KELDYSH INSTITUTE OF APPLIED MATHEMATICS
RUSSIAN ACADEMY OF SCIENCES

N.B.Petrovskaya

Newton's Method for High - Order Schemes: As Good As It Gets?

Moscow

N.B.Petrovskaya

Newton's Method for High-Order Schemes: As Good As It Gets?

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 to obtain non-oscillating convergent solutions.

Introduction. [1]

In resent 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 a 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 which solution has 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 which eliminates the oscillations near discontinuities should be addressed. A number of authors have contributed to the issue of limiters for DG schemes in resent years (*e.g.* see [5], [6], [9]). It has been demonstrated many times that stabilization of the scheme by means of local limiters allows to obtain accurate non-oscillating 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 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

[1] This work was supported by The Boeing Company under contract No 104AE and Russian Foundation for Basic Research, grant No 03-01-00063.

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 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 which allows to obtain convergent solutions.

1. 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 the equation (1), where $\mathbf{B}$ denotes a boundary condition operator.

For numerical solution of the boundary problem (1), (2) we introduce the element partition G of the region, $G=\bigcup\limits_{i=1}^{N} e_i$, $e_i=[x_i,x_{i+1}], 1\le i\le N$, 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 the equation (1) by test function $\phi_k(x)$, defined on the cell e_i for

$k = 0, 1, \ldots, K$ as

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

and integrating by parts over the cell e_i, we obtain

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

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, the equation (3) considered for the solution $u_h(x)$ requires 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

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

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 n-th 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 n-th iteration. The GMRES algorithm ([12], [2]) is used to solve numerically the algebraic system of linear equations obtained at each Newton's 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 i-th 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 n-th Newton iteration are as follows.

1. The numerical flux required to define the DG discretization on the cell e_i is correct, but the solution approximation is not consistent. 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 i-th cell. The incorrect flux approximation produces solution oscillations which will propagate over the domain at next Newton's 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.

2. 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 the equation (1) which illustrates the problem. Let the flux $F(x,u)$ be

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

The problem parameters x_0, x_1, 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

$$\int\limits_0^1 u(x)dx = B, \tag{9}$$

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

$$\tilde{F}^{EO}(u_l, u_r) = \int\limits_0^{u_r} \min(F'(s), 0)ds + \int\limits_0^{u_l} \max(F'(s), 0)ds + F(0). \tag{10}$$

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

Despite the curves u_0^I and u_0^{II} are close to each other, $||u_0^{II} - u_0^I||_{L^\infty} = |s_0^{II} - s_0^I|, \quad x \in [0, 1]$, the convergence results are quite different for the two functions. Starting with the initial guess u_0^I, the Newton 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 in more detail 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, which results 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 only due to the solution variation at grid interfaces, i.e. $F(u) = const$ within a grid cell. Evidently, this assumption will be correct, if the 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 the both interfaces x_i and x_{i+1}. Thus, the flux approximation on the cell e_i will not be correct, unless the flux is monitored over the cell.

To identify the extremum point for a DG discretization with polynomial degree $k > 0$, we consider the boundary values of the approximate solution on a given cell. For the initial guess u_0^{II}, the flux approximation at the extremum point is required in two cells. One of them is $e_{i_1} : u_h(x_i + 0) > A, \quad u_h(x_{i+1} - 0) < A$, and another one is $e_{i_2} : u_h(x_i + 0) < A$, $u_h(x_{i+1} - 0) > A$. The flux approximation near the extremum point in the cell e_{i_1} is illustrated for the numerical flux (11) in fig.1, where the solution values required to define the flux at the interfaces are shown in black. For the piecewise constant DG discretization displayed in fig.1a, the extremum point at the interface is taken into account in the definition of the numerical flux. The high order (e.g. linear) solution reconstruction is shown in fig.1b. It can be seen from the figure that the flux extremum is not detected by the discretization. For a high order DG scheme, the "phantom" solution on the cell e_{i_1} is not involved into the flux definition.

We define local Jacobian on the cell e_{i_1} as $j_{k_1 k_2} = \dfrac{\partial R_{k_1}}{\partial u_{k_2}}$, where local indices $k_1, k_2 = 0, 1, \ldots, K$ are used to denote the residual and solution components on the cell. Let us show that the "phantom" solution on the cell e_{i_1} results in the singular local Jacobian. According to (11), the DG equations on the cell e_{i_1} are as follows

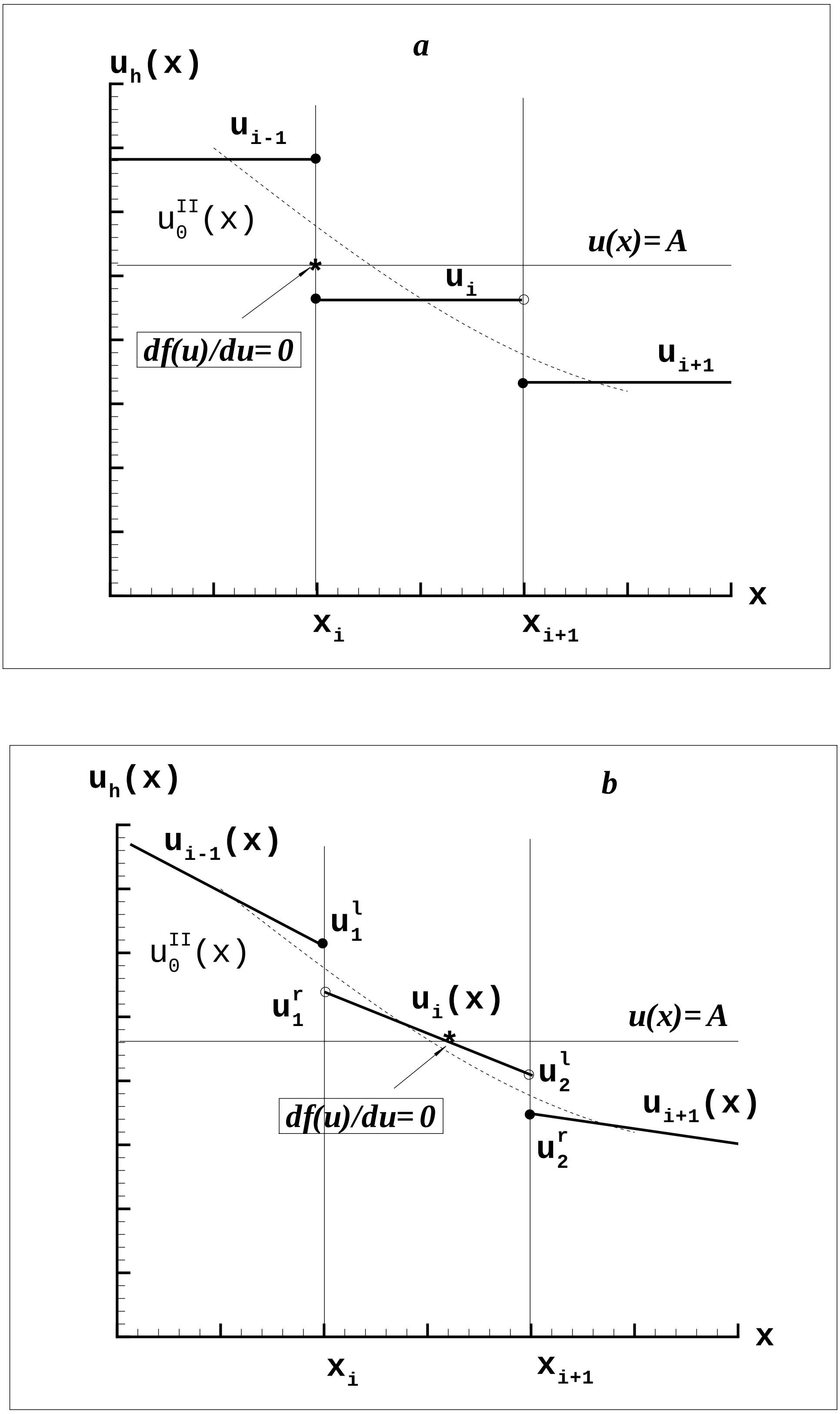

Figure 1: *The numerical flux near the extremum point for the problem (1), (8). The solution values at the interfaces required to define the numerical flux are schematically shown in black.(a) Piecewise constant solution approximation captures the flux extremum. (b) Higher order solution approximation misses the extremum point inside the grid cell.*

$$\begin{gathered}p(x_{i+1})f(u_2^r)-p(x_i)f(u_1^l)=0,\\ \vdots\\ p(x_{i+1})f(u_2^r)-\frac{K}{h_i^K}\int\limits_{x_i}^{x_{i+1}}p(x)f(u)(x-x_i)^{K-1}dx=0,\end{gathered}\tag{12}$$

where $u_1^l = u_h(x_i - 0)$ and $u_2^r = u_h(x_{i+1} + 0)$ are the solution values reconstructed in the adjacent cells (see fig.1b). Each of the DG equations (12) considered for the polynomial degree $k > 0$ contains an integral term, which linearization ensures nonzero entries in the local Jacobian. However, a discrete conservation law, i.e. the first of the equations (12), only requires the flux balance at the cell interfaces. Hence, if the definition of the numerical flux only concerns the solution on the adjacent cells, a zero row will appear in the matrix $j_{k_1k_2}$.

Consequently, the singularity of the local matrix affects the Jacobian in the case that the linearized system (7) is solved. Consider the block J_{lm} of the Jacobian which is related to the discretization on the cell e_{i_1}. Here the indices l and m are as follows

$$l \in L = 1, 2, \ldots, N(K+1)+1, \quad m \in M = M_1, M_1+1, \ldots, M_1+K,$$

where $M_1 = (i_1 - 1)(K + 1) + 1$, and N is the number of grid cells. Let now m_1 be a fixed number from the set M. Since the rank of the local Jacobian is $rank(j_{k_1k_2}) = K < dim(j_{k_1k_2}) = K + 1$, we can obtain by reordering the rows and columns of the matrix J_{lm} that

$$J_{lm_1} = 0, \quad \forall l \in L:\ M_1 \le l \le M_1 + K.$$

On the other hand, we have

$$J_{lm_1} = 0, \quad \forall l \in L:\ l < M_1 \text{ or } l > M_1 + K,$$

since the definition (11) provides the exact flux splitting for the problem. Hence, a zero column appears in the Jacobian of the system (7). The singular Jacobian leads to an incorrect transient solution (which appearance depends strongly on the robustness of the GMRES solver used in the problem). That solution, in turn, will impact on the flux at next Newton's iterations, so that the oscillations will rapidly propagate over the domain resulting in the divergence of the method.

An obvious way to correct the numerical flux in order to avoid nonphysical oscillations is to reduce the solution to piecewise constant approximation near a flux extremum. If the flux extremum generates a "phantom" solution in a given grid cell e_i (i.e. $u_l \equiv u_h(x_i+0) > A$ and $u_r \equiv u_h(x_{i+1}-0) < A$) at the n-th Newton iteration, then we compute $\bar{u} = \frac{1}{h_i}\int\limits_{x_i}^{x_{i+1}} u_h(x)dx$ and define the approximate solution on the cell e_i as $u_h(x) = \bar{u}$. In other words, we attach the extremum to the grid interface, so that the discretization (5), (11) is able to recognize it. The following 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 to obtain the convergent solution with the polynomial degree $k > 0$ for the function $u_0^{II}(x)$ considered as initial guess.

Let us mention again that the nature of oscillations arising in 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 correct approximation to the flux function $f(u)$ at the extremum point, while the solution remains a smooth monotone function near the flux extremum.

3. The numerical flux for time - dependent problems.

Approximate Riemann solvers have been successfully used many times in 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

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{13}$$

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 the equation (13) 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 $T < 0.4$, while the shock appears at later times.

For numerical solution of the conservation law (13), 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. That 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 solution at $T = 0.4$ is shown in fig.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 time-dependent problem (13) is readily explained based on the analysis of the Jacobian matrix. Consider the conservation law

$$\frac{\partial u}{\partial t} + F_x(x, u) = 0, \quad x \in \Omega. \tag{14}$$

The semi-discrete formulation of the equation (14) on the cell e_i is

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

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 (14) reads

$$\mathbf{M}(\mathbf{u}^{n+1} - \mathbf{u}^n) = -\Delta t \mathbf{R}^{DG}(\mathbf{u}^{n+1}) \tag{15}$$

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

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

on each grid cell.

The linearization of the residual vector yields the following system of equations

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

The Jacobian matrix is

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

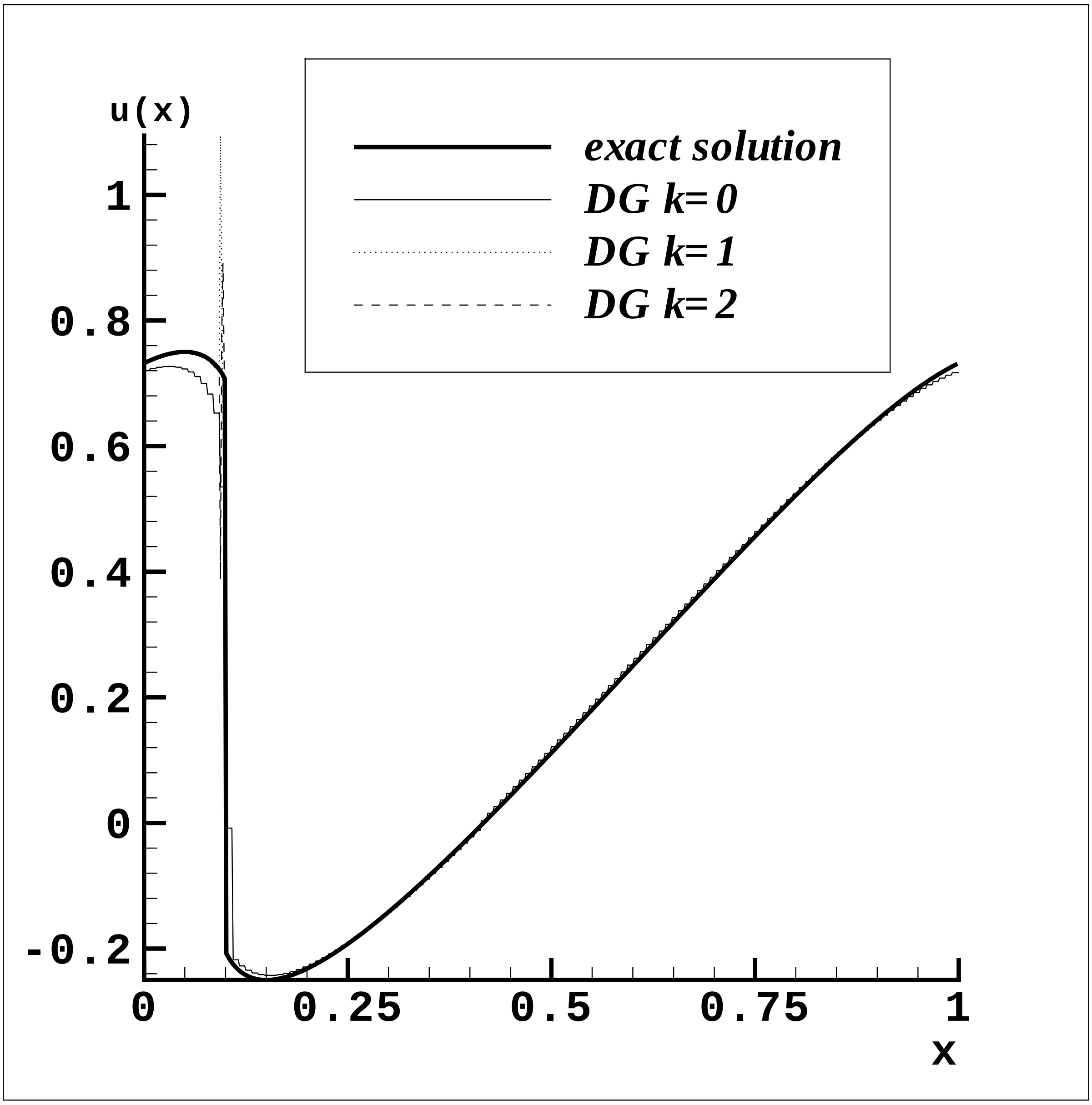

Figure 2: *Exact and numerical solution to the Burgers equation. Smooth approximation to the solution at the shock generates local oscillations for the DG solution with polynomial degree $k > 0$. The solution overshoots can be eliminated by means of local limiters.*

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 expression above 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.

4. The flux correction for steady state solutions.

Lack of stabilization terms in a steady state problem makes it difficult to use Newton's method for numerical solution. On the one hand, the problem of flux approximation at the extremum point cannot be completely solved by considering another numerical flux in a high order DG discretization. Any approximate Riemann solver which provides an exact flux splitting will result in a singular matrix near the extremum. On the other hand, it not sufficient to control the flux only near the extremum point to obtain a convergent solution. 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 $F(u)$, which is not monotone in the domain of definition, yields a multivalued solution to the steady state equation (1). (From a geometric point of view, this means that the solution $F(x, u) = C$ to the equation (1) intersects the curve $F(u)$ more than one time in the $(u, F(u))$ - plane.) For the boundary problem (1), (2), the uniqueness of the solution is defined by a boundary condition[2]. 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. Thus, the local bifurcations must be eliminated to avoid a "phantom" solution on the cell which leads to a singular Jacobian in the problem.

Let us denote the extremum points of the function $F(u)$ as $u_1, u_2, ..., u_{P-1}$. The domain of definition of the variable u can be partitioned as $\mathcal{D}_u = \bigcup\limits_{p=0}^{P-1} [u_p, u_{p+1}]$, where u_0 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 interface 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 $0 \leq p, q < P$.

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

[2] 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.

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 solution 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 state at the interface x_i as u_{1i} and u_{2i}, respectively. Given numerical flux $\tilde{F}(u_h)$, we define *state vector* $\mathrm{s}_i = (s_{1i}, s_{2i})^T$ at each grid interface $x_i,\ i = 1, ..., N+1$, as follows

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

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 problem known as the problem of mass flow in a convergent - divergent nozzle, [1]. Let A(x) be the area of the nozzle, $A(x) = 1/2 + 2(x-1/2)^2, \quad 0 \le x \le 1$, and $u(x)$ be the velocity deviation. The conservation law is

$$\frac{dF(x,u(x))}{dx} \equiv \frac{d(A(x)m(u))}{dx} = 0, \quad x \in [0,1], \tag{16}$$

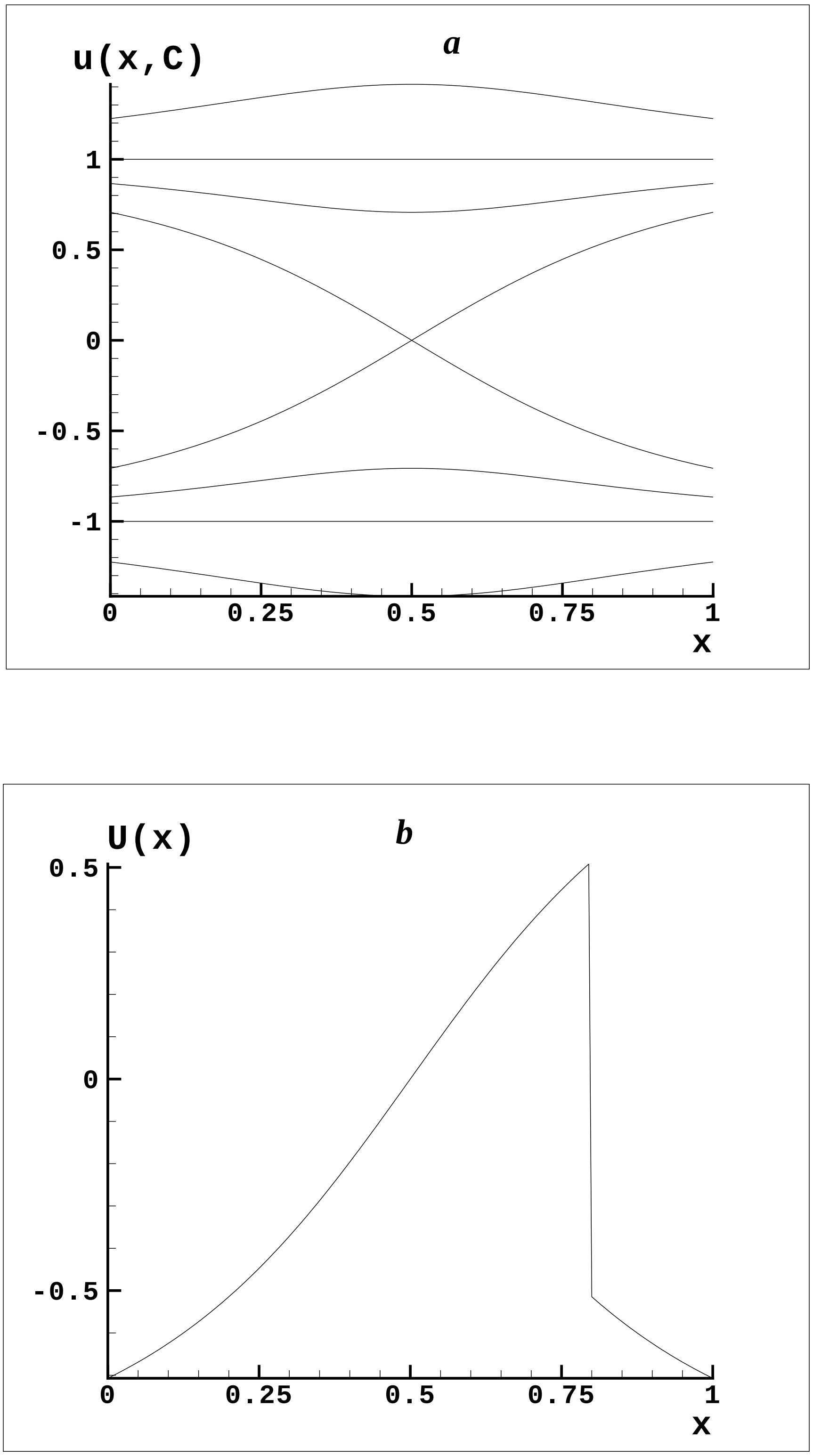

Figure 3: *The nozzle problem. (a) The solution parametric field $u(x, C)$. (b) A discontinuous solution to the boundary problem (16), (9).*

where the mass flux through the nozzle is given by

$$m(u) = \frac{1}{2}(1 - u^2). \tag{17}$$

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

A solution to the problem (16) is multivalued. The solutions are given by

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

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

The value C is a controlling parameter for the problem. Let us choose $C = 1/4$, so 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},

$$U(x) = \begin{cases} -\sqrt{(1 - 1/2A(x))}, & 0 \leq x \leq x_s, \ or \ x_{sh} + 0 \leq x \leq 1, \\ \sqrt{(1 - 1/2A(x))}, & x_s \leq x \leq x_{sh} - 0. \end{cases} \tag{18}$$

The equation (16) 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 $I_1 = -\int\limits_0^{x_s} \sqrt{1 - 1/2A(x)}dx$, $I_2(x_{sh}) = \int\limits_{x_s}^{x_{sh}} \sqrt{1 - 1/2A(x)}dx$, and $I_3(x_{sh}) = -\int\limits_{x_{sh}}^{1} \sqrt{1 - 1/2A(x)}dx$. 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 fig.3b.

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

$$\tilde{m}(u_l, u_r) = \begin{cases} m(u_r), & u_l < 0, \ u_r < 0, \ \text{(subsonic case)}, \\ m(u_l), & u_l > 0, \ u_r > 0, \ \text{(supersonic case)}, \\ m(0), & u_l < 0, \ u_r > 0, \ \text{(sonic case)}, \\ m(u_l) + m(u_r) - m(0), & u_l > 0, \ u_r < 0, \ \text{(shock case)}. \end{cases} \tag{19}$$

First, we use the standard DG approach to solve the boundary problem (16), (9). The problem is solved on a sequence of uniform grids. The initial guess on the first grid of 8 nodes is chosen as $u_0(x) = const = -1.0$. 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 in a shock cell. However, simple reducing 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, $i = 1, ..., N$ at each Newton step. The definition (19) gives us the following formal classification of the matrix $\mathbf{S}_i$:

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

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

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 (16), (9), is as follows.

1. *Compute the solution $u_h(x)$ on the cell e_i, $i = 1, ..., N$ at the n-th 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, $i = 1, ..., N$ 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 $\mathbf{S}_i$,

   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*

$$u_h^c(x) = \frac{1}{h_i}\int\limits_{x_i}^{x_{i+1}} u_h^{lin}(x)dx.$$

6. *Use the interpolated (piecewise linear or constant) solution on the marked cells to obtain $u_h(x)$ at the next Newton iteration.*

The above algorithm traces flux extrema over the grid for a transient solution at each Newton's iteration. For the subsonic and supersonic cases 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 fig.4a. The matrix $\mathbf{S}_i$ indicates that a local solution overshoot presents 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 fig.4a).

For the sonic case, we require to 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 fig.4b).

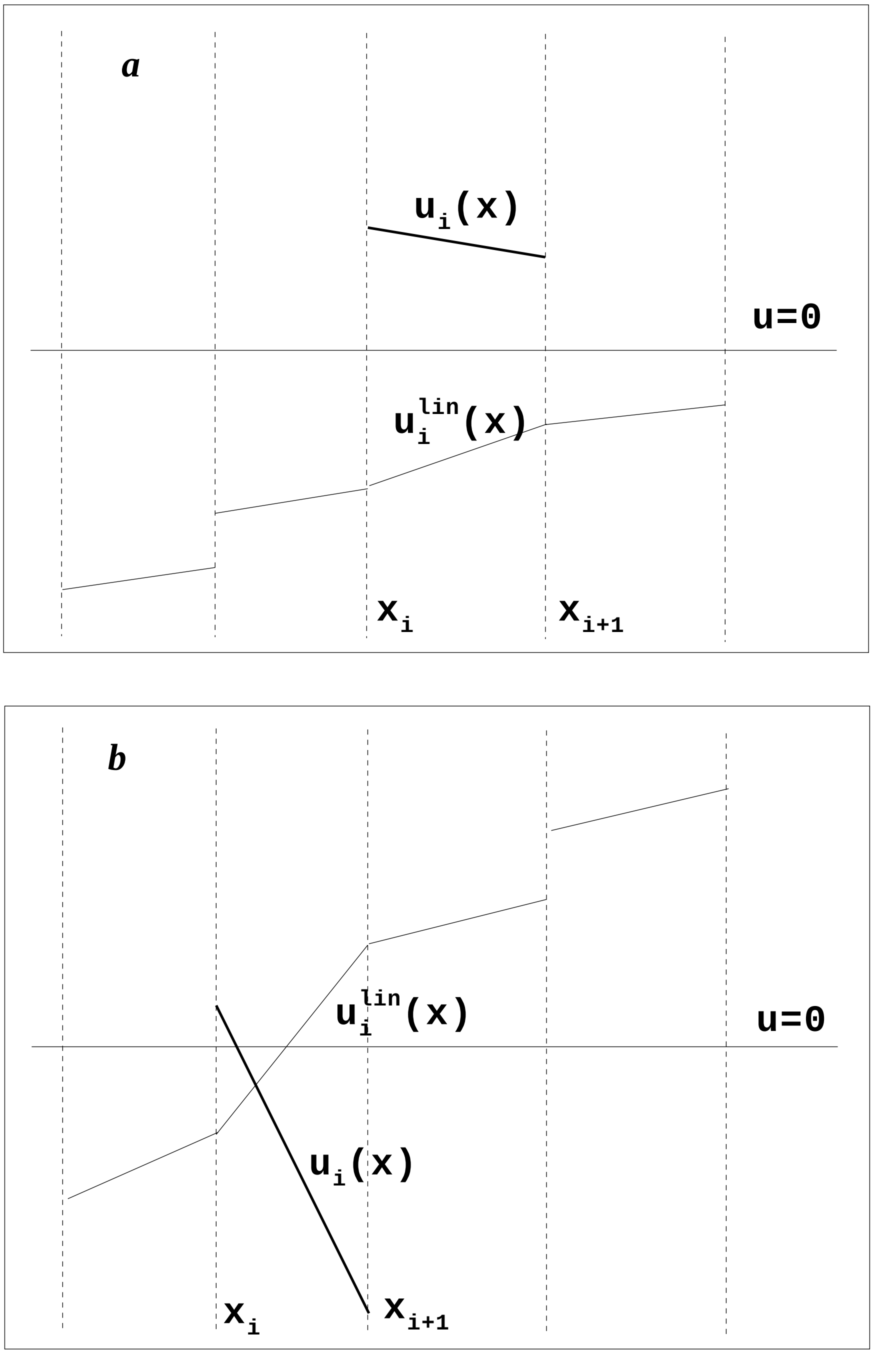

Figure 4: *Examples of nonphysical flux extrema in the problem (16), (9). (a) The subsonic solution overshoot produces a sonic point and a shock at the cell interfaces. (b) The sonic solution overshoot produces a shock at the interior point of the cell and another sonic point at the cell interface.*

The 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 two other 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 remains "phantom" in the presence of the shock. Thus, in our algorithm we reduce the solution to piecewise constant approximation at the shock. The correction procedure affords 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's iterations required to converge on a given grid is displayed in Table 1 for the polynomial degree $k \geq 0$.

Table 1

The number N of Newton's 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

The approximate DG solution with polynomial degree $k = 3$ obtained by the flux correction is shown in fig.5a on a coarse uniform grid of 16 cells and 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 fig.5b for polynomial degree $k \geq 0$. The L_1 - norm of the solution error, $||err||_{L_1} = \int\limits_0^1 |U(x) - u_h(x)|dx$, 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 only reduced for a transient solution at the current Newton step.

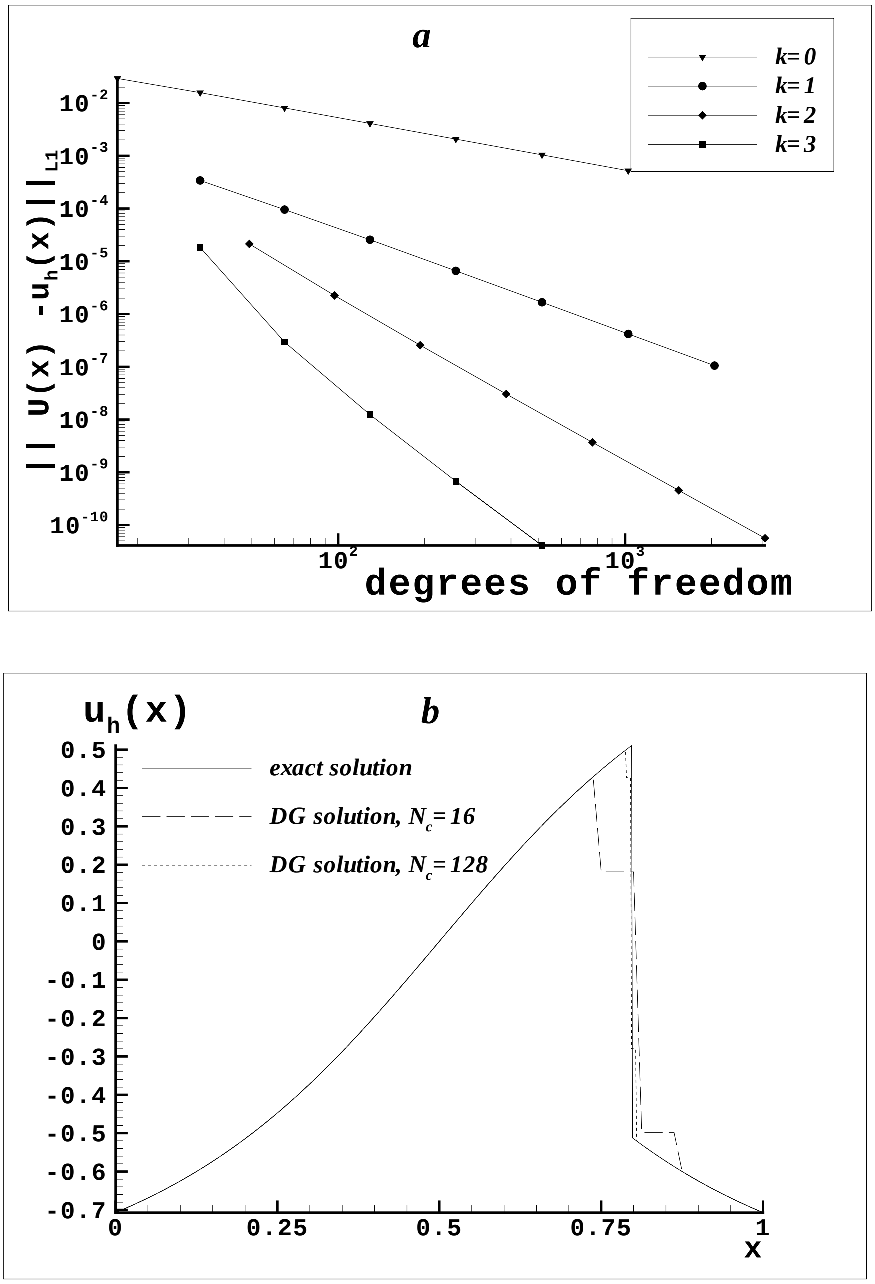

Figure 5: *Numerical solution to the problem (16), (9). (a) An example of the DG solution (polynomial degree $k = 3$) on a coarse and fine grid. N_c is the number of grid cells. (b) Convergence history for the DG solution with polynomial degree $k \geq 0$.*

Once the solution is correct, the original polynomial degree will be restored on the cell at next iterations. This approach allows to obtain the optimal order of the convergence for high order DG discretizations.

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 detect flux extremum points.

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 to avoid a singular matrix in the one-dimensional case, it does not seem to be always 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 and V.Venkatakrishnan.

References

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

[2] S.Balay, W.Gropp, L.C.McInnes, and D.Smith. *PETSc* 2.0 *Users Manual*, Technical Report ANL 95/11, Argonne National Laboratory, 1997, http://www.mcs.anl.gov/petsc/petsc.html

[3] B.Cockburn. *Discontinuous Galerkin Methods for Convection - Dominated Problems*, in High-Order Discretization Methods in Computational Physics, T.Barth and H.Deconinck, eds., Lecture Notes in Comput.Sci.Engrg., 9, 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, B.Cockburn, G.E.Karniadakis, and C.-W. Shu, eds., Lecture Notes in Comput.Sci.Engrg., 11, Springer-Verlag, New York, 2000, pp.3-50.

[5] B.Cockburn and C.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*, INRIA report No. 4491, INRIA Rennes, France, 2002.

[7] A.G.Kulikovskii, N.V.Pogorelov, A.Yu.Semenov. *Mathematical Aspects of Numerical Solution of Hyperbolic Systems*, Monographs and Surveys in Pure and Applied Mathematics, 188, Chapman and Hall/CRC, Boca Raton, Florida, 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*, PhD thesis, The University of Michigan, 1996.

[10] B.Engquist, 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, New Mexico, 1973.

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