SIAM J. SCI. COMPUT.
Vol. 41, No. 3, pp. A2037–A2063

POSITIVITY PRESERVING LIMITERS FOR TIME-IMPLICIT HIGHER ORDER ACCURATE DISCONTINUOUS GALERKIN DISCRETIZATIONS*

J. J. W. VAN DER VEGT†, YINHUA XIA‡, AND YAN XU§

Abstract. Currently, nearly all positivity preserving discontinuous Galerkin (DG) discretizations of partial differential equations are coupled with explicit time integration methods. Unfortunately, for many problems this can result in severe time-step restrictions. The techniques used to develop explicit positivity preserving DG discretizations cannot, however, easily be combined with implicit time integration methods. In this paper, we therefore present a new approach. Using Lagrange multipliers, the conditions imposed by the positivity preserving limiters are directly coupled to a DG discretization combined with a diagonally implicit Runge–Kutta time integration method. The positivity preserving DG discretization is then reformulated as a Karush–Kuhn–Tucker (KKT) problem, which is frequently encountered in constrained optimization. Since the limiter is only active in areas where positivity must be enforced, it does not affect the higher order DG discretization elsewhere. The resulting nonsmooth nonlinear algebraic equations have, however, a different structure compared to most constrained optimization problems. We therefore develop an efficient active set semismooth Newton method that is suitable for the KKT formulation of time-implicit positivity preserving DG discretizations. Convergence of this semismooth Newton method is proven using a specially designed quasi-directional derivative of the time-implicit positivity preserving DG discretization. The time-implicit positivity preserving DG discretization is demonstrated for several nonlinear scalar conservation laws, which include the advection, Burgers, Allen–Cahn, Barenblatt, and Buckley–Leverett equations.

Key words. positivity preserving, maximum principle, Karush–Kuhn–Tucker equations, discontinuous Galerkin methods, implicit time integration methods, semismooth Newton methods

AMS subject classifications. 65M60, 65K15, 65N22

DOI. 10.1137/18M1227998

1. Introduction.

The solution of many partial differential equations frequently must satisfy a maximum principle, or, more generally, certain variables must obey a lower and/or upper bound. In this paper, we will denote all these cases with positivity preserving. In particular, if the partial differential equations model physical processes, then these bounds are also crucial to obtain a meaningful physical solution. For example, a density, concentration, or pressure in fluid flow must be nonnegative, and a probability distribution should be in the range $[0, 1]$. A numerical solution should therefore strictly obey the bounds on the exact solution; otherwise, the problem can become ill-posed and the solution would be meaningless. Also, the numerical

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 21, 2018; accepted for publication (in revised form) April 4, 2019; published electronically June 25, 2019.

http://www.siam.org/journals/sisc/41-3/M122799.html

Funding: The first author's research was supported by the University of Science and Technology of China (USTC). The second author's research was supported by NSFC grants 11471306 and 11871449 and a grant from the Science & Technology on Reliability & Environmental Engineering Laboratory (6142A0502020817). The third author's research was supported by NSFC grants 11722112 and 91630207.

†Department of Applied Mathematics, Mathematics of Computational Science Group, University of Twente, Enschede, 7500 AE, The Netherlands (j.j.w.vandervegt@utwente.nl).

‡School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, People's Republic of China (yhxia@ustc.edu.cn).

§Corresponding author. School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, 230026, People's Republic of China (yxu@ustc.edu.cn).

algorithm can easily become unstable and lack robustness if the numerical solution violates these essential bounds.

In recent years, the development of positivity preserving discontinuous Galerkin (DG) finite element methods therefore has been a very active area of research. The standard approach to ensure that the numerical solution satisfies the bounds imposed by the partial differential equations is to use limiters, but this can easily result in loss of accuracy, especially for higher order accurate discretizations.

In a seminal paper, Zhang and Shu [34] showed how to design maximum principle and positivity preserving higher order accurate DG methods for first order scalar conservation laws. Their algorithm consists of a several important steps: (i) starting from a bounds preserving solution at time t_n, ensure that the element average of the solution satisfies the bounds at the next time level t_{n+1} by selecting a suitable time step in combination with a monotone first order scheme; (ii) limit the higher order accurate polynomial solution at the quadrature points in each element without destroying the higher order accuracy; (iii) higher order accuracy in time can then be easily obtained using explicit SSP Runge–Kutta methods [31]. This algorithm has been subsequently extended in many directions, e.g., various element shapes, the convection-diffusion equation, Euler and Navier–Stokes equations, and relativistic hydrodynamics [37, 38, 35, 36, 33, 29]. Other approaches to obtain higher order positivity preserving DG discretizations can be found in, e.g., [5, 13, 12].

All these DG discretizations use, however, an explicit time integration method. For many partial differential equations, this results in an efficient numerical discretization, where to ensure stability the time step is restricted by the Courant–Friedrichs–Lewy (CFL) condition. On locally dense meshes and for higher order partial differential equations, which often have a time step constraint $\triangle t \leq Ch^p$, with $p > 1$ and h the mesh size, these time-explicit algorithms can become computationally very costly. The alternative is to resort to implicit time integration methods, but positivity preserving time-implicit DG discretizations are still very much in their infancy. Meister and Ortleb developed in [22] a positivity preserving DG discretization for the shallow water equations using the Patankar technique [26]. Qin and Shu [28] extended the framework in [34, 35] to implicit positivity preserving DG discretizations of conservation laws in combination with an implicit Euler time integration method. An interesting result of the analysis in [28] is that to ensure positivity in the algorithm of Qin and Shu a lower bound on the time step is required. The approaches in [22, 28] require, however, a detailed analysis of the time-implicit DG discretization to ensure that the bounds are satisfied and are not so easy to extend to other classes of problems.

In this paper, we will present a very different approach to develop positivity preserving higher order accurate DG discretizations that are combined with a diagonally implicit Runge–Kutta (DIRK) time integration method. In analogy with obstacle problems, we consider the bounds imposed by a maximum principle or positivity constraint as a restriction on the DG solution space. The constraints are then imposed using a limiter and directly coupled to the time-implicit higher order accurate DG discretization using Lagrange multipliers. The resulting equations are the well-known Karush–Kuhn–Tucker (KKT) equations, which are frequently encountered in constrained optimization and solved with a semismooth Newton method [11, 17], and also used in constrained optimization-based discretizations of partial differential equation in, e.g., [3, 8, 10, 20]. The key benefit of the approach discussed in this paper, which we denote by KKT-Limiter and so far has not been applied to positivity preserving time-implicit DG discretizations, is that no detailed analysis is required to ensure that the DG discretization preserves the bounds for a particular partial differ-

ential equation. They are imposed explicitly and not part of the DG discretization. Also, since the limiter is only active in areas where positivity must be enforced, it does not affect the higher order DG discretization elsewhere since the Lagrange multipliers will be zero there. The approach discussed in this paper presents a general framework for how to couple DG discretizations with limiters and, very importantly, how to efficiently solve the resulting nonlinear algebraic equations.

The algebraic equations resulting from the KKT formulation of the positivity preserving time-implicit DG discretization are only semismooth. This excludes the use of standard Newton methods since they require C^1 continuity [9]. The obvious choice would be to use one of the many semismooth Newton methods available for nonlinear constrained optimization problems [11, 17], but the algebraic equations for the positivity preserving time-implicit DG discretization have a structure different from that for most constrained optimization problems. For instance, the conditions to ensure a nonsingular Jacobian [11] for methods based on the Fischer–Burmeister or related complementarity functions [23, 4] are not met by the KKT-Limiter in combination with a time-implicit DG discretization. This frequently results in nearly singular Jacobian matrices, poor convergence, and lack of robustness. We therefore developed an efficient active set semismooth Newton method that is suitable for the KKT formulation of time-implicit positivity preserving DG discretizations. Convergence of this semismooth Newton method can be proven using a specially designed quasi-directional derivative, as outlined in [15]; see also [17, 18].

The organization of this paper is as follows. In section 2, we formulate the KKT-equations, followed in section 3 by a discussion of an active set semismooth Newton method that is suitable to solve the nonlinear algebraic equations resulting from the positivity preserving time-implicit DG discretization. Special attention will be given to the quasi-directional derivative, which is an essential part to ensure convergence of the semismooth Newton method. In section 4, we discuss the DG discretization in combination with a DIRK time integration method and positivity constraints. In section 5, numerical experiments for the advection, Burgers, Allen–Cahn, Barenblatt, and Buckley–Leverett equations are provided. Conclusions are drawn in section 6. In Appendix B, more details on the quasi-directional derivative are given.

2. KKT limiting approach. In this section, we will directly couple the bounds preserving limiter to the time-implicit discontinuous Galerkin discretization using Lagrange multipliers. We will denote this approach as the KKT-Limiter.

Define the set

$$K := \{x \in \mathbb{R}^n \mid h(x) = 0,\ g(x) \leq 0\},$$

where $h : \mathbb{R}^n \to \mathbb{R}^l$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable functions denoting, respectively, the l equality and m inequality constraints to be imposed on the DG discretization. The variable x denotes the degrees of freedom and n the number of degrees of freedom in the unlimited DG discretization. For the continuously differentiable function $L : \mathbb{R}^n \to \mathbb{R}^n$, representing the unlimited discontinuous Galerkin discretization, the KKT-equations are

$$\mathcal{L}(x, \mu, \lambda) := L(x) + \nabla h(x)^T \mu + \nabla g(x)^T \lambda = 0, \tag{2.1a}$$

$$-h(x) = 0, \tag{2.1b}$$

$$0 \geq g(x) \perp \lambda \geq 0, \tag{2.1c}$$

with $\mu \in \mathbb{R}^l$, $\lambda \in \mathbb{R}^m$ the Lagrange multipliers. The compatibility condition (2.1c) is componentwise equal to

$$0 \geq g_j(x), \qquad \lambda_j \geq 0 \quad \text{and} \quad g_j(x)\lambda_j = 0, \quad j = 1, \ldots, m,$$

which is equivalent to

$$\min(-g(x), \lambda) = 0,$$

where the min-function is applied componentwise. The KKT-equations, with $F(z) \in \mathbb{R}^{n+l+m}$, can now be formulated as

$$0 = F(z) := \begin{pmatrix} \mathcal{L}(x, \mu, \lambda) \\ -h(x) \\ \min(-g(x), \lambda) \end{pmatrix}, \tag{2.2}$$

where $z := (x, \mu, \lambda)$. In the next section, we will discuss a global active set semismooth Newton method suitable for the efficient solution of (2.2) in combination with a DIRK-DG discretization. In section 4, the DG discretization and KKT-Limiter will be presented for a number of scalar conservation laws.

3. Semismooth Newton method. Standard Newton methods assume that $F(z)$ is continuously differentiable [9], but $F(z)$ given by (2.2) is only semismooth [11]. In this section, we will present a robust active set semismooth Newton method for (2.2) that is suitable for the efficient solution of the KKT-equations resulting from a higher order DG discretization combined with positivity preserving limiters and a DIRK time integration method [14].

3.1. Differentiability concepts. For the definition of the semismooth Newton method, we need several more general definitions of derivatives, which will be discussed in this section. For more details, we refer the reader to, e.g., [6, 11, 17, 30]. Since we use the semismooth Newton method directly on the algebraic equations of the limited DIRK-DG discretization, we only consider finite-dimensional spaces here.

Let $D \subseteq \mathbb{R}^m$ be an open subset in $\mathbb{R}^m$. Given $d \in \mathbb{R}^m$, the directional derivative of $F : D \to \mathbb{R}^n$ at $x \in D$ in the direction d is defined as

$$F'(x; d) := \lim_{t \downarrow 0^+} \frac{F(x + td) - F(x)}{t}. \tag{3.1}$$

A function $F : D \to \mathbb{R}^n$ is locally Lipschitz continuous if for every $x \in D$ there exist a neighborhood $N_x \subseteq D$ and a constant C_x, such that

$$|F(y) - F(z)| \leq C_x |y - z| \qquad \text{for all } y, z \in N_x.$$

If F is locally Lipschitz on D, then according to Rademacher's theorem, F is differentiable almost everywhere with derivative $F'(x)$. The B-subdifferential $\partial_B F(x)$ of $F(x)$ is then defined as

$$\partial_B F(x) := \lim_{\bar{x} \to x, \bar{x} \in D_F} F'(\bar{x}),$$

with D_F the points where F is differentiable, and the generalized derivative in the sense of Clarke is defined as

$$\partial F(x) := \text{ convex hull of } \partial_B F(x).$$

For example, $F(x) = |x|$ at $x = 0$ has $\partial_B F(0) = \{-1, 1\}$ and $\partial F(0) = [-1, 1]$. A function $F : D \to \mathbb{R}^n$ is called semismooth if [27]

$$\lim_{V \in \partial F(x+td'), d' \to d, t \downarrow 0^+} Vd' \quad \text{exists for all } d \in \mathbb{R}^m.$$

A function $F : D \to \mathbb{R}^n$ is Bouligand-differentiable (B-differentiable) at $x \in D$ if it is directionally differentiable at x and

$$\lim_{d \to 0} \frac{F(x+d) - F(x) - F'(x;d)}{|d|} = 0.$$

A locally Lipschitz continuous function F is B-differentiable at x if and only if it is directionally differentiable at x [30].

Given $d \in \mathbb{R}^m$, the Clarke generalized directional derivative of $F : D \to \mathbb{R}^n$ at $x \in D$ in the direction of d is defined by [6]

$$F^0(x;d) := \lim_{y \to x} \sup_{t \downarrow 0^+} \frac{F(y+td) - F(y)}{t}.$$

3.2. Global active set semismooth Newton method. For the construction of a global semismooth Newton method for (2.2), we will use the merit function $\theta(z) = \frac{1}{2}|F(z)|^2$, with $z = (x, \mu, \lambda)$. The Clarke directional derivatives of θ and F have the following relation.

Let $F : D \subseteq \mathbb{R}^p \to \mathbb{R}^p$, with D an open set and $p = n + l + m$, be a locally Lipschitz continuous function; then the Clarke generalized directional derivative of $\theta(z)$ can be expressed as [17]

$$\theta^0(z;d) = \limsup_{y \to z, t \downarrow 0^+} \frac{(F(z), (F(y+td) - F(y))}{t}, \tag{3.2}$$

and there exists an $F^0 : D \times \mathbb{R}^p \to \mathbb{R}^p$ such that

$$\theta^0(z;d) = (F(z), F^0(z;d)) \qquad \text{for } (z,d) \in D \times \mathbb{R}^p. \tag{3.3}$$

Here $(\cdot, \cdot)$ denotes the Euclidean inner product. The crucial point in designing a Newton method is to obtain proper descent directions for the Newton iterations. A possible choice is to use the Clarke derivative ∂F as the generalized Jacobian [11, 17], but this derivative is in general difficult to compute. In [24, 25], it was proposed to use d as the solution of

$$F(z) + F'(z;d) = 0, \tag{3.4}$$

which for the KKT-equations results in a mixed linear complementarity problem [25]. Unfortunately, (3.4) does not always have a solution, unless additional conditions are imposed. A better alternative is to use the quasi-directional derivative G of F [15, 17, 18].

Let $F : D \subseteq \mathbb{R}^p \to \mathbb{R}^p$ be directionally differentiable and locally Lipschitz continuous. Assume that $S = \{z \in D \mid |F(z)| \leq |F(z^0)|\}$ is bounded. Then $G : S \times \mathbb{R}^p \to \mathbb{R}^p$ is called the quasi-directional derivative of F on $S \subset \mathbb{R}^p$ if for all $z, \bar{z} \in S$ the following

conditions hold [15, 17, 18]:

$$(F(z), F'(z;d)) \leq (F(z), G(z;d)), \tag{3.5a}$$

$$G(z;td) = tG(z;d) \quad \text{for all } d \in \mathbb{R}^p,\ z \in S, \text{ and } t \geq 0, \tag{3.5b}$$

$$(F(\bar{z}), F^0(\bar{z};\bar{d})) \leq \limsup_{z\to\bar{z},d\to\bar{d}} (F(z), G(z;d)) \quad \text{for all } z \to \bar{z}, d \to \bar{d}. \tag{3.5c}$$

The search direction d in the semismooth Newton method is now the solution of

$$F(z) + G(z;d) = 0, \quad \text{with } z \in S, d \in \mathbb{R}^p, \tag{3.6}$$

which results for the KKT-equations (2.2) in a mixed linear complementarity problem. Using (3.3), (3.5c), and (3.6) this immediately results in the bound

$$\theta^0(\bar{z};\bar{d}) \leq \limsup_{z\to\bar{z},d\to\bar{d}} (F(z), G(z;d)) = -\lim_{z\to\bar{z}} |F(z)|^2 = -2\theta(\bar{z}).$$

Hence the search direction d obtained from (3.6) always provides a descent direction for the merit function $\theta(z)$. The merit function $\theta(z)$ and the quasi-directional derivative $G(z,d)$ can therefore be used to define a global line search semismooth Newton algorithm, which is stated in Algorithm 3.1. The key benefit of using the quasi-directional derivative G in this Newton algorithm is that, under the additional assumption $\|G(z;d)\| \geq L\|d\|$, with $L > 0$ constant, we immediately obtain a proof of the convergence of this algorithm, given by [15, Theorem 1].

In the next section, we will present the quasi-directional derivative G for the KKT-equations (2.2) and define the active sets used to solve (3.6) with the semismooth Newton algorithm presented in section 3.4. In section 4, Algorithm 3.1 will then be used to solve the nonlinear equations resulting from the DG discretization using a KKT-limiter in combination with a DIRK method.

3.3. Quasi-directional derivative. In order to compute the quasi-directional derivative G, satisfying the conditions stated in (3.5), we first need to compute the directional and Clarke generalized directional derivatives of the function $F(z)$ defined in (2.2).

Define $z \in \mathbb{R}^p$, with $p = n + l + m$ as $z = (x, \mu, \lambda)$ with $x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^l$, $\lambda \in \mathbb{R}^m$. Define $d \in \mathbb{R}^p$ as $d = (u, v, w)$, with $u \in \mathbb{R}^n$, $v \in \mathbb{R}^l$, $w \in \mathbb{R}^m$. The directional derivative $F'(z;d) \in \mathbb{R}^p \times \mathbb{R}^p$ of $F(z)$ defined in (2.2) in the direction d is equal to

$$F_i'(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, \qquad i \in N_n, \tag{3.7a}$$

$$F_{i+n}'(z;d) = -D_x h_i(x)\cdot u, \qquad i \in N_l, \tag{3.7b}$$

$$F_{i+n+l}'(z;d) = -D_x g_i(x)\cdot u, \qquad i \in \alpha(z), \tag{3.7c}$$

$$= \min(-D_x g_i(x)\cdot u, w_i), \qquad i \in \beta(z), \tag{3.7d}$$

$$= w_i, \qquad i \in \gamma(z), \tag{3.7e}$$

where the following sets are used:

$$\begin{aligned} N_q &= \{j \in \mathbb{N} \mid 1 \leq j \leq q\}, \\ \alpha(z) &= \{j \in \mathbb{N}_m \mid \lambda_j > -g_j(x)\}, \\ \beta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j = -g_j(x)\}, \\ \gamma(z) &= \{j \in \mathbb{N}_m \mid \lambda_j < -g_j(x)\}, \end{aligned}$$

with $q = n$ or $q = l$. The calculation of most of the terms in (3.7) is straightforward, except (3.7d), which can be computed using a Taylor series expansion of the arguments of $\min(-g_i(x), \lambda_i)$ in the limit of the directional derivative (3.1), combined with the relation $\min(a + b, a + d) - \min(a, a) = \min(b, d)$ and the fact that $i \in \beta(z)$.

The Clarke generalized derivative of $F(z)$ can be computed using the relations (3.2)–(3.3) and is equal to

$$
\begin{aligned}
&\text{(3.8a)} && F_i^0(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, && i \in N_n,\\
&\text{(3.8b)} && F_{i+n}^0(z;d) = -D_x h_i(x)\cdot u, && i \in N_l,\\
&\text{(3.8c)} && F_{i+n+l}^0(z;d) = -D_x g_i(x)\cdot u, && i \in \alpha(z),\\
&\text{(3.8d)} && = \max(-D_x g_i(x)\cdot u, w_i), && i \in \beta(z), F_{i+n+l}(z) > 0,\\
&\text{(3.8e)} && = \min(-D_x g_i(x)\cdot u, w_i), && i \in \beta(z), F_{i+n+l}(z) \le 0,\\
&\text{(3.8f)} && = w_i, && i \in \gamma(z).
\end{aligned}
$$

The calculation of (3.8d) and (3.8e) in $F^0(z;d)$ is nontrivial and is detailed in Appendix A.

Using the results for the directional derivative and the Clarke generalized directional derivative, we can now state a quasi-directional derivative $G : D \times \mathbb{R}^p \to \mathbb{R}^p$, satisfying the conditions (3.5), which for any $\delta > 0$ is equal to

$$
\begin{aligned}
&\text{(3.9a)} && G_i(z;d) = D_x\mathcal{L}_i(z)\cdot u + D_\mu\mathcal{L}_i(z)\cdot v + D_\lambda\mathcal{L}_i(z)\cdot w, && i \in N_n,\\
&\text{(3.9b)} && G_{i+n}(z;d) = -D_x h_i(x)\cdot u, && i \in N_l,\\
&\text{(3.9c)} && G_{i+n+l}(z;d) = -D_x g_i(x)\cdot u, && i \in \alpha_\delta(z),\\
&\text{(3.9d)} && = \max(-D_x g_i(x)\cdot u, w_i), && i \in \beta_\delta(z), F_{i+n+l}(z) > 0,\\
&\text{(3.9e)} && = \min(-D_x g_i(x)\cdot u, w_i), && i \in \beta_\delta(z), F_{i+n+l}(z) \le 0,\\
&\text{(3.9f)} && = w_i, && i \in \gamma_\delta(z),
\end{aligned}
$$

with the sets

$$
\begin{aligned}
\alpha_\delta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j > -g_j(x) + \delta\},\\
\beta_\delta(z) &= \{j \in \mathbb{N}_m \mid -g_j(x) - \delta \le \lambda_j \le -g_j(x) + \delta\},\\
\gamma_\delta(z) &= \{j \in \mathbb{N}_m \mid \lambda_j < -g_j(x) - \delta\}.
\end{aligned}
$$

The main benefit of introducing the δ-dependent sets is that in practice it is hard to test for the set $\beta(z)$, which would generally be ignored in real computations due to rounding errors. One would then miss a number of important components in the quasi-directional derivative, which can significantly affect the performance of the Newton algorithm. The set β_δ gives, however, a computational well-defined quasi-directional derivative $G(z;d)$. In Appendix B, a proof is given that $G(z;d)$ satisfies the conditions stated in (3.5), which is the condition required in [15, Theorem 1], to ensure convergence of the semismooth Newton method.

The formulation of the quasi-directional derivative G (3.9) is, however, not directly useful as a Jacobian in the semismooth Newton method due to the max and min functions. In order to eliminate these functions, we introduce the sets

$$
\begin{aligned}
I_{\beta_\delta}^{11}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) > 0, -D_x g_i(x)\cdot u > w_i\},\\
I_{\beta_\delta}^{12}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) > 0, -D_x g_i(x)\cdot u \le w_i\},\\
I_{\beta_\delta}^{21}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) \le 0, -D_x g_i(x)\cdot u > w_i\},\\
I_{\beta_\delta}^{22}(z,d) &:= \{i \in \beta_\delta(z) \mid F_{i+n+l}(z) \le 0, -D_x g_i(x)\cdot u \le w_i\}
\end{aligned}
$$

and define

$$I^1_\delta(z,d) := \alpha_\delta(z) \cup I^{11}_{\beta_\delta}(z,d) \cup I^{22}_{\beta_\delta}(z,d), \tag{3.10a}$$

$$I^2_\delta(z,d) := \gamma_\delta(z) \cup I^{12}_{\beta_\delta}(z,d) \cup I^{21}_{\beta_\delta}(z,d). \tag{3.10b}$$

The quasi-directional derivative $G(z;d)$ can now be written in a form suitable to serve as a Jacobian in the active set semismooth Newton method defined in Algorithm 3.1 to solve (2.2):

$$G(z;d) = \widehat{G}(z)d,$$

with

$$\widehat{G}(z) = \begin{pmatrix} D_x\mathcal{L}_i(z)|_{i\in N_n} & D_\mu\mathcal{L}_i(z)|_{i\in N_n} & D_\lambda\mathcal{L}_i(z)|_{i\in N_n} \\ -D_x h_i(x)|_{i\in N_l} & 0 & 0 \\ -D_x g_i(x)|_{i\in I^1_\delta(z,d)} & 0 & \delta_{ij}|_{i,j\in I^2_\delta(z,d)} \end{pmatrix} \in \mathbb{R}^{p\times p}, \tag{3.11}$$

with δ_{ij} the Kronecker symbol. By updating the sets $I^1_\delta(z;d)$ and $I^2_\delta(z;d)$ as part of the Newton method, the complementary problem (3.6) is simultaneously solved with the solution of (2.2). In general, after a few iterations the proper sets $I^{1,2}_\delta(z;d)$ will be found and the semismooth Newton method then converges like a regular Newton method. Also, one should note that *only* the contribution $D_x\mathcal{L}_i(z)$ in (3.11) depends on the DG discretization in $\mathcal{L}_i(z)$. Hence, the KKT-Limiter provides a general framework to impose limiters on time-implicit numerical discretizations and could, for instance, also be applied to time-implicit finite volume discretizations.

3.4. Active set semismooth Newton algorithm. As default values we use in Algorithm 3.1 $\bar\alpha = 10^{-12}$, $\beta = \gamma = \frac{1}{2}$, $\sigma = 10^{-9}$, $\delta = 10^{-12}$, and $\epsilon = 10^{-8}$.

An important aspect of Algorithm 3.1 is that we simultaneously solve the mixed linear complementarity equations (3.6) for the search direction d as part of the global Newton method using an active set technique. This was motivated by [16] and will reduce the mixed linear complementarity problem (3.6) into a set of linear equations. The use of the active set technique is also based on the observation in [18] of the close relation between an active set Newton method and a semismooth Newton method. After the proper sets $I^1_\delta(z;d)$, $I^2_\delta(z;d)$ are obtained for the quasi-directional derivative $G(z;d)$, the difference with a Newton method for smooth problems [9] will be rather small. The mixed linear complementarity problem can, however, have one, multiple, or no solutions, and, in order to deal also with cases where the matrix G is poorly conditioned, we will use a minimum norm least squares or Gauss–Newton method to solve the algebraic equations (3.12).

For the performance of a Newton algorithm, proper scaling of the variables is crucial. Here we use the approach outlined in [9] and the Newton method is applied directly to the scaled variables. Also, the matrix $\widehat{G}_k^T\widehat{G}_k + \bar\alpha\|F(z^k)/F(z^0)\|I$ in the Newton method will have a much larger condition number than the matrix $\widehat{G}_k$. In order to improve the conditioning of this matrix, we use simultaneous iterative row and column scaling in the L^∞-matrix norm, as described in [2]. This algorithm very efficiently scales the rows and columns such that an L^∞-matrix norm approximately equal to one is obtained. This gives a many orders of magnitude reduction in the matrix condition number and generally reduces the condition number of the matrix (3.12) to the same order as the condition number of the original matrix $\widehat{G}_k$.

Algorithm 3.1 Active set semismooth Newton method.

(A.0) (*Initialization*) Let $\bar{\alpha} \geq 0$, $\beta, \gamma \in (0,1)$, $\sigma \in (0, \bar{\sigma})$, $\delta > 0$, and $b > C \in \mathbb{R}^+$ arbitrarily large, but bounded. Choose $z^0, d^0 \in \mathbb{R}^p$ and tolerance ϵ.
(A.1) Scale z^0.
(A.2) (*Newton method*)
for $k = 0, 1, \ldots$ until $\|F(z^k)\| \leq \epsilon$ **and** $\|d^k\| \leq \epsilon$ **do**
    Compute the quasi-directional derivative matrix $\widehat{G}_k := \widehat{G}(z^k)$ given by (3.11) and the active sets $I^1_\delta(z;d)$, $I^2_\delta(z;d)$ of $\widehat{G}_k$ given by (3.10).
    Apply row-column scaling to $(\widehat{G}_k^T \widehat{G}_k + \bar{\alpha}\|F(z^k)/F(z^0)\|I)$, with I the identity matrix, such that the matrix has a norm $\|\cdot\|_{L^\infty} \cong 1$.
    if there exists a solution h^k to

$$(\widehat{G}_k^T \widehat{G}_k + \bar{\alpha}\|F(z^k)/F(z^0)\|I)h^k = -\widehat{G}_k^T F(z^k), \tag{3.12}$$

    with $|h^k| \leq b|F(z^k)|$ **and**

$$|F(z^k + h^k)| < \gamma |F(z^k)|,$$

    then
        Set $d^k = h^k$, $z^{k+1} = z^k + d^k$, $\alpha_k = 1$, and $m_k = 0$.
    else
        Choose $d^k = h^k$.
        Compute $\alpha_k = \beta^{m_k}$, where m_k is the first positive integer m for which

$$\theta(z^k + \beta^{m_k} d^k) - \theta(z^k) \leq -\sigma \beta^m \theta(z^k).$$

        Set $z^{k+1} = z^k + \alpha_k d^k$.
    end if
end for

4. KKT-Limiter DG discretization. Given a domain $\Omega \subseteq \mathbb{R}^d$, $d = \dim(\Omega)$, $d = 1, 2$, with Lipschitz continuous boundary $\partial\Omega$. As a general model problem we consider the following second order nonlinear scalar equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) + G(u) - \nabla \cdot (\nu(u)\nabla u) = 0, \tag{4.1}$$

with $u(x,t) : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}$ a scalar quantity, $F(u) : \mathbb{R} \to \mathbb{R}^d$ the flux, $G(u) : \mathbb{R} \to \mathbb{R}$ a reaction term, and $\nu(u) : \mathbb{R} \to \mathbb{R}^+$ a nonlinear diffusion term. By selecting different functions F, G, and ν in (4.1) we will demonstrate in section 5 the KKT-Limiter on various model problems that impose different positivity constraints on the solution.

For the DG discretization, we introduce the auxiliary variable $Q \in \mathbb{R}^d$ and rewrite (4.1) as a first order system of conservation laws

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) + G(u) - \nabla \cdot (\nu(u)Q) = 0, \tag{4.2a}$$

$$Q - \nabla u = 0. \tag{4.2b}$$

4.1. DG discretization. Let $\mathcal{T}_h$ be a tessellation of the domain Ω with shape regular line or quadrilateral elements K with maximum diameter $h > 0$. The total number of elements in $\mathcal{T}_h$ is N_K. We denote the union of the set of all boundary faces ∂K, $K \in \mathcal{T}_h$, as $\mathcal{F}_h$, denote all internal faces $\mathcal{F}_h^i$ and the boundary faces as $\mathcal{F}_h^b$, and hence get $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^b$. The elements connected to each side of a face

$S \in \mathcal{F}_h$ are denoted by the indices L and R, respectively. For the KKT-Limiter, it is important to use orthogonal basis functions; see section 4.2. In this paper, $\mathcal{P}_p(K)$ represent tensor product Legendre polynomials of degree p on d-dimensional rectangular elements $K \in \mathcal{T}_h$, when K is mapped to the reference element $(-1,1)^d$. For general elements, one can use Jacobi polynomials with proper weights to obtain an orthogonal basis; see [19, section 3.2]. Next, we define the finite element spaces

$$V_h^p := \Big\{ v \in L^2(\Omega) \mid v|_K \in \mathcal{P}_p(K) \ \forall K \in \mathcal{T}_h \Big\},$$
$$W_h^p := \Big\{ v \in (L^2(\Omega))^d \mid v|_K \in (\mathcal{P}_p(K))^d \ \forall K \in \mathcal{T}_h \Big\},$$

with $L^2(\Omega)$ the Sobolev space of square integrable functions. Equation (4.2) is discretized using the local discontinuous Galerkin discretization from [7]. Define $L_h^1 : V_h^p \times W_h^p \times V_h^p \to \mathbb{R}$ and $L_h^2 : V_h^p \times W_h^p \to \mathbb{R}$ as

$$\begin{aligned} L_h^1(u_h, Q_h; v) :=& -\big(F(u_h) - \nu(u_h)Q_h, \nabla_h v\big)_\Omega + \big(G(u_h), v\big)_\Omega \\ &+ \sum_{S \in \mathcal{F}_h^i} \big(H(u_h^L, u_h^R; n^L) - \widehat{\nu(u_h)} n^L \cdot \widehat{Q_h}, v^L - v^R\big)_S \\ &+ \sum_{S \in \mathcal{F}_h^b} \big(H(u_h^L, u_h^b; n^L) - \widehat{\nu(u_h)} n^L \cdot Q_h^b, v^L\big)_S, \end{aligned} \tag{4.3}$$

$$\begin{aligned} L_h^2(u_h; w) :=& \big(u_h, \nabla_h \cdot w\big)_\Omega - \sum_{S \in \mathcal{F}_h^i} \big(\widehat{u_h} n^L, w^L - w^R\big)_S \\ &- \sum_{S \in \mathcal{F}_h^b} \big(u_h^b n^L, w^L\big)_S, \end{aligned}$$

where $(\cdot,\cdot)_D$ is the $L^2(D)$ inner product, ∇_h is the elementwise ∇ operator, and the superscript b refers to boundary data. Here $n^L \in \mathbb{R}^d$ is the exterior unit normal vector at the boundary of the element $L \in \mathcal{T}_h$ that is connected to face S. The numerical flux H is the Lax–Friedrichs flux

$$H(u_h^L, u_h^R; n) = \frac{1}{2}\big(n \cdot (F(u_h^L) + F(u_h^R)) - C_{LF}(u_h^R - u_h^L)\big),$$

with Lax–Friedrichs coefficient $C_{LF} = \sup_{u_h \in [u_h^L, u_h^R]} |\frac{\partial}{\partial u_h}(n \cdot F(u_h))|$. For $\widehat{Q_h}$ and $\widehat{u_h}$, we use the alternating fluxes

$$\widehat{Q_h} = (1-\alpha)Q_h^L + \alpha Q_h^R, \tag{4.4a}$$
$$\widehat{u_h} = \alpha u_h^L + (1-\alpha)u_h^R, \tag{4.4b}$$

with $0 \le \alpha \le 1$. The numerical flux for the nonlinear diffusion is defined as

$$\widehat{\nu(u_h)} = \frac{1}{2}(\nu(u_h^L) + \nu(u_h^R)).$$

For $t \in (0, T]$, the semidiscrete DG formulation for (4.2) now can be expressed as follows: Find $u_h(t) \in V_h^p$, $Q_h(t) \in W_h^p$, such that for all $v \in V_h^p$, $w \in W_h^p$,

$$\left(\frac{\partial u_h}{\partial t}, v\right)_\Omega + L_h^1(u_h, Q_h; v) = 0, \tag{4.5a}$$
$$(Q_h, w)_\Omega + L_h^2(u_h; w) = 0. \tag{4.5b}$$

These equations are discretized in time with a DIRK method [14]. The main benefit of the DIRK method is that the RK stages can be computed successively, which significantly reduces the computational cost and memory overhead.

We represent u_h and Q_h in each element $K \in \mathcal{T}_h$, respectively, as $u_h|_K = \sum_{j=1}^{N_u} \widehat{U}_j^K \phi_j^K$ and $Q_h|_K = \sum_{j=1}^{N_Q} \widehat{Q}_j^K \psi_j^K$, with basis functions $\phi_j^K \in \mathcal{P}_p(K)$, $\psi_j^K \in \big(\mathcal{P}_p(K)\big)^d$ and DG coefficients $\widehat{U}_j^K \in \mathbb{R}$, $\widehat{Q}_j^K \in \mathbb{R}^d$. After replacing the test functions $v \in V_h^p$ in (4.5a) and $w \in W_h^p$ (4.5b) with, respectively, the independent basis functions $\phi_i^K \in \mathcal{P}_p(K)$, $i = 1, \dots, N_u$, and $\psi_i^K \in \big(\mathcal{P}_p(K)\big)^d$, $i = 1, \dots, N_Q$, we obtain the algebraic equations for the DG discretization.

In order to simplify notation, we introduce $\widehat{L}_h^1(\widehat{U}, \widehat{Q}) = L_h^1(u_h, Q_h; \phi) \in \mathbb{R}^{N_u N_K}$ and $\widehat{L}_h^2(\widehat{U}) = L_h^2(u_h; \psi) \in \mathbb{R}^{dN_Q N_K}$, with N_K the number of elements in $\mathcal{T}_h$ and $\phi = \phi_i^K$, $\psi = \psi_i^K$ the basis functions in element K. The algebraic equations for the DIRK stage vector $\widehat{K}^{(i)} \in \mathbb{R}^{N_u N_K}$, $i = 1, \dots, s$, with the DG coefficients can then be expressed as

$$\widehat{L}_h(\widehat{K}^{(i)}) := M_1\big(\widehat{K}^{(i)} - \widehat{U}^n\big) + \triangle t \sum_{j=1}^{i} a_{ij} \widehat{L}_h^1\big(\widehat{K}^{(j)}, -M_2^{-1} \widehat{L}_h^2(\widehat{K}^{(j)})\big) = 0. \tag{4.6}$$

Here we eliminated the DG coefficients for the auxiliary variable Q_h using (4.5b). The matrices $M_1 \in \mathbb{R}^{N_u N_K \times N_u N_K}$, $M_2 \in \mathbb{R}^{dN_Q N_K \times dN_Q N_K}$ are block-diagonal mass matrices since we use orthogonal basis functions and n denotes the index of time level $t = t_n$.

The coefficients a_{ij} are the coefficients in the Butcher tableau, which determine the properties of the RK method [14]. For DIRK methods, $a_{ij} = 0$ if $j > i$. The following DIRK methods are used: for basis functions with polynomial order $p = 1$ [1, page 1012, Theorem 5, first method with $\alpha = 1 - \frac{1}{2}$]; $p = 2$ [32, page 2117 (top)]; $p = 3$ [1, page 1012, Theorem 5, second method]; see also [32, page 2117 (top)]. The order of accuracy of these DIRK methods is $p + 1$, and their coefficients in the Butcher tableau satisfy $a_{sj} = b_j$, $j = 1, \dots, s$, which implies that these methods are stiffly accurate (see [14, section IV.6]), and the solution of the last DIRK stage is equal to the solution at the new time step

$$\widehat{U}^{n+1} = \widehat{K}^{(s)}.$$

Since each DIRK stage vector must satisfy the positivity constraints, this then also immediately applies to the solution at time t_{n+1}.

The Jacobian $D_x \mathcal{L}(\widehat{K}^{(i)}) \in \mathbb{R}^{N_u N_K \times N_u N_K}$, with $x = \widehat{K}^{(i)}$, in the quasi-directional derivative G (3.11) of DIRK stage i of the unlimited DIRK-DG discretization (4.6) is now equal to

$$D_x \mathcal{L}(\widehat{K}^{(i)}) = M_1 + \triangle t a_{ii} \left(\frac{\partial L_h^1}{\partial \widehat{K}^{(i)}} - \frac{\partial L_h^1}{\partial \widehat{Q}^{(i)}} M_2^{-1} \frac{\partial L_h^2}{\partial \widehat{K}^{(i)}} \right).$$

4.2. Limiter constraints. The limiter constraints for the DG discretization can be imposed directly by defining the inequality constraints in the KKT-equations. In each element $K \in \mathcal{T}_h$, we apply for each DIRK-stage $i = 1, \dots, s$ the following inequality constraints:

(i) *Positivity constraint*:

$$g_{1,k}^K(\widehat{K}^{K,(i)}) = u_{\min} - \sum_{q=1}^{N_u} \widehat{K}_q^{K,(i)} \phi_q^K(x_k), \quad k = 1, \dots, N_p. \tag{4.7}$$

(ii) *Maximum constraint*:

$$g_{2,k}^K(\widehat{K}^{K,(i)}) = \sum_{q=1}^{N_u} \widehat{K}_q^{K,(i)} \phi_q^K(x_k) - u_{\max}, \quad k = 1, \ldots, N_p. \tag{4.8}$$

Here the superscript K refers to element $K \in \mathcal{T}_h$, and (i) is the ith DIRK stage. The points x_k, $k = 1, \ldots, N_p$, are the points in element K where the inequality constraints are imposed and $u_{\min}$ and $u_{\max}$ denote, respectively, the allowed minimum and maximum values of u. The inequality constraints are imposed using the Lagrange multiplier λ; see (2.1c).

(iii) *Conservation constraint*:

Since the basis functions ϕ_j^K, $j = 1, \ldots, N_u$, are orthogonal in each element K, we have $(1, \phi_j^K)_K = 0$ for $j = 2, \ldots, N_u$. Hence, at each RK stage i, limiting the DG coefficients $\widehat{K}_j^{K,(i)}$, with $j = 2, \ldots, N_u$, has no effect on the element average $\bar{u}_h^{K,(i)} = \frac{1}{|K|}(u_h^{(i)}, 1)_K = \widehat{K}_1^{K,(i)}$, with $u_h^{(i)}$ the solution at stage i, and therefore does not influence the conservation properties of the DG discretization.

Limiting the DG coefficients $\widehat{K}_1^{K,(i)}$ can, however, affect the conservation properties of the DG discretization since $\bar{u}_h^{K,(i)} = \widehat{K}_1^{K,(i)}$. In order to ensure local conservation, we therefore need to impose in each element the local conservation constraint

$$\begin{aligned} h^K\big(\widehat{K}^{K,(i)}\big) &= \widehat{L}_{h,1}^K(\widehat{K}^{(i)}) \\ &= |K|\big(\widehat{K}_1^{K,(i)} - \widehat{U}_1^n\big) + (G(u_h^{(i)}), \phi_1^K)_K \\ &+ \sum_{S \in \mathcal{F}_h^i \cap \partial K} \big(H(u_h^{L,(i)}, u_h^{R,(i)}; n^L) \\ &- \widehat{\nu(u_h)} n^L \cdot ((1-\alpha) Q_h^{L,(i)} + \alpha Q_h^{R,(i)}), \phi_1^L - \phi_1^R\big)_S \\ &+ \sum_{S \in \mathcal{F}_h^b \cap \partial K} \big(H(u_h^{L,(i)}, u_h^b; n^L) - \widehat{\nu(u_h)} n^L \cdot Q_h^b, \phi_1^L\big)_S, \end{aligned} \tag{4.9}$$

with $\widehat{L}_{h,1}^K$ the equation for the element mean in element K in (4.6). The conservation constraint (4.9) is imposed using the Lagrange multiplier μ; see (2.1b). The conservation constraint explicitly ensures that at each RK stage the equation for the element mean $\bar{u}_h^{K,(i)}$ is exactly preserved in each element, and hence the KKT-Limiter does not affect the conservation properties of the DG discretization.

The remaining Jacobians $D_x h_i(x) \in \mathbb{R}^{N_K \times N_u N_K}$, $D_x g_i(x) \in \mathbb{R}^{N_p N_K \times N_u N_K}$ and $D_\mu \mathcal{L}_i(z) \in \mathbb{R}^{N_u N_K \times N_K}$, $D_\lambda \mathcal{L}_i(z) \in \mathbb{R}^{N_u N_K \times N_p N_K}$, with $x = \widehat{K}^{(i)}$, in the quasi-directional derivative matrix $\widehat{G}$ (3.11) are now straightforward to calculate.

It is important to ensure that the initial solution also satisfies the positivity constraints. An L^2-projection of the solution will in general not satisfy these constraints for a nonsmooth solution. To ensure that the initial solution also satisfies the positivity constraints, we apply a constrained projection using the active set semismooth Newton method given by Algorithm 3.1. The only difference is now that instead of

(4.6) we use L^2-projection

$$\widehat{L}_{hi}(\widehat{U}^0) = M^1\widehat{U}^0 - (u_0, \phi_i)_\Omega$$

and combine this with the positivity constraints (4.7)–(4.8). Here u_0 denotes the initial solution. As the initial solution for the constrained projection we use in Algorithm 3.1 the standard L^2-projection without constraints.

The positivity constraints are imposed at all element quadrature points since only the solution at these quadrature points is used in the DG discretization. In one dimension we use Gauss–Lobatto quadrature rules and in two dimensions product Gauss–Legendre quadrature rules. Since the number of quadrature points in an element is generally larger than the number of degrees of freedom in an element, this will result in an overdetermined set of algebraic equations and a rank deficit Jacobian matrix if the number of active constraints in an element is larger than the degrees of freedom N_u in the element. In order to obtain in Algorithm 3.1 accurate search directions h^k, we use the Gauss–Newton method given by (3.12). This approach can efficiently deal with the possible rank deficiency of the Jacobian matrix.

In practice, it will not be necessary to apply the inequality constraints in all elements, and one can significantly reduce the computational cost and memory overhead by excluding those elements for which it is obvious that they will meet the constraints anyway.

5. Numerical experiments. In this section, we will discuss a number of numerical experiments to demonstrate the performance of the DIRK-DG scheme with the positivity preserving KKT-Limiter. All computations were performed using the default values for the coefficients listed for Algorithm 3.1, except that for the accuracy tests discussed in section 5.1 we use $\epsilon = 10^{-10}$. The upwind coefficient α in (4.4) is set to $\alpha = 1$. In all 1D computations, the local conservation constraint is imposed and satisfied with an error less than 10^{-12}.

5.1. Accuracy tests. It is important to investigate whether the KKT-Limiter negatively affects the accuracy of the DG discretization in case the exact solution is smooth, but where also a positivity preserving limiter is required to ensure that the numerical solution stays within the bounds. To investigate this, we conduct the same accuracy tests as conducted in Qin and Shu [28, section 5.1]. Both the linear advection and the inviscid Burgers' equation are considered, which are obtained by setting $F(u) = u$ and $F(u) = \frac{1}{2}u^2$, respectively, and $G(u) = \nu(u) = 0$ in (4.1).

Example 5.1 (steady state solution to the linear advection equation). We consider

$$u_t + u_x = \sin^4 x, \qquad u(x,0) = \sin^2 x, \quad u(0,t) = 0, \tag{5.1}$$

with an outflow boundary condition at $x = 2\pi$. The exact solution $u(x,t)$ is positive for all $t > 0$; see [28]. As the steady state solution we use the solution at $t = 500$, when all residuals are approximately 10^{-16}. During the computations, the CFL number is dynamically adjusted between 10 and 89. For the time integration, an implicit Euler method is used. In Tables 1 and 2, the results of the accuracy tests, without and with the KKT-Limiter, are shown. The results in Table 2 show that the KKT-Limiter does not negatively affect the accuracy. For all test cases, the optimal accuracy in the L^2- and L^∞-norms is obtained. Also, the limiter is necessary, as can be seen from Table 1, and preserves the imposed positivity bound $u_{h\,\min} = 10^{-14}$ for the numerical solution.

TABLE 1
Error table for steady state linear advection equation (5.1) *without limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	1.461068e-02	-	2.044253e-02		–5.169578e-03
	40	3.702581e-03	1.98	5.287628e-03	1.95	–2.883487e-04
1	80	9.288342e-04	2.00	1.331962e-03	1.99	–1.208793e-05
	160	2.324090e-04	2.00	3.336614e-04	2.00	–4.036603e-07
	320	5.811478e-05	2.00	8.345620e-05	2.00	–1.282064e-08
	20	9.287703e-04	-	1.776878e-03	-	–4.952018e-05
	40	1.177042e-04	2.98	2.489488e-04	2.84	–1.627459e-06
2	80	1.476405e-05	3.00	3.200035e-05	2.96	–5.149990e-08
	160	1.847107e-06	3.00	4.027944e-06	2.99	–1.614420e-09
	320	2.309385e-07	3.00	5.043677e-07	3.00	–5.049013e-11
	20	5.653820e-05	-	1.230308e-04	-	–3.877467e-05
	40	3.583918e-06	3.98	7.803741e-06	3.98	–1.326415e-06
3	80	2.247890e-07	3.99	4.950122e-07	3.98	–4.237972e-08
	160	1.406175e-08	4.00	3.090593e-08	4.00	–1.331692e-09
	320	8.790539e-10	4.00	1.935324e-09	4.00	–4.167274e-11

TABLE 2
Error table for steady state linear advection equation (5.1) *with limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	1.464990e-02	-	2.044253e-02	-	9.998946e-15
	40	3.702367e-03	1.98	5.287628e-03	1.95	9.999813e-15
1	80	9.288338e-04	2.00	1.331962e-03	1.99	1.000000e-14
	160	2.324090e-04	2.00	3.336614e-04	2.00	1.000000e-14
	320	5.811478e-05	2.00	8.345620e-05	2.00	1.000000e-14
	20	9.290268e-04	-	1.776878e-03	-	1.000000e-14
	40	1.177053e-04	2.98	2.489488e-04	2.84	1.000000e-14
2	80	1.476406e-05	3.00	3.200035e-05	2.96	1.000000e-14
	160	1.847107e-06	3.00	4.027944e-06	2.99	1.000000e-14
	320	2.309385e-07	3.00	5.043677e-07	3.00	1.000000e-14
	20	5.742649e-05	-	1.230309e-04	-	9.999990e-15
	40	3.592170e-06	4.00	7.803745e-06	3.98	1.000000e-14
3	80	2.248562e-07	4.00	4.950122e-07	3.98	1.000000e-14
	160	1.406228e-08	4.00	3.090593e-08	4.00	1.000000e-14
	320	8.790580e-10	4.00	1.935323e-09	4.00	1.000000e-14

Example 5.2 (steady state solution to the inviscid Burgers' equation). We consider the inviscid Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = \sin^3\left(\frac{x}{4}\right), \qquad u(x,0) = \sin^2\left(\frac{x}{4}\right), \quad u(0,t) = 0, \tag{5.2}$$

with an outflow boundary condition at $x = 2\pi$. The exact solution $u(x,t)$ is positive for all $t > 0$; see [28]. As the steady state solution we use the solution at $t = 20.000$, when all residuals are approximately 10^{-16}. During the computations, the CFL number is dynamically adjusted between 10 and 954. For the time integration, an implicit Euler method is used. In Tables 3 and 4, the results of the accuracy tests, without and with the KKT-Limiter, show that the KKT-Limiter does not negatively

affect the accuracy. For all test cases, optimal accuracy in the L^2- and L^∞-norms is obtained. Also, the limiter is necessary and preserves the imposed positivity bound $u_{h\,\min} = 10^{-14}$ for the numerical solution.

TABLE 3
Error table for the steady state inviscid Burgers' equation (5.2) *without limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	2.110016e-03	-	3.387013e-03	-	–2.347303e-03
	40	5.230241e-04	2.01	8.577912e-04	1.98	–5.865522e-04
1	80	1.297377e-04	2.01	2.151386e-04	2.00	–1.466204e-04
	20	2.122765e-05	-	3.024868e-05	-	–1.048636e-05
	40	2.623666e-06	3.02	3.731754e-06	3.02	–6.681764e-07
2	80	3.266401e-07	3.01	4.634046e-07	3.01	–4.196975e-08
	20	2.985321e-07	-	1.895437e-06	-	1.895437e-06
	40	1.452601e-08	4.36	1.196963e-07	3.99	1.196963e-07
3	80	7.368455e-10	4.30	7.500564e-09	4.00	7.500564e-09
	160	3.948207e-11	4.22	4.346084e-10	4.11	4.346084e-10

TABLE 4
Error table for steady state inviscid Burgers' equation (5.2) *with limiter.*

p	N	L^2 error	Order	L^∞ error	Order	$\min u_h$
	20	2.208009e-03	-	3.637762e-03	-	9.999813e-15
	40	5.358952e-04	2.04	9.282398e-04	1.97	1.000003e-14
1	80	1.313948e-04	2.03	2.339566e-04	1.99	1.000003e-14
	20	2.116746e-05	-	3.024864e-05	-	1.000003e-14
	40	2.622584e-06	3.01	3.731752e-06	3.02	1.000139e-14
2	80	3.266221e-07	3.01	4.634046e-07	3.01	1.000040e-14
	20	2.985321e-07	-	1.895437e-06	-	1.895437e-06
	40	1.452601e-08	4.36	1.196963e-07	3.99	1.196963e-07
3	80	5.610147e-10	4.70	1.574760e-09	6.25	1.000105e-14
	160	3.232240e-11	4.11	9.038604e-11	4.12	1.000017e-14

5.2. Time-dependent tests. In this section, we will present results of simulations of the linear advection, Allen–Cahn, Barenblatt, and Buckley–Leverett equations. The order of accuracy of the DIRK time integration method is always $p+1$, with p the polynomial order of the spatial discretization. The minimum value of the residual $F(z)$ and Newton update d in Algorithm 3.1 to stop the Newton iterations is $\epsilon = 10^{-8}$ for each DIRK stage. This is a quite strong stopping criterion, and in practice the values are often smaller at the end of each DIRK stage. It is also important to make sure that the Newton stopping criterion is in balance with the accuracy required for the constraints. If the algebraic equations are not solved sufficiently accurate, then it is not likely that the KKT-constraints will be satisfied.

The time step for the DIRK method is dynamically computed, based on the CFL or diffusion number. If the Newton method does not converge within a predefined number of iterations, then the computation for the time step will be restarted with $\triangle t/2$. This is generally more efficient than conducting many Newton iterations. In the next time step, the time step will then be increased to $1.2\triangle t$, until the maximum

CFL number is obtained. In practice, depending on the severity of the nonlinearity, the time step will be constantly adjusted during the computations.

Example 5.3 (1D linear advection equation). We consider (5.1) with a zero right-hand side in the domain $\Omega = [0, 10]$ and periodic boundary conditions. The exact solution is

$$u(x,t) = \max(\cos(2\pi(x-t)/10), 0) \quad \text{for } x \in \Omega,\, t \in [0,T].$$

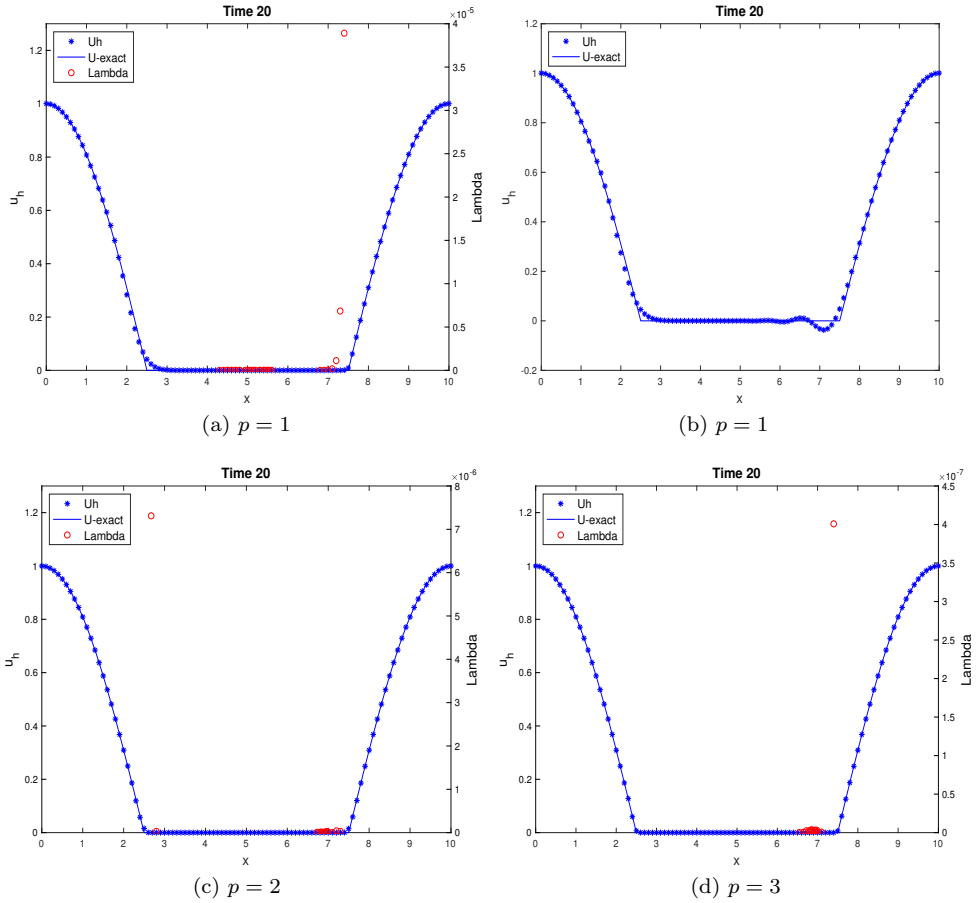
A constrained projection of $u(x,0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x,0)$. The computational mesh contains 100 elements, and the maximum CFL number is 1. In Figures 1a, 1c, and 1d, the exact and numerical solutions at time $t = 20$ are plotted for, respectively, polynomial orders 1, 2, and 3. At this time the wave has traveled twice through the domain and the numerical solution matches very well with the exact solution. Also plotted is the value of the Lagrange multipliers used to impose the positivity constraint $u_{h\,\min} = 10^{-10}$. These plots clearly show that the limiter is only active at locations where the constraint must be imposed and not in the smooth part of the solution. In Figure 1b, the solution for polynomial order $p = 1$ without the KKT-Limiter is plotted, which clearly shows that without the limiter the solution is significantly below the $u = 0$ minimum of the exact solution $u(x,t)$.

Example 5.4 (2D linear advection equation). The KKT-Limiter is also tested on a 2D linear advection equation, which is obtained by setting $F(u) = cu$, with $c = (-1,-2)$, and $G(u) = \nu(u) = 0$ in (4.1). The domain $\Omega = [0,3]^2$ with periodic boundary conditions is used in the computations. The computational mesh contains 30×30 elements. The exact solution is

$$u(x,t) = \max(\cos(2\pi(x+t)/3)\cos(2\pi(y+2t)/3), 0) \quad \text{for } x \in \Omega, t \in [0,T].$$

A constrained projection of $u(x,0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x,0)$. The maximum CFL number is 1. In Figure 2a the numerical solution is shown at $t = 6.3428$ and in Figure 2b the values of the Lagrange multipliers used to enforce the positivity constraint $u_{h\,\min} = 10^{-10}$. Comparing Figures 2a and 2b clearly shows that the KKT-Limiter is only active in those parts of the domain where the solution needs to satisfy the positivity constraint and not in the smooth part.

Example 5.5 (1D Burgers' equation). In order to test the KKT-Limiter on problems with time-dependent shocks, we consider the 1D Burgers' equation on a domain $\Omega = [-1,1]$ with initial condition $u_0 = \max(\cos(\pi x), 0)$ and periodic boundary conditions. The polynomial order is $p = 3$. As lower and upper bounds in the positivity preserving limiter we use, respectively, $u_{h\,\min} = 10^{-10}$ and $u_{h\,\max} = 1$, and no monotonicity constraint is imposed. The initially smooth part of the solution develops into a shock. The onset of the shock is shown in Figure 3a and the later stages of the shock at $t = 0.65$ in Figure 3b. Figure 3c shows the solution when the conservation constraint (4.9) is not explicitly enforced. The difference in the shock solution for the discretizations with and without the explicitly imposed conservation constraint is very small. The main reason for this is that the KKT-Limiter is only active in regions where the constraints must be imposed and does not affect the discretization at other places in the domain. This can be seen from the values of the Lagrange multipliers that are used to impose the positivity constraints, which are indicated with red



(a) $p = 1$ (b) $p = 1$

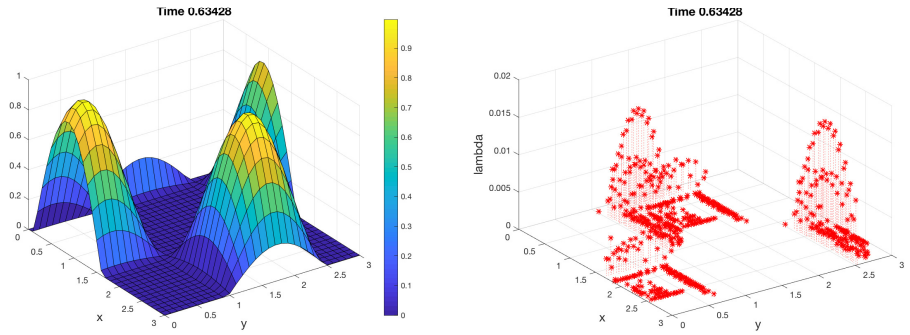
(c) $p = 2$ (d) $p = 3$

FIG. 1. *Example* 5.3, 1*D advection equation:* (a), (c), (d) *numerical solution* u_h *with positivity preserving limiter, polynomial order, respectively,* $p = 1$, 2, *and* 3*;* (b) *numerical solution* u_h *without positivity preserving limiter, polynomial order* $p = 1$. *Computational mesh* 100 *elements. Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (a), (c), *and* (d) *with a red (open) circle.*

circles, and are only nonzero in the vicinity of the shock and at locations where the solution has a discontinuous derivative. The KKT-Limiter to ensure the positivity constraints therefore has a very small effect on the conservation properties of the DG discretization, as can be seen by comparing Figures 3b and 3c.

Example 5.6 (Allen–Cahn equation). The Allen–Cahn equation is a reaction-diffusion equation that describes phase transition. The Allen–Cahn equation is obtained by setting $G(u) = u^3 - u$, $\nu(u) = \bar{\nu}$, and $F(u) = 0$ in (4.1). The solution of the Allen–Cahn equation should stay within the range $[0, 1]$. Hence, we apply both the positivity and the maximum preserving limiters, respectively, (4.7)–(4.8) with bounds $u_{h\min} = 10^{-14}$ and $u_{h\max} = 1 - 10^{-10}$. A constrained projection of $u(x, 0)$ onto the finite element space V_h^p is used as the initial solution $u_h(x, 0)$.

Example 5.6a (1D Allen–Cahn equation). As the test case we use the traveling



(a) (b)

FIG. 2. *Example* 5.4, 2*D advection equation:* (a) *solution* u_h, (b) *Lagrange multiplier. Computational mesh* 30×30 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

wave solution

$$u(x,t) = \frac{1}{2}\left(1 - \tanh\left(\frac{x - st}{2\sqrt{2\bar{\nu}}}\right)\right),$$

with wave velocity $s = 3\sqrt{\bar{\nu}/2}$. The computational domain is $\Omega = [-\frac{1}{2}, 2]$. If the mesh resolution is sufficiently dense such that the jump in the traveling wave solution is well resolved, then no limiter is required. For small values of the viscosity, the solution will, however, violate the positivity constraints, except on very fine meshes. In Figures 4a and 4b, respectively, the numerical solution u_h and its derivative Q_h and the exact solutions are shown for the viscosity $\bar{\nu} = 10^{-5}$ on a mesh with 100 elements and polynomial order 3 for the basis functions. The values of the Lagrange multiplier used to impose the positivity constraints are also shown in Figure 4a. The solution has a very thin and steep transition region, but the wave speed is still correctly computed by the LDG scheme and the KKT limiter ensures that both the positivity and the maximum constraints are satisfied.

Example 5.6b (2D Allen–Cahn equation). For the 2D test case, the computational domain is $\Omega = [-\frac{1}{2}, 2]^2$ and the computational mesh contains 30×30 elements. The viscosity coefficient is selected as $\bar{\nu} = 10^{-4}$. As the test case we use the initial solution

$$u(x,0) = \frac{1}{4}\left(1 - \tanh\left(\frac{x}{2\sqrt{2\bar{\nu}}}\right)\right)\left(1 - \tanh\left(\frac{y}{2\sqrt{2\bar{\nu}}}\right)\right),$$

whose values are also used as boundary conditions for $t > 0$. At this mesh resolution a positivity preserving limiter is necessary. The numerical solution shown in Figure 5a has steep gradients, and the positivity preserving limiter ensures that the bounds are satisfied. The locations where the limiter are active can be seen in Figure 5b, which shows the values and locations of the Lagrange multipliers used to impose the bounds in the DG discretization.

Example 5.7 (Barenblatt equation). The Barenblatt equation, which models a porous medium, is obtained by setting $\nu(u) = mu^{m-1}$, $m > 1$, and $F(u) = 0$,

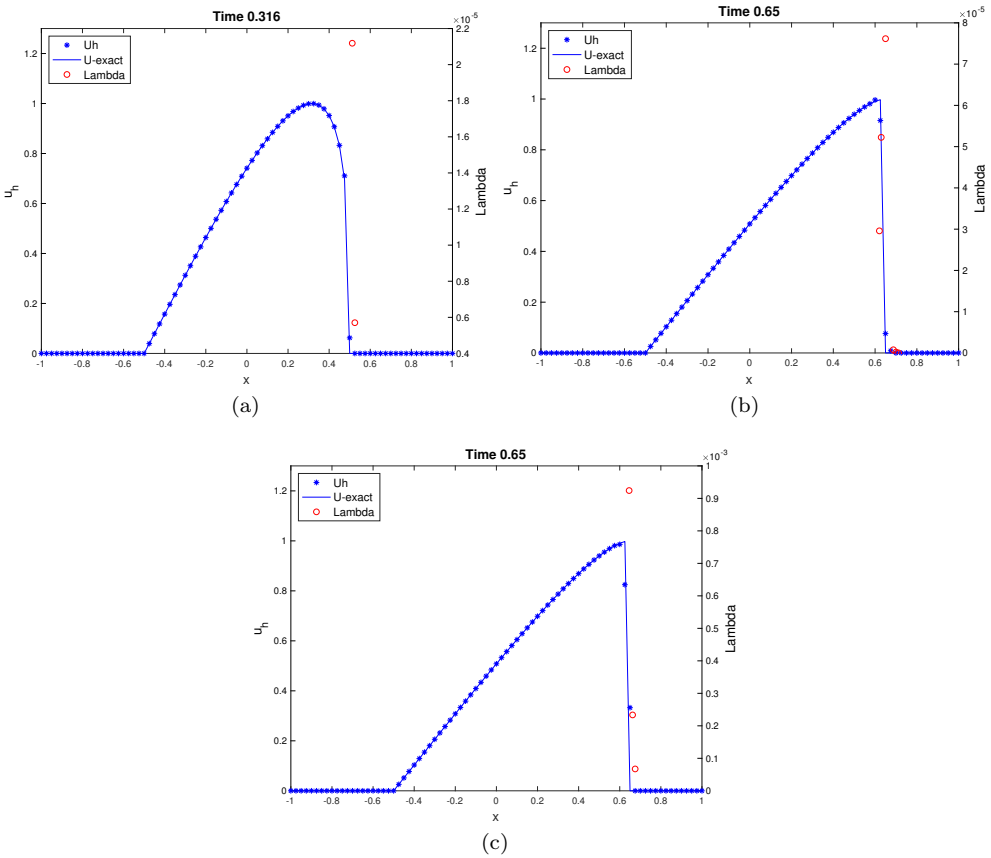


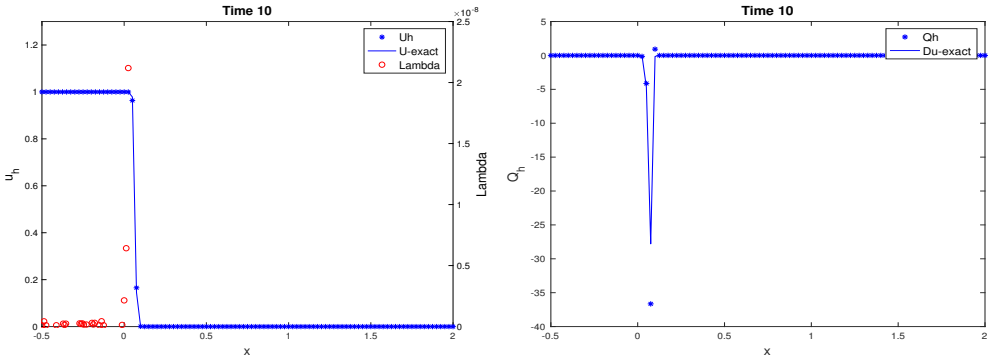
FIG. 3. *Example* 5.5, *1D Burgers' equation:* (a)–(c) *solution* u_h *and Lagrange multiplier. The solution in* (a) *and* (b) *is computed with local conservation imposed as an explicit constraint, whereas* (c) *shows the solution without explicitly imposing local conservation. Computational mesh* 80 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated with a red (open) circle.*

$G(u) = 0$ in (4.1). The exact solution is

$$u(t,x) = t^{\alpha}\left(\left(C - \frac{\beta(m-1)}{2m}\frac{|x|^2}{t^{2\beta}}\right)_+\right)^{\frac{1}{m-1}},$$

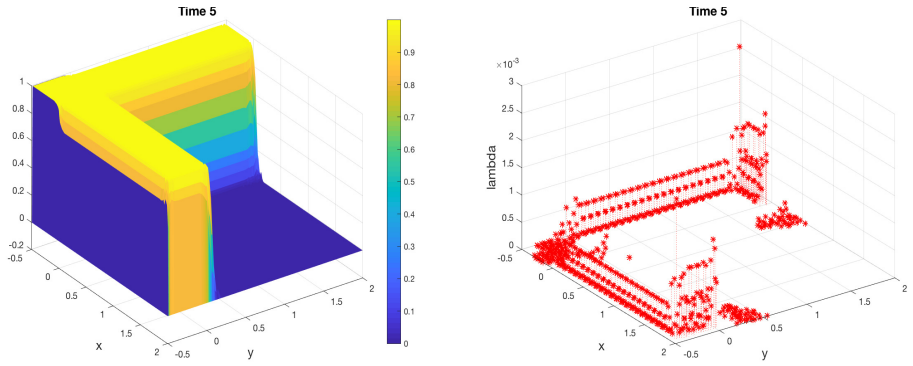
with $\alpha = \frac{n}{n(m-1)+2}$, $\beta = \frac{\alpha}{n}$, $n = \dim(\Omega)$, $(x)_+ = \max(x, 0)$, and $C > 0$. We selected $C = 1$ and $m = 8$. The solution should be positive or zero for $t > 0$. The initial solution for the computations is the constrained projection of $u(x, 1)$ onto the finite element space V_h^p. In the computations, Dirichlet boundary conditions are imposed, where the solution for $t > 0$ is fixed at the same level as the initial solution.

Example 5.7a (1D Barenblatt equation). We first consider the 1D Barenblatt equation on the domain $\Omega = [-7, 7]$ using a computational mesh of 100 elements. In Figure 6, the numerical solution without the use of a limiter is shown. It is clear that near the boundary of $u(t, x) > 0$, where the derivative of u becomes unbounded, significant negative values of u_h are obtained. These cause severe numerical problems



(a) u_h - with limiter (b) Q_h - with limiter

FIG. 4. *Example* 5.6a, *1D Allen–Cahn equation:* (a) *numerical solution* u_h *and exact solution* u, (b) *derivative of numerical solution* Q_h *and exact derivative* Du. *Computational mesh* 100 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated in* (a) *with a red (open) circle.*



(a) (b)

FIG. 5. *Example* 5.6b, *2D Allen–Cahn equation:* (a) *numerical solution* u_h *and* (b) *Lagrange multiplier. Computational mesh* 30×30 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

and do not allow the continuation of the computations.

Example 5.7b (2D Barenblatt equation). In Figures 7a and 7b, respectively, the numerical solution u_h of the 2D Barenblatt equation and the values of the Lagrange multiplier are shown at time $t = 2$ on a mesh of 50×50 elements. In these computations, the KKT-Limiter was used, which successfully prevents the numerical solution u_h from becoming negative, which is shown in Figure 7c. The imposed constraint is $u_{h\min} = 10^{-10}$. Figure 7c also shows an excellent agreement between the exact solution u and the numerical solution u_h.

Example 5.8 (1D Buckley–Leverett equation). The Buckley–Leverett equation models two phase flow in a porous medium. We consider two cases, respectively, with and without gravity. Since the solution has to be strictly inside the range $[0, 1]$, we use

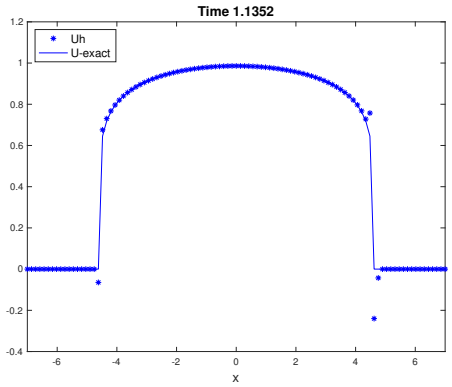


FIG. 6. *Example* 5.7a, *1D Barenblatt equation: numerical solution* u_h *without limiter and exact solution* u. *Computational mesh* 100 *elements, polynomial order* $p = 3$.

both the positivity and the maximum preserving limiters, with bounds $u_{h\min} = 10^{-10}$ and $u_{h\max} = 1 - 10^{-10}$, respectively. The computational domain is $\Omega = [0, 1]$. A Dirichlet boundary condition at $x = 0$, based on the initial solution, and an outflow boundary condition at $x = 1$ are imposed. The viscosity coefficient is $\bar{\nu} = 0.01$. Since we do not have an exact solution to compare with, we compute the numerical solution on two meshes, namely with 100 and 200 elements. The two test cases given by Examples 5.8a and 5.8b are also considered in [21].

Example 5.8a (1D Buckley–Leverett equation without gravity). The 1D Buckley–Leverett equation without gravity is obtained by setting $G(u) = 0$, and $\nu(u)$ and $F(u) = f(u)$, respectively, as

$$\nu(u) = \begin{cases} 4\bar{\nu}u(1-u) & \text{if } 0 \le u \le 1, \\ 0 & \text{otherwise;} \end{cases}$$

$$f(u) = \begin{cases} 0 & \text{if } u < 0, \\ \frac{u^2}{u^2+(1-u)^2} & \text{if } 0 \le u \le 1, \\ 1 & \text{if } u > 1. \end{cases} \tag{5.3}$$

The initial condition is

$$u(x,0) = \begin{cases} 0.99 - 3x, & 0 \le x \le 0.33, \\ 0, & \frac{1}{3} < x \le 1. \end{cases}$$

The numerical solution u_h and its derivative Q_h are shown in, respectively, Figures 8a and 8b. Also, the values of the Lagrange multiplier used to enforce the constraints are shown in Figure 8a. The limiter is only active in the thin layer between the phases and is crucial to obtain sensible physical solutions. The results of 100 and 200 elements match well.

Example 5.8b (1D Buckley–Leverett equation with gravity). A much more difficult test case is provided by the Buckley–Leverett equation with gravity, which is obtained

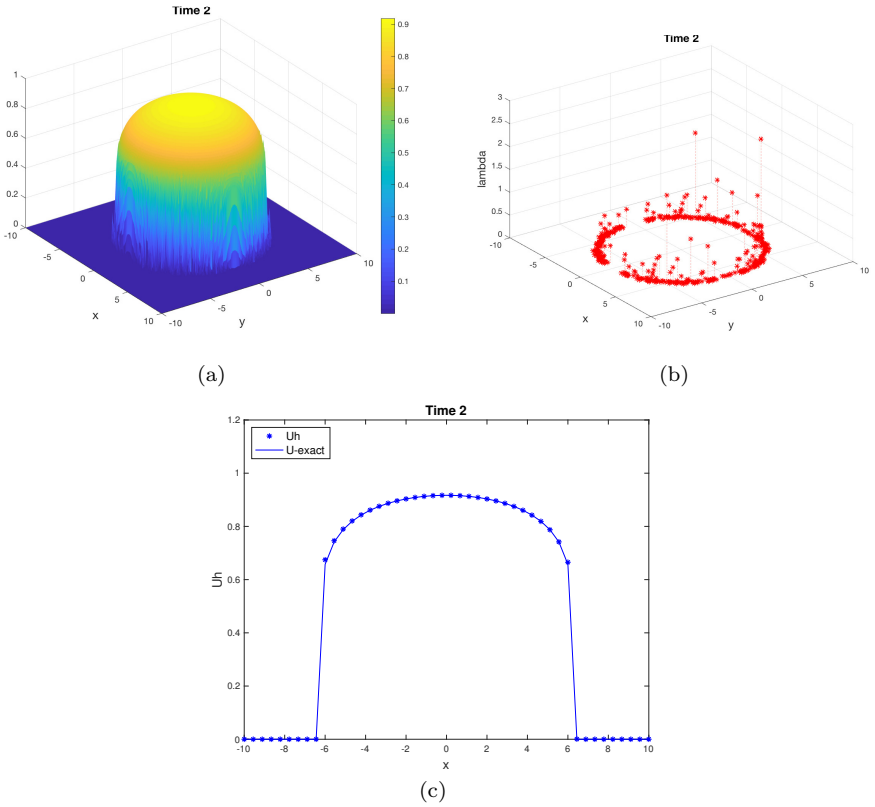


FIG. 7. *Example* 5.7b, *2D Barenblatt equation:* (a) *solution* u_h, (b) *Lagrange multiplier,* (c) *numerical solution* u_h *and exact solution* u *in cross-section at* $y = 0$. *Computational mesh* 50×50 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity preserving limiter larger than* 10^{-10} *are indicated in* (b) *with a red asterisk.*

by modifying the flux $F(u)$ as

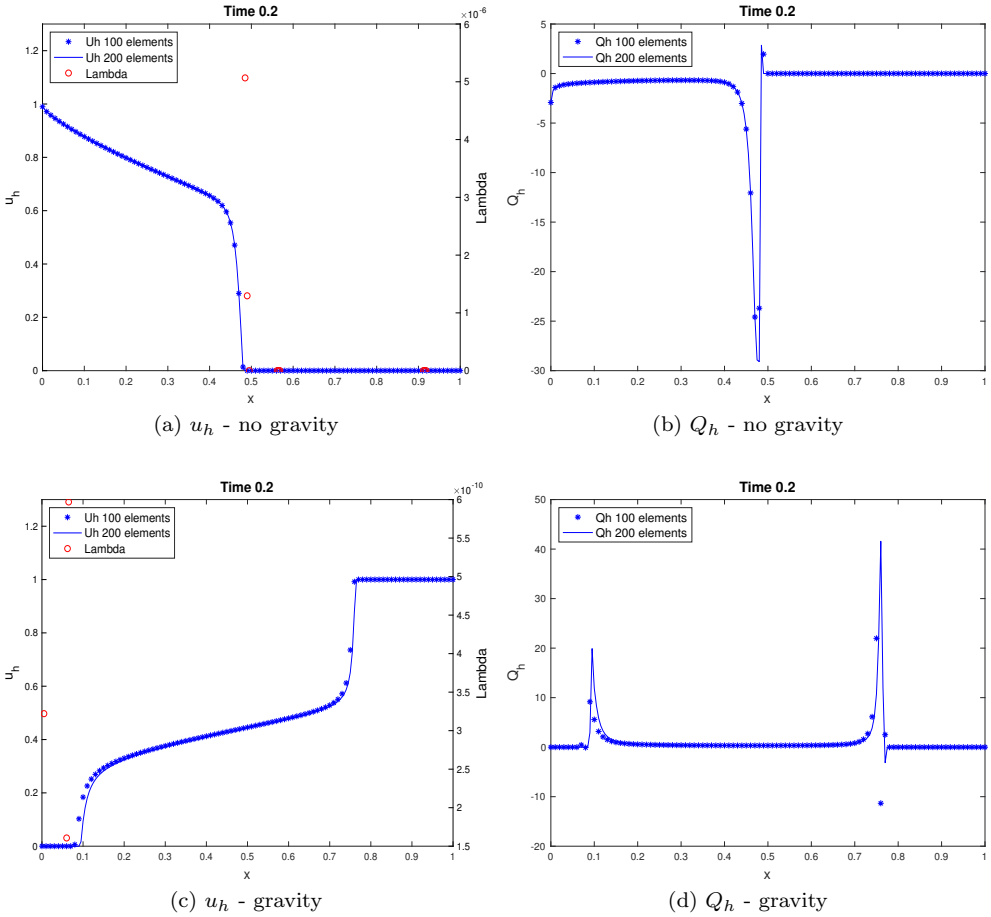
$$F(u) = \begin{cases} f(u)(1 - 5(1-u)^2), & u \le 1, \\ 1 & u > 1, \end{cases}$$

with $f(u)$ given by (5.3). The initial solution is

$$u(x,0) = \begin{cases} 0, & 0 \le x \le a, \\ \frac{1}{mh}(x-a), & a < x \le 1 - \frac{1}{\sqrt{2}}, \\ 1, & 1 - \frac{1}{\sqrt{2}} < x \le 1, \end{cases}$$

with $a = 1 - \frac{1}{\sqrt{2}} - mh$, h the mesh size, and $m = 3$. The linear transition for x in the range $[a, 1 - \frac{1}{\sqrt{2}}]$ is used to remove the infinite value in the derivative, which would otherwise result in unbounded values of Q_h at $t = 0$. The Buckley–Leverett equations with gravity result in a strongly nonlinear problem where the equations change type and are a severe test for the KKT-Limiter and semismooth Newton algorithm. The solution u_h and values of the Lagrange multiplier are shown in Figure 8c and the derivative Q_h in Figure 8d. The results on the two meshes compare well, and the

limiter ensures that the positivity and maximum bounds are satisfied.



(a) u_h - no gravity (b) Q_h - no gravity

(c) u_h - gravity (d) Q_h - gravity

FIG. 8. *Example* 5.8a, *2D Buckley–Leverett equation without gravity:* (a) *numerical solution* u_h, (b) *numerical solution derivative* Q_h. *Example* 5.8b, *1D Buckley–Leverett equation with gravity:* (c) *numerical solution* u_h, (d) *numerical solution derivative* Q_h. *Computational meshes* 100 *and* 200 *elements, polynomial order* $p = 3$. *Values of the Lagrange multiplier used in the positivity and maximum preserving limiters larger than* 10^{-10} *are indicated with a red (open) circle in* (a) *and* (c).

The number of Newton iterations necessary to obtain a minimum value 10^{-8} for the residual $F(z)$ and Newton update d in Algorithm 3.1 to stop the Newton iterations for each DIRK stage strongly varies. It depends on the type of equation, time step, and nonlinearity. In general, the time step is chosen such that the number of Newton iterations for each DIRK stage is between 5 and 20. For most time-dependent problems, the CFL number is then close to one, which is necessary to ensure time accuracy. Only for the Buckley–Leverett equation with gravity did the time step frequently have to be less than one in order to deal with the strong nonlinearity of the problem. In the computations, we did not observe a minimum time step to ensure positivity, as noticed in [28].

6. Conclusions. In this paper, we present a novel framework to combine positivity preserving limiters for DG discretizations with implicit time integration methods.

This approach does not depend on the specific type of DG discretization and is also applicable to, e.g., finite volume discretizations. The key features of the numerical method are the formulation of the positivity constraints as a KKT-problem and the development of an active set semismooth Newton method that accounts for the nonsmoothness of the algebraic equations. The algorithm was successfully tested on a number of increasingly difficult test cases, which required that the positivity constraints are satisfied in order to obtain meaningful results. The KKT-Limiter does not negatively affect the accuracy for smooth problems and accurately preserves the positivity constraints. Future work will focus on the extension of the KKT-Limiter to ensure also monotonicity of the solution.

Appendix A. Derivation of Clarke directional derivative. For completeness, we give here a derivation of the terms (3.8d) and (3.8e) in the Clarke directional derivative of $F(z)$ in (2.2). We will follow the approach outlined in [17]. Define $z := (x, \mu, \lambda)$, $\bar{z} := (\bar{x}, \bar{\mu}, \bar{\lambda})$, $d := (u, v, w) \in \mathbb{R}^p$, with $p = n + l + m$. Consider $\bar{F}(z) = F_{i+n+l}(z)$, $i \in \beta(z)$. The other Clarke directional derivatives of F are straightforward to compute. If we consider (3.2) only for the contribution of $\bar{F}(z)$ to the merit function to $\theta(z)$ and use (2.2) and a Taylor expansion of $\bar{F}(z)$ around z, then we obtain

$$\begin{aligned}\bar{\theta}^0(z;d) &= \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(\bar{F}(z), \min(-g(\bar{x}+tu), \bar{\lambda}+tw) - \min(-g(\bar{x}), \bar{\lambda})\Big)\\ &= \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(\bar{F}(z), \min(-g(x) - J(\bar{x}+tu-x), \bar{\lambda}+tw)\\ &\qquad - \min\big(-g(x) - J(\bar{x}-x), \bar{\lambda}\big)\Big),\end{aligned}$$

with $J := D_x g(x) \in \mathbb{R}^{m\times n}$. Here higher order terms are omitted since they will become zero in the limit. Define $h(x) := -g(x) + Jx$; then

$$\begin{aligned}\bar{\theta}^0(z;d) = \limsup_{\bar{z}\to z, t\downarrow 0^+} \frac{1}{t}\Big(&\bar{F}(z), \min(-J\bar{x} - tJu + h(x), \bar{\lambda}+tw)\\ &- \min(-J\bar{x} + h(x), \bar{\lambda})\Big).\end{aligned} \tag{A.1}$$

For $u \in \mathbb{R}^n, w \in \mathbb{R}^m$, define $r \in \mathbb{R}^m$ by

$$\begin{aligned}r_i < 0 \text{ on } S_1 :=& \{i \in \beta(z) \mid \bar{F}_i(z) > 0, -(Ju)_i > w_i\}\\ &\cup \{i \in \beta(z) \mid \bar{F}_i(z) \le 0, -(Ju)_i \le w_i\},\end{aligned} \tag{A.2a}$$

$$\begin{aligned}r_i > 0 \text{ on } S_2 :=& \{i \in \beta(z) \mid \bar{F}_i(z) > 0, -(Ju)_i \le w_i\}\\ &\cup \{i \in \beta(z) \mid \bar{F}_i(z) \le 0, -(Ju)_i > w_i\}.\end{aligned} \tag{A.2b}$$

Let $\bar{x} \in \mathbb{R}^n$ be such that

$$-J\bar{x} + h(x) = \bar{\lambda} + r. \tag{A.3}$$

Note that such an $\bar{x}$ exists for $i \in \beta(z)$ since (A.3) is equivalent to $-Ju = w + r$ with $u = \bar{x} - x$ and $w = \bar{\lambda} - \lambda$ as components of the search direction d. Choose $t \in (0, t_{\bar{x}})$ for $t_{\bar{x}} > 0$ such that

$$(-J\bar{x} + h(x) - tJu)_i < (\bar{\lambda} + tw)_i \quad \text{for } i \in S_1, \tag{A.4a}$$

$$(-J\bar{x} + h(x) - tJu)_i > (\bar{\lambda} + tw)_i \quad \text{for } i \in S_2. \tag{A.4b}$$

Note that such a $t_{\bar{x}}$ exists; see Remark A.1. We then obtain

$$\min((-J\bar{x}+h(x)-tJu)_i,(\bar{\lambda}+tw)_i)=\begin{cases}(-J\bar{x}+h(x)-tJu)_i & \text{for } i\in S_1,\\ (\bar{\lambda}+tw)_i & \text{for } i\in S_2.\end{cases}$$

Use now (A.3) and (A.2); then

$$\min((-J\bar{x}+h(x))_i,\bar{\lambda}_i)=\min(\bar{\lambda}_i+r_i,\bar{\lambda}_i)=\begin{cases}\bar{\lambda}_i+r_i & \text{for } i\in S_i,\\ \bar{\lambda}_i & \text{for } i\in S_2.\end{cases}$$

Combining the above results and using (A.3) again gives

$$\begin{aligned}\min((-J\bar{x}+h(x)-tJu)_i,(\bar{\lambda}+tw)_i)&-\min((-J\bar{x}+h(x))_i,\bar{\lambda}_i)\\ &=\begin{cases}-t(Ju)_i & \text{for } i\in S_1,\\ tw_i & \text{for } i\in S_2\end{cases}\\ &=\begin{cases}t\max(-(Ju)_i,w_i) & \text{if } \bar{F}_i(z)>0,\\ t\min(-(Ju)_i,w_i) & \text{if } \bar{F}_i(z)\le 0.\end{cases}\end{aligned}$$

Taking the limit in (A.1) and using (3.3) for $\bar{\theta}(z;d)$ then gives (3.8d) and (3.8e).

Remark A.1. Conditions (A.2) imply (A.4). Use $-J\bar{x}+h(x)=\bar{\lambda}+r$ in (A.4); then we obtain

$$(r-tJu)_i<tw_i \quad \text{for } i\in S_1, \tag{A.5}$$

$$(r-tJu)_i>tw_i \quad \text{for } i\in S_2. \tag{A.6}$$

I. If $i\in S_1$, $\bar{F}_i(z)>0$, then from (A.2a) we obtain $-(Ju)_i-w_i>0$ and (A.5) implies $r_i+t(-(Ju)_i-w_i)<0$. Choose $t<\frac{-r_i}{-(Ju)_i-w_i}=t_{\bar{x}}$. Since $r_i<0$ and $-(Ju)_i-w_i>0$ for $i\in S_1$, $\bar{F}_i(z)>0$, we obtain that $t_{\bar{x}}>0$.

II. If $i\in S_1$, $\bar{F}_i(z)\le 0$, then (A.2a) implies $-(Ju)_i-w_i\le 0$ and (A.5) gives $r_i+t(-(Ju)_i-w_i)<0$. Since both r_i and $-(Ju)_i-w_i<0$, any $t>0$ will imply (A.5).

The proof for $i\in S_2$ is completely analogous and is therefore omitted. Hence there exists a $t_{\bar{x}}>0$ for (A.4).

Appendix B. Verification of conditions for quasi-directional derivative. In this section, we show that the quasi-directional derivative (3.9) satisfies the conditions stated in (3.5), which are necessary to ensure convergence of the Newton algorithm defined in Algorithm 3.1.

Consider condition (3.5a): First note that

$$\begin{aligned}&F_i'(z;d)=F_i^0(z;d)=G_i(z;d), && i\in N_n,\\ &F_{i+n}'(z;d)=F_{i+n}^0(z;d)=G_{i+n}(z;d), && i\in N_l,\\ &F_{i+n+l}'(z;d)=F_{i+n+l}^0(z;d)=G_{i+n+l}(z;d), && i\in \alpha_\delta(z)\cup\gamma_\delta(z),\end{aligned}$$

since $\alpha_\delta(z)\cup\gamma_\delta(z)\subset\alpha(z)\cup\gamma(z)$. If $i\in\beta_\delta(z)$ and $F_{i+n+l}(z)\le 0$, then

$$\min(-(Ju)_i,w_i)\le -(Ju)_i,w_i.$$

Since $F_{i+n+l}(z) \leq 0$, this implies

$$F_{i+n+l}(z)\min(-(Ju)_i, w_i) \geq F_{i+n+l}(z)(-(Ju)_i), F_{i+n+l}(z)w_i.$$

If $i \in \beta_\delta(x)$ and $F_{i+n+l}(z) > 0$, then

$$-(Ju)_i, w_i \leq \max(-(Ju)_i, w_i).$$

Hence, since $F_{i+n+l}(z) > 0$, this implies

$$F_{i+n+l}(z)(-(Ju)_i), F_{i+n+l}(z)w_i \leq F_{i+n+l}(z)\max(-(Ju)_i, w_i).$$

Comparing all terms then immediately shows that $G(z; d)$ satisfies (3.5a) and (3.5c). Condition (3.5b) directly follows from the definition of G in (3.5).

Acknowledgment. We would like to acknowledge Mrs. Fengna Yan from USTC and the University of Twente for her contributions in testing the KKT-Limiter for several DG discretizations.

REFERENCES

[1] R. Alexander, *Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s*, SIAM J. Numer. Anal., 14 (1977), pp. 1006–1021, https://doi.org/10.1137/0714068.

[2] P. R. Amestoy, I. S. Duff, D. Ruiz, and B. Uçar, *A parallel matrix scaling algorithm*, in International Conference on High Performance Computing for Computational Science - VECPAR 2008, Springer, Berlin, Heidelberg, 2008, pp. 301–313.

[3] P. Bochev and D. Ridzal, *Optimization-based additive decomposition of weakly coercive problems with applications*, Comput. Math. Appl., 71 (2016), pp. 2140–2154, https://doi.org/10.1016/j.camwa.2015.12.032.

[4] J.-S. Chen, S. Pan, and T.-C. Lin, *A smoothing Newton method based on the generalized Fischer–Burmeister function for MCPs*, Nonlinear Anal., 72 (2010), pp. 3739–3758, https://doi.org/10.1016/j.na.2010.01.012.

[5] Z. Chen, H. Huang, and J. Yan, *Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes*, J. Comput. Phys., 308 (2016), pp. 198–217, https://doi.org/10.1016/j.jcp.2015.12.039.

[6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990, https://doi.org/10.1137/1.9781611971309.

[7] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463, https://doi.org/10.1137/S0036142997316712.

[8] M. D'Elia, M. Perego, P. Bochev, and D. Littlewood, *A coupling strategy for nonlocal and local diffusion models with mixed volume constraints and boundary conditions*, Comput. Math. Appl., 71 (2016), pp. 2218–2230, https://doi.org/10.1016/j.camwa.2015.12.006.

[9] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer, Heidelberg, 2011.

[10] J. A. Evans, T. J. Hughes, and G. Sangalli, *Enforcement of constraints and maximum principles in the variational multiscale method*, Comput. Methods Appl. Mech. Engrg., 199 (2009), pp. 61–76, https://doi.org/10.1016/j.cma.2009.09.019.

[11] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer Science & Business Media, New York, 2007.

[12] H. Guo and Y. Yang, *Bound-preserving discontinuous Galerkin method for compressible miscible displacement in porous media*, SIAM J. Sci. Comput., 39 (2017), pp. A1969–A1990, https://doi.org/10.1137/16M1101313.

[13] L. Guo and Y. Yang, *Positivity preserving high-order local discontinuous Galerkin method for parabolic equations with blow-up solutions*, J. Comput. Phys., 289 (2015), pp. 181–195, https://doi.org/10.1016/j.jcp.2015.02.041.

[14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II. *Stiff and Differential-Algebraic Problem*, Springer, Berlin, 2010.

[15] S.-P. Han, J.-S. Pang, and N. Rangaraj, *Globally convergent Newton methods for nonsmooth equations*, Math. Oper. Res., 17 (1992), pp. 586–607.
[16] P. T. Harker and J.-S. Pang, *A damped-Newton method for the linear complementarity problem*, in Computational Solution of Nonlinear Systems of Equations, Lectures in Appl. Math. 26, AMS, Providence, RI, 1990, pp. 265–284.
[17] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, SIAM, Philadelphia, 2008, https://doi.org/10.1137/1.9780898718614.
[18] K. Ito and K. Kunisch, *On a semi-smooth Newton method and its globalization*, Math. Program., 118 (2009), pp. 347–370, https://doi.org/10.1007/s10107-007-0196-3.
[19] G. Karniadakis and S. Sherwin, *Spectral/HP Element Methods for Computational Fluid Dynamics*, Oxford University Press, New York, 2013.
[20] P. Kuberry, P. Bochev, and K. Peterson, *An optimization-based approach for elliptic problems with interfaces*, SIAM J. Sci. Comput., 39 (2017), pp. S757–S781, https://doi.org/10.1137/16M1084547.
[21] A. Kurganov and E. Tadmor, *New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282, https://doi.org/10.1006/jcph.2000.6459.
[22] A. Meister and S. Ortleb, *On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows*, Internat. J. Numer. Methods Fluids, 76 (2014), pp. 69–94, https://doi.org/10.1002/fld.3921.
[23] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, and C. Kanzow, *The semismooth algorithm for large scale complementarity problems*, INFORMS J. Comput., 13 (2001), pp. 294–311, https://doi.org/10.1287/ijoc.13.4.294.9734.
[24] J.-S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.
[25] J.-S. Pang, *A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming, 51 (1991), pp. 101–131.
[26] S. Patankar, *Numerical Heat Transfer and Fluid Flow*, CRC Press, Boca Raton, FL, 1980.
[27] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.
[28] T. Qin and C.-W. Shu, *Implicit positivity-preserving high-order discontinuous Galerkin methods for conservation laws*, SIAM J. Sci. Comput., 40 (2018), pp. A81–A107, https://doi.org/10.1137/17M112436X.
[29] T. Qin, C.-W. Shu, and Y. Yang, *Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics*, J. Comput. Phys., 315 (2016), pp. 323–347, https://doi.org/10.1016/j.jcp.2016.02.079.
[30] A. Shapiro, *On concepts of directional differentiability*, J. Optim. Theory Appl., 66 (1990), pp. 477–487.
[31] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[32] L. Skvortsov, *Diagonally implicit Runge-Kutta methods for stiff problems*, Comput. Math. Math. Phys., 46 (2006), pp. 2110–2123, https://doi.org/10.1134/S0965542506120098.
[33] X. Zhang, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations*, J. Comput. Phys., 328 (2017), pp. 301–343, https://doi.org/10.1016/j.jcp.2016.10.002.
[34] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, J. Comput. Phys., 229 (2010), pp. 3091–3120, https://doi.org/10.1016/j.jcp.2009.12.030.
[35] X. Zhang and C.-W. Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, J. Comput. Phys., 229 (2010), pp. 8918–8934, https://doi.org/10.1016/j.jcp.2010.08.016.
[36] X. Zhang and C.-W. Shu, *Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms*, J. Comput. Phys., 230 (2011), pp. 1238–1248, https://doi.org/10.1016/j.jcp.2010.10.036.
[37] X. Zhang, Y. Xia, and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes*, J. Sci. Comput., 50 (2012), pp. 29–62, https://doi.org/10.1007/s10915-011-9472-8.
[38] Y. Zhang, X. Zhang, and C.-W. Shu, *Maximum-principle-satisfying second order discontinuous Galerkin schemes for convection–diffusion equations on triangular meshes*, J. Comput. Phys., 234 (2013), pp. 295–316, https://doi.org/10.1016/j.jcp.2012.09.032.