_i, $i = 1, \ldots, N$, at each Newton step. The definition (21) gives us the following formal classification of the matrix $\mathbf{S}_i$:

$$\mathbf{S}_i = \begin{bmatrix} 0 & s_{1\,i+1} \\ s_{2\,i} & 1 \end{bmatrix} \text{ (subsonic case)}, \qquad \mathbf{S}_i = \begin{bmatrix} 1 & s_{1\,i+1} \\ s_{2\,i} & 0 \end{bmatrix} \text{ (supersonic case)},$$

$$\mathbf{S}_i = \begin{bmatrix} 0 & s_{1\,i+1} \\ s_{2\,i} & 0 \end{bmatrix} \text{ (sonic case)}, \qquad \mathbf{S}_i = \begin{bmatrix} 1 & s_{1\,i+1} \\ s_{2\,i} & 1 \end{bmatrix} \text{ (shock case)},$$

where $s_{1\,i+1}$ and $s_{2\,i}$ may take the value 0 or 1.

Based on the analysis of the state matrix $\mathbf{S}_i$, the correction algorithm, which eliminates nonphysical flux extrema for the problem (18), (9), is as follows.

1. *Compute the solution $u_h(x)$ on the cell e_i, $i = 1, \ldots, N$, at the nth Newton iteration. Compute the left and right states at each cell interface.*
2. *Compute the state matrix $\mathbf{S}_i$ on the cell e_i, $i = 1, \ldots, N$, and define the type of $\mathbf{S}_i$.*
3. *Mark the cell e_i for linear interpolation if*
   3.1 *$s_{2i} \neq 1$ or $s_{1\,i+1} \neq 0$ for subsonic $\mathbf{S}_i$,*
   3.2 *$s_{2i} \neq 0$ or $s_{1\,i+1} \neq 1$ for supersonic $\mathbf{S}_i$,*
   3.3 *$s_{2i} = 0$ and $s_{1\,i+1} = 0$ for sonic $\mathbf{S}_i$, or*
   3.4 *$\mathbf{S}_i$ is a shock state matrix.*
4. *If the cell is marked for linear interpolation, then define the approximate solution on the cell e_i as*
$$u_h^{lin}(x) = u_{1i} + (u_{2\,i+1} - u_{1i})\phi_1(x).$$
5. *For shock cell e_i, define the approximate solution on the cell e_i as*
$$u_h^c(x) = \frac{1}{h_i}\int_{x_i}^{x_{i+1}} u_h^{lin}(x)dx.$$
6. *Use the interpolated (piecewise linear or constant) solution on the marked cells to obtain $u_h(x)$ at the next Newton iteration.*

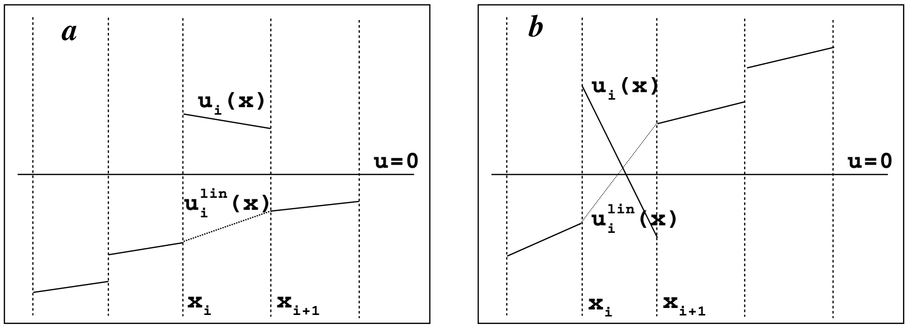

FIG. 4. *Examples of nonphysical flux extrema in the problem* (18), (9). (a) *The subsonic solution overshoot produces a sonic point and a shock at the cell interfaces.* (b) *The sonic solution overshoot produces a shock at the interior point of the cell and another sonic point at the cell interface.*

The above algorithm traces flux extrema over the grid for a transient solution at each Newton iteration. For the subsonic and supersonic cases, i.e., in the regions where the flux $m(u)$ is a monotone function, it is sufficient to control off-diagonal entries of $\mathbf{S}_i$ to ensure that there is no local bifurcation in the cell. We require that a transient solution generates the state matrix on the cell e_i as follows:

$$\mathbf{S}_i = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \text{ (subsonic case)}, \quad \mathbf{S}_i = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \text{ (supersonic case)}.$$

Let, for instance, the "subsonic" matrix be

$$\mathbf{S}_i = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

This matrix is related to the solution shown in Figure 4(a). The matrix $\mathbf{S}_i$ indicates that a local solution overshoot appears on the cell e_i. That overshoot produces two nonphysical flux extrema (a sonic point and a shock) which must be eliminated. For this purpose, we linearly interpolate a transient solution on the cell e_i between the points $u(x_i - 0)$ and $u(x_{i+1} + 0)$ (see Figure 4(a)).

For the sonic case, we must eliminate the state matrix

$$\mathbf{S}_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

which indicates a "phantom" sonic solution. Again, the solution on the cell e_i will be linearly interpolated between the points $u(x_i - 0)$ and $u(x_{i+1} + 0)$ if a sonic cell yields the above matrix (see Figure 4(b)). The other "sonic" matrices are legal. The matrix $\mathbf{S}_i$, which has $s_{1\,i+1} = 1$ and $s_{2\,i} = 1$, corresponds to the sonic point inside the cell e_i, while the other two matrices indicate the sonic point at the interface.

As it has been earlier discussed, the linear solution interpolation between the points $u(x_i - 0)$ and $u(x_{i+1} + 0)$ is not sufficient for the shock. The interpolated solution eliminates nonphysical flux extrema on the cell, but it remains "phantom" in the presence of the shock. In other words, the number of degrees of freedom is not equal to the number of equations on the shock cell for the interpolated solution. Thus, in our algorithm we reduce the solution to piecewise constant approximation at the shock.

Table 1

The number N of Newton iterations required to converge on a given grid. N_c is the number of grid cells.

$N(K, N_c)$:	$N_c = 8$	$N_c = 16$	$N_c = 32$	$N_c = 48$	$N_c = 64$
K=0	9	5	4	3	4
K=1	9	5	5	4	4
K=2	10	5	5	4	4
K=3	10	5	4	4	4

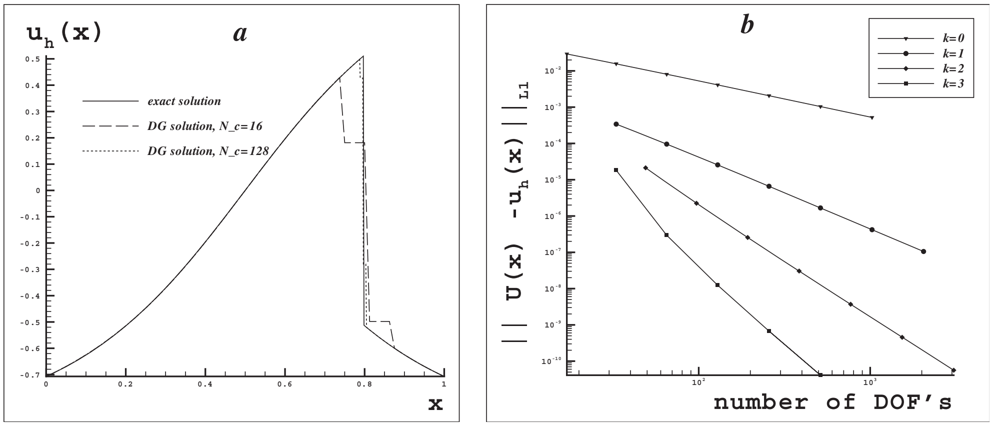

Fig. 5. *Numerical solution to the problem* (18), (9). (a) *An example of the DG solution (polynomial degree $K = 3$) on a coarse and a fine grid. N_c is the number of grid cells.* (b) *Convergence history for the DG solution with polynomial degree $K \geq 0$.*

The correction procedure allows one to obtain convergent solutions for DG $K > 0$ discretization schemes. Once the flux correction has been performed, the Newton method rapidly converges to the approximate solution. The number of Newton iterations required to converge on a given grid is displayed in Table 1 for the polynomial degree $K \geq 0$.

The approximate DG solution with polynomial degree $K = 3$ obtained by the flux correction is shown in Figure 5(a) on a coarse uniform grid of 16 cells and a fine grid of 128 cells. According to the correction algorithm, the shock is smeared over two adjacent grid cells, as an uncorrected solution has the shock at the grid interface at the final Newton step.

The convergence history on a sequence of uniform grids is plotted in Figure 5(b) for polynomial degree $K \geq 0$. The L_1-norm of the solution error,

$$||err||_{L_1} = \int_0^1 |U(x) - u_h(x)| dx,$$

is computed in regions where the solution is smooth (i.e., grid cells, which produce a shock state matrix, are not taken into account). The error norm is shown in the logarithmic scale. It can be seen from the convergence plots that the suggested algorithm keeps the order of approximation. The polynomial degree of the approximate solution is reduced only for a transient solution at the current Newton step. Once the solution is correct, the original polynomial degree will be restored on the cell at the next iterations. This approach allows one to obtain the optimal order of the convergence for high order DG discretizations.

6. Concluding remarks. We have considered high order DG schemes for steady state solutions. It has been shown that flux approximation near extremum points may generate spurious solution oscillations. Physical flux extrema require careful treatment to avoid a singular Jacobian in a steady state problem. Besides, false flux extrema may appear in a transient solution, when Newton's method is used to solve the problem. A high order DG discretization needs flux monitoring over each grid cell in order to eliminate nonphysical flux extrema.

The requirement of careful flux approximation makes Newton's method hardly appropriate for those steady state problems which do not have stabilization terms (e.g., diffusion and/or source terms) providing nonzero diagonal entries in the Jacobian. Although the flux control algorithm presented in the paper allows one to avoid a singular matrix in the one-dimensional case, it does not seem to always be efficient for multidimensional problems, where the construction of the state matrix on each grid cell becomes a complicated task.

The results of our paper confirm that a reasonable alternative to Newton's method is to use a time marching approach in order to obtain a steady state solution. It has been discussed in the paper that the time derivative can be considered as a stabilization term for high order DG schemes. However, an ill-conditioned Jacobian may appear at the end of the time stepping process when we approach "quasi-Newton" iterations. Thus, care should be taken of the flux approximation even in the case that a time stepping algorithm is used, and the issue of the numerical flux for high order DG discretizations requires further study when steady state problems are considered.

Acknowledgments. The author acknowledges the helpful discussions with F. T. Johnson, V. Venkatakrishnan, and other members of the GGNS team.

REFERENCES

[1] J. D. Anderson, Jr., *Fundamentals of Aerodynamics*, McGraw–Hill, New York, 1991.

[2] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang, *PETSc* Web page, http://www.mcs.anl.gov/petsc (2001).

[3] B. Cockburn, *Discontinuous Galerkin methods for convection-dominated problems*, in High-Order Discretization Methods in Computational Physics, Lect. Notes Comput. Sci. Eng. 9, T. Barth and H. Deconinck, eds., Springer-Verlag, Heidelberg, 1999, pp. 69–224.

[4] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, *The development of discontinuous Galerkin methods*, in Discontinuous Galerkin Methods. Theory, Computation and Applications, Lect. Notes Comput. Sci. Eng. 11, B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., Springer-Verlag, New York, 2000, pp. 3–50.

[5] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws.* II. *General framework*, Math. Comp., 52 (1989), pp. 411–435.

[6] H. Hoteit et al., *New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes*, Internat. J. Numer. Methods Engrg., 61 (2004), pp. 2566–2593.

[7] A. G. Kulikovskii, N. V. Pogorelov, and A. Yu. Semenov, *Mathematical Aspects of Numerical Solution of Hyperbolic Systems*, Chapman & Hall/CRC Monogr. Surv. Pure Appl. Math. 188, Chapman & Hall/CRC, Boca Raton, FL, 2001.

[8] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser-Verlag, Basel, Switzerland, 1992.

[9] R. B. Lowrier, *Compact Higher-Order Numerical Methods for Hyperbolic Conservation Laws*, Ph.D. thesis, The University of Michigan, Ann Arbor, MI, 1996.

[10] B. Engquist and S. Osher, *One-sided difference equations for nonlinear conservation laws*, Math. Comp., 36 (1981), pp. 321–352.

[11] W. H. Reed and T. R. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*, Technical Report LA-UR-73-479, Los Alamos National Laboratory, Los Alamos, NM, 1973.

[12] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Kent, UK, 1995.

[13] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[14] J. J. W. van der Vegt and H. van der Ven, *Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows.* I. *General formulation*, J. Comput. Phys., 182 (2002), pp. 546–585.