Local Discontinuous Galerkin Methods for One-Dimensional Second Order Fully Nonlinear Elliptic and Parabolic Equations

By: Xiaobing Feng, Thomas Lewis

Abstract:

This paper is concerned with developing accurate and efficient nonstandard discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic partial differential equations (PDEs) in the case of one spatial dimension. The primary goal of the paper to develop a general framework for constructing high order local discontinuous Galerkin (LDG) methods for approximating viscosity solutions of these fully nonlinear PDEs which are merely continuous functions by definition. In order to capture discontinuities of the first order derivative u_x of the solution u, two independent functions $q-$ and $q+$ are introduced to approximate one-sided derivatives of u. Similarly, to capture the discontinuities of the second order derivative u_{xx}, four independent functions $p--, p-+, p+-,$ and $p++$ are used to approximate one-sided derivatives of $q-$ and $q+$. The proposed LDG framework, which is based on a nonstandard mixed formulation of the underlying PDE, embeds a given fully nonlinear problem into a mostly linear system of equations where the given nonlinear differential operator must be replaced by a numerical operator which allows multiple value inputs of the first and second order derivatives u_x and u_{xx}. An easy to verify set of criteria for constructing "good" numerical operators is also proposed. It consists of consistency and generalized monotonicity. To ensure such a generalized monotonicity property, the crux of the construction is to introduce the numerical moment in the numerical operator, which plays a critical role in the proposed LDG framework. The generalized monotonicity gives the LDG methods the ability to select the viscosity solution among all possible solutions. The proposed framework extends a companion finite difference framework developed by Feng and Lewis (J Comp Appl Math 254:81–98, *2013*) and allows for the approximation of fully nonlinear PDEs using high order polynomials and non-uniform meshes. Numerical experiments are also presented to demonstrate the accuracy, efficiency and utility of the proposed LDG methods.

Keywords: Fully nonlinear PDEs | Viscosity solutions | Local discontinuous Galerkin methods

1 Introduction

This is the third paper in a series [*12*, *16*] which is devoted to developing finite difference (FD) and discontinuous Galerkin (DG) methods for approximating *viscosity solutions* of the following general one-dimensional fully nonlinear second order elliptic and parabolic equations:

$F(u_{xx}, u_x, u, x) = 0, x \in \Omega := (a,b) \subset \mathbf{R},$

(1.1)

and

$u_t + F(u_{xx}, u_x, u, x, t) = 0, (x,t) \in \Omega T := \Omega \times (0,T),$

(1.2)

which are complemented by appropriate boundary and initial conditions.

Fully nonlinear PDEs, which are nonlinear in the highest order derivatives of the solution functions in the equations, arise in many applications such as antenna design, astrophysics, differential geometry, fluid mechanics, image processing, meteorology, mesh generation, optimal control, optimal mass transport, etc (cf. [*11*, section 5]), and, as a result, the solution of each of these application problems critically depends on the solution of their underlying fully nonlinear PDEs. In particular, it calls for efficient and reliable numerical methods for computing the viscosity solutions of these fully nonlinear PDEs. Currently, the availability of such numerical methods is very limited (cf. [*11*]).

The goal of this paper is to design and implement a class of nonstandard local discontinuous Galerkin (LDG) methods for the fully nonlinear Eqs. (1.1) and (1.2). The more involved high dimensional generalizations of the nonstandard LDG methods of this paper will be reported elsewhere in [*17*]. The methods of this paper are designed to complement the interior penalty discontinuous Galerkin (IP-DG) methods for fully nonlinear second order equations developed in [*16*], where the focus is on approximating PDEs (1.1) and (1.2) whose viscosity solutions belong to $C1$.

To present the main ideas of our nonstandard LDG methods, we first briefly describe the nonstandard LDG method proposed by Yan and Osher [*22*] for approximating the viscosity solution of the Hamilton-Jacobi equation: $u_t + H(\nabla u, u, x, t) = 0$. The main ideas of [*22*] are to approximate the "left" and "right" side first order derivatives of the viscosity solution and to judiciously combine them through a monotone and consistent numerical Hamiltonian such as the Lax-Friedrichs numerical Hamiltonian. We note that the idea of pulling the highest order derivative(s) outside fully nonlinear PDEs is essential because it allows one to take advantages of

DG techniques to discretize the given fully nonlinear PDEs. Our nonstandard LDG methods to be presented below are exactly inspired by this idea, although the realization of this idea for fully nonlinear second order PDEs is more involved. Below we highlight the main steps/ideas of the construction of our nonstandard LDG framework.

Because of the full nonlinearity, integration by parts, which is the necessary tool for constructing any DG method, cannot be performed on Eq. (1.1). *The first key idea* of this paper is to introduce the auxiliary variables $p:=u_{xx}$ and $q:=u_x$ and rewrite the original fully nonlinear PDE in the following nonstandard mixed form:

$$F(p,q,u,x)=0,$$

(1.3)

$$q-u_x=0,$$

(1.4)

$$p-q_x=0.$$

(1.5)

Unfortunately, since u_x and u_{xx} may not exist for a viscosity solution $u \in C^0(\Omega)$, the above mixed form may not make sense. To overcome this difficulty, *our second key idea* is to replace $q=u_x$ by two possible values of u_x, namely, its left and right limits, and $p=q_x$ by two possible values for each possible q. Thus, we have the auxiliary
variables $q^-, q^+ : \Omega \to \mathbf{R}$ and $p^{--}, p^{-+}, p^{+-}, p^{++} : \Omega \to \mathbf{R}$ such that

$$q^-(x)-u_x(x-)=0,$$

(1.6)

$$q^+(x)-u_x(x+)=0,$$

(1.7)

$$p^{--}(x)-q^-_x(x-)=0,$$

(1.8)

$$p^{-+}(x)-q^-_x(x+)=0,$$

(1.9)

$$p^{+-}(x)-q^+_x(x-)=0,$$

(1.10)

$p++(x)-q+x(x+)=0.$

(1.11)

We remark that (1.6) paired with the Eq. (1.8) or (1.9), and (1.7) paired with Eq. (1.10) or (1.11), can each be regarded as a "one-sided" Poisson problem in u (in a mixed form) with source terms $p--,p-+,p+-,p++$, respectively.

To incorporate the multiple values of ux and uxx, Eq. (1.3) must be modified because F is only defined for single value functions p and q. To this end, we need *the third key idea* of this paper, that is, to replace (1.3) by

$F\hat{}(p--,p-+,p+-,p++,q-,q+,u,x)=0,$

(1.12)

where $F\hat{}$, which is called a *numerical operator*, should be some well-chosen approximation to F.

Natural questions now arise regarding the choice of $F\hat{}$. For example, what are criteria for $F\hat{}$ and how to construct such a numerical operator? These are two immediate questions which must be addressed. To do so, we need *the fourth key idea* of this paper, which is to borrow and adapt the notion of the numerical operators from our previous work [*12*] where a general finite difference framework has been developed for fully nonlinear second order PDEs. In summary, the criteria for $F\hat{}$ include *consistency* and *g-monotonicity*(generalized monotonicity), for which precise definitions can be found in Sect. 2. It should be pointed out that in order to construct the desired numerical operator $F\hat{}$, a fundamental idea used in [*12*] is to introduce the concept of *the numerical moment*, which can be regarded as a direct numerical realization for the moment term in *the vanishing moment methodology* introduced in [*10*] (also see [*11*, section 4], [*9*]).

Finally, we need to design a DG discretization for the mixed system (1.6)–(1.12) to complete the construction of our LDG method. This then calls for *the fifth key idea* of this paper, which is to use different *numerical fluxes* in the formulations of LDG methods for the four "one-sided" Poisson problems in their mixed forms described by (1.6)–(1.11). We remark that, to the best of our knowledge, this is one of a few scenarios in numerical PDEs where the flexibility and superiority (over other numerical methodologies) of the DG methodology makes a vital difference.

At this point, it is clear why we use the word "nonstandard" to indicate the differences between our LDG methods and the standard LDG methods [*6*] for linear and quasilinear PDEs. In the standard LDG methods, one only needs to introduce one copy of the vector $q:=\nabla u$, and there is no need to introduce multiple copies of $p:=D2u$ as independent unknowns to approximate. We also note that in the nonstandard IP-DG framework developed in [*16*], because u is assumed to be in $C1$, $q:=\nabla u$ is not introduced as an unknown function, instead, ∇uh is hard computed in the IP-DG methods and the approximated sided Hessians $p--h,p-+h,p+-h,p++h$ are defined directly

through *uh*. It is also interesting to point out that unlike the relationship between standard LDG and IP-DG methods (cf. [*1*]), due to fully nonlinear structure of the PDEs, no integration by parts can be performed, as a result, the "primal" forms of the nonstandard LDG methods of this paper are intrinsically different from the nonstandard IP-DG methods of [*16*] and they may exhibit different properties, especially in high dimensions (cf. [*17*]).

This paper consists of four additional sections. In Sect. 2 we collect some preliminaries including the definition of viscosity solutions, the definitions of the consistency and g-monotonicity of numerical operators, and the concept of the numerical moment. In Sect. 3 we give the detailed formulation of LDG methods for fully nonlinear elliptic equation (1.1) following the outline described above. In Sect. 4 we consider both explicit and implicit in time fully discrete LDG methods for the fully nonlinear parabolic equation (1.2). The explicit four stage classical Runge-Kutta method and the implicit trapezoidal method combined with the spatial LDG methods will be explicitly constructed. In Sect. 5 we present a number of numerical experiments for the proposed LDG methods for the fully nonlinear elliptic equation (1.1) and their fully discrete counterparts for the parabolic equation (1.2). These numerical experiments not only verify the accuracy of the proposed LDG methods but also demonstrate the efficiency and utility of these methods.

2 Preliminaries

Standard space notations are adopted in this paper. For example, $B(\Omega), USC(\Omega)$ and $LSC(\Omega)$ denote, respectively, the spaces of bounded, upper semi-continuous, and lower semicontinuous functions on Ω. For any $v \in B(\Omega)$, we define

$v*(x) := \limsup_{y \to x} v(y)$ and $v*(x) := \liminf_{y \to x} v(y)$.

Then, $v* \in USC(\Omega)$ and $v* \in LSC(\Omega)$, and they are called *the upper and lower semicontinuous envelopes* of v, respectively. If v is continuous, there obviously holds $v = v* = v*$.

Let $F: S^{d \times d} \times \mathbf{R}^d \times \mathbf{R} \times \overline{\Omega} \to \mathbf{R}$ be a bounded function, where $S^{d \times d}$ denotes the set of $d \times d$ symmetric real matrices. Then, the general second order fully nonlinear PDE takes the form

$F(D^2 u, \nabla u, u, x) = 0$ in $\overline{\Omega}$.

(2.1)

Note that here we have used the convention of writing the boundary condition as a discontinuity of the PDE (cf. [*3*, p.274]).

The following two definitions can be found in [*3*, *4*, *13*].

Definition 2.1

Equation (2.1) is said to be elliptic if, for all $(\mathbf{q}, \lambda, x) \in \mathbf{R}d \times \mathbf{R} \times \overline{\Omega}$, there holds

$$F(A, \mathbf{q}, \lambda, x) \leq F(B, \mathbf{q}, \lambda, x) \forall A, B \in Sd \times d, A \geq B,$$

(2.2)

where $A \geq B$ means that $A - B$ is a nonnegative definite matrix. We note that when F is differentiable, the ellipticity also can be defined by requiring that the matrix $\partial F \partial A$ is negative semi-definite (cf. [13, p. 441]).

Definition 2.2

A function $u \in B(\Omega)$ is called a viscosity subsolution (resp. supersolution) of (2.1) if, for all $\varphi \in C2(\overline{\Omega})$, if $u* - \varphi$ (resp. $u* - \varphi$) has a local maximum (resp. minimum) at $x0 \in \overline{\Omega}$, then we have

$$F*(D2\varphi(x0), \nabla\varphi(x0), u*(x0), x0) \leq 0$$

(resp. $F*(D2\varphi(x0), \nabla\varphi(x0), u*(x0), x0) \geq 0$). The function u is said to be a viscosity solution of (2.1) if it is simultaneously a viscosity subsolution and a viscosity supersolution of (2.1).

We note that if F and u are continuous, then the upper and lower $*$ indices can be removed in Definition 2.2. The definition of ellipticity implies that the differential operator F must be non-increasing in its first argument in order to be elliptic. It turns out that ellipticity provides a sufficient condition for Eq. (2.1) to fulfill a maximum principle (cf. [4, 13]). From the above definition it is clear that viscosity solutions in general do not satisfy the underlying PDEs in a tangible sense, and the concept of viscosity solutions is*nonvariational*. Such a solution is not defined through integration by parts against arbitrary test functions; hence, it does not satisfy an integral identity. This nonvariational nature of viscosity solutions is the main obstacle that prevents direct construction of Galerkin-type methods, which require variational formulations to start.

The following definitions are adapted from [12] in the case $d = 1$.

Definition 2.3

(i)

A function $F\hat{}: \mathbf{R}8 \rightarrow \mathbf{R}$ is called a *numerical operator*.

(ii)

A numerical operator $F\hat{}$ is said to be consistent (with the differential operator F) if $F\hat{}$ satisfies

$$\liminf_{p\mu\nu\to p,\mu,\nu=+,-q\pm\to q,\lambda1\to\lambda,\xi1\to\xi}\hat{F}(pp--,p-+,p+-,p++,q-,q+,\lambda1,\xi1)\geq F_*(p,q,\lambda,\xi),$$

(2.3)

$$\limsup_{p\mu\nu\to p,\mu,\nu=+,-q\pm\to q,\lambda1\to\lambda,\xi1\to\xi}\hat{F}(pp--,p-+,p+-,p++,q-,q+,\lambda1,\xi1)\leq F^*(p,q,\lambda,\xi),$$

(2.4)

where F_* and F^* denote respectively the lower and the upper semi-continuous envelopes of F.

(iii)

A numerical operator $\hat{F}$ is said to be *g-monotone* if $\hat{F}(p--,p-+,p+-,p++,q-,q+,\lambda,\xi)$ is monotone increasing in $p--$ and $p++$ and monotone decreasing in $p-+$ and $p+-$, that is, $\hat{F}(\uparrow,\downarrow,\downarrow,\uparrow,q-,q+,\lambda,\xi)$.

We remark that the above consistency and g-monotonicity play a critical role in the finite difference framework established in [*12*]. They also play an equally critical role in the LDG framework of this paper. In practice, the consistency is easy to fulfill and to verify, but the g-monotonicity is not. In order to ensure the g-monotonicity, one key idea of [*12*] is to introduce *the numerical moment*. The following is an example of a so-called Lax-Friedrichs-like numerical operator adapted from [*12*]:

$$\hat{F}(p--,p-+,p+-,p++,q-,q+,\lambda,\xi):=F(p-++p+-2,q-+q+2,\lambda,\xi)+\alpha(p---p-+-p+-+p++),$$

(2.5)

where $\alpha\in\mathbf{R}$ is an undetermined positive constant and the last term in (2.5) is called *the numerical moment*. It is trivial to verify that $\hat{F}$ is consistent with F. By choosing α correctly, we can also ensure g-monotonicity. For example, suppose F is differentiable and there exists a positive constant M such that

$$M>|||\partial F\partial uxx|||.$$

(2.6)

Then, it is trivial to check that the Lax-Friedrichs-like numerical operator is g-monotone provided that $\alpha\geq M$.

We conclude this section with a few remarks about the above definitions.

Remark 2.1

(a)

By the definition of the ellipticity for F, the monotonicity constraints on $\hat{F}$ with respect to $p-+$ and $p+-$ in the definition of g-monotonicity are natural.

(b)

By choosing the numerical moment correctly, the numerical operator $\hat{F}$ then behaves like a uniformly elliptic operator, even if the PDE operator F is a degenerate elliptic operator. The consistency assumption then guarantees that the numerical operator is still a reasonable approximation for the PDE operator.

(c)

Sometimes it may not be feasible to globally bound $\partial F \partial u_{xx}$; however, it is sufficient to chose a value for α such that the g-monotonicity property is preserved locally over each iteration of the nonlinear solver for a given initial guess.

(d)

The role of the numerical moment as well as the interpretation of the numerical moment will be further discussed in Sect. 5.3.

3 Formulation of LDG Methods for Elliptic Problems

We first consider the elliptic problem (1.1) with boundary conditions

$u(a)=u_a \text{ and } u(b)=u_b$

(3.1)

for two given constants u_a and u_b.

Let $\{x_j\}_{j=0}^{J} \subset \overline{\Omega}$ be a mesh for $\overline{\Omega}$ such that $x_0=a$ and $x_J=b$.
Define $I_j=(x_{j-1},x_j)$ and $h_j=x_j-x_{j-1}$ for all $j=1,2,\ldots,J, h_0=h_{J+1}=0$, and $h=\max_{1\leq j\leq J} h_j$.
Let T_h denote the collection of the intervals $\{I_j\}_{j=1}^{J}$ which form a partition of the domain $\overline{\Omega}$.
We also introduce the broken H^1-space and broken C^0-space

$H^1(T_h):=\prod_{I\in T_h} H^1(I), C^0(T_h):=\prod_{I\in T_h} C^0(\Gamma),$

and the broken $L2$-inner product

$(v,w)\mathrm{T}h:=\sum_{j=1}^{J}\int_{Ij}vw\,dx\,\forall v,w\in L2(\Omega).$

For a fixed integer $r\geq 0$, we define the standard DG finite element space $Vh\subset H1(\mathrm{T}h)\subset L2(\mathrm{T}h)$ as

$Vh:=\prod_{I\in\mathrm{T}h}\mathrm{P}r(I),$

where $\mathrm{P}r(I)$ denotes the set of all polynomials on I with degree not exceeding r. We also introduce the following standard jump notation:

$[vh(xj)]:=vh(x{-}j)-vh(x{+}j)\;\text{for}\;j=1,2,\cdots,J-1.$

We now are ready to formulate our LDG discretizations for Eqs. (1.6)–(1.12). First, for (fully) nonlinear equation (1.12) we simply approximate it by its broken $L2$-projection into Vh, namely,

$a0(uh,q{-}h,q{+}h,p{-}{-}h,p{-}{+}h,p{+}{-}h,p{+}{+}h;\phi0h)=0\,\forall\phi0h\in Vh,$

(3.2)

where

$a0(v,q{-},q{+},p{-}{-},p{-}{+},p{+}{-},p{+}{+};\phi)=(\hat{F}(p{-}{-},p{-}{+},p{+}{-},p{+}{+},q{-},q{+},v,\cdot),\phi)\mathrm{T}h.$

Next, we discretize the four *linear* equations (1.8)–(1.11). Notice that for given "sources" $\{p\mu v\}\mu,v=+,-$, (1.6) and (1.8), (1.6) and (1.9), (1.7) and (1.10), and (1.7) and (1.11) are four (different) Poisson equations for u. Thus, we can use the mixed upwinding LDG formulation for the Laplacian operator to discretize these equations. The only difference in the four equations will be how we choose our upwinding numerical fluxes for $uh,q{-}h$ and $q{+}h$. To realize the above strategy, we first define the element-wise LDG formulation, and we then define the whole domain LDG formulation afterward.

3.1 Element-Wise LDG Formulation

Suppose that values for $uh(a-),uh(b+),q{\pm}h(a-)$, and $q{\pm}h(b+)$ are given. We postpone explaining how these values are chosen until Sect. 3.2. Our LDG discretization of Eqs. (1.6)–(1.11) is defined as follows: for all $\phi{\pm}h\in Vh$,

$\int_{Ij}q{\pm}h\phi{\pm}h\,dx+\int_{Ij}uh(\phi{\pm}h)x\,dx=uh(x{\pm}j)\phi{\pm}h(x{-}j)-uh(x{\pm}j{-}1)\phi{\pm}h(x{+}j{-}1)$

(3.3)

and, for all $\psi\mu vh\in Vh$,

$\int_{Ij}p\mu vh\psi\mu vh\,dx+\int_{Ij}q\mu h(\psi\mu vh)x\,dx=q\mu h(xvj)\psi\mu vh(x{-}j)-q\mu h(xvj{-}1)\psi\mu vh(x{+}j{-}1)$

(3.4)

for $\mu,\nu=+,-$, for all $j=1,2,\cdots,J$. Notice that the nodal values from the right or left and the choice of $q\pm h$ in (3.4) follow directly from Eqs. (1.6)–(1.11).

3.2 Boundary Numerical Fluxes

To complete the construction, we must specify how the boundary numerical flux values for $uh,q-h$, and $q+h$ are determined in the above formulation. Due to the inherent jumps of piecewise constant functions, which corresponds to the case $r=0$, we shall consider the two cases $r\geq1$ and $r=0$ separately.

When $r\geq1$, we have freedom to control how the functions uh, $q-h$, and $q+h$ approach the boundary. Thus, we can assume continuity across the boundary for $uh,q-h$, and $q+h$. Considering the boundary conditions given by (3.1), the continuity requirement naturally leads to

$uh(a\pm)=ua,uh(b\pm)=ub.$

(3.5)

On the other hand, since no boundary data for $q-h$ or $q+h$ is given, any choice of the boundary numerical fluxes for them is a guess (unless one already knows the exact solution u). Here we choose

$q\pm h(a-)=q\pm h(a+),q\pm h(b+)=q\pm h(b-).$

(3.6)

It is important to note that both $q\pm h(a+)$ and $q\pm h(b-)$ are treated as unknowns in the above LDG formulation. The choice (3.6) is equivalent to requiring that $q\pm h$ is continuous at the boundary nodes $x=a$ and $x=b$.

We now consider the case $r=0$. To define the boundary numerical fluxes, we first examine the consequences of the interior flux choices represented by the above LDG formulation. Suppose Th is a uniform mesh and denote the midpoint of Ij by $x^\wedge j$ for all $Ij\in Th$. Define $Uj:=uh(x^\wedge j)$. Then, it follows from (3.3) and (3.4) that

$q-h(x^\wedge j)=Uj-Uj-1h:=\delta-xUj,$

(3.7)

$q+h(x^\wedge j)=Uj+1-Ujh:=\delta+xUj,$

(3.8)

$p--h(x^\wedge j)=q-h(x^\wedge j)-q-h(x^\wedge j-1)h=Uj-2-2Uj-1+Ujh2:=\delta2xUj-1,$

(3.9)

$$p_{-+}^h(x_j) = \frac{q_-^h(x_{j+1}) - q_-^h(x_j)}{h} = \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} =: \delta_x^2 U_j,$$

(3.10)

$$p_{+-}^h(x_j) = \frac{q_+^h(x_j) - q_+^h(x_{j-1})}{h} = \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} =: \delta_x^2 U_j,$$

(3.11)

$$p_{++}^h(x_j) = \frac{q_+^h(x_{j+1}) - q_+^h(x_j)}{h} = \frac{U_j - 2U_{j+1} + U_{j+2}}{h^2} =: \delta_x^2 U_{j+1},$$

(3.12)

for $j = 3, 4, \ldots, J-2$. Thus, in order to define boundary values for u^h, q_-^h, and q_+^h, we need to define ghost values U_{-1}, U_0, U_{J+1}, and U_{J+2} that are equivalent to extending the solution u to the outside of the domain Ω. Below we describe a natural way to do such an extension that is consistent with the interpretation of the auxiliary variables.

From the Dirichlet boundary data for u, a natural choice is that $U_0 = u_a$ and $U_{J+1} = u_b$. This is equivalent to assuming

$$u^h(a-) = u_a, \quad u^h(b+) = u_b.$$

(3.13)

In other words, we extend the boundary data for u away from the boundary over an interval of length h. Due to the inherent discontinuities of the piecewise constant functions, $u^h(a+)$ and $u^h(b-)$ are treated as unknowns. Otherwise, the boundary data would be extended into the interior of the domain over an interval of length h.

From (3.7) and (3.8) we see that $q_+^h(x_j) = q_-^h(x_{j+1})$ and $q_-^h(x_j) = q_+^h(x_{j-1})$ in the interior of the domain. Extending this relationship to the boundary yields

$$q_+^h(a-) = q_-^h(a+), \quad q_-^h(b+) = q_+^h(b-),$$

(3.14)

where both $q_-^h(a+)$ and $q_+^h(b-)$ are treated as unknowns in the above LDG formulation.

Finally, we need to define values for $q_-^h(a-)$ and $q_+^h(b+)$. Using [10] as a guide, we are led to choosing

$$q_-^h(a-) = q_-^h(a+), \quad q_+^h(b+) = q_+^h(b-).$$

(3.15)

We note that this is consistent with discretizing the auxiliary boundary conditions

$(q-h)x(a)=(q+h)x(b)=0.$

(3.16)

In order words, we require that $q-h$ and $q+h$ are constant across the boundary. Using ghost values, the above requirements are equivalent to imposing the constraints

$U_{-1}=2u_a-U_1, U_{J+2}=2u_b-U_J.$

Remark 3.1

From the imposed boundary conditions we can see that the relationship $p^-+h=p^+-h$ has been extended to the boundaries. Thus, using the ghost values defined above and substituting the equations (3.7)–(3.12) into (3.2), we successfully recover the convergent finite difference method defined in [*12*] for the grid function U. Hence, for $r=0$, the convergence of the proposed LDG method is obtained. Heuristically, using higher order elements should increase the rate and/or accuracy of convergence.

3.3 Whole Domain LDG Formulation

Using the above element-wise LDG formulation (3.3) and (3.4), and substituting the boundary numerical flux values from Sect. 3.2, we get the following whole domain LDG discretization of (1.6)–(1.11):

$(q_{\pm h},\phi_{\pm h})_{T_h}+a_{\pm}(u_h,\phi_{\pm h})=f_{\pm}(\phi_{\pm h}),\forall\phi_{\pm h}\in V_h,$

(3.17)

$(p_{\mu\nu h},\psi_{\mu\nu h})_{T_h}+b_{\mu\nu}(q^-h,q^+h;\psi_{\mu\nu h})=0,\forall\psi_{\mu\nu h}\in V_h,\mu,\nu=+,-,$

(3.18)

where

$a_-(v,\varphi) a_+(v,\varphi) b_{--}(v^-,v^+;\varphi) b_{++}(v^-,v^+;\varphi) b_{-+}(v^-,v^+;\varphi) b_{+-}(v^-,v^+;\varphi)=(v,\varphi_x)_{T_h}-(1-\kappa_r)v(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v(x_j^-)[\varphi(x_j)],=(v,\varphi_x)_{T_h}+(1-\kappa_r)v(a^+)\varphi(a^+)-\sum_{j=1}^{J-1}v(x_j^+)[\varphi(x_j)],=(v^-,\varphi_x)_{T_h}+v^-(a^+)\varphi(a^+)-v^-(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^-(x_j^-)[\varphi(x_j)],=(v^+,\varphi_x)_{T_h}+v^+(a^+)\varphi(a^+)-v^+(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^+(x_j^+)[\varphi(x_j)],=(v^-,\varphi_x)_{T_h}+v^-(a^+)\varphi(a^+)-(1-\kappa_r)v^+(b^-)\varphi(b^-)-\kappa_r v^-(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^-(x_j^+)[\varphi(x_j)],=(v^+,\varphi_x)_{T_h}+(1-\kappa_r)v^-(a^+)\varphi(a^+)+\kappa_r v^+(a^+)\varphi(a^+)-v^+(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^+(x_j^-)[\varphi(x_j)],$

and

$f_-(\phi) f_+(\phi)=\kappa_r u_b\phi(b^-)-u_a\phi(a^+),=u_b\phi(b^-)-\kappa_r u_a\phi(a^+),$

for

$\kappa_r=\begin{cases}0\\1\end{cases}\text{if } r=0,\text{otherwise.}$

(3.19)

In summary, our nonstandard LDG methods for the fully nonlinear Dirichlet problem (1.1) and (3.1) are defined as seeking $(u_h, q_h^-, q_h^+, p_h^{--}, p_h^{-+}, p_h^{+-}, p_h^{++}) \in (V_h)^7$ such that (3.2), (3.17), and (3.18) hold.

We conclude the section with a few remarks.

Remark 3.2

(a)

Looking backwards, (3.17) and (3.18) provide a proper interpretation for each of $q_h^\pm$ and $p_h^{\mu\nu}$ for $\mu, \nu = +, -$, for a given function u_h. Each $q_h^\pm$ defines a discrete derivative for u_h and each $p_h^{\mu\nu}$ defines a discrete second-order derivative for u_h. The functions q_h^- and q_h^+ should be very close to each other if u_x exists. Similarly, the functions $p_h^{--}, p_h^{-+}, p_h^{+-}$, and p_h^{++} should be very close to each other if u_{xx} exists. However, their discrepancies are expected to be large if u_x or u_{xx}, respectively, do not exist. The auxiliary functions $q_h^\pm$ defined by (3.17) and the auxiliary functions $p_h^{\mu\nu}$ defined by (3.18) can be regarded as high order extensions of their lower order finite difference counterparts defined in [12].

(b)

It is easy to check that the linear equations defined by (3.17)–(3.18) are linearly independent.

(c)

Notice that (3.2), (3.17), and (3.18) form a nonlinear system of equations, with the nonlinearity only appearing in a_0. Thus, a nonlinear solver is necessary in implementing the above scheme. In Sect. 5, an iterative method is used with initial guess given by the linear interpolant of the boundary data. Since a good initial guess is essential for most nonlinear solvers to converge, another possibility is to first linearize the nonlinear operator and solve the resulting linear system first. However, we show in our numerical tests that the simple initial guess works well in many cases. We suspect that the g-monotonicity of $\hat{F}$ enlarges the domain of "good" initial guesses over which the iterative method converges.

4 Formulation of Fully Discrete LDG Methods for Parabolic Problems

The goal of this section is to extend the LDG methods for elliptic problems to solving the initial-boundary value problem (1.2) using the method of lines. Let the initial condition be given by

$$u(x,0)=u0(x),\forall x\in\Omega,$$

(4.1)

and the boundary conditions be given by

$$u(a,t)=ua(t),u(b,t)=ub(t),\forall t\in(0,T].$$

(4.2)

We shall consider both the implicit trapezoidal rule and the (explicit) fourth order classical Runge–Kutta method (i.e., RK4) for the time-discretization. In practice, the time-discretization scheme should be chosen to match the order of the spatial discretization. Thus, when using a piecewise constant element, a sufficient choice for the time discretization would be forward or backward Euler. However, when using higher order elements, a higher order scheme such as RK4 should be chosen.

We first formulate the semi-discrete in space discretization for Eq. (1.2), which is a straightforward adaptation of the one described in Sect. 3. Let $\phi\in Vh$. Replacing the PDE operator with a numerical operator in (1.2), and using the LDG framework of Sect. 3, we obtain the semi-discrete equation

$$((uh)t,\phi h)Th=-(F\hat{}(p--h,p-+h,p+-h,p++h,q-h,q+h,uh,\cdot,t),\phi h)Th,\forall\phi h\in Vh,$$

(4.3)

where, given uh at time t, corresponding values for $q\pm h$ and $p\mu vh$ can be found using the methodology below.

We now describe a full discretization procedure for (1.2) by applying an ODE solver to the semi-discrete equations given in (4.3). For a fixed integer $M>0$, let $\Delta t=TM$ and $tk:=k\Delta t$ for $k\in(0,M]$. Notationally, $unh(x)\in Vh$ will be an approximation for $u(x,tn)$ for $n=0,1,\ldots,M$. We define the initial value $u0h$ to be the $L2$-projection of $u0$, namely,

$$(u0h,\phi h)Th=(u0,\phi h)Th,\forall\phi h\in Vh.$$

(4.4)

Next, we introduce several "one-sided" discrete differential operators, which will be used to define explicit time-stepping schemes and to define the auxiliary variables at time $t=0$ in implicit time-stepping schemes. We first define two "one-sided" first-order discrete derivatives $Q-,khv,Q+,khv\in Vh$ for a given function $v(\cdot,tk)\in H1(Th)$ by

$(Q^{-,k}_h v, \phi_h)_{T_h} = \kappa_r u_b(t_k)\phi_h(b^-) - u_a(t_k)\phi_h(a^+) + (1-\kappa_r)v(b^-)\phi_h(b^-) - (v,(\phi_h)_x)_{T_h} + \sum_{j=1}^{J-1} v(x_j^-)[\phi_h(x_j)], \forall \phi_h \in V_h,$

(4.5)

$(Q^{+,k}_h v, \phi_h)_{T_h} = u_b(t_k)\phi_h(b^-) - \kappa_r u_a(t_k)\phi_h(a^+) - (1-\kappa_r)v(a^+)\phi_h(a^+) - (v,(\phi_h)_x)_{T_h} + \sum_{j=1}^{J-1} v(x_j^+)[\phi_h(x_j)], \forall \phi_h \in V_h.$

(4.6)

The above definitions are inspired by (3.17). The super-index k on $Q^{\pm,k}_h$ indicates that the definitions are t-dependent because of the boundary terms.

We also define four discrete "one-sided" second order discrete derivatives $P^{\mu\nu,k}_h v \in V_h, \mu,\nu=+,-,$ at time t_k by

$(P^{--,k}_h v, \psi_h)_{T_h} = Q^{-,k}_h v(b^-)\psi_h(b^-) - Q^{-,k}_h v(a^+)\psi_h(a^+) - (Q^{-,k}_h v,(\psi_h)_x)_{T_h} + \sum_{j=1}^{J-1} Q^{-,k}_h v(x_j^-)[\psi_h(x_j)], \forall \psi_h \in V_h,$

(4.7)

$(P^{++,k}_h v, \psi_h)_{T_h} = Q^{+,k}_h v(b^-)\psi_h(b^-) - Q^{+,k}_h v(a^+)\psi_h(a^+) - (Q^{+,k}_h v,(\psi_h)_x)_{T_h} + \sum_{j=1}^{J-1} Q^{+,k}_h v(x_j^+)[\psi_h(x_j)], \forall \psi_h \in V_h,$

(4.8)

and

$(P^{-+,k}_h v, \psi_h)_{T_h} = (1-\kappa_r)Q^{+,k}_h v(b^-)\psi_h(b^-) + \kappa_r Q^{-,k}_h v(b^-)\psi_h(b^-) - Q^{-,k}_h v(a^+)\psi_h(a^+) - (Q^{-,k}_h v,(\psi_h)_x)_{T_h} + \sum_{j=1}^{J-1} Q^{-,k}_h v(x_j^+)[\psi_h(x_j)], \forall \psi_h \in V_h,$

(4.9)

$(P^{+-,k}_h v, \psi_h)_{T_h} = Q^{+,k}_h v(b^-)\psi_h(b^-) - (1-\kappa_r)Q^{-,k}_h v(a^+)\psi_h(a^+) - \kappa_r Q^{+,k}_h v(a^+)\psi_h(a^+) - (Q^{+,k}_h v,(\psi_h)_x)_{T_h} + \sum_{j=1}^{J-1} Q^{+,k}_h v(x_j^-)[\psi_h(x_j)], \forall \psi_h \in V_h,$

(4.10)

where κ_r is defined by (3.19). The above four definitions are motivated by (3.18).

Lastly, to simplify the presentation, we introduce the operator notation

$\hat{F}^k[v] := \hat{F}(P^{--,k}_h v, P^{-+,k}_h v, P^{+-,k}_h v, P^{++,k}_h v, Q^{-,k}_h v, Q^{+,k}_h v, v, x, t_k).$

(4.11)

Using the new notation, the semi-discrete equation can be rewritten compactly as

$((u_h)_t(\cdot,t_k),\phi_h)_{T_h}=-(\hat{F}^k[u_h(\cdot,t_k)],\phi_h)_{T_h},\forall\phi_h\in V_h,k\in(0,M].$

(4.12)

4.1 The Fourth Order Classical Runge–Kutta Method

A straightforward application of the fourth order classical Runge–Kutta (RK4) method to (4.12) yields

$(u_h^n,\phi_h)_{T_h}=(u_h^{n-1},\phi_h)_{T_h}+\frac{1}{6}(\xi_1+2\xi_2+2\xi_3+\xi_4,\phi_h)_{T_h},\forall\phi\in V_h,n=1,2,\ldots,M,$

where

$(\xi_1,\phi_h)_{T_h}(\xi_2,\phi_h)_{T_h}(\xi_3,\phi_h)_{T_h}(\xi_4,\phi_h)_{T_h}=-\Delta t(\hat{F}^{n-1}[u_h^{n-1}],\phi_h)_{T_h},=-\Delta t(\hat{F}^{n-\frac{1}{2}}[u_h^{n-1}+\frac{1}{2}\xi_1],\phi_h)_{T_h},=-\Delta t(\hat{F}^{n-\frac{1}{2}}[u_h^{n-1}+\frac{1}{2}\xi_2],\phi_h)_{T_h},=-\Delta t(\hat{F}^n[u_h^{n-1}+\xi_3],\phi_h)_{T_h}.$

Notice that in the above explicit time-stepping scheme, the function u_h^n is defined as an L^2-projection of the source data based on u_h^{n-1}. However, the boundary conditions are not enforced in the definition for u_h^n. To take care of the boundary conditions, we choose to enforce them weakly, which requires the introduction of a modified L^2-projection. Specifically, for any $v\in L^2(\Omega)$, we recall that the standard L^2-projection $P_h v\in V_h$ of v is defined by

$(P_h v,\phi_h)_{T_h}=(v,\phi_h)_{T_h},\forall\phi_h\in V_h.$

(4.13)

For any $v\in C^0(T_h)$, we introduce a modified L^2-projection $P_{kh}:L^2(\Omega)\cap C^0(T_h)\rightarrow V_h$ at time $t_k\in(0,T]$ by

$(P_{kh}v,\phi_h)_{T_h}+\frac{1}{h}\sqrt{}(P_{kh}v(a)\phi_h(a)+P_{kh}v(b)\phi_h(b))=(v,\phi_h)_{T_h}+\frac{1}{h}\sqrt{}(u_a(t_k)\phi_h(a+)+u_b(t_k)\phi_h(b-)),\forall\phi_h\in V_h.$

(4.14)

Clearly, the boundary conditions (4.2) are weakly enforced in (4.14) via a penalty technique, an idea which dates back to Nitsche [20].

Using the above discrete differential operators $Q_{\pm,kh}$ and $P_{\mu v,kh}$ for $\mu,v=+,-$, the projection operators $P_{kh}$and P_h, and using the notation given in (4.11), our fully-discrete RK4 method for the initial-boundary value problem (1.2), (4.2), and (4.1) is defined as follows: for $n=1,2,\ldots,M$,

$u_h^n=P_{nh}(u_h^{n-1}+\frac{1}{6}(\xi_{n-1}^1+2\xi_{n-1}^2+2\xi_{n-1}^3+\xi_{n-1}^4)),$

(4.15)

$\xi_{n-1}^1=-\Delta t P_h\hat{F}^{n-1}[u_h^{n-1}],$

(4.16)

$$\xi_{n-1}^2 = -\Delta t P_h \hat{F}^{n-\frac{1}{2}}[u_h^{n-1} + \frac{1}{2}\xi_{n-1}^1],$$

(4.17)

$$\xi_{n-1}^3 = -\Delta t P_h \hat{F}^{n-\frac{1}{2}}[u_h^{n-1} + \frac{1}{2}\xi_{n-1}^2],$$

(4.18)

$$\xi_{n-1}^4 = -\Delta t P_h \hat{F}^{n}[u_h^{n-1} + \xi_{n-1}^3],$$

(4.19)

$$u_h^0 = P_h u^0.$$

(4.20)

We remark that it is easy to verify that the value ξ_{n-1}^4 actually already takes into account the boundary conditions at time t_n because of the evaluation calls $Q^{-,n}_h$ and $Q^{+,n}_h$, which in turn have the boundary conditions built-in. Thus, in the above formulation, the boundary condition enforcement can actually be successfully relaxed by replacing P_h^n with P_h in (4.15). However, for other explicit methods a weak boundary condition enforcement method such as the above modified L^2-projection is necessary, especially if the boundary conditions are not consistent with the initial condition. For example, in the forward Euler method, defined by

$$u_h^n = P_h^n(u_h^{n-1} - \Delta t \hat{F}^{n-1}[u_h^{n-1}]),$$

(4.21)

we can see that the approximation at time t_n relies upon the modified L^2-projection in order to see the Dirichlet boundary condition at the current time.

4.2 The Trapezoidal Method

Applying the trapezoidal rule to (4.12), we obtain

$$(u_h^n + \frac{\Delta t}{2}\hat{F}^n[u_h^n], \phi_h)_{T_h} = (u_h^{n-1} - \frac{\Delta t}{2}\hat{F}^{n-1}[u_h^{n-1}], \phi_h)_{T_h}, \forall \phi_h \in V_h,$$

and $n = 1, 2, \ldots, M$. Thus, using the trapezoidal rule to discretize (4.12), and using the implicit equalities

$$q^{-,n}_h = Q^{-,n}_h u_h^n, q^{+,n}_h = Q^{+,n}_h u_h^n,$$

and

$$p^{--,n}_h = P^{--,n}_h u_h^n, p^{-+,n}_h = P^{-+,n}_h u_h^n, p^{+-,n}_h = P^{+-,n}_h u_h^n, p^{++,n}_h = P^{++,n}_h u_h^n,$$

for $n=1,2,\ldots,M$, the fully discrete trapezoidal LDG method for approximating solutions to (1.2), (4.1), and (4.2) is defined by seeking $(u_{nh},q^-,{nh},q^+,{nh},p^{--},{nh},p^{-+},{nh},p^{+-},{nh}, p^{++},{nh})\in(V_h)^7$ such that

$$(u_{nh}+\tfrac{\Delta t}{2}\hat{F}^n[u_{nh}],\phi_{0h})_{Th}=(u_{n-1h}-\tfrac{\Delta t}{2}\hat{F}^{n-1}[u_{n-1h}],\phi_{0h})_{Th},\forall\phi_{0h}\in V_h,$$

(4.22)

$$(q^\pm,{nh},\phi^\pm h)_{Th}+\hat{a}^\pm(u_{nh},\phi^\pm h)=g^\pm(t_n,\phi^\pm h),\forall\phi^\pm h\in V_h,$$

(4.23)

$$(p^{\mu v},{nh},\psi^{\mu v}h)_{Th}+\hat{b}^{\mu v}(q^-,{nh},q^+,{nh};\psi^{\mu v}h)=0,\forall\psi^{\mu v}h\in V_h,\mu,v=+,-,$$

(4.24)

where $u_{0h}=P_h u_0,q^\pm,{0h}=Q^\pm,{0h}u_{0h}, p^{\mu v},{0h}=P^{\mu v},{0h}u_{0h}$ for $\mu,v=+,-$, and

$$\hat{a}^-(v_n,\varphi)\hat{a}^+(v_n,\varphi)\hat{b}^{--}(v^-,n,v^+,n;\varphi)\hat{b}^{++}(v^-,n,v^+,n;\varphi)\hat{b}^{-+}(v^-,n,v^+,n;\varphi)\hat{b}^{+-}(v^-,n,v^+,n;\varphi)g^-(t_n,\phi)g^+(t_n,\phi)=(v_n,\varphi_x)_{Th}-(1-\kappa_r)v_n(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v_n(x_{-j})[\varphi(x_j)],=(v_n,\varphi_x)_{Th}+(1-\kappa_r)v_n(a^+)\varphi(a^+)-\sum_{j=1}^{J-1}v_n(x_{+j})[\varphi(x_j)],=(v^-,n,\varphi_x)_{Th}+v^-,n(a^+)\varphi(a^+)-v^-,n(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^-,n(x_{-j})[\varphi(x_j)],=(v^+,n,\varphi_x)_{Th}+v^+,n(a^+)\varphi(a^+)-v^+,n(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^+,n(x_{+j})[\varphi(x_j)],=(v^-,n,\varphi_x)_{Th}+v^-,n(a^+)\varphi(a^+)-(1-\kappa_r)v^+,n(b^-)\varphi(b^-)-\kappa_r v^-,n(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^-,n(x_{+j})[\varphi(x_j)],=(v^+,n,\varphi_x)_{Th}+(1-\kappa_r)v^-,n(a^+)\varphi(a^+)+\kappa_r v^+,n(a^+)\varphi(a^+)-v^+,n(b^-)\varphi(b^-)-\sum_{j=1}^{J-1}v^+,n(x_{-j})[\varphi(x_j)],=\kappa_r u_b(t_n)\phi(b^-)-u_a(t_n)\phi(a^+),=u_b(t_n)\phi(b^-)-\kappa_r u_a(t_n)\phi(a^+).$$

Again, κ_r is defined by (3.19). Notice that the above fully discrete formulation amounts to approximating a non-homogeneous fully nonlinear elliptic equation at each time step using the LDG method defined in Sect. 3.

5 Numerical Experiments

In this section, we present a series of numerical tests to demonstrate the utility of the proposed LDG methods for fully nonlinear PDEs of the types (1.1) and (1.2). In all of our tests we shall use uniform spatial meshes as well as uniform temporal meshes for the time-dependent problems. To solve the resulting nonlinear algebraic systems, we use the Matlab built-in nonlinear solver *fsolve*. For the elliptic problems we choose the initial guess as the linear interpolant of the boundary data. For parabolic problems, we let $u_{0h}=P_h u_0$, and then define all auxiliary variables by $q^\pm,{0h}=Q^\pm,{0h}u_{0h}$ and $p^{\mu v},{0h}=P^{\mu v},{0h}u_{0h}$ for $\mu,v=+,-$. Also, the initial guess for *fsolve* at the nth time step will be chosen as the computed solution at the previous time step when using implicit methods. The role of the numerical moment will be further explored in Sect. 5.3.

For our numerical tests, errors will be measured in the L^∞ norm and the L^2 norm, where the errors are measured at the current time step for the time-dependent problems. For both elliptic

and parabolic test problems where the error is dominated by the spatial discretization errors, it appears that the spatial errors are of optimal order $O(h^s)$ for most problems, where $s=\min\{r+1,k\}$ for the viscosity solution $u\in H^k(\Omega)$. However, for a couple of problems, we observe less than optimal rates of convergence. We note that the actual convergence rates have not yet been analyzed, and may also depend on the regularity of the operator F in addition to the regularity of u.

5.1 Elliptic Test Problems

We first present the results for four test problems of type (1.1). Both Monge–Ampère and Bellman types of equations will be tested.

Test 1 Consider the elliptic Monge–Ampère problem

$$-u_{xx}^2+1=\frac{1}{2}, \quad u(0)=0, u(1)=0, \quad 0<x<1.$$

It is easy to check that this problem has exactly two classical solutions:

$$u_+(x)=\tfrac{1}{2}x^2, \quad u_-(x)=-\tfrac{1}{2}x^2+x,$$

where u_+ is convex and u_- is concave. Note that u_+ is the unique viscosity solution which we want our numerical schemes to converge to. In Sect. 5.3 we shall give some insights about the selectiveness of our schemes.

We approximate the given problem for various degree elements ($r=0,1,2$) to see how the approximation converges with respect to h. Note, when $r=0,1$, the solution is not in the DG space V_h. The numerical results are shown in Fig. 1. We observe that the approximations using $r=2$ are almost exact for each mesh size. This is consistent with the fact $u_+\in V_h$ when $r=2$.
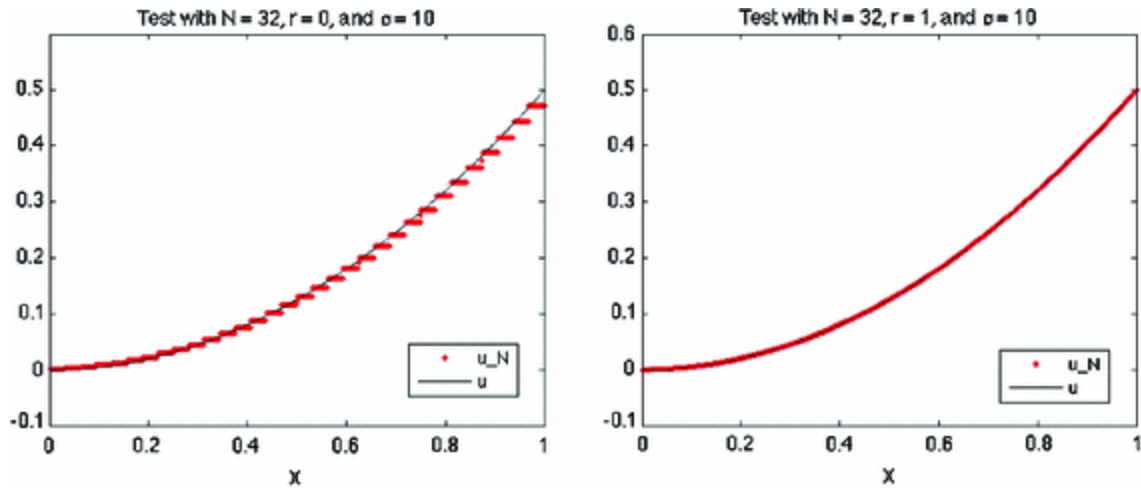
r	Norm	$h=1/4$	$h=1/8$		$h=1/16$		$h=1/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	7.1e-02	3.5e-02	1.02	1.4e-02	1.30	7.5e-03	0.92
	L^∞	1.3e-01	8.7e-02	0.57	5.3e-02	0.73	2.9e-02	0.87
1	L^2	1.6e-02	5.0e-03	1.67	1.3e-03	1.90	3.4e-04	1.95
	L^∞	2.2e-02	6.3e-03	1.84	1.6e-03	2.00	3.9e-04	2.00
2	L^2	3.1e-13	3.0e-13	0.03	3.0e-13	-0.01	3.1e-13	-0.01
	L^∞	7.4e-13	6.1e-13	0.28	6.7e-13	-0.14	7.1e-13	-0.08

Fig. 1 Test 1 with $\alpha=10$

Test 2 Consider the problem

$$-u3_{xx}+u_{xx}+S(x)3-S(x)=0,-1<x<1,u(-1)=-\sin(1)-8\cos(0.5)+9,u(1)=\sin(1)-8\cos(0.5)+9,$$

where

$$S(x)= \begin{cases} 2x|x|\cos(x2)-4x2\sin(x|x|)+2\cos(x2)+2,-4x2\sin(x|x|)+2\cos(x2)+2, & x\neq 0, \\ & x=0. \end{cases}$$

This problem has the exact solution $u(x)=\sin(x|x|)-8\cos(x2)+x2+8\in H2(-1,1)$. Note that this problem is not monotone decreasing in u_{xx}, and the exact solution is not twice differentiable at $x=0$. However, the derivative of F with respect to u_{xx} is uniformly bounded. The numerical results are shown in Fig. 2. As expected, we can see from the plot that the error appears largest around the point $x=0$, and both the accuracy and order of convergence improve as the order of the element increases. For finer meshes, we see the rates of convergence begin to deteriorate. Theoretically, we expect the rates of convergence to be bounded by two for high-order bases due to the lower regularity of the solution.
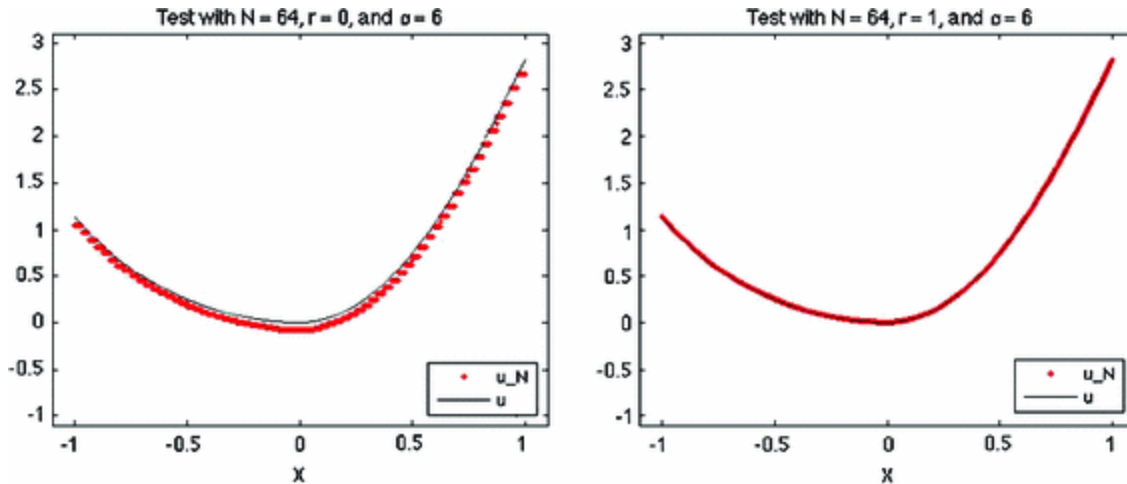
r	Norm	$h = 1/2$	$h = 1/4$		$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order	Error	Order
0	L^2	1.8	9.0e-01	1.04	4.3e-01	1.05	2.1e-01	1.04	1.0e-01	1.02
	L^∞	2.5	1.2	0.99	6.2e-01	1.00	3.1e-01	1.00	1.6e-01	1.00
1	L^2	2.9e-01	6.3e-02	2.20	1.9e-02	1.73	7.0e-03	1.44	2.8e-03	1.34
	L^∞	2.9e-01	6.4e-02	2.17	2.0e-02	1.66	7.6e-03	1.42	3.1e-03	1.30
2	L^2	5.7e-03	8.2e-04	2.80	1.3e-04	2.66	3.2e-05	2.03	9.1e-06	1.81
	L^∞	2.0e-02	3.1e-03	2.70	4.2e-04	2.87	5.5e-05	2.94	8.0e-06	2.77
3	L^2	8.8e-04	7.7e-05	3.51	3.0e-06	4.68	1.4e-07	4.42	1.0e-08	3.76
	L^∞	2.1e-03	1.4e-04	3.90	8.6e-06	4.01	5.6e-07	3.94	9.5e-08	2.57

Fig. 2 Test 2 with $\alpha=6$

Test 3 Consider the stationary Hamilton-Jacobi-Bellman problem with finite dimensional control set

$$\min_{\theta(x)\in\{1,2\}}\{-\theta u_{xx}+u_x-u+S(x)\}u(-1)=-1, u(1)=0, -1<x<1, =1,$$

where

$$S(x)=\{-12x2-4|x|3+x|x|3, 24x2-4|x|3+x|x|3, x<0 x\geq0.$$

This problem has the exact solution $u(x)=x|x|3\in H4(-1,1)$ corresponding to $\theta*(x)=1$ for $x<0$ and $\theta*(x)=2$ for $x\geq0$. Approximating the problem using various order elements, we have the following results recorded in Fig. 3. Due to the low regularity of the solution, we expect the rates of convergence to be bounded by four for high-order bases. We observe that the rates of convergence for $r=0,1,2$ appear to be optimal on average, while the rates of convergence for $r=3$ appear to be limited to three. However, we still see increased accuracy for $r=3$ when compared to $r=2$.
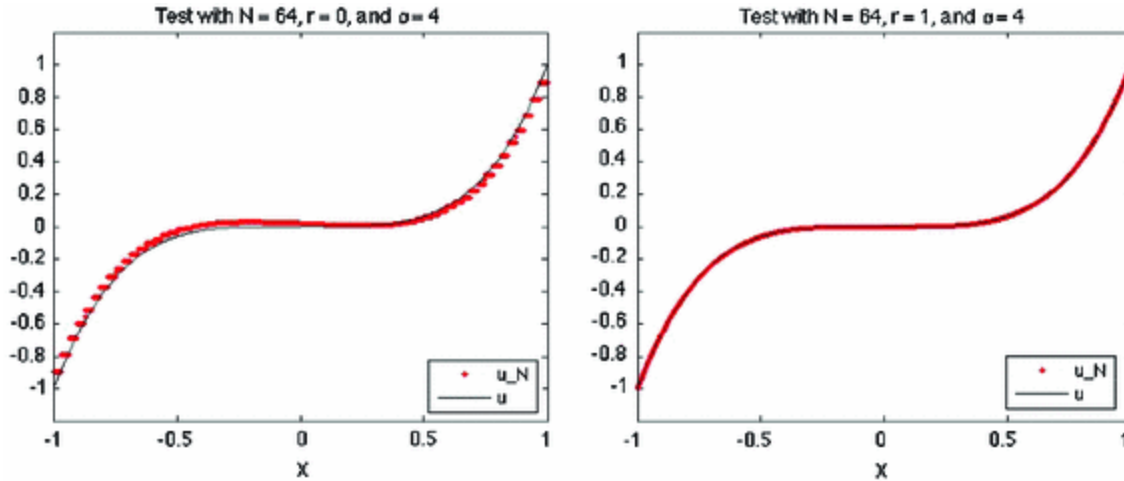
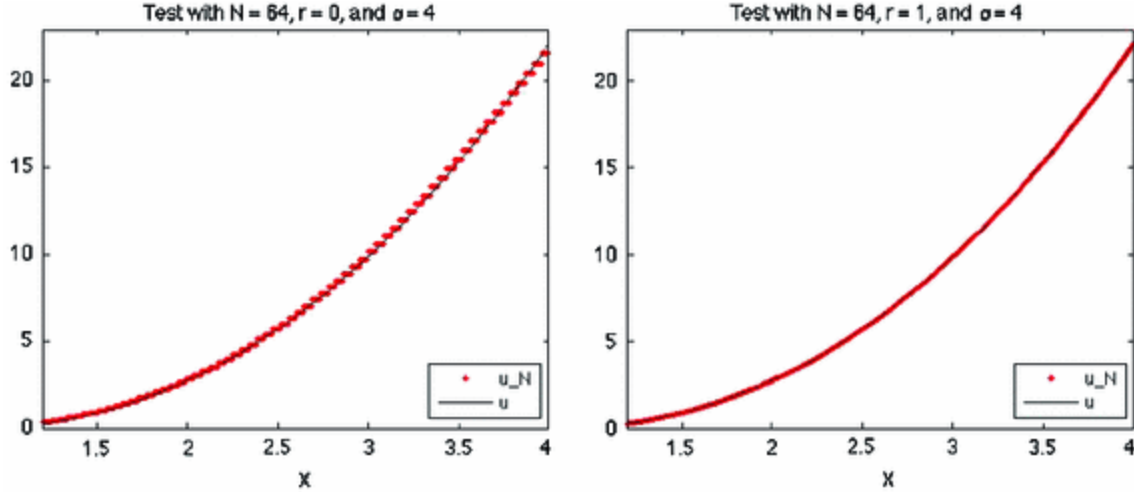| r	Norm	$h = 1/2$	$h = 1/4$		$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order	Error	Order
0	L^2	5.0e-01	3.4e-01	0.53	1.3e-01	1.37	6.1e-02	1.12	4.4e-02	0.49
	L^∞	8.5e-01	5.7e-01	0.57	3.5e-01	0.72	2.0e-01	0.79	1.1e-01	0.86
1	L^2	1.4e-01	4.3e-02	1.72	9.7e-03	2.16	2.7e-03	1.85	7.3e-04	1.87
	L^∞	3.6e-01	1.1e-01	1.74	3.0e-02	1.87	7.8e-03	1.94	2.0e-03	1.97
2	L^2	2.8e-02	3.2e-03	3.10	4.0e-04	3.00	5.1e-05	2.99	6.4e-06	2.99
	L^∞	4.0e-02	5.5e-03	2.84	7.3e-04	2.93	9.3e-05	2.97	1.2e-05	2.98
3	L^2	9.4e-03	1.3e-03	2.91	1.6e-04	3.01	1.9e-05	3.01	2.4e-06	3.01
	L^∞	1.1e-02	1.5e-03	2.91	1.9e-04	3.02	2.3e-05	3.01	2.9e-06	3.01

Fig. 3 Test 3 with $\alpha=4$

Test 4 Consider the stationary Hamilton-Jacobi-Bellman problem with infinite dimensional control set

$$\inf_{0\leq\theta(x)\leq1}\{-\theta u_{xx}+\theta^2 x^2 u_x+\tfrac{1}{x}u+S(x)\} \quad u(1.2)=1.44\ln1.2, \quad u(4)=0, \quad 1.2<x<4, \quad =16\ln4,$$

where

$$S(x)=4\ln(x)^2+12\ln(x)+9-8x^4\ln(x)^2-4x^4\ln(x)4x^3[2\ln(x)+1].$$

This problem has the exact solution $u(x)=x^2\ln x$ corresponding to the control function $\theta*(x)=2\ln(x)+\tfrac{3}{2}x^3[2\ln(x)+1]$. Approximating the problem using various order elements, we obtain the results recorded in Fig. 4.

Fig. 4 Test 4 with $\alpha=4$

The figure contains two plots and a data table.

Test with N = 64, r = 0, and σ = 4 (left plot) and **Test with N = 64, r = 1, and σ = 4** (right plot), both with legend entries u_N and u.

| r	Norm	$h=2.8/4$	$h=2.8/8$		$h=2.8/16$		$h=2.8/32$		$h=2.8/64$	
		Error	Error	Order	Error	Order	Error	Order	Error	Order
0	L^2	5.2	3.3	0.67	1.5	1.08	6.1e-01	1.34	2.6e-01	1.25
	L^∞	5.7	3.6	0.65	1.9	0.91	1.1	0.84	5.7e-01	0.91
1	L^2	2.6e-01	8.6e-02	1.60	2.6e-02	1.72	7.4e-03	1.83	2.0e-03	1.90
	L^∞	3.3e-01	1.1e-01	1.56	3.5e-02	1.67	1.1e-02	1.71	3.4e-03	1.65
2	L^2	2.6e-03	3.9e-04	2.77	6.6e-05	2.55	1.4e-05	2.26	3.2e-06	2.09
	L^∞	7.3e-03	1.0e-03	2.85	1.4e-04	2.91	1.9e-05	2.81	4.1e-06	2.25
3	L^2	6.4e-05	4.2e-06	3.93	3.1e-07	3.75	1.2e-07	1.35	1.2e-07	0.08
	L^∞	2.7e-04	2.1e-05	3.72	1.4e-06	3.84	8.7e-07	0.72	8.8e-07	-0.01

5.2 Parabolic Test Problems

We now implement the proposed fully discrete RK4 and trapezoidal LDG methods for approximating fully nonlinear parabolic equations of the form (1.2). While the above formulation makes no attempt to formally quantify a CFL condition for the RK4 method, for the tests we assume a CFL constraint of the form $\Delta t = \kappa_t h^2$, and note that the constant κ_t appears to decrease as the order of the element increases. Below we implement both the RK4 method and the trapezoidal method for each test problem. However, we make no attempt to classify and compare the efficiency of the two methods. Instead, we focus on testing and demonstrating the usability of both fully discrete schemes and their promising potentials. For explicit scheme tests, we record the parameter κ_t, and for implicit scheme tests, we record the time step size Δt. Note that the row $0*$ in the figures corresponding to the RK4 method refers to elements with $r=0$ that use the standard L^2 projection operator in (4.15).

Test 5 Let $\Omega=(0,1), u_a(t)=t^4, u_b=12+t^4$, and $u_0(x)=12x^2$. We consider the problem (1.2), (4.1), and (4.2) with

$$F(u_{xx}, u_x, u, t, x) = -u_{xx}u + 12x^2 + t^4 - 4t^3 + 1.$$

It is easy to verify that this problem has a unique classical solution $u(x,t)=0.5x^2+t^4+1$. Notice that the PDE is actually quasi-linear, but does provide a measure of the effectiveness of the implementation. The numerical results for RK4 are presented in Fig. 5 and the results for the

trapezoidal method are shown in Fig. 6. As expected, RK4 appears to recover the exact solution when r=2.
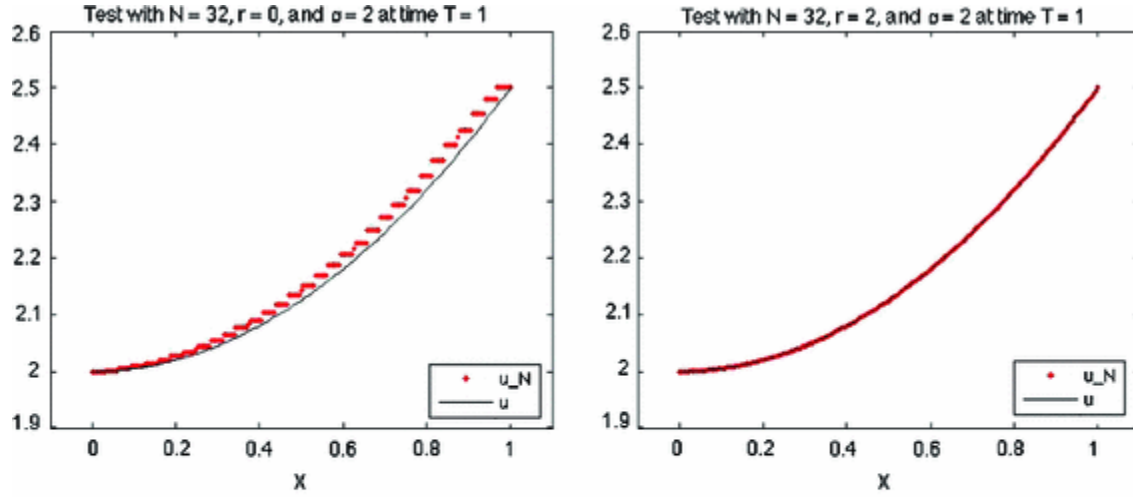
r	Norm	$h = 1/4$	$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	9.9e-02	6.4e-02	0.64	3.6e-02	0.81	1.9e-02	0.92
	L^∞	2.2e-01	1.4e-01	0.64	8.0e-02	0.81	4.3e-02	0.89
1	L^2	5.7e-03	1.5e-03	1.98	3.7e-04	1.99	9.2e-05	1.99
	L^∞	8.0e-03	2.0e-03	1.99	5.1e-04	1.99	1.3e-04	1.99
2	L^2	2.4e-08	2.4e-08	0.00	2.4e-08	0.00	2.4e-08	-0.00
	L^∞	3.6e-08	3.7e-08	-0.03	3.7e-08	-0.01	3.7e-08	-0.01

Fig. 5 Test 5: Computed solutions at T=1 using κt=0.001,α=2
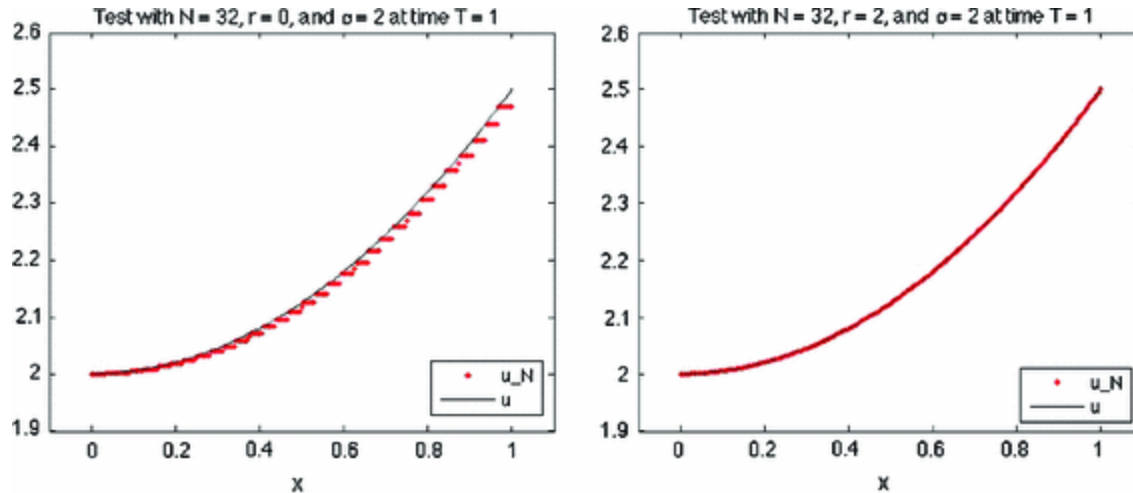
r	Norm	$h=1/4$	$h=1/8$		$h=1/16$		$h=1/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	5.9e-02	3.4e-02	0.78	1.9e-02	0.84	1.0e-02	0.91
	L^∞	1.9e-01	1.1e-01	0.79	5.9e-02	0.90	3.0e-02	0.95
1	L^2	5.0e-03	1.4e-03	1.86	3.6e-04	1.94	9.1e-05	1.97
	L^∞	1.1e-02	2.8e-03	2.00	7.1e-04	2.00	1.8e-04	2.00
2	L^2	1.1e-07	1.1e-07	0.00	1.1e-07	0.00	1.1e-07	0.00
	L^∞	1.5e-07	1.6e-07	-0.04	1.6e-07	-0.01	1.6e-07	-0.00

**Fig. 6** Test 5: Computed solution at T=1 using Δt=0.001 and α=2

Test 6 Let Ω=(0,2),$ua(t)$=1,ub=$e2(t+1)$, and $u0(x)$=ex. We consider the problem (1.2), (4.1), and (4.2) with

$F(uxx,ux,u,t,x)$=$-ux\ln(uxx+1)+S(x,t)$,

and

$S(x,t)$=$e(t+1)x(x-(t+1)\ln((t+1)2e(t+1)x+1))$.

It is easy to verify that this problem has a unique classical solution $u(x,t)$=$e(t+1)x$. Notice that this problem is nonlinear in both uxx and ux. Furthermore, the exact solution u cannot be factored into the form $u(x,t)$=$G(t)Y(x)$ for some functions G and Y. Results for RK4 are given in Fig. 7, and results for the trapezoidal method are shown in Fig. 8. We note that RK4 was unstable without using the very restrictive values for κt recorded in Fig. 7. However, for RK4, we observe optimal rates of convergence in the spatial variable while the rates for the trapezoidal method appear to be limited by the lower rate of convergence for the time-stepping scheme.

Test with N = 32, r = 0, and σ = 4 at time T = 0.5

Test with N = 32, r = 3, and σ = 4 at time T = 0.5
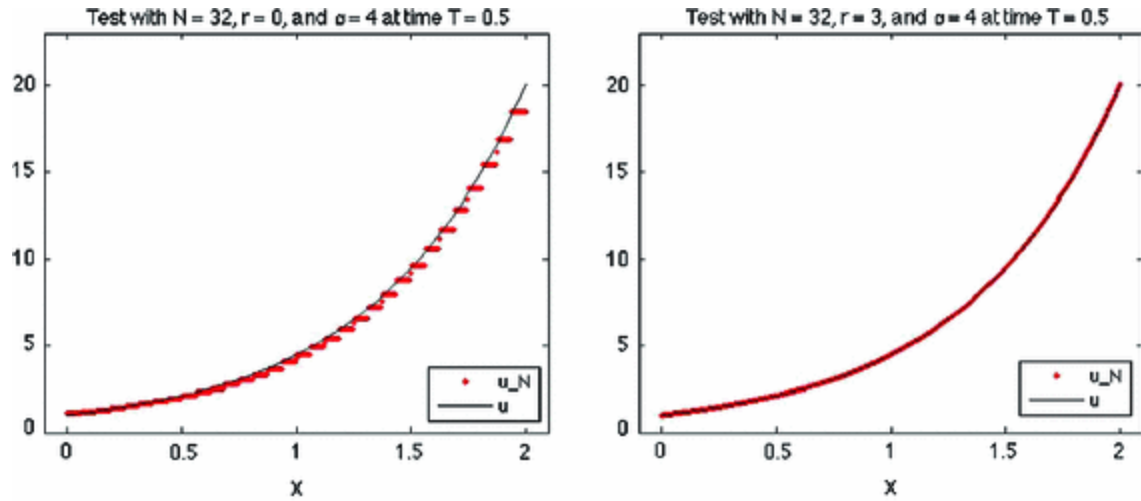
r	Norm	$h = 2/4$	$h = 2/8$		$h = 2/16$		$h = 2/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	6.9e+00	5.7e+00	0.28	4.1e+00	0.47	2.6e+00	0.65
	L^∞	1.0e+01	7.9e+00	0.40	5.6e+00	0.50	3.7e+00	0.60
0*	L^2	2.8e+00	1.5e+00	0.89	8.6e-01	0.82	5.1e-01	0.76
	L^∞	7.8e+00	5.0e+00	0.65	2.9e+00	0.76	1.6e+00	0.85
1	L^2	4.8e-01	1.2e-01	2.05	3.0e-02	1.93	8.2e-03	1.88
	L^∞	8.4e-01	2.3e-01	1.87	5.8e-02	1.99	1.5e-02	1.92
2	L^2	3.7e-02	7.8e-03	2.23	1.9e-03	2.03	4.8e-04	2.00
	L^∞	5.9e-02	1.0e-02	2.53	2.2e-03	2.19	5.2e-04	2.11
3	L^2	1.1e-03	7.4e-05	3.95	4.7e-06	3.99	3.01e-07	3.96
	L^∞	2.5e-03	1.6e-04	3.93	1.1e-05	3.86	7.66e-07	3.84

Fig. 7 Test 6: Computed solutions at
time T=0.5 using κt=0.005,0.001,0.0005,0.0001 for r=0,1,2,3, respectively, and α=4. *Left figure* uses the standard *L*2 projection operator

Fig. 8 Test 6: Computed solutions at time T=0.5 using Δt=0.005 and α=4
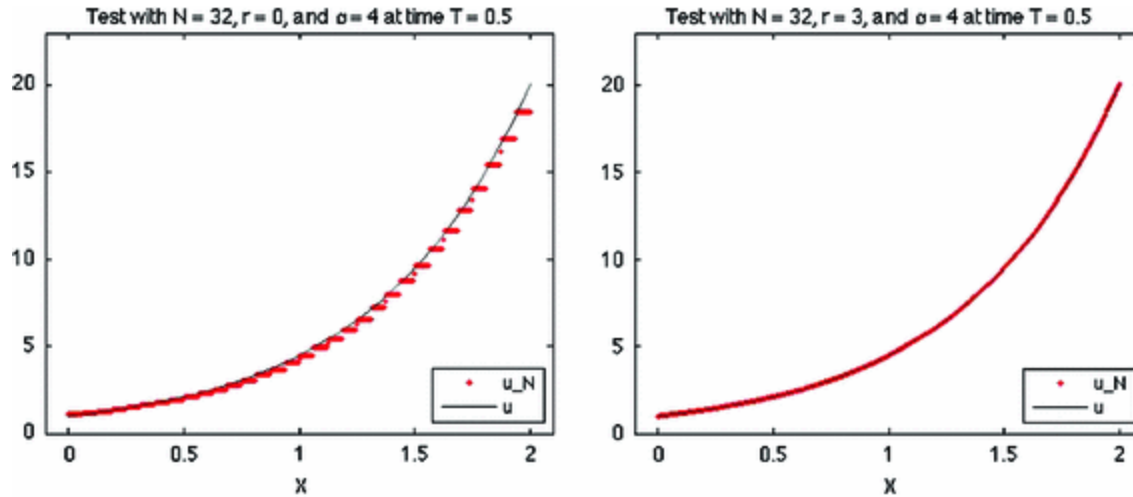
r	Norm	$h=2/4$	$h=2/8$		$h=2/16$		$h=2/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	2.8e+00	1.5e+00	0.89	8.6e-01	0.82	5.1e-01	0.77
	L^∞	7.8e+00	5.0e+00	0.65	2.9e+00	0.76	1.6e+00	0.85
1	L^2	3.8e-01	1.3e-01	1.62	4.5e-02	1.49	1.5e-02	1.60
	L^∞	1.3e+00	4.0e-01	1.74	1.1e-01	1.85	3.0e-02	1.91
2	L^2	2.7e-02	6.7e-03	2.04	1.9e-03	1.82	5.3e-04	1.85
	L^∞	1.0e-01	1.5e-02	2.77	2.1e-03	2.80	5.4e-04	1.99
3	L^2	1.1e-03	7.2e-05	3.89	1.3e-05	2.46	1.2e-05	0.09
	L^∞	5.5e-03	4.0e-04	3.76	2.7e-05	3.92	1.3e-05	1.05

Test 7 Let $\Omega=(0,2\pi)$, $u_a(t)=0$, $u_b=0$, and $u_0(x)=\sin(x)$. We consider the problem (1.2), (4.1), and (4.2) with

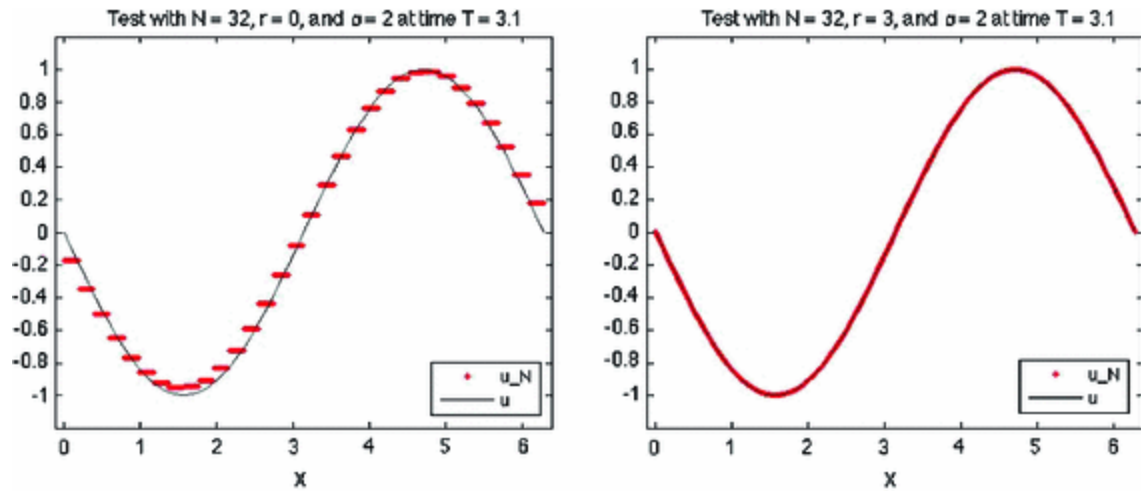$$F(u_{xx},u_x,u,t,x)=-\min_{\theta(t,x)\in\{1,2\}}\{A_\theta u_{xx}-c(x,t)\cos(t)\sin(x)-\sin(t)\sin(x)\},$$

where $A_1=1$, $A_2=12$, and

$$c(x,t)=\begin{cases} 1,\\ 12, \end{cases} \text{if } 0<t\leq\frac{\pi}{2} \text{ and } 0<x\leq\pi \text{ or } \frac{\pi}{2}<t\leq\pi \text{ and } \pi<x<2\pi, \text{otherwise.}$$

It is easy to check that this problem has a unique classical solution $u(x,t)=\cos(t)\sin(x)$. Notice that this problem has a finite dimensional control parameter set, and the optimal control is given by
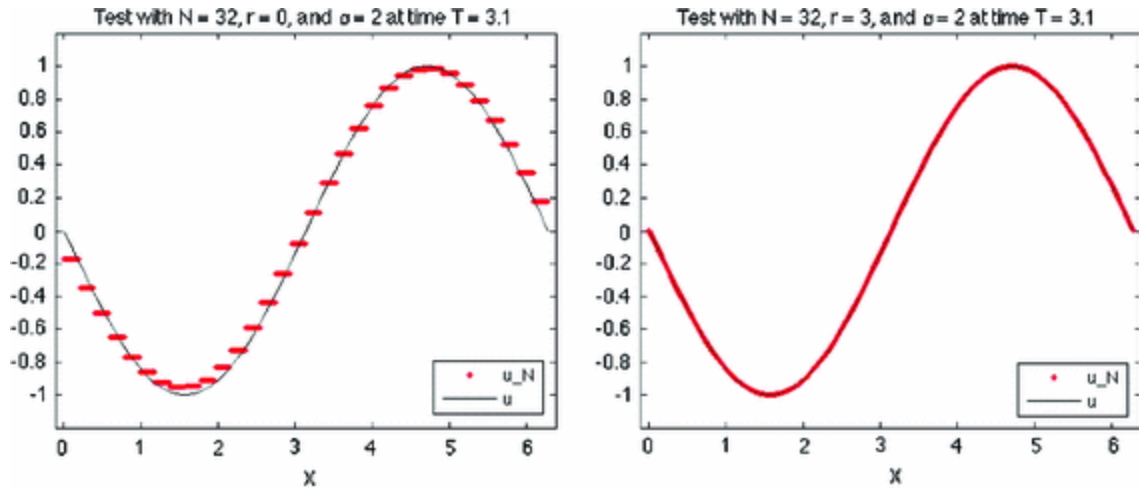
$$\theta*(t,x)=\begin{cases} 1,\\ 2, \end{cases} \text{if } c(x,t)=1, \text{if } c(x,t)=12.$$

The numerical results are reported in Fig. 9 for RK4 and in Fig. 10 for the trapezoidal method.

Fig. 9 Test 7: Computed solutions at
time T=3.10 using κt=0.05,0.005,0.001,0.0005 for r=0,1,2,3, respectively, and α=2. *Left plot* uses the standard $L2$ projection operator

Plot titles:
Test with N = 32, r = 0, and σ = 2 at time T = 3.1
Test with N = 32, r = 3, and σ = 2 at time T = 3.1

r	Norm	$h=\pi/2$ Error	$h=\pi/4$ Error	Order	$h=\pi/8$ Error	Order	$h=\pi/16$ Error	Order
0	L^2	1.5e+00	1.3e+00	0.24	7.1e-01	0.82	3.3e-01	1.12
	L^∞	9.6e-01	7.7e-01	0.31	4.9e-01	0.65	2.9e-01	0.78
0*	L^2	1.2e+00	6.9e-01	0.78	3.0e-01	1.19	1.4e-01	1.16
	L^∞	9.2e-01	5.8e-01	0.67	3.2e-01	0.85	1.8e-01	0.83
1	L^2	2.7e-01	7.6e-02	1.81	2.0e-02	1.94	5.0e-03	1.99
	L^∞	1.9e-01	6.6e-02	1.52	1.7e-02	1.97	4.2e-03	2.00
2	L^2	7.2e-02	1.8e-02	1.98	4.5e-03	2.02	1.1e-03	2.01
	L^∞	6.8e-02	1.8e-02	1.93	4.3e-03	2.04	1.1e-03	2.03
3	L^2	8.3e-03	5.7e-04	3.87	3.6e-05	3.97	2.2e-06	4.02
	L^∞	7.6e-03	5.1e-04	3.92	3.2e-05	4.00	1.9e-06	4.04

r	Norm	$h = \pi/2$	$h = \pi/4$		$h = \pi/8$		$h = \pi/16$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	1.2e+00	6.9e-01	0.78	3.0e-01	1.19	1.4e-01	1.16
	L^∞	9.2e-01	5.8e-01	0.67	3.2e-01	0.85	1.8e-01	0.83
1	L^2	2.3e-01	7.5e-02	1.63	2.0e-02	1.89	7.9e-03	1.36
	L^∞	2.1e-01	6.6e-02	1.66	1.7e-02	1.95	5.8e-03	1.56
2	L^2	6.9e-02	1.8e-02	1.93	4.7e-03	1.95	2.5e-03	0.90
	L^∞	6.5e-02	1.8e-02	1.87	4.5e-03	2.00	1.9e-03	1.21
3	L^2	8.1e-03	5.9e-04	3.79	1.1e-04	2.42	1.1e-04	0.03
	L^∞	7.5e-03	5.2e-04	3.87	7.6e-05	2.77	7.3e-05	0.06

**Fig. 10** Test 7: Computed solutions at time T=3.10 using Δt=0.031 and α=2

Test 8 Let Ω=(0,3),$ua(t)$=e−t,ub=8e−t, and $u0(x)$=|x−1|3. We consider the problem (1.2), (4.1), and (4.2) with

$F(uxx,ux,u,t,x)$=−inf−1≤$\theta(t,x)$≤1 {|x−1|uxx+θux}−|x−1|2(|x−1|−3)e−t,

It is easy to verify that the problem has the exact solution $u(t,x)$=|x−1|3e−t. Notice that the above operator is not elliptic for x=1. For each value of t, we have $u \in H3$(0,3). Also, this problem has a bang-bang type control with the optimal control given by
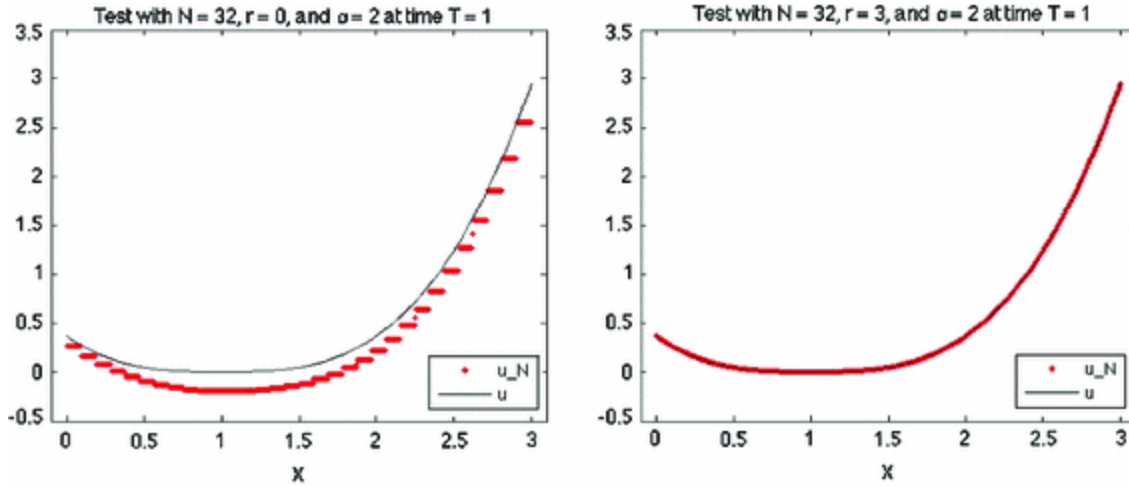
$\theta*(t,x)$={1,−1,if x<1,if x>1.

We can see from the results for the RK4 method in Fig. 11 and the results for the trapezoidal method in Fig. 12 that the spatial rates of convergence appear to be limited to two instead of the optimal rate of three for$r$≥2, while the accuracy appears to increase with respect to the element degree.
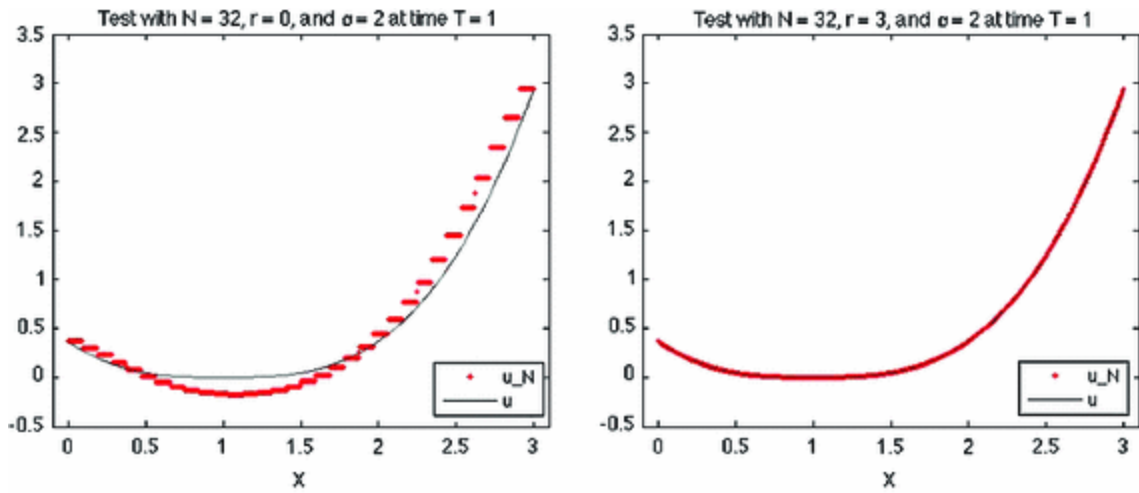
r	Norm	$h=3/4$	$h=3/8$		$h=3/16$		$h=3/32$	
		Error	Error	Order	Error	Order	Error	Order
0	L^2	2.1e+00	1.4e+00	0.51	5.3e-01	1.44	2.9e-01	0.88
	L^∞	2.1e+00	1.5e+00	0.44	8.5e-01	0.86	4.9e-01	0.81
0*	L^2	8.6e-01	8.8e-01	-0.02	5.9e-01	0.58	2.9e-01	1.00
	L^∞	1.8e+00	1.3e+00	0.53	7.3e-01	0.80	3.9e-01	0.92
1	L^2	2.0e-01	7.3e-02	1.45	1.3e-02	2.44	3.2e-03	2.06
	L^∞	3.9e-01	1.0e-01	1.92	2.7e-02	1.95	6.8e-03	1.98
2	L^2	1.1e-01	2.0e-02	2.49	5.2e-03	1.98	1.3e-03	2.02
	L^∞	1.1e-01	2.0e-02	2.42	5.2e-03	1.99	1.3e-03	2.00
3	L^2	3.0e-02	6.8e-03	2.13	1.7e-03	2.00	4.2e-04	2.03
	L^∞	2.9e-02	6.9e-03	2.08	1.7e-03	2.02	4.2e-04	2.02

Fig. 11 Test 8: Computed solutions at time $T=1$ using $\kappa t=0.05,0.005,0.001,0.0005$ for $r=0,1,2,3$, respectively, and $\alpha=2$

r	Norm	$h=3/4$ Error	$h=3/8$ Error	$h=3/8$ Order	$h=3/16$ Error	$h=3/16$ Order	$h=3/32$ Error	$h=3/32$ Order
0	L^2	8.6e-01	8.8e-01	-0.02	5.9e-01	0.58	2.9e-01	1.00
	L^∞	1.8e+00	1.3e+00	0.53	7.3e-01	0.80	3.9e-01	0.92
1	L^2	2.0e-01	7.3e-02	1.45	1.3e-02	2.44	3.2e-03	2.05
	L^∞	3.9e-01	1.0e-01	1.92	2.7e-02	1.95	6.8e-03	1.98
2	L^2	1.1e-01	2.0e-02	2.49	5.2e-03	1.98	1.3e-03	2.02
	L^∞	1.1e-01	2.0e-02	2.42	5.2e-03	1.99	1.3e-03	2.00
3	L^2	3.0e-02	6.8e-03	2.13	1.7e-03	2.00	4.2e-04	2.03
	L^∞	2.9e-02	6.9e-03	2.08	1.7e-03	2.02	4.2e-04	2.02

Fig. 12 Test 8: Computed solutions at time $T=1$ using $\Delta t=0.001$ and $\alpha=2$

5.3 The Role of the Numerical Moment

In this section, we focus on understanding the role of the numerical moment. We first give two ways to interpret the numerical moment, and then we explore the utility of the numerical moment. The role of the numerical moment can heuristically be understood as follows when the numerical moment is rewritten in the form

$$h2\alpha(p--h-p-+h-p+-h+p++hh2).$$

Letting $r=0$, we see that $p--h-p-+h-p+-h+p++hh2$ is an $O(h2)$ approximation to $uxxxx$. Then, we can see that the numerical moment acts as a centered difference approximation for $\Delta2u$ multiplied by a factor that tends to zero with rate $O(h2)$. Thus, at the PDE level, we are in essence approximating the fully nonlinear second order elliptic operator

$$F(uxx,ux,u,x)$$

by the quasilinear fourth order operator $F\tilde{}\rho$, where

$$F\tilde{}\rho(uxxxx,uxx,ux,u,x)=\rho uxxxx+F(uxx,ux,u,x).$$

In the limit as $\rho \to 0$, we heuristically expect that the solution of the fourth order problem converges to the unique viscosity solution of the second order problem. Using a convergent family of fourth order quasilinear PDEs to approximate a fully nonlinear second order PDE has previously been considered for PDEs such as the Monge–Ampère equation, the prescribed Gauss curvature equation, the infinity-Laplace equation, and linear second order equations of non-divergence form. The method is known as the vanishing moment method and serves as an analogue to the vanishing viscosity method for first order Hamilton-Jacobi equations where the PDE is perturbed by a second order derivative term. We refer the reader to [9, 11] for a detailed exposition.

We now express the numerical moment using jumps defined on the interior nodes. Observe, by equations (3.18),

$$(p^{--}h - p^{-+}h - p^{+-}h + p^{++}h, \varphi h)_{T_h} = -b^{--}(q^-h, q^+h; \varphi h) + b^{-+}(q^-h, q^+h; \varphi h) + b^{+-}(q^-h, q^+h; \varphi h) - b^{++}(q^-h, q^+h; \varphi h) = (1-\kappa r)(q^-h(a^+) - q^+h(a^+))\varphi h(a^+) + (1-\kappa r)(q^-h(b^-) - q^+h(b^-))\varphi h(b^-) + \sum_{j=1}^{J-1}([q^-h(x_j)] - [q^+h(x_j)])[\varphi h(x_j)]$$

for all $\varphi h \in V_h$, where κr is defined in (3.19). Thus, the Lax-Friedrichs-like numerical operator with a numerical moment amounts to the addition of nonstandard jump/stabilization terms to an $L2$ projection of the fully nonlinear PDE operator into V_h. The jump/stabilization terms penalize the differences in q^-h and q^+h. For comparison, the numerical moment using the IP-DG formulation in [16] takes the form

$$\sum_{j=1}^{J-1}[uh(x_j)][\varphi h(x_j)],$$

a standard jump/stabilization term. Due to only having a single approximation for ∇u in the IP-DG framework, a standard jump/stabilization term that corresponds to q^-h and q^+h would take the form

$$\sum_{j=1}^{J-1}\frac{1}{2}[q^-h(x_j) + q^+h(x_j)][\varphi h(x_j)],$$

where we penalize the sum of q^-h and q^+h. Therefore, the above nonstandard LDG formulation that uses two independent approximations for ∇u allows for the formulation of a new jump/stabilization term that is not analogous to penalty terms from the IP-DG literature. We also note that while the numerical moment can be rewritten using jumps, numerical experiments indicate generic solvers such as *fsolve* appear to succeed in finding a root more frequently when using the presented mixed formulation with the numerical moment implemented as a function of the various second-order derivative approximations.

For the remainder of the section, we focus on observed consequences of the numerical moment in our test problems. In particular, we observe that the proposed schemes using a numerical moment can eliminate the numerical artifacts that arise as consequences from using a standard discretization, and in certain cases when the numerical artifacts are not fully eliminated, the

algebraic system has enough structure to design solvers that emphasize the monotonicity in $p^-{+}h+p^+{-}h2$ and are consistent in searching for solutions at which the nonlinear PDE problem is elliptic.

We again consider the Monge–Ampère type problem from Test 1 in Sect. 5.1. The given problem has two classical PDE solutions $u+$ and $u-$. However, there exists infinitely many $C1$ functions that satisfy the PDE and boundary conditions almost everywhere, as seen by $\hat{\mu}$ in (5.1). These almost everywhere solutions will correspond to what we call numerical artifacts in that algebraic solutions for a given discretization may correspond to these functions. It is well known that using a standard discretization scheme for the Monge–Ampère problem can yield multiple solutions, many of which are numerical artifacts that do not correspond to PDE solutions [*11*]. For example, let $\mu \in H2(0,1) \backslash C2(0,1)$ be defined by

$$\hat{\mu}(x) = \begin{cases} 12x2+14x, & \text{for } x<0.5, \\ -12x2+54x-14, & \text{for } x\geq 0.5. \end{cases}$$

(5.1)

Furthermore, suppose $xj=0.5$ for some $j=2,3,\ldots,J-1$. Then, when using a standard central difference discretization, $\hat{\mu}$ corresponds to a numerical solution, yielding a numerical artifact.

We now consider our discretization that uses a numerical moment. When $\alpha=0$, we have no numerical moment. As a result, we have numerical artifacts as in the standard central difference discretization case. Suppose $r=0$. Then, for $\alpha \neq 0$, inspection of (3.9)–(3.12) yields the fact that $p^-{}h$ cannot jump from a value of 1 to a value of -1 when going across $xj=0.5$. Thus, the numerical moment penalizes discontinuities in $p\mu vh, \mu, v=+,-$, and, as a result, the numerical moment eliminates numerical artifacts such as $\hat{\mu}$. Similarly, for $r=1$, we can see that $\hat{\mu}$ does not correspond to a numerical solution. However, in this case, the algebraic system does have a small residual that may trap solvers such as *fsolve*. Thus, for $r=0$ and $r=1$, the numerical moment penalizes differences in $p\mu vh, \mu, v=+,-$, that arise from discontinuities in uh, q^-h, and q^+h. Hence, it eliminates numerical artifacts such as $\hat{\mu}$. We note that the same test using the IP-DG framework in [*16*] does appear to have a numerical artifact corresponding to $\hat{\mu}$ when using $r=1$. Thus, the LDG formulation is more successful at removing numerical artifacts for this example when using $r=1$. Furthermore, the IP-DG formulation does not allow for $r=0$.

We now consider $r\geq 2$, in which case $\hat{\mu} \in Vh$. Furthermore, since $\hat{\mu} \in C1$, we will end up with $uh=\hat{\mu}, q^-h=q+h$, and $p\pm\mp h=p\pm\pm h$ being a numeric solution, where

$$q^-h(x) = \begin{cases} x+14, & \text{for } x<0.5, \\ -x+54, & \text{for } x>0.5, \end{cases} \text{and} \quad p^-{-}h(x) = \{1, -1, \text{for } x<0.5, \text{for } x>0.5.$$

Thus, by the equalities of $p\mu vh, \mu, v=+,-$, the numerical moment is always zero and we do have numerical artifacts. These equalities are a consequence of the continuity of uh, q^-h, and q^+h. With the extra degrees of freedom for $r\geq 2$, we allow $C1$ to be embedded into our approximation space Vh, thus creating possible solutions with a zero-valued numerical moment. The numerical

moment acts as a penalty term for differences in $p_{\mu\nu}^h, \mu, \nu = +, -$, which are a consequence of differences in q^-h and q^+h that naturally arise for nontrivial functions when $r=0$ or $r=1$.

Even with the possible presence of numerical artifacts for the above discretization when $r \geq 2$, the numerical moment can be exploited at the solver level. We now present a splitting algorithm for solving the resulting nonlinear algebraic system that uses the numerical moment to strongly emphasize the fact that the viscosity solution of the PDE should preserve the monotonicity required by the definition of ellipticity. Again, we emphasize that the following algorithm is based upon the mixed formulation where the numerical moment is a function of the various second-order derivative approximations.

Algorithm 5.1

(1) Pick an initial guess for u_h.

(2) Form initial guesses for $q^-h, q^+h, p^{--}h, p^{-+}h, p^{+-}h$, and $p^{++}h$ using equations (3.17) and (3.18).

(3) Solve Eq. (3.2) for $p^{-+}h + p^{+-}h2$.

(4) Solve Eq. (3.17) for $+$ and $-$ and the equation formed by averaging (3.18) for $\mu=-, \nu=+$ and $\mu=+, \nu=-$ for u_h, q^-h, and q^+h.

(5) Solve Eq. (3.18) for $\mu, \nu=+, -$ for $p^{--}h, p^{-+}h, p^{+-}h$, and $p^{++}h$.

(6) Repeat Steps 3–5 until the change in $p^{-+}h + p^{+-}h2$ is sufficiently small.

We note that step (3) requires solving a nonlinear equation. However, the nonlinear equation is both monotone and entirely local with respect to the unknown function. Also, step (4) of Algorithm 5.1 requires solving a nonstandard Poisson discretization. The analysis for the corresponding Poisson discretization can be found in [*18*].

For the next numerical tests, we will show that using Algorithm 5.1 with a sufficiently large coefficient for the numerical moment destabilizes numerical artifacts such as $\hat{\mu}$ and steers the approximation towards the viscosity solution of the PDE. Let $u^-(x)=x2$. Then, u^- is the secant line formed by the boundary data for the given boundary value problem. We now approximate the solution of the Monge–Ampère type problem from Test 1 in Sect. 5.1 by using 100 iterations of Algorithm 5.1 followed by using *fsolve* on the full system to solve the global discretization given by (3.2), (3.17), and (3.18). We take the initial guess to be

$u(0)h=34\hat{\mu}+14u^-,$

where, for $r=0, u0h$ is first projected into V_h. From Fig. 13, we see that the numerical moment drives the solution towards the viscosity solution $u+$ when $r=0$ and α is positive. From Fig. 14, we see that the numerical moment also drives the solution towards the viscosity

solution $u+$ when $r=2$ and α is positive, despite the presence of numerical artifacts. From Fig. 15, we see that the moment drives the solution towards the viscosity solution of $F(u_{xx},u_x,u,x):=u^2_{xx}-1$ given by $u-$ for $r=0$ and $r=2$ when α is chosen to be negative. In each figure, the middle graph corresponds to $\hat{\mu}$. Clearly, we recover the numerical artifact corresponding to $\hat{\mu}$ when $\alpha=0$. Thus, the numerical moment plays an essential role in either eliminating numerical artifacts at the discretization level or handling numerical artifacts at the solver level.
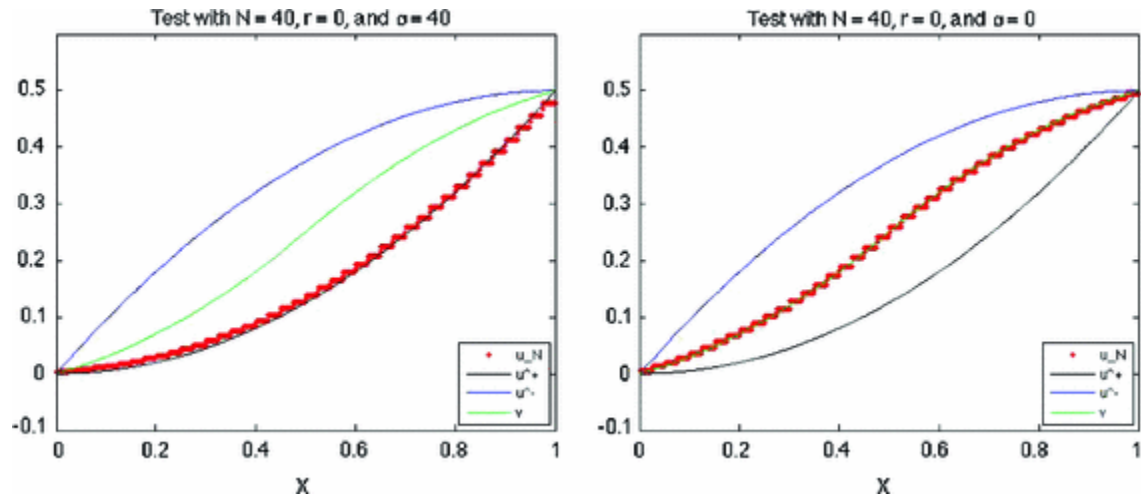

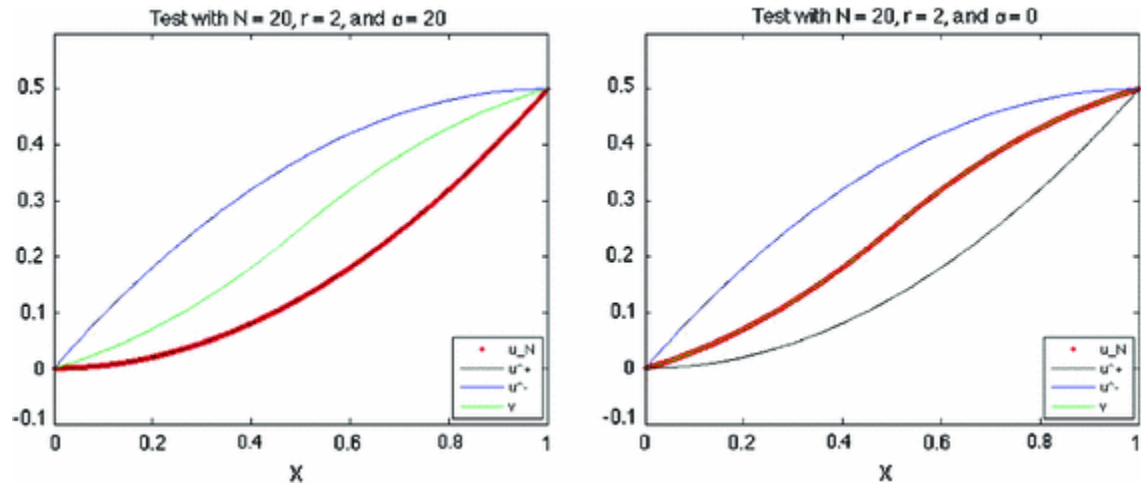**Fig. 13** *Left $\alpha=40,h=1/40$, and $r=0$. Right $\alpha=0,h=1/40$, and $r=0$*

Fig. 14 *Left $\alpha=20,h=1/20$, and $r=2$. Right $\alpha=0,h=1/20$, and $r=2$*
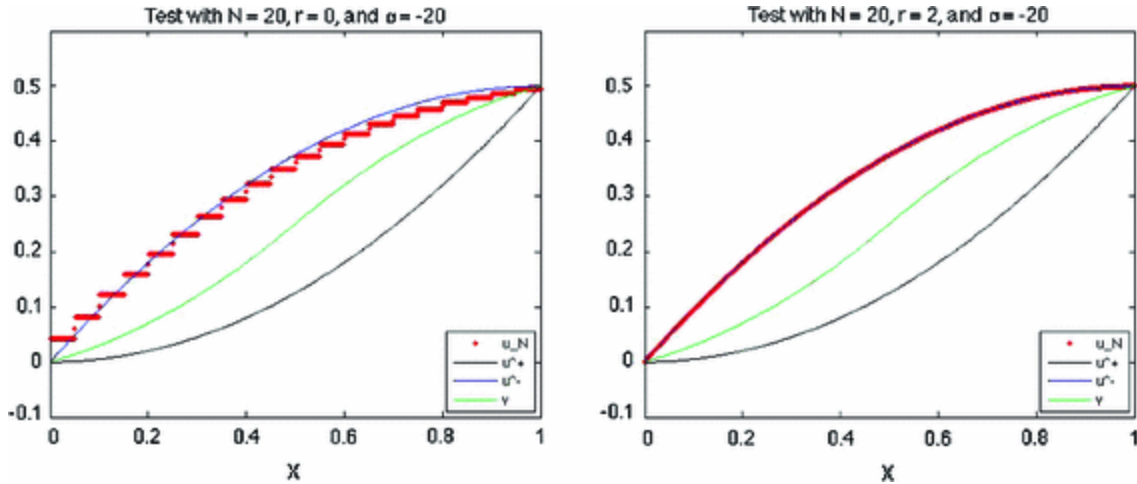
Fig. 15 *Left $\alpha=-40$, $h=1/40$, and $r=0$. Right $\alpha=-20$, $h=1/20$, and $r=2$*

The IP-DG framework presented in [*16*] has similar behavior for the preceding numerical tests when using $r=2$ with a solver analogous to Algorithm 5.1, where again we have a numerical artifact corresponding to $\mu\hat{}$. However, due to the wider availability of IP-DG Poisson solvers in the literature, there is currently greater potential to speed up step (4) in Algorithm 5.1 using the IP-DG formulation. Again, we see that the IP-DG formulation has potential for increased speed at the solver level over the corresponding LDG formulation.

We make one final note about using the iterative solver given by Algorithm 5.1. Using *fsolve* to solve the full system with the initial guess given by $u(0)h$ resulted in either not finding a root for many tests ($r=1$) or converging to a numerical artifact with a discontinuous second order derivative at another node in the mesh ($r=2$). In order to use *fsolve* for the given test problem, the initial guess should either be restricted to the class of functions where $p-+h$ and $p+-h$ preserve the ellipticity of the nonlinear operator, the initial guess should be preconditioned by first using *fsolve* with $r=0,1$, or the initial guess should be preconditioned using Algorithm 5.1. When using $r=0$ and a non-ellipticity-preserving initial guess, solving the full system of equations with *fsolve* still has the potential to converge to $u-$ even for $\alpha>0$. The strength of Algorithm 5.1 is that it strongly enforces the requirement that $F\hat{}$ is monotone decreasing in $p-+$ and $p+-$ over each iteration. Thus, a sufficiently large value for α drives the approximation towards the class of ellipticity-preserving functions if the algorithm converges.

References

1. Arnold, D., Brezzi, F., Cockburn, B., Marini, D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. **39**, 1749–1779 (2001)

2. Bardi, M., Capuzzo-Dolcetta, I.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations, Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA (1997)

3. Barles, G., Souganidis, P.E.: Convergence of approximation schemes for fully nonlinear second order equations. Asymptot. Anal. **4**(3), 271–283 (1991)

4. Caffarelli L.A., Cabré X.: Fully Nonlinear Elliptic Equations, Vol. 43 of American Mathematical Society Colloquium Publications. American Mathematical Society, Providence, RI, (1995)

5. Caffarelli, L.A., Souganidis, P.A.: A rate of convergence for monotone finite difference approximations to fully nonlinear, uniformly elliptic PDEs. Comm. Pure Appl. Math. **61**, 1–17 (2008)

6. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal. **35**, 2440–2463 (1998)

7. Crandall, M.G., Lions, P.-L.: Viscosity solutions of Hamilton–Jacobi equations. Trans. Am. Math. Soc. **277**(1), 1–42 (1983)

8. Crandall, M.G., Lions, P.L.: Two approximations of solutions of Hamilton–Jacobi equations. Math. Comp. **43**, 1–19 (1984)

9. Feng, X., Neilan, M.: The Vanishing Moment Method for Fully Nonlinear Second Order Partial Differential Equations: Formulation, Theory, and Numerical, Snalysis. arxiv.org/abs/1109.1183v2

10. Feng, X., Neilan, M.: Vanishing moment method and moment solutions for second order fully nonlinear partial differential equations. J. Sci. Comput. **38**, 74–98 (2008)

11. Feng, X., Glowinski, R., Neilan, M.: Recent developments in numerical methods for second order fully nonlinear PDEs. SIAM Rev. **55**(2), 205–267 (2013)

12. Feng, X., Kao, C., Lewis, T.: Convergent finite difference methods for one-dimensional fully nonlinear second order partial differential equations. J. Comp. Appl. Math. **254**, 81–98 (2013). doi:10.1016/j.cam.2013.02.001

13. Gilbarg, D., Trudinger, N.S.: Elliptic Partial Differential Equations of Second Order, Classics in Mathematics (reprint of the 1998 edition). Springer, Berlin (2001)

14. Krylov, N.V.: The rate of convergence of finite-difference approximations for Bellman equations with Lipschitz coefficients. Appl. Math. Optim.**52**(3), 365–399 (2005)

15. Kuo, H.J., Trudinger, N.S.: Discrete methods for fully nonlinear elliptic equations. SIAM J. Numer. Anal. **29**(1), 123–135 (1992)

16. Lewis, T., Feng, X.: Mixed interior penalty discontinuous Galerkin methods for one-dimensional fully nonlinear second order elliptic and parabolic equations. J. Comput. Math. (under review)

17. Lewis, T., Feng, X.: Nonstandard local discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. (In preparation)

18. Lewis, T., Neilan, M.: Convergence analysis of a symmetric dual-wind discontinuous Galerkin method. J. Sci. Comput. (to appear)

19. Lieberman, G.M.: Second order parabolic differential equations. World Scientific Publishing Co. Inc., River Edge, NJ (1996)

20. Nitsche, J.A.: Über ein Variationsprinzip zur Lösung von Dirichlet Problemen bei Verwendung von Teilraumen, die keinen Randbedingungen unterworfen sind. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **36**:9–15 (1970/1971)

21. Rivière, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations, Vol. 35 of Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2008)

22. Yan, J., Osher, S.: Direct discontinuous local Galerkin methods for Hamilton–Jacobi equations. J. Comp. Phys. **230**, 232–244 (2011)