

MIXED INTERIOR PENALTY DISCONTINUOUS GALERKIN METHODS FOR ONE-DIMENSIONAL FULLY NONLINEAR SECOND ORDER ELLIPTIC AND PARABOLIC EQUATIONS*

Xiaobing Feng

Department of Mathematics, The University of Tennessee, Knoxville, TN 37996, U.S.A.

Email: xfeng@math.utk.edu

Thomas Lewis

Department of Mathematics, The University of Tennessee, Knoxville, TN 37996, U.S.A.

*Department of Mathematics and Statistics, The University of North Carolina at Greensboro,
NC 27402, U.S.A.*

Email: tllewis3@uncg.edu

Abstract

This paper is concerned with developing accurate and efficient numerical methods for one-dimensional fully nonlinear second order elliptic and parabolic partial differential equations (PDEs). In the paper we present a general framework for constructing high order interior penalty discontinuous Galerkin (IP-DG) methods for approximating viscosity solutions of these fully nonlinear PDEs. In order to capture discontinuities of the second order derivative u_{xx} of the solution u , three independent functions p_1, p_2 and p_3 are introduced to represent numerical derivatives using various one-sided limits. The proposed DG framework, which is based on a nonstandard mixed formulation of the underlying PDE, embeds a nonlinear problem into a mostly linear system of equations where the nonlinearity has been modified to include multiple values of the second order derivative u_{xx} . The proposed framework extends a companion finite difference framework developed by the authors in [9] and allows for the approximation of fully nonlinear PDEs using high order polynomials and non-uniform meshes. In addition to the nonstandard mixed formulation setting, another main idea is to replace the fully nonlinear differential operator by a numerical operator which is consistent with the differential operator and satisfies certain monotonicity (called g-monotonicity) properties. To ensure such a g-monotonicity, the crux of the construction is to introduce the numerical moment, which plays a critical role in the proposed DG framework. The g-monotonicity gives the DG methods the ability to select the mathematically “correct” solution (i.e., the viscosity solution) among all possible solutions. Moreover, the g-monotonicity allows for the possible development of more efficient nonlinear solvers as the special nonlinearity of the algebraic systems can be explored to decouple the equations. This paper also presents and analyzes numerical results for several numerical test problems which are used to gauge the accuracy and efficiency of the proposed DG methods.

Mathematics subject classification: 65N30, 65M60, 35J60, 35K55.

Key words: Fully nonlinear PDEs, Viscosity solutions, Discontinuous Galerkin methods.

1. Introduction

Fully nonlinear partial differential equations (PDEs) refer to a class nonlinear PDEs which is nonlinear in the highest order derivatives of the unknown functions in the equations. Due

* Received January 11, 2013 / Revised version received December 23, 2013 / Accepted January 15, 2014 / Published online March 31, 2014 /

to their strong nonlinearity, this class of PDEs are most difficult to analyze analytically and to approximate numerically. In the mean time, fully nonlinear PDEs arise in many applications such as antenna design, astrophysics, differential geometry, fluid mechanics, image processing, meteorology, mesh generation, optimal control, optimal mass transport, etc [8], which calls for the development of efficient and reliable numerical methods for solving their underlying fully nonlinear PDE problems.

This is the second paper in a series [9] 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:

$$F(u_{xx}, u_x, u, x) = 0, \quad x \in \Omega := (a, b), \quad (1.1)$$

and

$$u_t + F(u_{xx}, u_x, u, t, x) = 0, \quad (x, t) \in \Omega_T := \Omega \times (0, T], \quad (1.2)$$

which are complemented by appropriate boundary and initial conditions. The *goal* of this paper is to design and implement a class of interior penalty discontinuous Galerkin (IP-DG) methods which is based on a nonstandard mixed formulation; the proposed IP-DG methods are named mIP-DG methods. For the ease of presenting the main ideas and avoiding the technicalities, in this paper we confine our attention to the *one dimensional* fully nonlinear second order PDE problem. The generalization and extension to the high dimensional case of the mIP-DG methods of this paper will be presented in a forthcoming work [11]. In fact, it will be seen later that even in the one dimensional case, the construction and analysis of the proposed mIP-DG methods is already quite complicated.

It is well known [8] that the primary challenges for approximating viscosity solutions of fully nonlinear PDEs are caused by the very notion of viscosity solutions themselves (see section 2 for the definition). Unlike the notion of weak solutions for linear and quasilinear PDEs, the notion of viscosity solutions by design is non-variational, and, in general, viscosity solutions do not satisfy the underlying PDEs in a tangible sense. The non-variational nature of viscosity solutions immediately prevents any attempt to directly and straightforwardly construct Galerkin-type (including DG) methods for approximating fully nonlinear PDEs; in other words, nonlinearity in the highest order derivatives of the unknown function does not allow one to perform integration by parts to transfer one order of derivatives to test functions as often done with linear and quasilinear PDEs. Another big challenge for approximating viscosity solutions of fully nonlinear PDEs is caused by the conditional uniqueness of viscosity solutions; namely, viscosity solutions may only be unique in a restricted function class. Requiring numerical solutions to stay or approximately stay in the same function class often imposes a difficult constraint for designing numerical methods. Finally, we like to mention that as expected, solving the resulting strongly nonlinear (algebraic) systems, regardless which discretization method is used, is another difficult issue encountered with numerical fully nonlinear PDEs.

The mIP-DG methods proposed in this paper aim to approximate viscosity solutions of (1.1) and (1.2) which belong to $H^1(\Omega)$ in the spatial variable. We note that such a viscosity still does not satisfy the underlying PDEs in a tangible sense. We also mention that in order to approximate viscosity solutions that do not have H^1 regularity in the spatial variable, we refer the reader to a companion paper [12] in which we propose another class of more complicated mixed discontinuous Galerkin that incorporates a local discontinuous Galerkin (LDG) approach instead of the IP-DG approach. Such an alternate LDG approach is also more appropriate when a more accurate approximation for u_x is desired.

Several novel ideas are utilized to design the mIP-DG methods in this paper which are briefly described below. Since integration by parts, which is the necessary tool for constructing any DG method, cannot be performed on equation (1.1), *the first key idea* is to introduce the auxiliary variable $p := u_{xx}$ and rewrite the original fully nonlinear PDE as a system of PDEs:

$$F(p, u_x, u, x) = 0, \quad (1.3)$$

$$p - u_{xx} = 0. \quad (1.4)$$

Unfortunately, since u_{xx} may not exist for a viscosity solution $u \in H^1(\Omega)$, the above mixed formulation may not make sense. To overcome this difficulty, *the second key idea* is to replace $p := u_{xx}$ by three possible values of u_{xx} , namely, the left and right limits, as well as their average. Thus, we have

$$p_1(x) - u_{xx}(x^-) = 0, \quad (1.5)$$

$$p_2(x) - u_{xx}(x^a) = 0, \quad (1.6)$$

$$p_3(x) - u_{xx}(x^+) = 0, \quad (1.7)$$

where $u_{xx}(x^a)$ can be thought of as the arithmetic average of $u_{xx}(x^-)$ and $u_{xx}(x^+)$. We remark that (1.5) and (1.7) can be regarded as two “one-sided” Poisson problems in u , and (1.6) can be thought of as the “regular” Poisson problem. To incorporate the multiple values of u_{xx} , equation (1.3) must be modified because F is only defined for a single value function p . To this end, we need *the third key idea* of this paper, which is to replace (1.3) by

$$\widehat{F}(p_1, p_2, p_3, u_x, u, x) = 0, \quad (1.8)$$

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

Natural questions that now arise are what are the criterions for \widehat{F} , and how can such a numerical operator \widehat{F} be constructed? 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 [9] where a general finite difference framework has been developed for fully nonlinear second order PDEs. In summary, the criterions for \widehat{F} include *consistency* and *g-monotonicity* (generalized monotonicity), for which precise definitions can be found in section 2. It should be pointed out that in order to construct the desired numerical operator \widehat{F} , a fundamental idea used in [9] 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 [13] (also see [8, section 4], [14]). Finally, we need to design a DG discretization for the mixed system (1.5)–(1.8) to accomplish the goal. *The fifth key idea* of this paper is to use different *numerical fluxes* in the formulations of IP-DG methods for the “one-sided” Poisson problems (1.5) and (1.7) as well as for the “regular” Poisson problem (1.6). 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.

The remainder of this paper is organized as follows. In Section 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 Section 3 we present the detailed formulation of mIP-DG methods for fully nonlinear elliptic equation (1.1) following the outline described above. In Section 4 we consider both explicit and implicit in time fully discrete mIP-DG methods for fully nonlinear parabolic equation (1.2) based on

the method of lines approach. Euler methods and more general Runge-Kutta methods combined with the spatial mIP-DG methods will be specifically formulated. In Section 5 we present many numerical experiments for the proposed mIP-DG methods and their fully discrete counterparts for the parabolic equation (1.2). These numerical experiments verify the accuracy of the proposed mIP-DG methods and also demonstrate the efficiency of these methods. Finally, we complete the paper with a brief summary and some concluding remarks in Section 6.

2. Preliminaries

For a bounded open domain $\Omega \subset \mathbf{R}^d$, let $B(\Omega)$, $USC(\Omega)$ and $LSC(\Omega)$ denote, respectively, the spaces of bounded, upper semicontinuous, and lower semicontinuous functions on Ω . For any $v \in B(\Omega)$, we define

$$v^*(x) := \limsup_{y \rightarrow x} v(y) \quad \text{and} \quad v_*(x) := \liminf_{y \rightarrow x} v(y).$$

Then, $v^* \in USC(\Omega)$ and $v_* \in LSC(\Omega)$, and they are called *the upper and lower semicontinuous envelopes* of v , respectively.

Given a bounded function $F : \mathcal{S}^{d \times d} \times \mathbf{R}^d \times \mathbf{R} \times \bar{\Omega} \rightarrow \mathbf{R}$, where $\mathcal{S}^{d \times d}$ denotes the set of $d \times d$ symmetric real matrices, the general second order fully nonlinear PDE takes the form

$$F(D^2u, \nabla u, u, x) = 0, \quad \text{in } \bar{\Omega}. \quad (2.1)$$

Note that here we have used the convention of writing the boundary condition as a discontinuity of the PDE (cf. [2, p.274]).

The following two definitions can be found in [2, 3, 7].

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

$$F(A, \mathbf{q}, \lambda, x) \leq F(B, \mathbf{q}, \lambda, x), \quad \forall A, B \in \mathcal{S}^{d \times d}, \quad A \geq B, \quad (2.2)$$

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 $\frac{\partial F}{\partial A}$ is negative semi-definite (cf. [7, p. 441]).

Definition 2.2. A function $u \in B(\Omega)$ is called a viscosity subsolution (resp. supersolution) of (2.1) if, for all $\varphi \in C^2(\bar{\Omega})$, if $u^* - \varphi$ (resp. $u_* - \varphi$) has a local maximum (resp. minimum) at $x_0 \in \bar{\Omega}$, then we have

$$F_*(D^2\varphi(x_0), \nabla\varphi(x_0), u^*(x_0), x_0) \leq 0,$$

(resp. $F^*(D^2\varphi(x_0), \nabla\varphi(x_0), u_*(x_0), x_0) \geq 0$). 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 remark 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 equation (2.1) to fulfill a maximum principle (cf. [3, 7]). It is clear from the above definition 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. As pointed out in section 1, the nonvariational nature of viscosity solutions is the main obstacle that prevents direct construction of Galerkin-type methods, which are based on variational formulations.

The following definitions are adapted from [9] in the case $d = 1$.

Definition 2.3. (i) A function $\widehat{F} : \mathbf{R}^6 \rightarrow \mathbf{R}$ is called a numerical operator.

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

$$\liminf_{\substack{p_k \rightarrow p, k=1,2,3 \\ q_1 \rightarrow q, \lambda_1 \rightarrow \lambda, \xi_1 \rightarrow \xi}} \widehat{F}(p_1, p_2, p_3, q_1, \lambda_1, \xi_1) \geq F_*(p, q, \lambda, \xi), \quad (2.3)$$

$$\limsup_{\substack{p_k \rightarrow p, k=1,2,3 \\ q_1 \rightarrow q, \lambda_1 \rightarrow \lambda, \xi_1 \rightarrow \xi}} \widehat{F}(p_1, p_2, p_3, q_1, \lambda_1, \xi_1) \leq F^*(p, q, \lambda, \xi), \quad (2.4)$$

where F_* and F^* denote respectively the lower and the upper semicontinuous envelopes of F .

(iii) A numerical operator \widehat{F} is said to be g-monotone if $\widehat{F}(p_1, p_2, p_3, q, \lambda, \xi)$ is monotone increasing in p_1 and p_3 and monotone decreasing in p_2 , that is, $\widehat{F}(\uparrow, \downarrow, \uparrow, q, \lambda, \xi)$.

We note that the above consistency and g-monotonicity play a critical role in the finite difference framework established in [9]. They also play an equally critical role in the mIP-DG methods of this paper. We also note that 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 [9] is to introduce the concept of *the numerical moment* to help. The following are two examples of so-called Lax-Friedrichs-like numerical operators [9]:

$$\widehat{F}_1(p_1, p_2, p_3, q, \lambda, \xi) := F(p_2, q, \lambda, \xi) + \alpha_1(p_1 - 2p_2 + p_3), \quad (2.5)$$

$$\widehat{F}_2(p_1, p_2, p_3, q, \lambda, \xi) := F\left(\frac{p_1 + p_2 + p_3}{3}, q, \lambda, \xi\right) + \alpha_2(p_1 - 2p_2 + p_3), \quad (2.6)$$

where α_1 and α_2 are undetermined positive constants and the last term in (2.5) and (2.6) is called *the numerical moment*. It is trivial to verify that \widehat{F}_1 and \widehat{F}_2 are consistent with F . To ensure \widehat{F}_1 to be g-monotone, we need

$$\alpha > \frac{1}{2} \left| \frac{\partial F}{\partial u''} \right|, \quad (2.7)$$

assuming adequate regularity for the operator F . We remark that it is natural to require that \widehat{F}_1 is decreasing in p_2 because by the definition of ellipticity, F is decreasing in u'' .

3. Formulation of mIP-DG Methods for Elliptic Problems

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

$$u(a) = u_a \quad \text{and} \quad u(b) = u_b, \quad (3.1)$$

for two given constants u_a and u_b .

Let $\{x_j\}_{j=0}^J \subset \bar{\Omega}$ be a mesh for $\bar{\Omega}$ 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 $j = 1, 2, \dots, J$, $h_0 = h_{J+1} = 0$ and $h = \max_{1 \leq j \leq J} h_j$. Let \mathcal{T}_h denote the collection of the intervals $\{I_j\}_{j=1}^J$ which form a partition of the domain $\bar{\Omega}$. We also introduce the broken H^1 -space

$$H^1(\mathcal{T}_h) := \prod_{I \in \mathcal{T}_h} H^1(I)$$

and the broken L^2 -inner product

$$(v, w)_{\mathcal{T}_h} := \sum_{j=1}^J \int_{I_j} vw \, dx, \quad \forall v, w \in H^1(\mathcal{T}_h).$$

For a fixed integer $r \geq 1$, we define the standard DG finite element space $V^h \subset H^1(\mathcal{T}_h) \subset L^2(\mathcal{T}_h)$ by

$$V^h := \prod_{I \in \mathcal{T}_h} \mathcal{P}_r(I),$$

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

$$\begin{aligned} [v_h(x_j)] &:= v_h(x_j^-) - v_h(x_j^+), & \text{for } j = 1, 2, \dots, J-1, \\ [v_h(x_0)] &:= -v_h(x_0), & [v_h(x_J)] &:= v_h(x_J); \\ \{v_h(x_j)\} &:= \frac{1}{2} \left(v_h(x_j^-) + v_h(x_j^+) \right), & \text{for } j = 1, 2, \dots, J-1, \\ \{v_h(x_0)\} &:= v_h(x_0), & \{v_h(x_J)\} &:= v_h(x_J). \end{aligned}$$

It is trivial to verify the following so-called ‘‘magic formulas’’:

$$[v(x_j)w(x_j)] = v(x_j^-)[w(x_j)] + [v(x_j)]w(x_j^+), \quad (3.2)$$

$$[v(x_j)w(x_j)] = \{v(x_j)\}[w(x_j)] + [v(x_j)]\{w(x_j)\}, \quad (3.3)$$

$$[v(x_j)w(x_j)] = v(x_j^+)[w(x_j)] + [v(x_j)]w(x_j^-). \quad (3.4)$$

Let $\gamma_{0i} > 0$ for $i = 1, 2, 3$ denote interior penalty parameters. It will be clear later that to avoid redundancy of three equations for p_1, p_2 and p_3 , we need to require that $\gamma_{02} > \max\{\gamma_{01}, \gamma_{03}\}$. Define the interior penalty terms

$$J_{0i}(v, w) = \sum_{j=0}^J \frac{\gamma_{0i}}{h_{j,j+1}} [v(x_j)] [w(x_j)], \quad (3.5)$$

for $i = 1, 2, 3$, where

$$h_{j,j+1} = \max\{h_j, h_{j+1}\}, \quad \text{for } j = 0, 1, 2, \dots, J.$$

We now are ready to formulate our DG discretizations for equations (1.5)–(1.8). First, for (fully) nonlinear equation (1.8) we simply approximate it by its broken L^2 -projection into V^h , namely,

$$a_0(u_h, p_{1h}, p_{2h}, p_{3h}; \phi_{0h}) = 0, \quad \forall \phi_{0h} \in V^h, \quad (3.6)$$

where

$$a_0(u, p_1, p_2, p_3; \phi_0) = \left(\widehat{F}(p_1, p_2, p_3, u', u, \cdot), \phi_0 \right)_{\mathcal{T}_h}.$$

Next, we discretize the three *linear* equations (1.5)–(1.7). Notice that for given “sources” $\{p_i\}_{i=1}^3$, (1.5)–(1.7) are three (different) Poisson equations for u . Thus, we can use the standard IP-DG formulation for the Laplacian operator to discretize these equations. However, there is a crucial distinction for doing so on the three equations. We use, respectively, “magic formulas” (3.2), (3.3), and (3.4) when we add the local integration by parts formula to handle the jump terms at the interior nodes. To realize the above strategy, we define the bilinear forms $b_i : H^1(\mathcal{T}_h) \times H^1(\mathcal{T}_h) \rightarrow \mathbb{R}$ by

$$b_i(v, w) := (v', w')_{\mathcal{T}_h} + v'(a)w(a) - \epsilon v(a)w'(a) - v'(b)w(b) + \epsilon v(b)w'(b) + J_{0i}(v, w) \quad \forall v, w \in H^1(\mathcal{T}_h), \quad i = 1, 2, 3, \quad (3.7)$$

where $\epsilon \in \{-1, 0, 1\}$ is often called the “symmetrization” parameter [19]. Note, b_i is symmetric if $\epsilon = -1$, nonsymmetric if $\epsilon = 1$, and incomplete if $\epsilon = 0$. Using the bilinear forms b_i , we define the following DG discretizations of (1.5)–(1.7):

$$a_i(u_h, p_{ih}; \phi_{ih}) = f_i(\phi_{ih}), \quad \forall \phi_{ih} \in V^h, \quad i = 1, 2, 3, \quad (3.8)$$

where

$$\begin{aligned} a_1(u, p_1; \phi_1) &= (p_1, \phi_1)_\Omega + b_1(u, \phi_1) - \sum_{j=1}^{J-1} \left(u'(x_j^-) [\phi_1(x_j)] - \epsilon [u(x_j)] \phi_1'(x_j^-) \right), \\ a_2(u, p_2; \phi_2) &= (p_2, \phi_2)_\Omega + b_2(u, \phi_2) - \sum_{j=1}^{J-1} \left(\{u'(x_j)\} [\phi_2(x_j)] - \epsilon [u(x_j)] \{\phi_2'(x_j)\} \right), \\ a_3(u, p_3; \phi_3) &= (p_3, \phi_3)_\Omega + b_3(u, \phi_3) - \sum_{j=1}^{J-1} \left(u'(x_j^+) [\phi_3(x_j)] - \epsilon [u(x_j)] \phi_3'(x_j^+) \right), \end{aligned}$$

and

$$f_i(\phi_i) = \left(\frac{\gamma_{0i}}{h_{0,1}} \phi_i(a) - \epsilon \phi_i'(a) \right) u_a + \left(\frac{\gamma_{0i}}{h_{J,J+1}} \phi_i(b) + \epsilon \phi_i'(b) \right) u_b,$$

for $i = 1, 2, 3$.

In summary, our mIP-DG methods for the fully nonlinear Dirichlet problem (1.1), (1.2), and (3.1) are defined as seeking $(u_h, p_{1h}, p_{2h}, p_{3h}) \in [V^h]^4$ such that (3.6) and (3.8) hold.

We conclude this section with a few remarks.

Remark 3.1. (a) Looking backwards, (3.8) provides a proper interpretation for each of p_{1h} , p_{2h} , and p_{3h} for a given function u_h . Each p_{ih} defines a discrete second order derivative of u_h . The functions p_{1h} , p_{2h} , and p_{3h} should be very close to each other if u_{xx} exists; however, their discrepancies are expected to be large if u_{xx} does not exist. p_{1h} , p_{2h} , and p_{3h} defined by (3.8) can be regarded as high order extensions of their lower order counterparts defined in [9].

(b) It is easy to check that the three equations defined by (3.8) are linearly independent provided that $\gamma_{02} > \max\{\gamma_{01}, \gamma_{03}\}$.

(c) The reason for $r \neq 0$ can be explained as follows. When $r = 0$, the piecewise constant basis functions have piecewise zero derivatives on the given mesh. After eliminating the jump terms containing derivatives in (3.8), it is clear that the ability for p_1 and p_3 to carry information from the left and the right, respectively, is lost. Furthermore, if $\gamma_{01} = \gamma_{02} = \gamma_{03}$, then $a_1 = a_2 = a_3$, which in turn implies that on a uniform mesh, $p_{1h} = p_{2h} = p_{3h}$. In fact, the variables are all equal to the centered difference approximation for the second order derivative of u_h . As a

result, the numerical moment term vanishes and we are left with the standard three-point finite difference approximation for (1.1) and (3.1). On the other hand, when $r \geq 1$, the numerical operator maintains the directional interpretations for p_1 and p_3 , allowing the numerical operator to take advantage of the numerical moment.

(e) Notice that (3.6)–(3.8) is 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 section 5, an iterative method is used with initial guess given by projecting the secant line resulting from the boundary conditions into V^h . 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 \widehat{F} enlarges the domain of “good” initial values over which the iterative method converges.

4. Formulation of Fully Discrete mIP-DG Methods for Parabolic Problems

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

$$u(0, x) = u_0(x), \quad \forall x \in \Omega, \quad (4.1)$$

and the boundary conditions be given by

$$u(t, a) = u_a(t), \quad u(t, b) = u_b(t), \quad \forall t \in (0, T). \quad (4.2)$$

For the ease of the presentation, we first consider the forward and backward Euler methods. Next, we present generic Runge-Kutta methods.

For a fixed integer $M > 0$, let $\Delta t = \frac{T}{M}$ be the time-step size. Define $t^n := n\Delta t$ for $n = 0, 1, \dots, M$. Then, the forward Euler method (in operator form) for (1.2) is defined by seeking $u^{n+1} : \Omega \rightarrow \mathbf{R}$ such that

$$u^{n+1} = u^n - \Delta t F(u_{xx}^n, u_x^n, u^n, t^n, \cdot), \quad \text{in } \Omega, \quad (4.3)$$

and the standard backward Euler method (in operator form) for (1.2) is defined by seeking $u^{n+1} : \Omega \rightarrow \mathbf{R}$ such that

$$u^{n+1} = u^n - \Delta t F(u_{xx}^{n+1}, u_x^{n+1}, u^{n+1}, t^{n+1}, \cdot), \quad \text{in } \Omega, \quad (4.4)$$

for $n = 0, 1, \dots, M - 1$, where

$$u^0 = u_0, \quad \text{in } \Omega.$$

The compatibility condition between the initial and boundary data immediately implies that

$$u^n(a) = u_a(t^n), \quad u^n(b) = u_b(t^n), \quad n = 1, 2, \dots, M. \quad (4.5)$$

We next apply the mIP-DG framework developed in the previous section to equations (4.3) and (4.4) for their spatial discretizations. We present these two cases separately below because they require different treatments and involve different technicalities.

4.1. Forward Euler method

It turns out that the forward Euler method is tricky to formulate because the variables u_h^{n+1} and p_{jh}^{n+1} ($j = 1, 2, 3$) are not determined simultaneously. Instead, they are constructed sequentially. For a given u_h^n , we first construct p_{jh}^n for $j = 1, 2, 3$ using (3.8). We then define u_h^{n+1} to be a modified L^2 -projection of the right-hand side of (4.3). To take care of the boundary condition, we choose to enforce the boundary condition for u_h^{n+1} weakly in the definition of the modified L^2 -projection.

Specifically, for any $v \in L^2(\Omega)$, we recall that the standard L^2 -projection $\mathcal{P}_h v \in V^h$ of v is defined by

$$(\mathcal{P}_h v, \phi_h)_{\mathcal{T}_h} = (v, \phi_h)_{\mathcal{T}_h}, \quad \forall \phi_h \in V^h. \quad (4.6)$$

For any $v \in C^0(\bar{\Omega})$, we define a modified L^2 -projection $\tilde{\mathcal{P}}_h v \in V^h$ of v by

$$\begin{aligned} & (\tilde{\mathcal{P}}_h v, \phi_h)_{\mathcal{T}_h} + \frac{1}{\sqrt{h}} \left(\tilde{\mathcal{P}}_h v(a) \phi_h(a) + \tilde{\mathcal{P}}_h v(b) \phi_h(b) \right) \\ &= (v, \phi_h)_{\mathcal{T}_h} + \frac{1}{\sqrt{h}} \left(v(a) \phi_h(a) + v(b) \phi_h(b) \right), \quad \forall \phi_h \in V^h, \end{aligned} \quad (4.7)$$

and the corresponding modified L^2 -projection operator $\tilde{\mathcal{P}}_h : L^2(\Omega) \cap C^0(\bar{\Omega}) \rightarrow V^h$. In the above definition, the boundary condition (4.2) is weakly enforced via a penalty technique which is due to Nitsche [18].

For a given function $v \in H^1(\mathcal{T}_h)$ which satisfies the boundary condition (4.2), we define its three discrete ‘‘one-sided’’ second order derivatives $\mathcal{Q}_h^- v, \mathcal{Q}_h^a v, \mathcal{Q}_h^+ v \in V^h$ ($j = 1, 2, 3$) using (3.8) as follows:

$$\begin{aligned} (\mathcal{Q}_h^- v, \phi_h)_\Omega &= \left(\frac{\gamma_{01}}{h_{0,1}} \phi_h(a) - \epsilon \phi_h'(a) \right) u_a(t) + \left(\frac{\gamma_{01}}{h_{J,J+1}} \phi_h(b) + \epsilon \phi_h'(b) \right) u_b(t) \\ &\quad - b_1(v, \phi_h) + \sum_{j=1}^{J-1} \left(v'(x_j^-) [\phi_h(x_j)] - \epsilon [v(x_j)] \phi_h'(x_j^-) \right), \quad \forall \phi_h \in V^h, \quad (4.8) \\ (\mathcal{Q}_h^a v, \phi_h)_\Omega &= \left(\frac{\gamma_{02}}{h_{0,1}} \phi_2(a) - \epsilon \phi_h'(a) \right) u_a(t) + \left(\frac{\gamma_{02}}{h_{J,J+1}} \phi_h(b) + \epsilon \phi_h'(b) \right) u_b(t) \\ &\quad - b_2(v, \phi_h) + \sum_{j=1}^{J-1} \left(\{v'(x_j)\} [\phi_h(x_j)] - \epsilon [v(x_j)] \{ \phi_h'(x_j) \} \right), \quad \forall \phi_h \in V^h, \quad (4.9) \\ (\mathcal{Q}_h^+ v, \phi_h)_\Omega &= \left(\frac{\gamma_{03}}{h_{0,1}} \phi_h(a) - \epsilon \phi_h'(a) \right) u_a(t) + \left(\frac{\gamma_{03}}{h_{J,J+1}} \phi_h(b) + \epsilon \phi_h'(b) \right) u_b(t) \\ &\quad - b_3(v, \phi_h) + \sum_{j=1}^{J-1} \left(v'(x_j^+) [\phi_3(x_j)] - \epsilon [v(x_j)] \phi_h'(x_j^+) \right), \quad \forall \phi_h \in V^h, \end{aligned} \quad (4.10)$$

and the corresponding operators $\mathcal{Q}_h^-, \mathcal{Q}_h^a, \mathcal{Q}_h^+ : H^1(\mathcal{T}_h) \rightarrow V^h$.

With the help of the operators $\mathcal{Q}_h^-, \mathcal{Q}_h^a, \mathcal{Q}_h^+$, and $\tilde{\mathcal{P}}_h$, we now define our fully discrete forward Euler method for the initial-boundary value problem (1.2), (4.1), (4.2) as follows: for $n = 0, 1, \dots, M-1$,

$$u_h^{n+1} = \tilde{\mathcal{P}}_h \left(u_h^n - \Delta t \hat{F}(\mathcal{Q}_h^- u_h^n, \mathcal{Q}_h^a u_h^n, \mathcal{Q}_h^+ u_h^n, u_{hx}^n, u_h^n, t^n, \cdot) \right), \quad (4.11)$$

$$u_h^0 = \mathcal{P}_h u_0, \quad (4.12)$$

where the operators \mathcal{Q}_h^- , \mathcal{Q}_h^a , and \mathcal{Q}_h^+ are evaluated at time t^n over each iteration and the modified projection operator is understood at time t^{n+1} . Thus, the projection operator is used to both enforce the boundary condition and map the value of \widehat{F} back into V^h .

4.2. Backward Euler method

We rewrite the backward Euler scheme (4.4) as

$$u^n + \Delta t F(u_{xx}^n, u_x^n, u^n, t^n, \cdot) = u^{n-1}, \quad (4.13)$$

for $n = 1, \dots, M$, where $u^0 = u_0$.

Clearly, (4.13) and (1.2) have the same form. Thus, the spatial discretization of the backward Euler scheme (4.4) is a straightforward adaptation of the mIP-DG framework for elliptic PDEs developed in section 3. To the end, we define a new numerical operator \widehat{G} by

$$\widehat{G}(p_1, p_2, p_3, u, t, x) := u(x, t) + \Delta t \widehat{F}(p_1, p_2, p_3, u_x, u, t, x), \quad \forall (x, t) \in \Omega_T. \quad (4.14)$$

Then, analogous to the formulation for (1.1) and (3.1), the fully discrete backward Euler mIP-DG methods for (1.2), (4.1), and (4.2) is defined by seeking $(u_h^n, p_{1h}^n, p_{2h}^n, p_{3h}^n) \in (V^h)^4$ such that, for $n = 1, 2, \dots, M$,

$$\widehat{a}_0(t^n, u_h^n, p_{1h}^n, p_{2h}^n, p_{3h}^n; \phi_{0h}) = (u_h^{n-1}, \phi_{0h})_{\mathcal{T}_h}, \quad \forall \phi_{0h} \in V^h, \quad (4.15)$$

$$\widehat{a}_i(u_h^n, p_{ih}^n; \phi_{ih}) = g_i(t^n, \phi_{ih}), \quad \forall \phi_{ih} \in V^h, \quad i = 1, 2, 3, \quad (4.16)$$

$$u_h^0 = \mathcal{P}_h u_0. \quad (4.17)$$

where

$$\begin{aligned} \widehat{a}_0(t, u, p_1, p_2, p_3; \phi_0) &= \left(\widehat{G}(p_1, p_2, p_3, u, t, \cdot), \phi_0 \right)_{\mathcal{T}_h}, \\ \widehat{a}_1(u, p_1; \phi_1) &= (p_1, \phi_1)_\Omega + b_1(u, \phi_1) - \sum_{j=1}^{J-1} \left(u'(x_j^-) [\phi_1(x_j)] - \epsilon [u(x_j)] \phi_1'(x_j^-) \right), \\ \widehat{a}_2(u, p_2; \phi_2) &= (p_2, \phi_2)_\Omega + b_2(u, \phi_2) - \sum_{j=1}^{J-1} \left(\{u'(x_j)\} [\phi_2(x_j)] - \epsilon [u(x_j)] \{ \phi_2'(x_j) \} \right), \\ \widehat{a}_3(u, p_3; \phi_3) &= (p_3, \phi_3)_\Omega + b_3(u, \phi_3) - \sum_{j=1}^{J-1} \left(u'(x_j^+) [\phi_3(x_j)] - \epsilon [u(x_j)] \phi_3'(x_j^+) \right), \end{aligned}$$

and

$$g_i(t, \phi_i) = \left(\frac{\gamma_{0i}}{h_{0,1}} \phi_i(a) - \epsilon \phi_i'(a) \right) u_a(t) + \left(\frac{\gamma_{0i}}{h_{J,J+1}} \phi_i(b) + \epsilon \phi_i'(b) \right) u_b(t),$$

for $i = 1, 2, 3$. That is, a nonhomogeneous fully nonlinear elliptic problem is solved at each time step.

4.3. Runge-Kutta methods

We can also formulate Runge-Kutta (RK) methods for approximating the initial value problem corresponding to (1.2) as follows. For simplicity, we first introduce a semi-discrete version of (1.2) that has been spatially discretized. To this end, we introduce some convenient notation.

Define $t^k := k \Delta t$ for $0 \leq k \leq M$. Thus, $t^k \in [0, T]$ for all k . We let $\mathcal{Q}_h^{-,k}$, $\mathcal{Q}_h^{a,k}$, and $\mathcal{Q}_h^{+,k}$ denote the discrete second order derivative operators evaluated at time t^k in (4.8), (4.9), and (4.10), respectively. We also define the operator $\widehat{F}^k : V^h \times \Omega \rightarrow \mathbf{R}$ by

$$\widehat{F}^k[v_h](x) := \widehat{F}(\mathcal{Q}_h^{-,k}v_h, \mathcal{Q}_h^{a,k}v_h, \mathcal{Q}_h^{+,k}v_h, v_{hx}, v_h, t^k, x), \quad (4.18)$$

for all $0 \leq k \leq M$. Then, the semi-discrete form of (1.2) is

$$\frac{\partial}{\partial t} u_h(x, t^k) = -\widehat{F}^k[u_h](x), \quad (4.19)$$

for all $0 < k \leq M$, $x \in \Omega$.

Let s be a positive integer, $A \in \mathbb{R}^{s \times s}$, and $b, c \in \mathbb{R}^s$ such that

$$\sum_{\ell=1}^s a_{k,\ell} = c_k,$$

for each $k = 1, 2, \dots, s$. Then, a generic s -stage RK method for approximating (4.19) can be written

$$u_h^{n+1} = \widetilde{\mathcal{P}}_h \left(u_h^n - \Delta t \sum_{\ell=1}^s b_\ell \widehat{F}^{n+c_\ell}[\xi_h^{n,\ell}] \right), \quad (4.20)$$

with

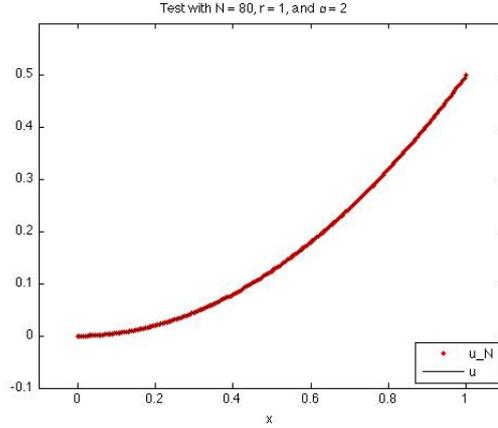
$$\xi_h^{n,\ell} = \mathcal{P}_h \left(u_h^n - \Delta t \sum_{k=1}^s a_{k,\ell} \widehat{F}^{n+c_k}[\xi_h^{n,k}] \right),$$

for all $n = 0, 1, \dots, N-1$ and $u_h^0 = \mathcal{P}_h u_0$. We note that (4.20) corresponds to an explicit method when A is strictly lower diagonal and an implicit method otherwise. Also, we can interpret $\xi_h^{n,\ell}$ in (4.20) as an approximation for $u_h^{n+c_\ell}$. Since the boundary condition at time t^{n+1} is enforced by \widehat{F}^{n+1} , we can replace $\widetilde{\mathcal{P}}_h$ with \mathcal{P}_h in (4.20) if $c_s = 1$. In the next section, we implement the classical (explicit) RK4 method to test the effectiveness of our fully-discrete formulation when paired with a high-order time stepping scheme.

5. Numerical Experiments

In this section, we present a series of numerical tests to demonstrate the utility of the proposed mIP-DG 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 dynamic 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 u_a and u_b . For dynamic problems, we let $u_h^0 = \mathcal{P}_h u_0$, $p_{1h}^0 = u_{hxx}^0(x^-)$, $p_{2h}^0 = \{u_{hxx}^0(x)\}$, and $p_{3h}^0 = u_{hxx}^0(x^+)$. Also, the initial guess for u_h^n will be provided by u_h^{n-1} , and the initial guesses for p_{1h}^n , p_{2h}^n , and p_{3h}^n will be provided by p_{1h}^{n-1} , p_{2h}^{n-1} , and p_{3h}^{n-1} , respectively. For convenience, we set $\epsilon = 0$ for all tests. We remark that similar results can be obtained when $\epsilon \neq 0$, and the actual benefit of the symmetrization parameter is unclear in the context of nonlinear algebraic systems. The role of α and the numerical moment will be further explored in Section 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 dynamic problems. For the dynamic test problems, we shall see that the lower order time discretization dominates the approximation



r	Norm	h = 1/10	h = 1/20		h = 1/40		h = 1/80	
		Error	Error	Order	Error	Order	Error	Order
1	L^2	2.9e-03	7.3e-04	2.00	1.8e-04	1.99	4.7e-05	1.97
	L^∞	3.8e-03	9.4e-04	2.00	2.4e-04	1.99	6.1e-05	1.96

Fig. 5.1. Test 1: $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 1$, and $\gamma_{02} = 1.1$. The plot corresponds to $r = 1$ and $h = 1/80$.

error for reasonable time step size Δt when using Euler methods. For the elliptic test problems and for the dynamic test problems where the error is dominated by the spatial discretizations, it appears that the spatial error is of order $\mathcal{O}(h^\ell)$, where

$$\ell = \begin{cases} r + 1, & \text{for } r \text{ odd,} \\ r, & \text{for } r \text{ even.} \end{cases}$$

Furthermore, we observe that when using odd order elements, the schemes exhibit optimal rate of convergence in both norms.

5.1. Elliptic test problems

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

Test 1. Consider the stationary Monge-Ampère problem

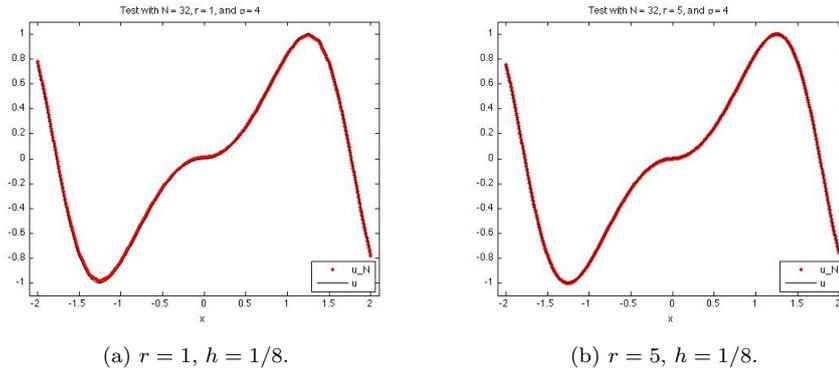
$$\begin{aligned} -u_{xx}^2 + 1 &= 0, & 0 < x < 1, \\ u(0) &= 0, & u(1) = \frac{1}{2}. \end{aligned}$$

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

$$u^+(x) = \frac{1}{2}x^2, \quad u^-(x) = -\frac{1}{2}x^2 + x,$$

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 Section 5.3 we shall give some insights about the selectiveness of our schemes.

We approximate the given problem using the linear element ($r = 1$) to see how the approximation converges with respect to h when the solution is not in the approximation space.



r	Norm	$h = 1$	$h = 1/2$		$h = 1/4$		$h = 1/8$	
		Error	Error	Order	Error	Order	Error	Order
1	L^2	8.1e-01	2.4e-01	1.73	8.0e-02	1.60	2.8e-02	1.52
	L^∞	1.0e+00	2.3e-01	2.14	7.8e-02	1.58	2.7e-02	1.54
2	L^2	1.1e+00	2.9e-01	1.88	4.2e-02	2.78	2.9e-02	0.56
	L^∞	8.1e-01	2.4e-01	1.76	4.5e-02	2.40	1.8e-02	1.30
3	L^2	6.4e-01	2.7e-02	4.55	1.4e-03	4.33	6.5e-05	4.38
	L^∞	4.9e-01	3.1e-02	3.99	1.6e-03	4.32	9.1e-05	4.09
4	L^2	5.6e-02	3.2e-03	4.14	2.4e-04	3.72	1.7e-05	3.83
	L^∞	4.9e-02	3.0e-03	4.02	2.6e-04	3.56	1.6e-05	4.02
5	L^2	2.3e-02	8.5e-04	4.79	1.5e-05	5.82	2.4e-07	5.96
	L^∞	2.1e-02	9.3e-04	4.49	1.8e-05	5.67	2.6e-07	6.11

Fig. 5.2. Test 2: $\epsilon = 0, \alpha = 4, \gamma_{01} = \gamma_{03} = 2,$ and $\gamma_{02} = 2.5.$

The numerical results are shown in Fig. 5.1. The results for the quadratic element ($r = 2$) are presented in Fig. 5.10. We note that the approximations using $r = 2$ are almost exact for each mesh size. This is expected since $u^+ \in V^h$ when $r = 2$.

Test 2. Consider the problem

$$\begin{aligned}
 -u_{xx}^3 + |u_x| + S(x) &= 0, & -2 < x < 2, \\
 u(-2) &= \sin(4), & u(2) &= -\sin(4),
 \end{aligned}$$

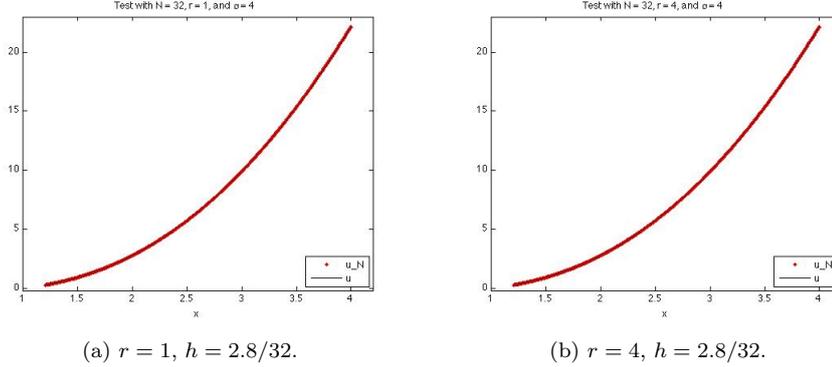
where

$$S(x) = [2\text{sign}(x) \cos(x^2) - 4x^2 \sin(x|x|)]^3 - 2|x \cos(x^2)|.$$

This problem has the exact (viscosity) solution $u(x) = \sin(x|x|)$. Notice that the equation is nonlinear in both u_{xx} and u_x , and the exact solution is not twice differentiable at $x = 0$. The numerical results are shown in Fig. 5.2. 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.

Test 3. Consider the stationary Hamilton-Jacobi-Bellman problem

$$\begin{aligned}
 \inf_{0 < \theta(x) \leq 1} \left\{ -\theta u_{xx} + \theta^2 x^2 u_x + \frac{1}{x} u + S(x) \right\} &= 0, & 1.2 < x < 4, \\
 u(1.2) &= 1.44 \ln 1.2, & u(4) &= 16 \ln 4,
 \end{aligned}$$



r	Norm	$h = 2.8/4$		$h = 2.8/8$		$h = 2.8/16$		$h = 2.8/32$	
		Error	Error	Order	Error	Order	Error	Order	
1	L^2	3.5e-01	9.8e-02	1.83	2.6e-02	1.93	6.6e-03	1.97	
	L^∞	3.9e-01	1.2e-01	1.70	3.4e-02	1.81	9.0e-03	1.91	
2	L^2	9.1e-03	1.9e-03	2.28	4.2e-04	2.18	9.6e-05	2.11	
	L^∞	9.9e-03	1.7e-03	2.53	3.6e-04	2.23	8.2e-05	2.15	
3	L^2	3.5e-04	2.7e-05	3.69	1.9e-06	3.85	4.2e-07	2.14	
	L^∞	5.1e-04	4.2e-05	3.61	3.3e-06	3.69	3.7e-07	3.15	
4	L^2	2.5e-05	1.4e-06	4.14	7.7e-08	4.19	8.5e-09	3.18	
	L^∞	3.3e-05	1.5e-06	4.46	7.6e-08	4.30	1.3e-08	2.51	

Fig. 5.3. Test 3: $\epsilon = 0$, $\alpha = 4$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

where

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

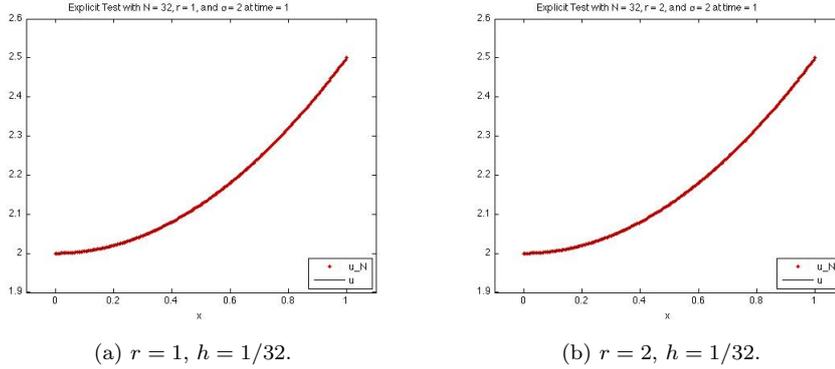
It can be shown that the exact (viscosity) solution of this problem is given by $u(x) = x^2 \ln x$, which occurs when

$$\theta^*(x) = \frac{2 \ln(x) + 3}{2x^3 [2 \ln(x) + 1]}.$$

We solve this problem using various order elements and record the numerical results in Fig. 5.3, which shows that our mIP-DG methods can also handle the Bellman-type fully nonlinear PDEs very well.

5.2. Parabolic test problems

We now implement the proposed fully discrete forward and backward Euler and RK4 mIP-DG 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 either the forward Euler method or RK4, our test problems generally require $\Delta t = \mathcal{O}(h^2)$ to ensure the stability. In fact, the constant for the CFL condition appears to decrease as the order of the element increases. Below we implement both the implicit and explicit methods 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 the proposed fully discrete schemes and their promising potentials. For explicit tests, we record the parameter κ_t



r	Norm	$h = 1/4$		$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order	
1	L^2	5.7e-03	1.4e-03	1.98	3.7e-04	1.99	9.2e-05	1.99	
	L^∞	7.9e-03	2.0e-03	1.99	5.0e-04	1.99	1.3e-04	1.99	
2	L^2	3.3e-05	8.2e-06	2.00	2.1e-06	2.00	5.1e-07	2.00	
	L^∞	4.5e-05	1.1e-05	2.00	2.8e-06	2.00	7.1e-07	2.00	
3	L^2	3.3e-05	8.2e-06	2.00	2.1e-06	2.00	5.1e-07	2.00	
	L^∞	4.5e-05	1.1e-05	2.00	2.8e-06	2.00	7.1e-07	2.00	

Fig. 5.4. Test 4: Computed solutions at $T = 1$ using the explicit Euler method with $\kappa_t = 0.002$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$. Note, the scheme is unstable for $r = 2, 3$ when $\kappa_t = 0.01$.

which serves as the scaling constant for the CFL condition, so we have $\Delta t = \kappa_t h^2$. For implicit tests, we record computed solutions with various time step Δt .

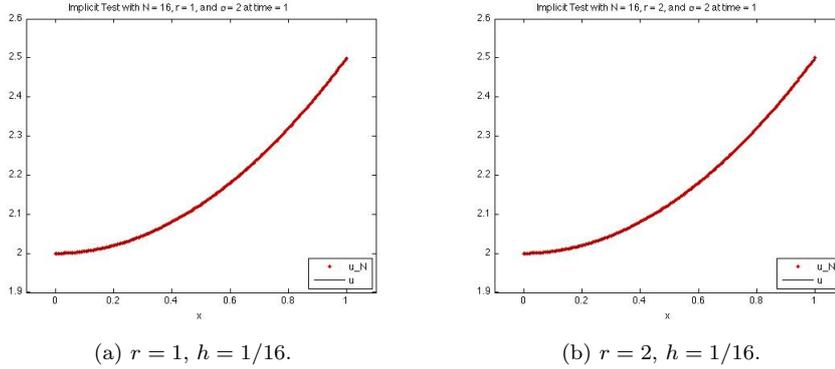
Test 4. Let $\Omega = (0, 1)$, $u_a(t) = t^4$, $u_b = \frac{1}{2} + t^4$, and $u_0(x) = \frac{1}{2}x^2$. We consider the (benchmark) quasilinear problem (1.2), (4.1), and (4.2) with

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

It is easy to verify that this problem has a unique classical solution $u(x, t) = 0.5x^2 + t^4 + 1$. The numerical results for the fully discrete forward Euler method are presented in Fig. 5.4, and the results for the backward Euler method are shown in Fig. 5.5. We observe that the errors for the backward Euler method are dominated by the relatively small size of the time step when

Table 5.1. Test 4: Computed solutions at $T = 1$ using RK4 with $\kappa_t = 0.002$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

r	Norm	$h = 1/4$		$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order	
1	L^2	4.7e-03	1.2e-03	2.01	2.9e-04	2.01	7.2e-05	2.01	
	L^∞	1.4e-02	3.5e-03	1.98	8.9e-04	1.99	2.2e-04	2.00	
2	L^2	5.2e-10	3.9e-10	0.40	3.3e-10	0.23	3.1e-10	0.12	
	L^∞	8.9e-10	6.1e-10	0.54	4.9e-10	0.31	4.4e-10	0.17	
3	L^2	7.0e-10	5.5e-10	0.35	4.7e-10	0.21	4.4e-10	0.11	
	L^∞	1.2e-09	8.6e-10	0.44	7.0e-10	0.29	6.3e-10	0.16	



(a) $r = 1, h = 1/16$.

(b) $r = 2, h = 1/16$.

r	Norm	$h = 1/4$	$h = 1/8$		$h = 1/16$	
		Error	Error	Order	Error	Order
1	L^2	4.4e-03	9.6e-04	2.20	1.8e-04	2.40
	L^∞	9.4e-03	2.4e-03	2.00	5.9e-04	2.00
2	L^2	2.6e-04	2.6e-04	-0.00	2.6e-04	-0.00
	L^∞	3.6e-04	3.6e-04	-0.00	3.6e-04	-0.00
3	L^2	2.6e-04	2.6e-04	-0.00	2.6e-04	-0.00
	L^∞	3.6e-04	3.6e-04	-0.00	3.6e-04	-0.00

Fig. 5.5. Test 4: Computed solution at $T = 1$ using the implicit Euler method with $\Delta t = 0.001$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

compared to the forward Euler method. For smaller time step sizes, the errors are similar. However, the backward Euler method appears unstable for $\kappa_t > 0.01$. We also recorder the results for RK4 in Table 5.1 where we observe optimal convergence rates for $r = 1$ and recover the exact solution for $r \geq 2$.

We now consider the error for the approximation resulting from using Euler time stepping methods. Note that the solution u is a quadratic in space and a quartic in time. Letting $r = 2$, we limit the approximation error almost entirely to the time discretization scheme. In fact, setting $t = 0$ and solving the stationary form of the PDE, we have

$$\|u - u_h\|_{L^2((0,1))} \approx 1.6 \times 10^{-9} \quad \text{and} \quad \|u - u_h\|_{L^\infty((0,1))} \approx 2.4 \times 10^{-9}$$

using the elliptic solver with $h = 1/4$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 1$, $\gamma_{02} = 1.1$, and initial guess given by the secant line for the boundary data. Then, approximating the problem for varying Δt , we have the results recorded in Table 5.2 for the forward Euler method and in Table 5.3 for the backward Euler method. We observe that the convergence rate in time appears to have order 1 as expected when using the Euler methods. We also see in Table 5.4 that we recover the exact solution when using RK4.

Table 5.2. Test 4: Computed solutions at time $T = 1$ using the explicit Euler method with $h = 1/16$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 1$, and $\gamma_{02} = 1.1$.

r	Norm	$\kappa_t = 0.008$	$\kappa_t = 0.004$		$\kappa_t = 0.002$		$\kappa_t = 0.001$	
		Error	Error	Order	Error	Order	Error	Order
2	L^2	8.2e-06	4.1e-06	1.00	2.1e-06	1.00	1.0e-06	1.00
	L^∞	1.1e-05	5.7e-06	1.00	2.8e-06	1.00	1.4e-06	1.00

Table 5.3. Test 4: Computed solutions at $T = 1$ using the implicit Euler method with $h = 1/4$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 1$, and $\gamma_{02} = 1.1$.

r	Norm	$\Delta t = 1/10$	$\Delta t = 1/20$		$\Delta t = 1/40$		$\Delta t = 1/80$	
		Error	Error	Order	Error	Order	Error	Order
2	L^2	2.4e-02	1.3e-02	0.93	6.4e-03	0.96	3.2e-03	0.98
	L^∞	3.3e-02	1.7e-02	0.93	8.8e-03	0.96	4.5e-03	0.98

Table 5.4. Test 4: Computed solutions at time $T = 1$ using RK4 with $h = 1/16$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 1$, and $\gamma_{02} = 1.1$.

r	Norm	$\kappa_t = 0.008$	$\kappa_t = 0.004$		$\kappa_t = 0.002$		$\kappa_t = 0.001$	
		Error	Error	Order	Error	Order	Error	Order
2	L^2	8.9e-10	8.9e-10	-0.00	8.9e-10	0.00	8.9e-10	0.00
	L^∞	1.3e-09	1.3e-09	0.00	1.3e-09	-0.00	1.3e-09	0.00

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

$$F(u_{xx}, u_x, u, t, x) = -u_x \ln(u_{xx} + 1) + S(x, t),$$

$$S(x, t) = e^{(t+1)x} \left(x - (t+1) \ln((t+1)^2 e^{(t+1)x} + 1) \right).$$

It is easy to verify that this problem has a unique classical solution $u(x, t) = e^{(t+1)x}$. Notice, the exact solution u cannot be factored into the form $u(x, t) = G(t)Y(x)$ for some functions G and Y . The numerical results for the fully discrete forward Euler method are recorded in Fig. 5.6 and the results for the backward Euler method are given in Fig. 5.7. The error appears to be dominated by the low order time discretization given the relatively large value for Δt in the backward Euler test. However, using a smaller Δt for the forward Euler test, we were able to achieve a higher order of accuracy. We remark that even for $\Delta t = 0.005h^2$, the forward Euler scheme is not stable for $h = \frac{1}{4}$ and $r = 1$. We also record the results for RK4 in Table 5.5 where we observe optimal rates of convergence in h until the the error reaches a level of 1.0e-09, which appears to be the floor for our implementation as suggested in Table 5.4.

Test 6. 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

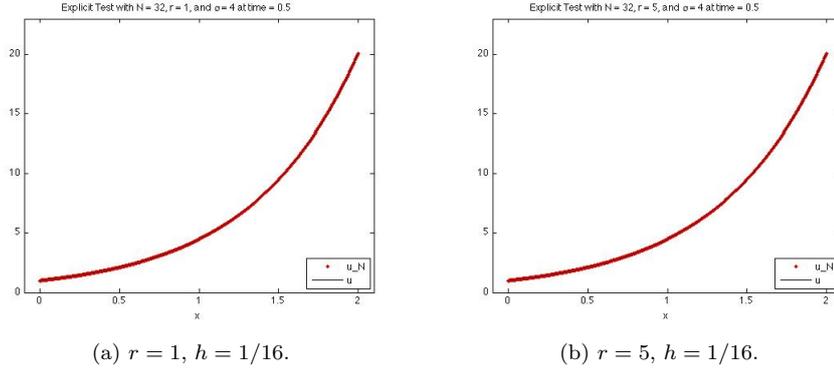
$$F(u_{xx}, u_x, u, t, x) = - \min_{\theta(t,x) \in \{1,2\}} \left\{ A_\theta u_{xx} - c(x, t) \cos(t) \sin(x) - \sin(t) \sin(x) \right\},$$

where $A_1 = 1, A_2 = \frac{1}{2}$, and

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

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

$$\theta^*(t, x) = \begin{cases} 1, & \text{if } c(x, t) = 1, \\ 2, & \text{if } c(x, t) = \frac{1}{2}. \end{cases}$$



r	Norm	$h = 1/2$		$h = 1/4$		$h = 1/8$		$h = 1/16$	
		Error	Error	Order	Error	Order	Error	Order	
1	L^2	5.0e-01	3.6e-02	1.73	1.2e-02	1.32	3.6e-03	1.67	
	L^∞	8.2e-01	2.8e-01	1.57	1.0e-01	1.47	3.1e-02	1.69	
2	L^2	4.5e-02	1.2e-02	1.89	3.3e-03	1.87	8.7e-04	1.93	
	L^∞	6.0e-02	1.4e-02	2.11	3.6e-03	1.96	9.0e-04	1.98	
3	L^2	1.5e-03	2.8e-04	2.39	7.1e-05	1.98	1.8e-05	1.98	
	L^∞	2.7e-03	3.5e-04	2.97	7.6e-05	2.21	1.8e-05	2.05	
4	L^2	1.2e-03	2.9e-04	2.06	7.2e-05	2.02	1.8e-05	2.01	
	L^∞	1.3e-03	3.0e-04	2.13	7.3e-05	2.02	1.8e-05	2.01	
5	L^2	1.2e-03	2.9e-04	2.00	7.2e-05	2.00	1.8e-05	2.00	
	L^∞	1.2e-03	2.9e-04	2.00	7.3e-05	2.00	1.8e-05	2.00	

Fig. 5.6. Test 5: Computed solutions at time $T = 3.10$ using the explicit Euler method with $\kappa_t = 0.0025$, $\epsilon = 0$, $\alpha = 4$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$. The method is not stable for $\kappa_t = 0.005$.

The numerical results are recorded in Fig. 5.8 for the fully discrete forward Euler method and in Fig. 5.9 for the backward Euler method. We observe that the accuracy of the implicit method appears to suffer from the lower order accuracy of the Euler method. For $h = \frac{\pi}{8}$, the explicit method requires $\Delta t \approx 3.1 \times 10^{-4}$, while the implicit method only uses $\Delta t = 0.062$. When Δt increases, the explicit method demonstrates instability. We also observe optimal rates of convergence in Table 5.6 for RK4 timestepping.

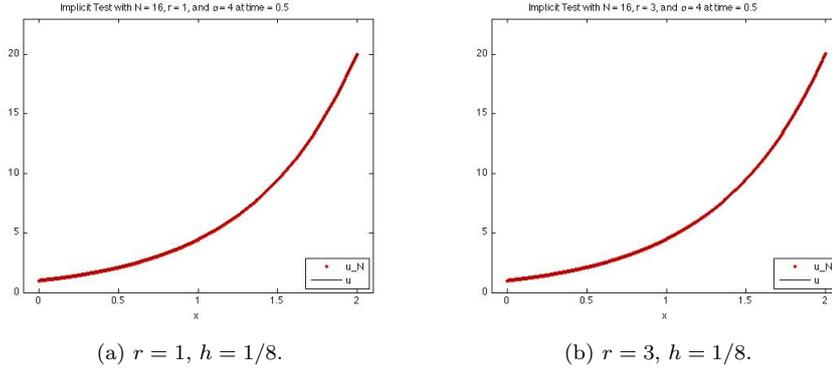
5.3. The role of the numerical moment

We now discuss the role and utility of the numerical moment in forming an appropriate numerical operator. Consider the stationary Monge-Ampère problem from Test 1, which has the following two solutions:

$$u^+(x) = \frac{1}{2}x^2, \quad u^-(x) = -\frac{1}{2}x^2 + x,$$

where u^+ is convex and u^- is concave. The solution u^+ is the unique viscosity solution.

To demonstrate the role of the numerical moment, we approximate the given problem using $\alpha > 0$, $\alpha = 0$, and $\alpha < 0$. Notice that multiplying the PDE by -1 , we see that u^- is the unique viscosity solution of the equation with the operator $F(u) = u_{xx}^2 - 1$. Then, for $\alpha > 0$, our



r	Norm	$h = 1/2$	$h = 1/4$		$h = 1/8$	
		Error	Error	Order	Error	Order
1	L^2	4.2e-01	1.4e-01	1.54	4.6e-02	1.66
	L^∞	8.3e-01	2.4e-01	1.77	7.9e-02	1.63
2	L^2	7.3e-02	1.6e-02	2.21	3.0e-03	2.40
	L^∞	9.6e-02	1.8e-02	2.41	3.2e-03	2.49
3	L^2	2.8e-03	7.8e-04	1.82	9.1e-04	-0.22
	L^∞	5.6e-03	8.5e-04	2.71	9.2e-04	-0.11

Fig. 5.7. Test 5: Computed solutions at time $T = 0.5$ using the implicit Euler method with $\Delta t = 0.0005$, $\epsilon = 0$, $\alpha = 4$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

scheme should converge to u^+ , and for $\alpha < 0$, our scheme should converge to u^- provided $|\alpha|$ is sufficiently large. However, for $\alpha = 0$, the scheme may converge to either u^+ or u^- depending on the initial guess used for the nonlinear solver. Note that while we cannot globally bound $\partial_{u''} F$ for the operator $F(u'') = 1 - (u'')^2$, we can locally bound $\partial_{u''} F$. Thus, the necessary magnitude for α to allow for selective convergence depends on the initial guess and the solver. Without a global bound on $\partial_{u''} F$, the numerical operator is only locally g-monotone.

Let \bar{u} be the linear interpolant of the boundary data and let the initial guess for u be given by $u^{(0)} = \frac{1}{3}\bar{u} + \frac{2}{3}u^-$ and the initial guesses for p_i be given by $p_i^{(0)} = 0$ for $i = 1, 2, 3$. Thus, the initial guess is closer to u^- . From Fig. 5.10 we see that the scheme converges to u^+ for $\alpha = 4$ and the scheme converges to u^- for $\alpha = 0$ and $\alpha = -4$ for the given parameters. If we change the initial guess to $u^{(0)} = \frac{1}{3}\bar{u} + \frac{2}{3}u^+$, the scheme converges to u^+ for $\alpha = 0$ and $\alpha = 4$ and the scheme converges to u^- for $\alpha = -4$ for the given parameters. Furthermore, for $u^{(0)} = \bar{u}$, *fsolve* does not find a root for $\alpha = 0$, whereas the scheme converges for $\alpha = \pm 4$.

Therefore, the numerical moment plays two major roles. It allows the scheme to converge for a wider range of initial guesses, and it enables the scheme to address the issue of the conditional uniqueness of viscosity solutions. Given the form of the numerical moment, $\alpha(p_1 - 2p_2 + p_3)$, these benefits are even more substantial given the way in which p_1 , p_2 , and p_3 are formed. The three variables only differ in their jump terms. When $\gamma_{01} = \gamma_{02} = \gamma_{03}$, the three different choices for the numerical fluxes (or jump terms) are all equivalent at the PDE level, and often the various jump formulations are presented as interchangeable when discretizing linear and quasilinear PDEs using the DG methodology. Yet, for our schemes for fully nonlinear PDEs, we see that the three different choices of the numerical fluxes all play an essential role at the numerical level when combined to form the numerical moment, even in the degenerative case

Table 5.5. Test 5: Computed solutions at time $T = 3.10$ using RK4 with $\kappa_t = 0.0025$, $\epsilon = 0$, $\alpha = 4$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

r	Norm	$h = 1/4$	$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order
1	L^2	5.0e-01	1.4e-01	1.75	4.3e-02	1.73	1.2e-02	1.81
	L^∞	9.9e-01	3.2e-01	1.63	9.8e-02	1.71	2.8e-02	1.81
2	L^2	7.5e-02	1.7e-02	2.16	3.9e-03	2.10	9.3e-04	2.06
	L^∞	1.0e-01	1.9e-02	2.41	4.1e-03	2.20	9.6e-04	2.10
3	L^2	3.3e-03	2.3e-04	3.87	1.5e-05	3.96	9.1e-07	4.00
	L^∞	6.7e-03	5.2e-04	3.69	3.6e-05	3.85	2.4e-06	3.93
4	L^2	2.2e-04	1.1e-05	4.33	5.7e-07	4.23	2.0e-08	4.87
	L^∞	3.4e-04	1.3e-05	4.65	6.6e-07	4.35	2.4e-08	4.80
5	L^2	6.4e-06	1.1e-07	5.89	1.9e-09	5.83	1.4e-09	0.47
	L^∞	1.4e-05	2.5e-07	5.75	6.5e-09	5.28	1.9e-09	1.80

Table 5.6. Test 6: Computed solutions at time $T = 3.10$ using the explicit Euler method with $\kappa_t = 0.002$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

r	Norm	$h = 1/4$	$h = 1/8$		$h = 1/16$		$h = 1/32$	
		Error	Error	Order	Error	Order	Error	Order
1	L^2	1.9e-01	4.9e-02	1.91	1.3e-02	1.95	3.2e-03	1.98
	L^∞	1.9e-01	4.6e-02	2.01	1.2e-02	1.95	3.0e-03	1.99
2	L^2	7.8e-02	1.9e-02	2.03	4.6e-03	2.05	1.1e-03	2.03
	L^∞	7.0e-02	1.6e-02	2.12	3.6e-03	2.17	8.3e-04	2.11
3	L^2	1.0e-02	5.7e-04	4.16	3.4e-05	4.07	2.1e-06	4.03
	L^∞	8.4e-03	4.7e-04	4.14	2.9e-05	4.01	1.8e-06	4.01
4	L^2	1.3e-03	8.4e-05	3.94	5.1e-06	4.05	3.1e-07	4.05
	L^∞	1.1e-03	6.9e-05	3.93	3.9e-06	4.15	2.2e-07	4.13
5	L^2	1.0e-04	1.4e-06	6.20	2.0e-08	6.12	2.4e-10	6.38
	L^∞	8.6e-05	1.1e-06	6.25	1.5e-08	6.20	2.6e-10	5.89

where $\gamma_{01} = \gamma_{02} = \gamma_{03}$ which will be discussed below.

The role of the numerical moment can heuristically be understood as follows when the numerical moment is rewritten in the form

$$\alpha h^2 \left(\frac{p_1 - 2p_2 + p_3}{h^2} \right).$$

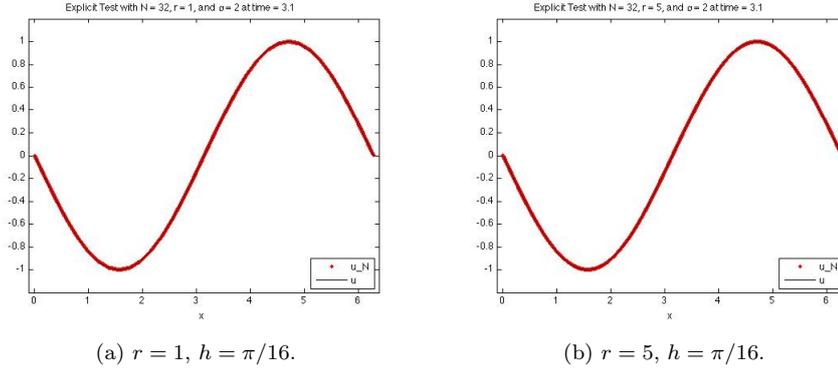
From here, we can see that the numerical moment acts as a centered difference approximation for u_{xxxx} multiplied by a factor that tends to zero with rate $\mathcal{O}(h^2)$. Thus, at the PDE level, we are in essence approximating the nonlinear elliptic operator

$$F(u_{xx}, u_x, u, x)$$

by the quasilinear fourth order operator \widehat{F}_ρ , where

$$\widehat{F}_\rho(u_{xxxx}, u_{xx}, u_x, u, x) = \rho u_{xxxx} + F(u_{xx}, u_x, u, x).$$

In the limit as $\rho \rightarrow 0$, we heuristically expect the unique limit of the fourth order problem to converge to the unique viscosity solution of the second order problem. Using a converging



r	Norm	$h = \pi/2$		$h = \pi/4$		$h = \pi/8$		$h = \pi/16$	
		Error	Error	Order	Error	Order	Error	Order	
1	L^2	2.2e-01	5.3e-02	2.07	1.3e-02	2.02	3.3e-03	2.01	
	L^∞	1.7e-01	4.8e-02	1.87	1.2e-02	1.98	3.1e-03	1.99	
2	L^2	6.0e-02	1.6e-02	1.90	4.2e-03	1.94	1.1e-03	1.97	
	L^∞	6.4e-02	1.5e-02	2.07	3.5e-03	2.13	8.2e-04	2.09	
3	L^2	7.4e-03	6.9e-04	3.43	1.4e-04	2.32	3.5e-05	2.00	
	L^∞	8.0e-03	5.6e-04	3.82	1.0e-04	2.46	2.3e-05	2.14	
4	L^2	2.5e-03	5.7e-04	2.10	1.4e-04	2.03	3.5e-05	2.01	
	L^∞	1.4e-03	3.5e-04	2.01	8.9e-05	1.98	2.2e-05	1.99	
5	L^2	2.2e-03	5.6e-04	2.00	1.4e-04	2.00	3.5e-05	2.00	
	L^∞	1.4e-03	3.6e-04	1.99	8.9e-05	2.00	2.2e-05	2.00	

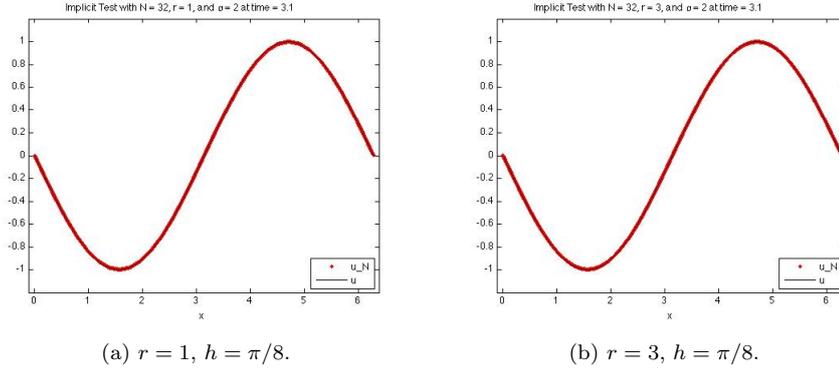
Fig. 5.8. Test 6: Computed solutions at time $T = 3.10$ using the explicit Euler method with $\kappa_t = 0.002$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

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. We refer the reader to [8, 14] for a detailed exposition.

In addition to the connection with the numerical moment to quasilinear fourth order PDEs, we also mention another benefit of the numerical moment. By the choice of α , we can enlarge the domain for which the numerical operator \widehat{F} is increasing in p_1 and p_3 and decreasing in p_2 . Since the definition of ellipticity is based on the monotonicity of the operator, and the issue of conditional uniqueness stems from whether the solution preserves the monotonicity of the operator, building monotonicity into the discretization is important when trying to preserve the nature of the operator we are approximating.

We can demonstrate the power of the monotonicity of the numerical operator with two simple tests. For both tests, we shall again approximate the Monge-Ampère problem from Test 1. However, now we let $\gamma_{01} = \gamma_{02} = \gamma_{03}$. Then, we have $p_2 = \frac{p_1 + p_3}{2}$, which in turn implies that the equation for p_2 is redundant in the formulation and the numerical moment should be zero upon convergence to a root.

For the first test, we again approximate the Monge-Ampère problem from Test 1 while plotting the norm of $p_1 - 2p_2 + p_3$ after each iteration of *fsolve*. From Fig. 5.11, we can see



r	Norm	h = π/2		h = π/4		h = π/8		h = π/16	
		Error	Error	Order	Error	Order	Error	Order	
1	L ²	1.7e-01	4.9e-02	1.82	1.4e-02	1.84	4.8e-03	1.50	
	L [∞]	1.5e-01	4.4e-02	1.78	1.3e-02	1.82	4.1e-03	1.60	
2	L ²	8.0e-02	2.0e-02	2.00	5.9e-03	1.76	3.2e-03	0.87	
	L [∞]	7.0e-02	1.6e-02	2.14	4.0e-03	1.98	1.9e-03	1.06	
3	L ²	1.1e-02	3.0e-03	1.91	2.8e-03	0.09	2.8e-03	0.00	
	L [∞]	8.1e-03	1.8e-03	2.16	1.8e-03	0.01	1.8e-03	0.00	

Fig. 5.9. Test 6: Computed solutions at time $T = 3.10$ using the implicit Euler method with $\Delta t = 0.0062$, $\epsilon = 0$, $\alpha = 2$, $\gamma_{01} = \gamma_{03} = 2$, and $\gamma_{02} = 2.5$.

that even though we expect the moment to be zero based on the redundancy of the equation for p_2 given the equations for p_1 and p_3 , the Newton-based solver *fsolve* treats p_1 , p_2 , and p_3 as independent variables when searching for a root. The monotonicity of each variable appears to aid *fsolve* in the search for a root.

For the second test, instead of using *fsolve*, a Newton-based solver, for solving the nonlinear system of equations, we use the following splitting algorithm:

Algorithm 5.1.

- (1) Pick an initial guess for u , p_1 , and p_3 .
- (2) Solve equation (3.6) for p_2 .
- (3) Solve equation (3.8) for $i = 2$ for u .
- (4) Solve equation (3.8) for $i = 1$ for p_1 .
- (5) Solve equation (3.8) for $i = 3$ for p_3 .
- (6) Repeat Steps 2 - 5 until the change in p_2 is sufficiently small.

We observe that only Step (2) involves the use of a nonlinear solver. Each of Steps (3)-(5) only requires solving a linear system with a constant matrix that can be pre-factored. Thus, the above solver fully decouples the entire system of equations and minimizes the number of unknowns in the nonlinear system. Because this paper is concerned mainly with the discretiza-

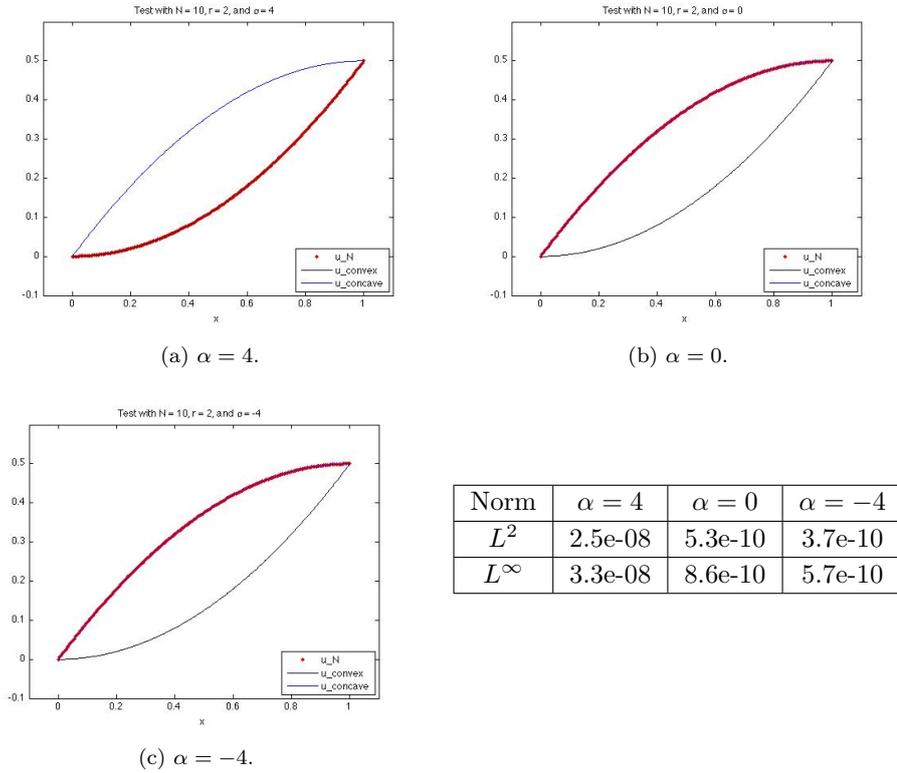


Fig. 5.10. Test 1: $r = 2$, $h = 1/10$, $\epsilon = 0$, $\gamma_{01} = \gamma_{03} = 1.1$, and $\gamma_{02} = 1.5$.

tion of fully nonlinear PDEs, we do not make an effort to compare solvers. The simple solver presented here is meant to demonstrate a potential benefit of using the numerical moment to create monotone numerical operators.

We use Algorithm 5.1 with *fsolve* to execute Step (2) for Test 1. Let the initial guesses be given by $u = u^-$ and $p_1 = p_2 = p_3 = -0.99$. For $p_2 = -0.99$, F is increasing while \hat{F} is decreasing for $\alpha > 0.99$. Since $F(-0.99) > 0$ and \hat{F} is decreasing for $p_2 \geq -1$ when

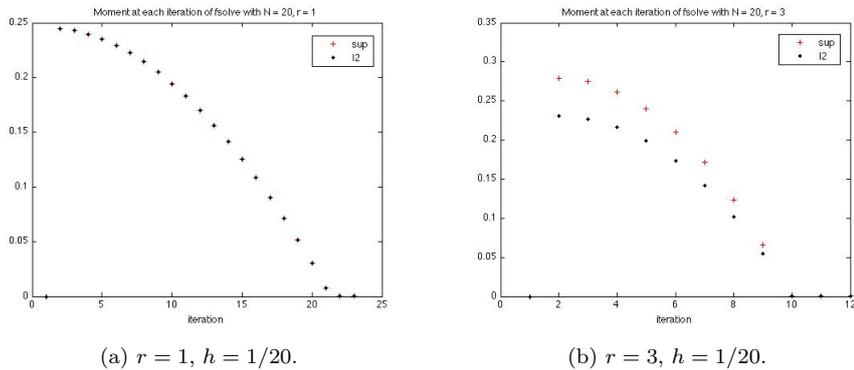
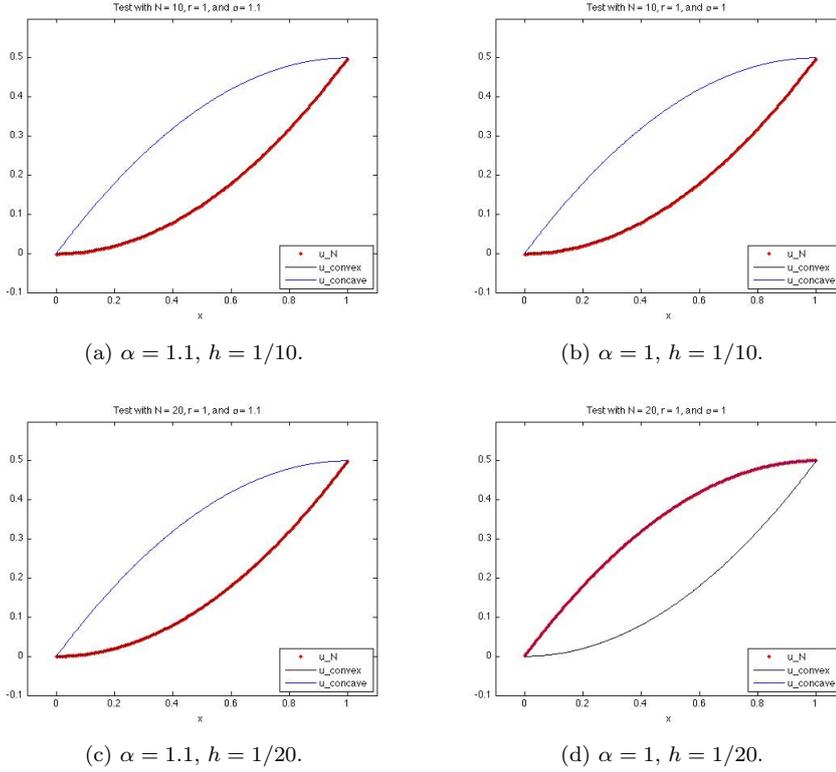


Fig. 5.11. Plots of the norm of $p_1 - 2p_2 + p_3$ with $\alpha = 4$, $\epsilon = 0$, and $\gamma_{01} = \gamma_{02} = \gamma_{03} = 2$ at each iteration of *fsolve*.



α	Norm	$h = 1/5$		$h = 1/10$		$h = 1/20$		$h = 1/40$	
		Error	Error	Order	Error	Order	Error	Order	
4	L^2	6.8e-03	1.7e-03	2.00	4.3e-04	2.00	1.0e-04	2.08	
	L^∞	1.0e-02	2.5e-03	2.00	6.2e-04	2.00	1.6e-04	2.00	
2	L^2	6.8e-03	1.7e-03	2.00	4.3e-04	2.01	9.8e-05	2.12	
	L^∞	1.0e-02	2.5e-03	2.00	6.2e-04	2.00	1.6e-04	2.00	
1.1	L^2	6.8e-03	1.7e-03	2.00	4.2e-04	2.01	9.7e-05	2.13	
	L^∞	1.0e-02	2.5e-03	2.00	6.2e-04	2.00	1.6e-04	2.00	
1	L^2	6.8e-03	1.7e-03	2.00	5.7e-04	1.58	8.2e-04	-0.53	
	L^∞	1.0e-02	2.5e-03	2.00	9.4e-04	1.42	1.2e-03	-0.32	
0.99	L^2	6.0e-03	9.7e-04	2.62	5.7e-04	0.77	8.2e-04	-0.53	
	L^∞	9.9e-03	2.5e-03	2.00	9.4e-04	1.40	1.2e-03	-0.32	
0	L^2	6.8e-03	1.7e-03	2.00	4.3e-04	1.99	1.1e-04	1.96	
	L^∞	1.0e-02	2.5e-03	2.00	6.3e-04	1.99	1.6e-04	1.96	

Fig. 5.12. Test 1 is solved using Algorithm 5.1 with $r = 1$, $\epsilon = 0$, and $\gamma_{01} = \gamma_{02} = \gamma_{03} = 2$. For $\alpha \geq 1.1$, the scheme converges to u^+ . For $\alpha \leq 0.99$, the scheme converges to u^- . When $\alpha = 1.0$, the scheme converges to u^+ for $h \geq \frac{1}{10}$ and the scheme converges to u^- for $h \leq \frac{1}{20}$.

$\alpha > 1$, we expect the splitting algorithm will move away from the concave root $p_2 = -1$. The numerical results are presented in Fig. 5.12. We note that even with the initial guess close to u^- , the solver, with the aid of the numerical moment, converges to u^+ . Similarly, the solver converges to u^+ when $p_1 = p_2 = p_3 > -0.99$ are used as initial guesses. For initial guesses $p_1 = p_2 = p_3 < -1.0$, the solver does not converge. Thus, we see that even for the above simple solver, the monotonicity of \widehat{F} provided by the numerical moment allows the scheme to either

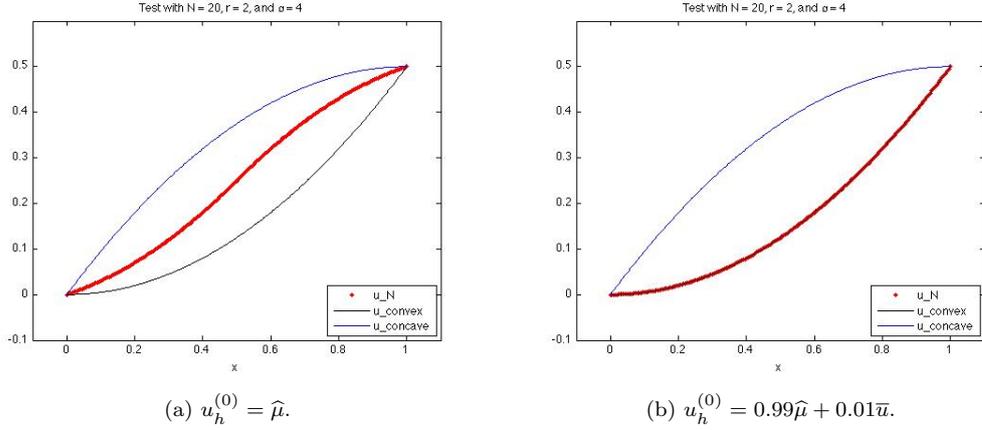


Fig. 5.13. Computed solutions for Test 1 using $r = 2$, $\alpha = 4$, $h = 5.000\text{e-}02$, $\gamma_{01} = \gamma_{03} = 10.0$, $\gamma_{02} = 20.0$, $\epsilon^i = 0$, and Algorithm 5.1.

selectively converge to u^+ or diverge and find no solution. Hence, we again see the benefit of including the numerical moment when tackling the issue of conditional uniqueness for viscosity solutions.

A key concern when discretizing a fully nonlinear PDE is the presence of numerical artifacts, i.e., algebraic solutions that do not correspond to PDE solutions. We continue considering the Monge-Ampère equation from Test 1. In addition to the two classical PDE solutions, there exists infinitely many C^1 functions that satisfy the PDE and boundary conditions almost everywhere with respect to the Lebesgue measure, as seen by $\hat{\mu}$ defined below in (5.1). These almost everywhere solutions will correspond to numerical artifacts due to the fact algebraic solutions for a given discretization may correspond to these functions. It is well known that using a standard linear discretization scheme for the Monge-Ampère problem can yield multiple solutions, many of which are numerical artifacts that do not correspond to PDE solutions (cf. [8]). For example, let $\hat{\mu} \in H^2(0,1) \setminus C^2(0,1)$ be defined by

$$\hat{\mu}(x) = \begin{cases} \frac{1}{2}x^2 + \frac{1}{4}x, & \text{for } x < 0.5, \\ -\frac{1}{2}x^2 + \frac{5}{4}x - \frac{1}{4}, & \text{for } x \geq 0.5. \end{cases} \quad (5.1)$$

Furthermore, suppose $x = 0.5$ is a node in the partition. Then, when using a standard IPDG discretization, $\hat{\mu}$ corresponds to a numerical solution, yielding a numerical artifact.

We first explore the possibility that the discretization contains numerical artifacts such as $\hat{\mu}$. Let $r = 2$ in V^h . Then, $\hat{\mu} \in C^1(\Omega) \cap V^h$. Thus, we have $[\hat{\mu}] = [\hat{\mu}'] = 0$ on \mathcal{E}_h^I and $\hat{\mu} = g$ on $\partial\Omega$. Therefore, whenever $x = 0.5$ is a node in the partition, we have $u_h = \hat{\mu}$ and $p_{ih}(x) = 1$ if $x < 0.5$ and $p_{ih}(x) = -1$ if $x > 0.5$ for $i = 1, 2, 3$ is a numerical solution, and it follows that our discretization does have numerical artifacts when $r \geq 2$. We can see the presence of a numerical artifact in Fig. 5.13. The function $\hat{\mu}$ corresponds to a fixed point for the solver. However, if we slightly perturb the initial guess away from $\hat{\mu}$, we see that Algorithm 5.1 converges to u^+ . Unfortunately, the Newton algorithm *fsolve* does still converge to $\hat{\mu}$ with the same slightly perturbed initial guess. Thus, for $r = 2$, our discretization does contain numerical artifacts that must be addressed at the solver level. We do note that in all testing, Algorithm 5.1 only converged to a numerical artifact when the numerical artifact corresponded to an initial guess.

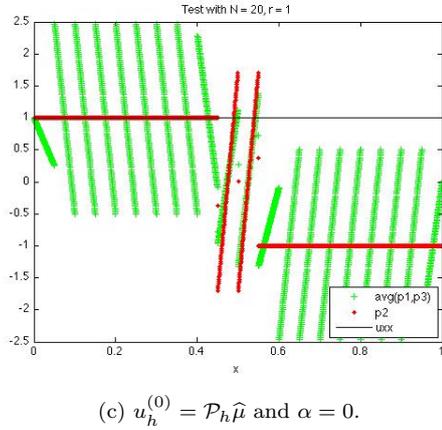
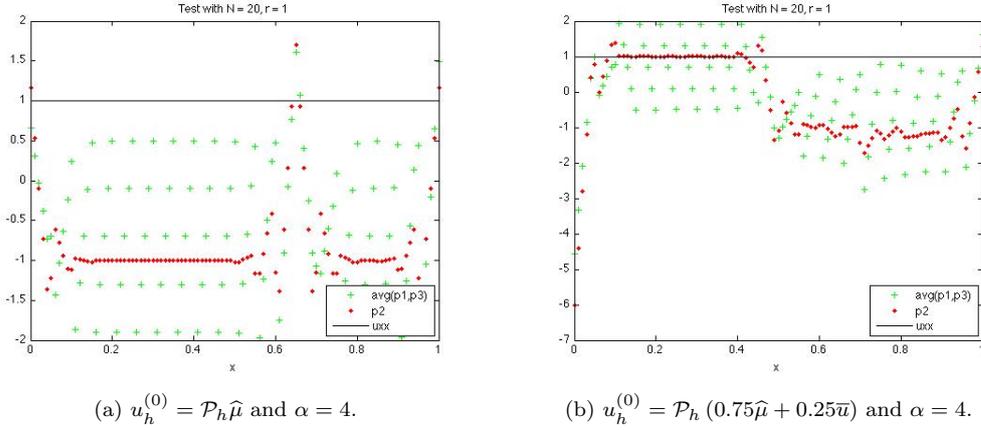


Fig. 5.14. Computed solutions for Test 1 using $r = 1$, $h = 5.000\text{e-}02$, $\gamma_{01} = \gamma_{03} = 10.0$, $\gamma_{02} = 20.0$, $\epsilon^i = 0$, and *fsolve*.

We now consider the presence of numerical artifacts when $r = 1$ in V^h . Then, $\hat{\mu} \notin V^h$. Furthermore, the only C^1 function in V^h that satisfies the boundary condition is \bar{u} . Thus, we expect the numerical moment to have an effect since a good approximation for u^\pm or $\hat{\mu}$ must have jumps in the gradient along \mathcal{E}_h^I . Since we cannot determine analytically if there is a numerical solution in V^h that corresponds to $\hat{\mu}$, we will explore the possibility numerically. To this end, we approximate Test 1 with $r = 1$ using *fsolve* and initial guesses that correspond to functions near $\hat{\mu}$. The results can be found in Fig. 5.14, where we plot the resulting values of p_{2h} and note that p_{2h} near -1 corresponds to u^- and p_{2h} near 1 corresponds to u^+ . Observe, for the initial guess $u_h^{(0)} = \mathcal{P}_h \hat{\mu}$, the solution appears to converge to a function near u^- . While the final approximation has a small residual, $\mathcal{O}(10^{-26})$, the last step for the solver was ineffective according to the error flags for *fsolve*. When approximating u^- with $r = 1$ by using a negative value for the coefficient of the numerical moment, the plot for p_{2h} is near -1 over the entire domain. Thus, while it is unclear if the approximation actually converged to u^- , it is clear that the approximation converged away from $\hat{\mu}$. For the initial guess $u_h^{(0)} = \mathcal{P}_h (0.75\hat{\mu} + 0.25\bar{u})$, the solver does not find a root after 106 iterations. Instead, *fsolve* appears to be trapped in a small-residual well. After the 100th iteration, the residual is about 0.007. Thus, the discretization

does not appear to have a numerical solution that corresponds to $\widehat{\mu}$ when $r = 1$. In contrast, when we set $\alpha = 0$, we clearly converge to a numerical artifact that corresponds to $\widehat{\mu}$. One last observation from Fig. 5.14 is that $p_{1h} - 2p_{2h} + p_{3h}$ is nonzero in all three plots, as expected when using $r = 1$ paired with the lower regularity of $\widehat{\mu}$. The same tests using Algorithm 5.1 all converged to the viscosity solution.

6. Conclusion

In this paper we present a general framework for constructing high order interior penalty discontinuous Galerkin methods for approximating viscosity solutions of fully nonlinear second order elliptic and parabolic PDEs. The proposed framework extends the (second order) finite difference framework developed by the authors in [9] to a more flexible DG framework, allowing for the approximation of fully nonlinear PDEs using high order polynomials and non-uniform meshes. Various numerical experiments are provided to show the performance of the proposed methodology. The proposed DG framework is based on a nonstandard mixed formulation of the underlying fully nonlinear PDE. In order to capture discontinuities of the second order derivative u_{xx} of the solution u , three independent functions p_1, p_2 and p_3 are introduced to accomplish the goal, where p_1 and p_3 measure the left and right limits of u'' . If u'' is discontinuous, p_1 and p_3 can be used to gain insight into the discontinuity upon convergence. Thus, the methodology has the ability to capture some of the more interesting aspects of the viscosity solutions. The proposed mIP-DG methodology takes the most important aspects of the companion finite difference framework of [9] and extends them in multiple directions. For example, by adopting and expanding the idea of numerical operators, the mIP-DG formulation allows for even more flexibility than finite difference methods in construction.

The proposed mIP-DG discretizations touch the inner core and make use of the full potential of the DG methodology. This is because there is a natural match among the three choices of possible numerical fluxes and the three numerical second order derivatives p_1, p_2 , and p_3 . The flexibility of DG methods allows the implantation of this connection into the formulation of the proposed mIP-DG methods.

As in the finite difference framework, the g-monotonicity (generalized monotonicity) and the numerical moment play a central role in the proposed mIP-DG framework. The g-monotonicity gives the mIP-DG methods the ability to select the mathematically “correct” solution (i.e., the viscosity solution) among all possible solutions, and the numerical moment is the catalyst which facilitates the g-monotonicity of the proposed mIP-DG methods. Moreover, the g-monotonicity allows for the possible development of more efficient/selective (than generic Newton) solvers. The special nonlinearity of the algebraic systems can be explored to decouple the equations as seen in Algorithm 5.1. We believe that one of the main strengths of the mIP-DG formulation presented in this paper lies in the way in which the discretization handles the nonlinearity. The discretization takes a nonlinear problem and embeds it into a mostly linear system of equations where the nonlinearity has been modified to ensure g-monotonicity. The added monotonicity can theoretically enlarge the domain of valid initial guesses over which a solver will converge. Thus, the weak coupling with linear equations is only a small penalty for the added structure in the nonlinearity.

We also remark that the role and benefit of the symmetrizing parameter is unclear for nonlinear systems of equations. When γ_0 is sufficiently large, we observe that numerical results seem independent of the choice for ϵ . However, when the penalty constant γ_0 is not sufficiently

large, the inclusion of ϵ can be detrimental to the approximation. For small γ_0 , the approximation is allowed to have larger jumps occur. When the jumps become too large, the effect from having ϵ present becomes exaggerated, and the overall accuracy of the approximation begins to suffer beyond just the presence of jumps. For elliptic problems, we can see the formation of a boundary layer. For dynamic problems, we see the approximation actually diverging (almost instantaneously) along the interior of the domain even with small timesteps. As expected, when γ_0 increases and becomes sufficiently large, these phenomena disappear. Thus, at a numerical level, the presence of the symmetrizing constant ϵ seems important, even though at the continuous level for the PDE, the symmetrization terms all become zero.

Conceptually, the mIP-DG framework presented in this paper can be easily extended to the high dimensional fully nonlinear PDE problems, the detailed exposition will be given in a forthcoming paper [11]. On the other hand, the proposed mIP-DG framework may not work in the case when the viscosity solution does not belong to $H^1(\Omega)$. In such a case, a more involved mixed local discontinuous Galerkin (mLDG) framework must be invoked. We refer the interested reader to [12] for a detailed exposition.

Acknowledgments. The work of the both authors was partially supported by the NSF grants DMS-1016173 and DMS-1318486.

References

- [1] M. Bardi, I. Capuzzo-Dolcetta. Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations, Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA, 1997.
- [2] G. Barles, P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, **4**:3 (1991), 271-283.
- [3] L.A. Caffarelli, X. Cabré. Fully nonlinear elliptic equations, Vol. 43 of American Mathematical Society Colloquium Publications. American Mathematical Society, Providence, RI, 1995.
- [4] L.A. Caffarelli, P.A. Souganidis. A rate of convergence for monotone finite difference approximations to fully nonlinear, uniformly elliptic PDEs. *Comm. Pure Appl. Math.*, **61** (2008), 1-17.
- [5] M.G. Crandall, P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, **277**:1 (1983), 1-42.
- [6] M.G. Crandall, P.L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, **43** (1984), 1-19.
- [7] D. Gilbarg, N. S. Trudinger. Elliptic partial differential equations of second order, Classics in Mathematics. Springer-Verlag, Berlin, 2001, reprint of the 1998 edition.
- [8] X. Feng, R. Glowinski, M. Neilan. Recent developments in numerical methods for second order fully nonlinear PDEs. *SIAM Review*, **55**:2 (2013), 205-267.
- [9] X. Feng, C. Kao, T. Lewis. Convergent finite difference methods for one-dimensional fully nonlinear second order partial differential equations. *J. Comput. Appl. Math.*, **254** (2013), 81-98.
- [10] X. Feng, C.Y. Kao, T. Lewis. Monotone finite difference methods for the Monge-Ampère equation in high dimensions. in preparation.
- [11] T. Lewis, X. Feng, Mixed interior penalty discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions, *Numer. Methods for PDEs* (to appear).
- [12] T. Lewis, X. Feng, Local discontinuous Galerkin methods for one-dimensional second order fully nonlinear elliptic and parabolic equations, *J. Scient. Comput.*, DOI 10.1007/s10915-013-9763-3.
- [13] X. Feng, M. Neilan. Vanishing moment method and moment solutions for second order fully nonlinear partial differential equations. *J. Scient. Comput.*, **38** (2008), 74-98.

- [14] X. Feng, M. Neilan. The vanishing moment method for fully nonlinear second order partial differential equations: formulation, theory, and numerical analysis. arxiv.org/abs/1109.1183v2.
- [15] N.V. Krylov. The rate of convergence of finite-difference approximations for Bellman equations with Lipschitz coefficients. *Appl. Math. Optim.*, **52**:3 (2005), 365-399.
- [16] H.J. Kuo, N.S. Trudinger. Discrete methods for fully nonlinear elliptic equations. *SIAM J. Numer. Anal.*, **29**:1 (1992), 123-135.
- [17] G.M. Lieberman. Second order parabolic differential equations. World Scientific Publishing Co. Inc., River Edge, NJ, 1996.
- [18] J.A. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1970/71.
- [19] B. Rivière. Discontinuous Galerkin methods for solving elliptic and parabolic equations, volume 35 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
- [20] J. Yan, S. Osher. Direct discontinuous local Galerkin methods for Hamilton-Jacobi equations. *J. Comp. Phys.*, **230** (2011), 232-244.