

Mixed Interior Penalty Discontinuous Galerkin Methods for Fully Nonlinear Second Order Elliptic and Parabolic Equations in High Dimensions

By: Xiaobing Feng, [Thomas Lewis](#)

This is the accepted version of the following article:

X. Feng and T. Lewis. Mixed interior penalty discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. *Numer. Methods Partial Differential Equations*, 30(5), 1538-1557. doi: 10.1002/num.21856,

which has been published in final form at <http://dx.doi.org/10.1002/num.21856>.

***© Wiley. Reprinted with permission. No further reproduction is authorized without written permission from Wiley. This version of the document is not the version of record. Figures and/or pictures may be missing from this format of the document. ***

Abstract:

This article is concerned with developing efficient discontinuous Galerkin methods for approximating viscosity (and classical) solutions of fully nonlinear second-order elliptic and parabolic partial differential equations (PDEs) including the Monge–Ampère equation and the Hamilton–Jacobi–Bellman equation. A general framework for constructing interior penalty discontinuous Galerkin (IP-DG) methods for these PDEs is presented. The key idea is to introduce multiple discrete Hessians for the viscosity solution as a means to characterize the behavior of the function. The PDE is rewritten in a mixed form composed of a single nonlinear equation paired with a system of linear equations that defines multiple Hessian approximations. To form the single nonlinear equation, the nonlinear PDE operator is replaced by the projection of a numerical operator into the discontinuous Galerkin test space. The numerical operator uses the multiple Hessian approximations to form a numerical moment which fulfills consistency and g-monotonicity requirements of the framework. The numerical moment will be used to design solvers that will be shown to help the IP-DG methods select the “correct” solution that corresponds to the unique viscosity solution. Numerical experiments are also presented to gauge the effectiveness and accuracy of the proposed mixed IP-DG methods.

Keywords: Fully nonlinear PDEs | viscosity solutions | discontinuous Galerkin methods

Article:

I Introduction

General second-order partial differential equations (PDEs) have the form

$$F(D^2u, \nabla u, u, x) = 0,$$

where $D^2u(x)$ and $\nabla u(x)$ denote the Hessian and gradient of u at x , respectively. PDEs are often classified based on the nonlinearity of the PDE operator F . A fully nonlinear PDE corresponds to an equation where the operator F is nonlinear in the highest-order derivative(s) appearing in the PDE. The theory for linear, semilinear, and quasi-linear PDEs is well studied and can be considered classical in many situations. In contrast, fully nonlinear PDEs are still at the forefront of PDE analysis, and the area of numerical PDEs for fully nonlinear PDEs is still in its infancy. An overview of numerical methods for fully nonlinear second-order PDEs and their applications can be found in the survey paper [1].

This is the fourth paper in a series [2-4] which is devoted to developing finite difference (FD) and discontinuous Galerkin (DG) methods for approximating viscosity solutions of fully nonlinear second-order elliptic and parabolic equations:

$$F[u](x) := F(D^2u, \nabla u, u, x) = 0, \quad x \in \Omega \subset \mathbb{R}^d \quad (1.1)$$

and

$$u_t + F(D^2u, \nabla u, u, x, t) = 0, \quad (x, t) \in \Omega_T := \Omega \times (0, T], \quad (1.2)$$

which are complemented by appropriate Dirichlet boundary conditions and initial conditions. The goal of this article is to design and implement a class of interior penalty DG (IP-DG) methods for high-dimension ($d \geq 2$) problems.

The IP-DG methods proposed in this article aim to directly approximate viscosity solutions of (1.1) and (1.2) which belong to $H^1(\Omega)$ in the spatial variables. As in the 1-d case [3], the methods of this article will be based on a nonstandard mixed formulation. The main idea is to introduce multiple Hessian approximations using an IP-DG methodology and then introduce a numerical operator that incorporates the multiple discrete Hessians in forming a numerical moment. On the other hand, there are a couple of additional difficulties associated with the high-dimensional problems that were not addressed in [3]. First, the concept of left and right Hessians at a face/edge is ambiguous. Therefore, a systematic scheme must be introduced to yield the correct interpretation. Second, the Hessian matrix D^2u now contains mixed second-order derivatives. The mixed second-order derivatives must be handled appropriately in the discretization, and as we will see, when choosing a solver.

The remainder of this article is organized as follows. In Section II, we record some preliminaries including the definition of viscosity solutions and our notation. In Section III, we formulate and discuss the IP-DG methods for elliptic problems, and then we extend our formulation to parabolic problems in Section IV. A discussion regarding solving the resulting nonlinear equations is given in Section V. Finally, in Section VI, we present some numerical experiments for the proposed IP-DG methods that verify the accuracy and efficiency of our methods. The

numerical experiments will also target understanding the role of the numerical moment, a key component that is used to design schemes that fulfill the proposed framework.

II Preliminaries

We begin by introducing some basic definitions and background regarding fully nonlinear PDEs. We also introduce some (standard) DG notation that will be used throughout the article.

A Background

For presentation purposes, we adopt standard function and space notations as in [5] and [6]. For example, for a bounded open domain $\Omega \subset \mathbb{R}^d$, $B(\Omega)$, $USC(\Omega)$, and $LSC(\Omega)$ are used to denote, respectively, the spaces of bounded, upper semicontinuous, and lower semicontinuous functions on Ω . Also, 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.

In general, solutions may not exist for fully nonlinear second-order problems. Thus, we choose to impose some structure on fully nonlinear second-order problems represented by (1.1). More precisely, we impose an ellipticity requirement. The following definition is standard (cf. [5-7]).

Definition 2.1.. Eq. ((1.1)) is said to be elliptic if, for all $(\mathbf{p}, \lambda, x) \in \mathbb{R}^d \times \mathbb{R} \times \bar{\Omega}$, there holds

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

where $A \geq B$ means that $A - B$ is a nonnegative definite matrix.

Definition 2.2.. Assume the second-order operator F in (1.1) is elliptic in a function class $\mathcal{A} \subset B(\Omega)$. The function $u \in \mathcal{A}$ is called a viscosity subsolution (resp. supersolution) of (1.1) if $\forall \varphi \in C^2(\bar{\Omega})$, when $u^ - \varphi$ (resp. $u_* - \varphi$) has a local maximum (resp. maximum) at $x_0 \in \bar{\Omega}$,*

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

(resp. $F^(D^2\varphi(x_0), \nabla\varphi(x_0), u_*(x_0), x_0) \geq 0$). The function $u \in \mathcal{A}$ is called a viscosity solution of (1.1) if u is both a viscosity subsolution and a viscosity supersolution of (1.1).*

We note that the ellipticity assumption on the operator F in the definition is not necessary. However, the ellipticity assumption is used when proving the existence of viscosity solutions. A self-contained overview of viscosity solution theory for second-order problems can be found in [8].

We end this section with a few comments regarding the numerical approximation of viscosity solutions. By the definition of viscosity solutions, we see that an approximation method must be able to capture low-regularity functions. However, to make the situation more difficult, an approximation method must also have a mechanism for filtering the possibly infinite number of lower-regularity functions that satisfy the PDE almost everywhere whenever the viscosity solution has higher regularity. Such low-regularity almost everywhere solutions can correspond to algebraic solutions of the system of equations that results from discretizing a fully nonlinear PDE. These (false) algebraic solutions are referred to as numerical artifacts resulting from the discretization, and these numerical artifacts are known to plague the numerical discretization of fully nonlinear PDEs when using or adapting standard numerical methods for linear, semilinear, and quasilinear PDEs. Additionally, viscosity solutions may be unique only in a restrictive function class \mathcal{A} , a property referred to as conditional uniqueness. Consequently, the numerical solutions must also belong to a discrete function class that is consistent with \mathcal{A} . Also, due to the nonlinearity of the PDE, multiplication by a test function and using integration by parts is not possible. In fact, the definition of viscosity solutions is entirely nonvariational. Thus, Galerkin-based methodologies are not immediately applicable for approximating fully nonlinear PDEs. Instead, the viscosity solution concept is based on a “differentiation by parts” approach, an entirely local definition that has no known discrete analogue.

B. Notation

To develop our schemes, we first introduce some (standard) notation for DG methods. We denote $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. Let \mathcal{T}_h denote a locally quasi-uniform and shape-regular partition of the domain Ω , see [9], with $h = \max_{K \in \mathcal{T}_h} \text{diam } K$. We define the following broken H^1 -space and broken C^0 -space

$$H^1(\mathcal{T}_h) := \prod_{K \in \mathcal{T}_h} H^1(K), \quad C^0(\mathcal{T}_h) := \prod_{K \in \mathcal{T}_h} C^0(\bar{K}),$$

and the broken L^2 -inner product

$$(v, w)_{\mathcal{T}_h} := \sum_{K \in \mathcal{T}_h} \int_K v w \, dx \quad \forall v, w \in L^2(\mathcal{T}_h).$$

Let \mathcal{E}_h^I denote the set of all interior faces/edges of \mathcal{T}_h , \mathcal{E}_h^B denote the set of all boundary faces/edges of \mathcal{T}_h , and $\mathcal{E}_h := \mathcal{E}_h^I \cup \mathcal{E}_h^B$. Then, for a set $\mathcal{S}_h \subset \mathcal{E}_h$, we define the broken L^2 -inner product over \mathcal{S}_h by

$$(v, w)_{\mathcal{S}_h} := \sum_{e \in \mathcal{S}_h} \int_e v w \, ds \quad \forall v, w \in L^2(\mathcal{S}_h).$$

For a fixed integer $r \geq 0$, we define the standard DG finite element space $V^h \subset H^1(\mathcal{T}_h) \subset L^2(\mathcal{T}_h)$ by

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

where $\mathbb{P}_r(K)$ denotes the set of all polynomials on K with degree not exceeding r .

We now define (standard) interior face/edge-dependent functions. Choose $K, K' \in \mathcal{T}_h$, and let $e = \partial K \cap \partial K' \in \mathcal{E}_h^I$. Without loss of generality, we assume that the global labeling number of K is smaller than that of K' and define the following (standard) jump and average notation:

$$[v]_e := v|_K - v|_{K'}, \quad \{v\}_e := \frac{v|_K + v|_{K'}}{2} \quad (2.1)$$

and the (standard) left and right limit notation:

$$v^+|_e := v|_{K'}, \quad v^-|_e := v|_K \quad (2.2)$$

for any $v \in H^1(\mathcal{T}_h)$. We also define $n_e := n_K = -n_{K'}$ as the normal vector to e . For $e \in \mathcal{E}_h^B$, we define n_e as the unit outward normal for the underlying boundary simplex. By convention, we define function values on \mathcal{E}_h^B as interior limits. Observe, when Ω is a d -rectangle and \mathcal{T}_h corresponds to a Cartesian partition with the simplexes labeled according to the natural ordering, v^- corresponds to a limit from the “left” and v^+ corresponds to a limit from the “right.”

Last, we define the L^2 -projection operator $\mathcal{P}_h : L^2(\mathcal{T}_h) \rightarrow V^h$ by

$$(\mathcal{P}_h v, \phi_h)_{\mathcal{T}_h} = (v, \phi_h)_{\mathcal{T}_h} \quad \forall \phi_h \in V^h \quad (2.3)$$

for all $v \in L^2(\mathcal{T}_h)$.

III A Mixed IP-DG Framework for Second-Order Elliptic Problems

We now develop a class of IP-DG methods for the following boundary value problem

$$2F[u](x) := F(D^2u, \nabla u, u, x) = 0, \quad x \in \Omega \subset \mathbb{R}^d, \quad (3.1a)$$

$$u(x) = g(x), \quad x \in \partial\Omega, \quad (3.1b)$$

where F is a fully nonlinear elliptic operator; Ω is an open, bounded, polygonal domain; and the viscosity solution $u \in H^1(\Omega)$.

A Motivation

Due to the lower regularity of the viscosity solution, D^2u may not exist. Thus, we will use three possible approximations for D^2u , namely, the left and right limits, as well as their average. Using a numerical operator that can handle multiple approximations for D^2u , we rewrite (3.1) in mixed form as

$$\hat{F}(P^+, P, P^-, \nabla u, u, x) = 0, \quad (3.2a)$$

$$P^+(x) - D^2u(x^+) = 0, \quad (3.2b)$$

$$P(x) - D^2u(x^a) = 0, \quad (3.2c)$$

$$P^-(x) - D^2u(x^-) = 0 \quad (3.2d)$$

for all $x \in \Omega$, where $D^2u(x^a)$ can be thought of as the arithmetic average of $D^2u(x^+)$ and $D^2u(x^-)$.

We now formalize the definition and properties of a numerical operator. The following definitions are high-dimensional counterparts of the one-dimensional (1D) definitions in [3].

Definition 3.1..

i. The function $\hat{F} : (\mathbb{R}^{d \times d})^3 \times \mathbb{R}^d \times \mathbb{R} \times \Omega \rightarrow \mathbb{R}$ in (3.2a) is called a numerical operator.

ii. Let $P \in \overline{\mathbb{R}^{d \times d}}, q \in \overline{\mathbb{R}^d}, v \in \mathbb{R}$, and $x \in \overline{\Omega}$. The numerical operator \hat{F} in (3.2a) is said to be consistent if \hat{F} satisfies

$$\liminf_{\substack{P^+, P^-, P \rightarrow P, \zeta \rightarrow q; \\ v \rightarrow v, \xi \rightarrow x}} \hat{F}(P^+, P^-, P, \zeta, v, \xi) \geq F_*(P, q, v, x),$$

$$\limsup_{\substack{P^+, P^-, P \rightarrow P, \zeta \rightarrow q; \\ v \rightarrow v, \xi \rightarrow x}} \hat{F}(P^+, P^-, P, \zeta, v, \xi) \leq F^*(P, q, v, x),$$

where F_* and F^* denote, respectively, the lower and the upper semicontinuous envelopes of F . Thus, we have

$$F_*(P, q, v, x) := \liminf_{\substack{\tilde{P} \rightarrow P, \tilde{q} \rightarrow q, \\ \tilde{v} \rightarrow v, \tilde{x} \rightarrow x}} F(\tilde{P}, \tilde{q}, \tilde{v}, \tilde{x}),$$

$$F^*(P, q, v, x) := \limsup_{\substack{\tilde{P} \rightarrow P, \tilde{q} \rightarrow q, \\ \tilde{v} \rightarrow v, \tilde{x} \rightarrow x}} F(\tilde{P}, \tilde{q}, \tilde{v}, \tilde{x}),$$

where $\tilde{P} \in \mathbb{R}^{d \times d}, \tilde{q} \in \mathbb{R}^d, \tilde{v} \in \mathbb{R}$, and $\tilde{x} \in \Omega$.

iii. The numerical operator \hat{F} in (3.2a) is said to be generalized monotone (g-monotone) if $\hat{F}(P^+, P, P^-, q, v, x)$ is monotone increasing in P^+ and P^- and monotone decreasing in P . More precisely, for all $P^+, P, P^- \in \mathbb{R}^{d \times d}, q \in \mathbb{R}^d, v \in \mathbb{R}$, and $x \in \Omega$, there holds

$$\hat{F}(A, P, P^-, q, v, x) \leq \hat{F}(B, P, P^-, q, v, x),$$

$$\hat{F}(P^+, A, P^-, q, v, x) \geq \hat{F}(P^+, B, P^-, q, v, x),$$

$$\hat{F}(P^+, P, A, q, v, x) \leq \hat{F}(P^+, P, B, q, v, x),$$

for all $A, B \in \mathcal{S}^{d \times d}$ such that $A \leq B$, where $A \leq B$ means that $B - A$ is a nonnegative definite matrix. In other words, $\hat{F}(\uparrow, \downarrow, \uparrow, q, v, x)$.

To ensure the g-monotonicity property of the numerical operator in (3.2a), we introduce a numerical moment and we propose the following Lax–Friedrichs-like numerical operators:

$$\hat{F}_1(P^+, P, P^-, q, \lambda, \xi) := F(P, q, \lambda, \xi) + \alpha_1 : (P^+ - 2P + P^-), \quad (3.4a)$$

$$\hat{F}_2(P^+, P, P^-, q, \lambda, \xi) := F\left(\frac{P^+ + P + P^-}{3}, q, \lambda, \xi\right) + \alpha_2 : (P^+ - 2P + P^-), \quad (3.4b)$$

where $\alpha_1, \alpha_2 \in \mathbb{R}^{d \times d}$ are positive constant matrices chosen to enforce the g-monotonicity property of \hat{F} and the last term in (3.4a) and (3.4b) is called the numerical moment. Notationally, we denote the Frobenius inner product; that is,

$$A : B := \sum_{k, \ell=1}^d a_{k\ell} b_{k\ell}$$

for all $A, B \in \mathbb{R}^{d \times d}$. To ensure \hat{F}_1 is g-monotone, we require α_1 to be symmetric and

$$2\alpha_1 > \pm \frac{\partial F}{\partial D^2 u}, \quad (3.5)$$

assuming adequate regularity for the differential operator F . Observe, for F differentiable, \hat{F}_1 is g-monotone with respect to q if $\alpha_1 > \mathbf{0}$, and \hat{F}_1 is g-monotone with respect to P if

$$\mathbf{0} > \frac{\partial \hat{F}_1}{\partial P} = \frac{\partial F}{\partial D^2 u} - 2\alpha_1,$$

where we again assume $\alpha_1 > \mathbf{0}$.

Remark 3.1.

- a. The g-monotonicity property is motivated by the definition of ellipticity. We wish to impose that the numerical operator \hat{F} is unconditionally elliptic with respect to a Hessian argument even in cases where the fully nonlinear operator F is conditionally elliptic, that is, the Monge–Ampère operator considered in the numerical tests found in Section VI. However, we must balance any artificially imposed monotonicity using our other Hessian arguments if we wish for our numerical operator to be faithful to the original PDE operator. The choice that \hat{F} is decreasing with respect to P and increasing with respect to P^- and P^+ is motivated both by symmetry and the results of [2].

- b. The H^1 assumption emphasizes the idea that a single gradient approximation is sufficient to capture the behavior corresponding to the gradient of the viscosity solution. When the gradient of the viscosity solution does not exist or has jumps, we advocate the local discontinuous Galerkin (LDG) methods found in [4] and a forthcoming high-dimensional paper where multiple gradient approximations are introduced as unknowns. Such an approach is not possible using the IP-DG methodology that is based on a piecewise gradient operator that can be hard-computed when using a piecewise polynomial bases. However, by introducing a numerical moment that compares Hessian approximations, the H^1 assumption may not be necessary, and it can be thought of as a formal assumption for motivating the formulation of the IP-DG methods.

B Formulation

We now formalize our IP-DG discretization of (3.2). We discretize (3.2a) by simply using its broken L^2 -projection into V_h , namely,

$$a_0(u_h, P_h^+, P_h, P_h^-; \phi_{0h}) = 0 \quad \forall \phi_{0h} \in V^h, \quad (3.6)$$

where

$$a_0(u, P^+, P, P^-; \phi) = (\hat{F}(P^+, P, P^-, \nabla u, u, \cdot), \phi)_{T_h}.$$

Note, for $u_h \in V^h$, ∇u_h is defined piecewise on each simplex $K \in T_h$.

Next, we discretize the three linear equations in (3.2). To this end, we first introduce some well-known identities (often referred to as “magic formulas” in the literature) defined on \mathcal{E}_h^I for functions in V_h :

$$[v w] = v^- [w] + [v] w^+, \quad (3.7a)$$

$$\{v w\} = \{v\} [w] + [v] \{w\}, \quad (3.7b)$$

$$[v w] = v^+ [w] + [v] w^- \quad (3.7c)$$

on \mathcal{E}_h^I for all $v, w \in V^h$, where $[\cdot]$ and $\{\cdot\}$ are defined in (2.1) and v^+ and v^- are defined in (2.2).

We also introduce (standard) C^0 interior-penalty terms. Let $\gamma_{k\ell}^{0*} \geq 0$ for $k, \ell = 1, 2, \dots, d$ denote interior-penalty parameters, where $*$ takes $+$, $-$, and empty value. It will be clear later that to avoid redundancy of the three equations for P_h^+ , P_h , and P_h^- , we need to require that $\gamma_{k\ell}^0 \geq \max\{\gamma_{k\ell}^{0+}, \gamma_{k\ell}^{0-}\}$ for all $k = 1, 2, \dots, d$ and $\gamma_{kk}^0 > \max\{\gamma_{kk}^{0+}, \gamma_{kk}^{0-}\}$ for all $k = 1, 2, \dots, d$. Then, we define the interior-penalty terms by

$$J_{k\ell}^{0*}(v, w) := \sum_{e \in \mathcal{E}_h^I} \frac{\hat{\gamma}_{k\ell}^{0*}}{h_e} \langle [v], [w] \rangle_e, \quad (3.8)$$

where $h_e := \text{diam } e$ when $d \geq 2$ and $*$ takes $+$, $-$, and empty value. A C^1 interior-penalty term could also be introduced for higher regularity solutions; however, we will see in the numerical tests that a C^1 penalty may be too restrictive inside the viscosity solution setting.

To discretize the three linear equations in (3.2), we use the integration by parts formula

$$\int_S v_{x_k x_\ell} \varphi \, dx = \int_{\partial S} v_{x_k} \varphi n_\ell \, ds - \int_S v_{x_k} \varphi_{x_\ell} \, dx$$

for all $v, \varphi \in H^2(S)$, for $k, \ell = 1, 2, \dots, d$. Thus, we formally have

$$(u_{x_k x_\ell}, \phi)_{\mathcal{T}_h} = \langle [u_{x_k} \phi], n_\ell \rangle_{\mathcal{E}_h^I} + \langle u_{x_k}, \phi n_\ell \rangle_{\mathcal{E}_h^B} - (u_{x_k}, \phi_{x_\ell})_{\mathcal{T}_h} \quad \forall \phi \in H^1(\mathcal{T}_h) \quad (3.9)$$

for all $u \in H^2(\mathcal{T}_h)$, for $k, \ell = 1, 2, \dots, d$.

Combining the integral identity (3.9) with (3.7) and introducing standard penalty and jump terms, we can now fully discretize the auxiliary linear equations in (3.2). To this end, we let $\epsilon_{k\ell}^* \in \{-1, 0, 1\}$, $k, \ell = 1, 2, \dots, d$, where $*$ takes $+$, and empty value, denote the ‘‘symmetrization’’ parameters [10]. Then, we define the auxiliary variables $P_h^+, P_h, P_h^- \in [V^h]^{d \times d}$ by

$$a_{k\ell}^+(u, \phi) = b_{k\ell}^+(u, \phi) - \langle u_{x_k}^+, [\phi] n_\ell \rangle_{\mathcal{E}_h^I} + \epsilon_{k\ell}^+ \langle [u], \phi_{x_k}^+ n_\ell \rangle_{\mathcal{E}_h^I}, \quad (3.10)$$

for all $k, \ell = 1, 2, \dots, d$ and $*$ takes $+$, and empty value, where

$$a_{k\ell}^+(u, \phi) = b_{k\ell}^+(u, \phi) - \langle u_{x_k}^+, [\phi] n_\ell \rangle_{\mathcal{E}_h^I} + \epsilon_{k\ell}^+ \langle [u], \phi_{x_k}^+ n_\ell \rangle_{\mathcal{E}_h^I}, \quad (3.11a)$$

$$a_{k\ell}(u, \phi) = b_{k\ell}(u, \phi) - \langle \{u_{x_k}\}, [\phi] n_\ell \rangle_{\mathcal{E}_h^I} + \epsilon_{k\ell} \langle [u], \{\phi_{x_k}\} n_\ell \rangle_{\mathcal{E}_h^I}, \quad (3.11b)$$

$$a_{k\ell}^-(u, \phi) = b_{k\ell}^-(u, \phi) - \langle u_{x_k}^-, [\phi] n_\ell \rangle_{\mathcal{E}_h^I} + \epsilon_{k\ell}^- \langle [u], \phi_{x_k}^- n_\ell \rangle_{\mathcal{E}_h^I}, \quad (3.11c)$$

$$f_{k\ell}^*(\phi) = \epsilon_{k\ell}^* \sum_{e \in \mathcal{E}_h^B} \langle g, \phi_{x_k} n_\ell \rangle_e + \sum_{e \in \mathcal{E}_h^B} \frac{\hat{\gamma}_{k\ell}^{0*}}{h_e} \langle g, \phi \rangle_e, \quad (3.11d)$$

and $b_{k\ell}^* : H^1(\mathcal{T}_h) \times H^1(\mathcal{T}_h) \rightarrow \mathbb{R}$ is defined by

$$\begin{aligned} b_{k\ell}^*(v, w) &= (v_{x_k}, w_{x_\ell})_{\mathcal{T}_h} - \langle v_{x_k}, w n_\ell \rangle_{\mathcal{E}_h^B} + \epsilon_{k\ell}^* \langle v, w_{x_k} n_\ell \rangle_{\mathcal{E}_h^B} \\ &+ J_{k\ell}^{0*}(v, w) + \sum_{e \in \mathcal{E}_h^B} \frac{\hat{\gamma}_{k\ell}^{0*}}{h_e} \langle v, w \rangle_e \end{aligned}$$

for all $v, w \in H^1(\mathcal{T}_h)$, and $P_{k\ell}^{*h} = [P_h^*]_{k\ell}$ for $k, \ell = 1, 2, \dots, d$ and $*$ takes $+, -,$ and empty value.

Suppose $r \geq 1$ in the definition of V^h . Then, our IP-DG method for the fully nonlinear Dirichlet problem (3.1) is defined as seeking $u_h \in V^h$ and $P_h^+, P_h, P_h^- \in [V^h]^{d \times d}$ such that (3.6) and (3.10) hold for all $k, \ell = 1, 2, \dots, d$ and for $*$ taking $+, -,$ and empty value.

C The Numerical Moment

We now take a closer look at the numerical moment used in the definition of the Lax–Friedrichs-like numerical operator, (3.4a). For simplicity, we assume the three symmetrization constants are the same, that is, $\epsilon^+ = \epsilon = \epsilon^-$. Then,

$$\begin{aligned} (P_{k\ell}^{+h} - 2P_{k\ell}^h + P_{k\ell}^{-h}, \phi)_{\mathcal{T}_h} &= f_{k\ell}^+(\phi) - 2f_{k\ell}(\phi) + f_{k\ell}^-(\phi) - a_{k\ell}^+(u_h, \phi) + 2a_{k\ell}(u_h, \phi) - a_{k\ell}^-(u_h, \phi) \\ &= \sum_{e \in \mathcal{E}_h^B} \frac{\gamma_{k\ell}^{0+} - 2\gamma_{k\ell}^0 + \gamma_{k\ell}^{0-}}{h_e} \langle g - u_h, \phi \rangle_e \\ &\quad - \sum_{e \in \mathcal{E}_h^I} \frac{\gamma_{k\ell}^{0+} - 2\gamma_{k\ell}^0 + \gamma_{k\ell}^{0-}}{h_e} \langle [u_h], [\phi] \rangle_e \end{aligned}$$

for all $\phi \in V^h$. Thus, we have

$$\begin{aligned} a_0(u_h, P_h^+, P_h, P_h^-; \phi) &= (F(P_h, \nabla u_h, u_h, \cdot), \phi)_{\mathcal{T}_h} + \sum_{e \in \mathcal{E}_h^I} \frac{\alpha_1 : (2\gamma^0 - \gamma^{0+} - \gamma^{0-})}{h_e} \langle [u_h], [\phi] \rangle_e \\ &\quad + \sum_{e \in \mathcal{E}_h^B} \frac{\alpha_1 : (2\gamma^0 - \gamma^{0+} - \gamma^{0-})}{h_e} \langle u_h - g, \phi \rangle_e, \end{aligned}$$

and it follows that our IP-DG discretization amounts to replacing the continuous Hessian operator with a discrete Hessian operator, projecting our fully nonlinear operator into the DG space, and adding a penalization term to the nonlinear equation.

We end this section with a few remarks.

Remark 3.2.

- a. Looking backwards, (3.10) provides a proper interpretation for each variable $P_h^+, P_h,$ and P_h^- for a given function uh . Each P_h^* defines a discrete Hessian for uh . The functions $P_h^+, P_h,$ and P_h^- should be very close to each other if D^2u exists and is continuous. However, their discrepancies are expected to be large if D^2u does not exist.

- b. We have the three equations for approximating $u_{x_h x_\ell}$ are linearly independent provided that $\gamma_{k\ell}^0 > \max\{\gamma_{k\ell}^{0+}, \gamma_{k\ell}^{0-}\}$. The system of equations is comprised of (3.10) for $k, \ell \in \{1, 2, \dots, d\}$ fixed and * with +, -, and empty value.
- c. The reason for requiring $r \geq 1$ can be explained as follows. When $r = 0$, the piecewise constant functions have zero derivatives on the given partition. After eliminating the jump terms containing derivatives in (3.10), it is clear that the ability for P_h^+ and P_h^- to carry information from the right and the left, respectively, is lost. Furthermore, if $\gamma_{k\ell}^{0+} = \gamma_{k\ell}^0 = \gamma_{k\ell}^{0-}$, then for all $a_{k\ell}^+ = a_{k\ell}^- = a_{k\ell}$. As a result, the numerical moment term vanishes and we are left with a trivial discretization for (3.1), which is known not to work well in general.
- d. Let $\Lambda_h^* := \sum_{k=1}^d [P_h^*]_{kk}$, where * takes +, -, and empty value. Then, we have

$$\sum_{k=1}^d a_{kk}^* (u_h, \phi_h^*) = \sum_{k=1}^d f_{kk}^* (\phi_h^*) - (\Lambda_h^*, \phi_h^*)_{\mathcal{T}_h} \quad (3.12)$$

for all $\phi_h^* \in V^h$. Treating $\{\Lambda_h^+, \Lambda_h, \Lambda_h^-\}$ as “sources,” (3.12) represents three different Poisson discretizations for u . Thus, γ^{0*} for sufficiently large, we have (3.12) forms an invertible linear mapping between Λ_h^* and u/h when * is +, -, or empty value. Furthermore, the mapping is symmetric for a given value of * when $\epsilon_{kk}^* = -1$ for all $k = 1, 2, \dots, d$. We call the mapping “nonsymmetric” if $\epsilon_{kk}^* = 1$ for all $k = 1, 2, \dots, d$, and we call the mapping “incomplete” if $\epsilon_{kk}^* = 0$ for all $k = 1, 2, \dots, d$.

- e. Notice that (3.6) and (3.10) forms a nonlinear system of equations, with the nonlinearity only appearing in a_0 . Thus, a nonlinear solver is necessary in implementing the above scheme. We will perform numerical tests in Section VI using both a straight-forward Newton solver on the entire system and a solver to be proposed in the following section. We will see that our proposed discretizations either remove or destabilize many of the numerical artifacts that plague a trivial discretization of a nonlinear PDE problem, especially when paired with an appropriate solver.
- f. Eq. ((1.2).10) represents a block diagonal system where the auxiliary variables P_h^* are defined locally. The local structure can be exploited in both the design and the implementation of nonlinear solvers. More detailed consequences of the structure of (3.10) will be discussed in the following three sections.

IV Extensions of the Ip-Dg Framework for Second-Order Parabolic Problems

Using the above IP-DG methodology for elliptic problems, we now develop a class of fully discrete methods for second-order initial-boundary value problems of the form

$$2u_t + F(D^2u, \nabla u, u, x, t) = 0, \quad (x, t) \in \Omega_T := \Omega \times (0, T], \quad (4.1a)$$

$$u(x, t) = g(x, t), \quad (x, t) \in \partial\Omega \times (0, T], \quad (4.1b)$$

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

where F is a fully nonlinear elliptic operator; Ω is an open, bounded, polygonal domain; and T is a positive number. Assume the viscosity solution $u \in H^1(\Omega \times (0, T])$. The methodology will be based on using the method of lines for the time discretization.

To partition the time domain, we fix an integer $M > 0$ and let $\Delta t = \frac{T}{M}$. Then, we define $t_k := k \Delta t$ for $0 \leq k \leq M$. Notationally, $u_h^k(x) \in V^h$ and $P_h^{*,k} \in [V^h]^{d \times d}$ will be approximations for $u(x, t_k)$ and $D^2u(x, t_k)$, respectively, for all $0 \leq k \leq M$, where $*$ takes $+, -,$ and empty value. For both implicit and explicit schemes, we define the initial value, u_h^0 , by

$$u_h^0 = \mathcal{P}_h u_0, \quad (4.2)$$

where the projection operator \mathcal{P}_h is defined by (2.3). We also define a modified projection operator $\mathcal{P}_h^k : L^2(\mathcal{T}_h) \rightarrow V^h$ by

$$(\mathcal{P}_h^k v, \varphi_h)_{\mathcal{T}_h} + \delta \sum_{i=1}^d \langle \mathcal{P}_h^k v, \varphi_h n_i \rangle_{\mathcal{E}_h^B} = (v, \varphi_h)_{\mathcal{T}_h} + \delta \sum_{i=1}^d \langle g(\cdot, t_k), \varphi_h n_i \rangle_{\mathcal{E}_h^B} \quad \forall \varphi_h \in V^h \quad (4.3)$$

for all $v \in L^2(\mathcal{T}_h)$, where δ is a nonnegative penalty constant and $0 \leq k \leq M$. We note that, for $\delta = 0$, $\mathcal{P}_h^k = \mathcal{P}_h$, yielding the L^2 -projection operator. The modified projection operator is used to dynamically enforce the boundary conditions in explicit methods using a penalty technique introduced in [11].

To simplify the appearance of the methods and to make them more transparent for use with a given ODE solver, we define discrete Hessian operators $D_h^{2*,k} : V^h \rightarrow [V^h]^{d \times d}$, where $*$ takes $+, -,$ and empty value, at time tk using (3.10), where $0 \leq k \leq M$. Then, we define $D_h^{2*,k} v$ by

$$\begin{aligned} ([D_h^{2*,k} v]_{\ell m}, \phi_{\ell m}^{*h})_{\mathcal{T}_h} &= -a_{\ell m}^* (v, \phi_{\ell m}^{*h}) + \epsilon_{\ell m}^* \langle g(\cdot, t_k), (\phi_{\ell m}^{*h})_{x_\ell} n_m \rangle_{\partial\Omega} \\ &+ \sum_{e \in \mathcal{E}_h^B} \frac{\gamma_{\ell m}^{0*}}{h_e} \langle g(\cdot, t_k), \phi_{\ell m}^{*h} \rangle_e \end{aligned} \quad (4.4)$$

for all $\phi_{\ell m}^{*h} \in V^h$. We also introduce the operator notation

$$\hat{F}^k[v] := \hat{F}(D_h^{2+,k} v, D_h^{2-,k} v, D_h^{2\emptyset,k} v, \nabla v, v, x, k\Delta t) \quad (4.5)$$

for all $v \in V^h$. Then, we have the semi-discrete equation

$$\frac{\partial}{\partial t} u_h(x, t_k) = -\mathcal{P}_h \hat{F}^k [u_h(x, t_k)] \quad (4.6)$$

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

Letting \mathcal{P}_h^k denote the modified projection operator defined by (4.3), we can define fully discrete methods for approximating problem (4.1) based on approximating (4.6) using the forward Euler method, backward Euler method, and the trapezoidal method. Hence, we have the following fully discrete schemes for approximating (4.1):

$$u_h^{n+1} = \mathcal{P}_h^{n+1} \left(u_h^n - \Delta t \hat{F}^n [u_h^n] \right), \quad (4.7)$$

$$u_h^{n+1} + \Delta t \mathcal{P}_h \hat{F}^{n+1} [u_h^{n+1}] = u_h^n, \quad (4.8)$$

and

$$u_h^{n+1} + \frac{\Delta t}{2} \mathcal{P}_h \hat{F}^{n+1} [u_h^{n+1}] = u_h^n - \frac{\Delta t}{2} \mathcal{P}_h \hat{F}^n [u_h^n] \quad (4.9)$$

for $n = 0, 1, \dots, M-1$, where $u_h^0 := \mathcal{P}_h u_0$ and, for (4.8) and (4.9), we also have the implied equations $\mathcal{P}_h^{n*} = D_h^{2*,n} u_h^n$ for $*$ taking $+$, $-$, and empty value. Observe, (4.7), (4.8), and (4.9) correspond to the forward Euler method, backward Euler method, and trapezoidal method, respectively.

We can also formulate Runge–Kutta (RK) methods for approximating (4.6) as follows. Let s be a positive integer, $A \in \mathbb{R}^{s \times s}$, and $b, c \in \mathbb{R}^s$ such that

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

for each $k = 1, 2, \dots, s$. Then, a generic s -stage RK method for approximating (4.6) can be written as

$$u_h^{n+1} = \mathcal{P}_h^{n+1} \left(u_h^n - \Delta t \sum_{\ell=1}^s b_\ell \hat{F}^{n+c_\ell} [\xi_h^{n,\ell}] \right) \quad (4.10)$$

with

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

for all $n = 0, 1, \dots, M-1$ and $u_h^0 = \mathcal{P}_h u_0$. We note that (4.10) 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

(4.10) as an approximation for $u_h^{n+c_\varepsilon}$. Since the boundary condition at time t_{n+1} is enforced by \tilde{F}^{n+1} , we can set $\delta = 0$ in (4.3) if $c_s = 1$.

Remark 4.1. The linear Hessian operators D_h^{2*} correspond to sparse matrices. The construction of D_h^{2*} requires local computations as well as the inversion of only the reference mass matrix. Thus, the linear Hessian operators can be constructed offline and involve only sparse matrix multiplication when invoked.

V Nonlinear Solvers

In this section, we discuss different strategies for solving the nonlinear system of equations that results from the IP-DG discretization for the elliptic problem, (3.1). We have a nonlinear equation that is complemented by a linear system of equations. Furthermore, the nonlinear equation is monotone in three of its five arguments at every given point in the domain. In this section, we propose an algorithm for a solver that has been more tailored toward the IP-DG discretization, and we will discuss the benefits of the proposed algorithm in Remark 5.1.

First, we observe that the numerical operators given by (3.4) are symmetric in P_h^+ and P_h^- . Thus, there is the possibility for variable reduction. Assuming $\gamma_{k\ell}^{0+} = \gamma_{k\ell}^{0-}$ for all $k, \ell = 1, 2, \dots, d$, we can form a new variable $\bar{P}_h = \frac{P_h^+ + P_h^-}{2} \in V^h$. Then, we have P_h and \bar{P}_h correspond to two IP-DG approximations for the Hessian of u that both use averaged flux values on all interior faces/edges; i.e., P_h and \bar{P}_h are both solutions to (3.10) using the empty value for $*$, where the only difference is the value of the penalty parameter γ . An essential theme in the series [2-4] devoted to approximating fully nonlinear second-order PDEs is the ability to form at least two different Hessian approximations.

Second, we observe that the matrix-valued variables P_h^* are not symmetric, even for symmetric ε^* and γ^* . This is due to the fact that in the formulation of IP-DG methods, we only keep one of the terms in (3.7). To save computational cost, we could define symmetric versions of P_h^* by letting only $P_{k\ell}^*$ be defined by (3.10) for $k \geq \ell$ and letting $P_{\ell k}^* = P_{k\ell}^*$ for $k < \ell$. However, for twice-differentiable functions that are not C^2 functions, the Hessian is not symmetric. Thus, in our formulation, we chose not to artificially symmetrize our Hessian approximations. Large discrepancies in the mixed derivative approximations for a given value of P_h^* may actually serve as a local low-regularity indicator that can be paired with adaptivity techniques.

Third, we present a splitting algorithm that relies upon the observation in Remark 3.2 part (d) and the invertibility of (3.12). The algorithm is based on using the numerical moment as a means to split the system of equations.

Algorithm 5.1.

1. Pick initial guesses for u_h , P_h^+ , and P_h^- .

2. Set

$$[G]_{\ell} := F(P_h, \nabla u_h, u_h, x) + \hat{\alpha}[P_h^+ - 2P_h + P_h^-]_{\ell\ell}$$

for a fixed constant $\hat{\alpha} > 0$, and solve

$$([G]_{\ell}, \phi_{\ell})_{T_h} = 0 \quad \forall \phi_{\ell} \in V^h$$

for $[P_h]_{\ell\ell}$ for all $\ell = 1, 2, \dots, d$.

3. Set $\Lambda_h = \sum_{\ell=1}^d [P_h]_{\ell\ell}$. Find u_h by solving (3.12) for the given value of Λ_h .

4. Set $P_h^+ = D_h^{2+} u_h$ and $P_h^- = D_h^{2-} u_h$.

5. Repeat Steps 2–4 until the change in P_h is sufficiently small.

We end the section with some remarks concerning the observed performance of the proposed algorithm.

Remark 5.1.

- a. Algorithm 5.1 appears to be more selective than using a standard Newton solver on the full system of equations that results from the mixed formulation. However, the algorithm also appears to converge more slowly than the standard Newton solver when the Newton solver does converge. Thus, the algorithm may be best utilized as a way to generate an initial guess for a more efficient solver.
- b. There is potential to speed up Algorithm 5.1. Step 2 of Algorithm 5.1 requires solving a nonlinear system that is entirely monotone with respect to the unknowns for $\hat{\alpha}$ sufficiently large. Furthermore, the nonlinear equation is entirely local with respect can be solved in parallel. Step 3 of Algorithm 5.1 requires inverting a sparse matrix that is symmetric and positive definite when choosing $\epsilon_{kk} = -1$ sufficiently large and $k = 1, 2, \dots, d$ for all.
- c. From Section IIIC, we can see that there is a possibility the discretization contains C^0 numerical artifacts, especially when allowing nontrivial C^0 functions into the test space by choosing higher-degree polynomial basis functions. Algorithm 5.1 can be interpreted as a fixed-point solver that iterates over the discrete Laplacian. We will see in Section VI that by iterating over a high-order term in the discretization, Algorithm 5.1 appears to “destabilize” numerical artifacts even when such artifacts are present.
- d. By summing the diagonal elements of the discrete Hessian, we are able to map a second order derivative function in V_h back to u_h . Thus, we will have

$$\sum_{\ell=1}^d [P_h]_{\ell\ell} = \sum_{\ell=1}^d [D_h^2 u_h]_{\ell\ell}.$$

However, when uh is an approximation for a low regularity function, we will not have $[P_h]_{\ell\ell} = [D_h^2 u_h]_{\ell\ell}$ for all $\ell = 1, 2, \dots, d$. In fact, we would expect inverting the Laplacian operator to have a “smoothing” effect. Therefore, we would have $P_h^\pm \neq P_h$, and large discrepancies can serve as an indicator for low-regularity and/or adaptivity. This observation will be seen in Example 6.3 below.

- e. We may not be able to enforce the g-monotonicity requirement globally on a given fully nonlinear PDE such as the Monge–Ampère equation where the differential operator is only elliptic when acting on a particular class of functions. Thus, for such problems, we propose enforcing the g-monotonicity requirement “locally,” that is, over each iteration of the nonlinear solver, as described in the following definition.

Definition 5.1. The numerical operator \hat{F} in (3.2a) is said to be locally generalized monotone (locally g-monotone) for a function $v_h \in V^h$ if $\hat{F}(D_h^{2+} v_h, D_h^2 v_h, D_h^{2-} v_h)$ is monotone increasing in $(\nabla_h v_h, v_h, x)$ and $D_h^{2+} v_h$ and monotone decreasing in $D_h^{2-} v_h$.

VI Numerical Experiments

In this section, we present numerical tests to demonstrate the utility of the proposed IP-DG methods for fully nonlinear elliptic PDEs. Numerical tests for the proposed IP-DG methods for parabolic PDEs of type (1.2) can be found in [3]. In all of our tests, we use uniform Cartesian partitions on rectangular domains. To solve the resulting nonlinear algebraic systems, we use the Matlab built-in nonlinear solver *fsolve*. When recording the coefficient α for the numerical moment, we let I denote the identity matrix and $\mathbf{1}$ denote the ones matrix. We choose the initial guess as the zero function. For convenience, we set $\epsilon^* = \mathbf{0}$ for all tests except Example 6.2 with $r = 1$. We remark that similar results can be obtained when $\epsilon^* \neq \mathbf{0}$ and the penalty constants are sufficiently large. However, the actual benefit of the symmetrization parameter is unclear in the context of nonlinear algebraic systems. All errors will be measured in the L^∞ norm and the L^2 norm. Unless otherwise stated, reported residual values are those recorded by *fsolve*, which corresponds to the ℓ^2 norm of the vector-valued system of equations evaluated at the current approximation value. Since the numerical moment is the key to designing a consistent, g-monotone numerical operator, we will also use the conditionally elliptic Monge–Ampère test problems to better understand the contribution of the numerical moment. The tests in the 1D case, see [3], indicate the spatial error may have order $\mathcal{O}(h^\ell)$, where

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

However, the rates are not perfectly clear from the test data in this article.

Example 6.1. Consider the stationary Hamilton–Jacobi–Bellman problem

$$\min \{-\Delta u, -\Delta u/2\} = f \quad \text{in } \Omega,$$

$$u = g \quad \text{on } \partial\Omega,$$

$$\text{where } \Omega = (0, \pi) \times (-\pi/2, \pi/2),$$

$$f(x, y) = \begin{cases} 2 \cos(x) \sin(y), & \text{if } (x, y) \in S, \\ \cos(x) \sin(y), & \text{otherwise,} \end{cases}$$

$$S = (0, \pi/2] \times (-\pi/2, 0] \cup (\pi/2, \pi] \times (0, \pi/2)$$

, and g is chosen such that the viscosity solution is given by $u(x, y) = \cos(x) \sin(y)$.

The results for approximating the problem for $r = 1$, $r = 2$, and $r = 3$ are recorded in Table 1. The calculated rates for $r = 1$ appear less than the predicted rates, and the calculated rates for $r = 2$ appear greater than the predicted rates. The calculated rates for $r = 3$ appear to agree with the predicted rate of 4 when averaged.

Example 6.2.. Consider the 2D Monge–Ampère problem

$$- \det D^2 u = -u_{xx} u_{yy} + u_{xy} u_{yx} = f \quad \text{in } \Omega,$$

$$u = g \quad \text{on } \partial\Omega,$$

where $f = -(1 + x^2 + y^2)e^{x^2+y^2}$, $\Omega = (0, 1) \times (0, 1)$, and g is chosen such that the viscosity solution is given by $u(x, y) = e^{-\frac{x^2+y^2}{2}}$.

Table 1. Rates of convergence for Example 6.1 using $\alpha = 10 \mathbf{1}$, $\gamma_{0+} = \gamma_{0-} = 20 \mathbf{1}$, $\gamma_0 = 40 \mathbf{1}$, $\epsilon^* = \mathbf{0}$, and 4 iterations of Algorithm 5.1 followed by fsolve with initial guess $u_h^{(0)} = \mathbf{0}$

		$h = 2.22\text{E}+00$	$h = 1.48\text{E}+00$		$h = 1.11\text{E}+00$		$h = 8.89\text{E}-01$	
r	Norm	Error	Error	Order	Error	Order	Error	Order
1	L^2	4.86E-01	2.79E-01	1.37	1.92E-01	1.29	1.51E-01	1.09
		2.86E-01	2.33E-01	0.51	1.69E-01	1.12	1.20E-01	1.55
2	L^2	1.97E-01	7.81E-02	2.28	3.83E-02	2.48	2.49E-02	1.92
		1.51E-01	5.95E-02	2.29	3.02E-02	2.36	1.72E-02	2.52
3	L^2	7.60E-02	1.25E-02	4.46	7.77E-03	1.65	1.81E-03	6.53
		5.56E-02	1.06E-02	4.08	5.74E-03	2.14	1.54E-03	5.89

We approximate the given problem for $r = 1$ and $r = 2$, with the results recorded in Table 2. We observe that the rate of convergence is suboptimal for $r = 1$, and the last approximation showed little improvement even with a refined mesh. Similar results were obtained for $\epsilon^* = \mathbf{0}$. However, for $r = 2$, we observe rates between 2.0 and 2.5, which are superior to the predicted rates of convergence.

Table 2. Rates of convergence for Example 6.2 using $r = 1$ with $\alpha = 10I$, $\gamma_0 = 200\mathbf{1}$, $\epsilon^* = -1\mathbf{1}$, $u_h^{(0)} = 0$, and $fsolve$ with initial guess $\alpha = 24I$ and $r = 2$ with $\alpha = 24I$, $\gamma_{0+} = \gamma_{0-} = 20\mathbf{1}$, $\gamma_0 = 40\mathbf{1}$, $\epsilon^* = 0$, and 5 iterations of Algorithm 5.1 followed by $fsolve$ with initial guess $u_h^{(0)} = 0$

		$h = 7.07E-01$	$h = 4.71E-01$		$h = 3.54E-01$		$h = 2.83E-01$	
r	Norm	Error	Error	Order	Error	Order	Error	Order
1	L^2	4.88E-02	2.04E-02	2.15	1.31E-02	1.55	1.33E-02	-0.09
		1.47E-01	7.82E-02	1.56	4.32E-02	1.65	4.32E-02	0.53
2	L^2	6.37E-03	2.52E-03	2.29	1.30E-03	2.29	7.63E-04	2.39
		1.99E-02	7.68E-03	2.35	3.79E-03	2.45	2.17E-03	2.51

We also observe that the numerical moment plays a key role in preventing a Newton solver from encountering a singularity when solving the resulting system of nonlinear equations. In fact, for this example, $fsolve$ does not converge when the numerical moment is not present, even with a good initial guess. We let $r = 1$, $\gamma^0 = 100\mathbf{1}$, $\epsilon = 0$, and $h = 2.83e-01$. Using the mixed formulation for only uh and Ph with $\alpha = 0$ and solving the resulting system of equations directly with $fsolve$ has an initial residual of 357,315 with a residual of 197.73 after 50 iterations when the initial guess is given by $u_h^{(0)} = 0$ and has an initial residual of 90.35 with a residual of 1.41 after 50 iterations when the initial guess is given by $u_h^{(0)} = \mathcal{P}_h u$, the broken L^2 projection of the exact solution. Both attempts report the system of equations is close to singular, an error message that was not reported when performing the same tests with $\alpha \neq 0$.

Example 6.3.. Consider the 2D Monge–Ampère problem

$$- \det D^2 u = -u_{xx} u_{yy} + u_{xy} u_{yx} = 0 \quad \text{in } \Omega,$$

$$u = g \quad \text{on } \partial\Omega,$$

where $\Omega = (-1, 1) \times (-1, 1)$ and g is chosen such that the viscosity solution is given by $u(x, y) = |x| \in H^1(\Omega)$.

We first approximate the example by partitioning Ω using an odd number of rectangles in both the x and y coordinate directions. Thus, the line $x = 0$ does not correspond to an interior edge for any of the partitions. From Table 3, we observe better than optimal rates of convergence for $r = 1$, using the fact that $u \in H^1(\Omega)$. Partitioning the domain into 64 uniform rectangles such that the line $x = 0$ always corresponds to an interior edge, we recover the exact solution for $r = 1$. In fact, using $r = 1$, $\alpha = 100I$, $\gamma^{0+} = \gamma^{0-} = 20\mathbf{1}$, $\gamma^0 = 40\mathbf{1}$, $\epsilon^* = 0$, $h = 3.54e - 01$, and 10 iterations of Algorithm 5.1 followed by $fsolve$ with initial guess $u_h^{(0)} = 0$, we have $\|h - u\|_{L^\infty(\Omega)} \approx 6.88e-15$ and $\|h - u\|_{L^2(\Omega)} \approx 4.67e-15$. Note, for such a partition, we have $u \in V^h$.

Table 3. Rates of convergence for Example 6.3 using $\alpha = 100 I$, $\gamma_{0+} = \gamma_{0-} = 20 \mathbf{1}$, $\gamma_0 = 40 \mathbf{1}$, $\epsilon^* = \mathbf{0}$, and 10 iterations of Algorithm 5.1 followed by fsolve with initial guess $u_h^{(0)} = \mathbf{0}$. Note, the line $x = 0$ does not correspond to an interior edge for any of the partitions

		$h = 9.43\text{E-}01$	$h = 5.66\text{E-}01$		$h = 4.04\text{E-}01$		$h = 2.57\text{E-}01$	
r	Norm	Error	Error	Order	Error	Order	Error	Order
1	L^2	1.52E-01	7.25E-02	1.45	4.39E-02	1.49	2.24E-02	1.49
		2.81E-01	1.73E-01	0.96	1.22E-01	1.02	7.73E-02	1.02

We now perform a series of three tests that focus on both the choice of solver and the presence of a numerical moment. We will see that for this particular problem, the choice of the nonlinear solver has a larger impact on whether or not the proposed IP-DG methods successfully approximate the given viscosity solution. We first approximate Example 6.3 using Algorithm 5.1 to solve the system of nonlinear equations. We can see in Figure 1 that the residuals measured by the L^∞ norm of $F[u_h]$ converge to zero quickly for $r = 1$ and appear to be converging toward zero slowly for $r = 2$. In fact, after 24 iterations, we have $\|h - u\|_{L^\infty(\mathcal{T}_h)} \approx 2.20 \text{e-}04$ and $\|h - u\|_{L^2(\mathcal{T}_h)} \approx 2.04 \text{e-}04$ for $r = 1$. After 1 iteration of Algorithm 5.1 with an initial guess of $u_h^{(0)} = \mathbf{0}$, we have $\|(D_h^{2*} u_h^{(1)})\|_{L^\infty(\mathcal{T}_h)} \approx 13.54$. Furthermore, if $c_{k\ell}^*$ denotes the coefficients for $[D_h^{2*} u_h^{(1)}]_{k\ell}$, then we have $\|c_{11} + c_{22}\|_{\ell^2} = 0$, as expected from the initial guess, and $\|c_{11}^+ + c_{11}^- + c_{22}^+ + c_{22}^-\|_{\ell^2} \approx 22.9912$, indicating a nonzero numerical moment. From Figure 2, we can see that as Algorithm 5.1 iterates, the approximation does in fact become less smooth and appears to be converging toward the viscosity solution u .

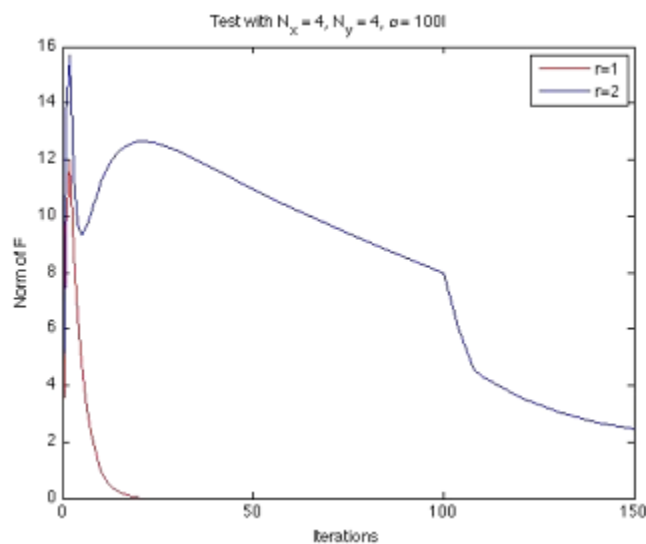


Figure 1. Computed residuals using the L^∞ norm on $F[u_h]$ for Example 6.3 using $\alpha = 100 I$, $\gamma^{0+} = \gamma^{0-} = 20 \mathbf{1}$, $\gamma^0 = 40 \mathbf{1}$, $\epsilon^* = 0$, $h = 7.071068e-01$, and Algorithm 5.1 with initial guess $u_h^{(0)} = 0$.

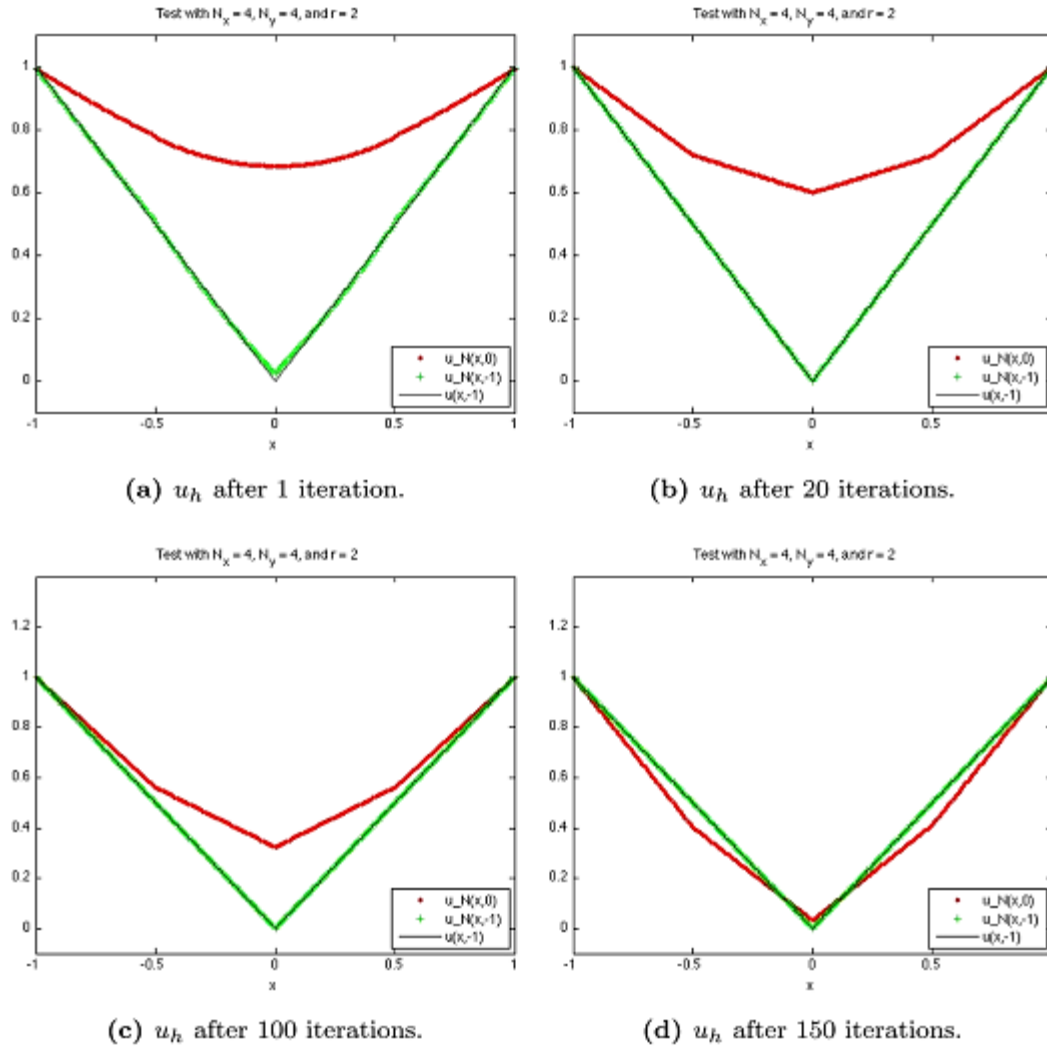


Figure 2. Computed solutions for Example 6.3 using $r = 2$, $\alpha = 100 I$, $\gamma^{0+} = \gamma^{0-} = 20 \mathbf{1}$, $\gamma^0 = 40 \mathbf{1}$, $\epsilon^* = 0$, $h = 7.071068e-01$, and Algorithm 5.1 with initial guess $u_h^{(0)} = 0$.

We now approximate Example 6.3 using *fsolve* to solve the system of nonlinear equations. The results for using *fsolve* directly or *fsolve* after 20 iterations of Algorithm 5.1 can be found in Figure 3. We see that neither approximation converges to the viscosity solution, yet the residuals for *fsolve* are given by $3.34007e-26$ after 11 iterations when we use *fsolve* directly and $1.63896e-26$ after a maximum of 20 iterations when we first use Algorithm 5.1 to precondition the initial guess. Thus, using a Newton solver appears to yield C^0 numerical artifacts for the given problem. We do note that even with the small residuals, *fsolve* does return an error flag that indicates a

possible lack of convergence for both tests. Also, while the first test that used *fsolve* fulfilled stopping criteria, the second test that used Algorithm 5.1 to precondition the initial guess had a trust-region radius for *fsolve* that was less than $O(10^{-10})$ for the last eight iterations causing the solver to stop prematurely.

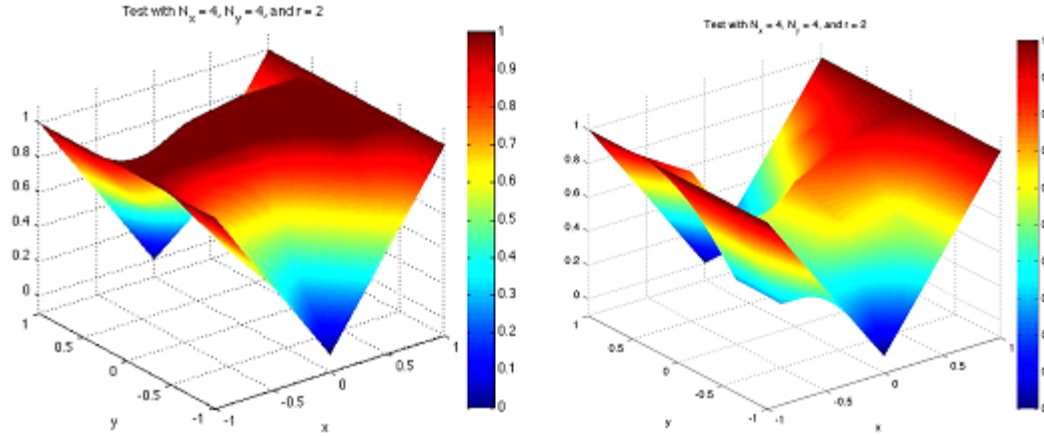


Figure 3. Computed solutions for Example 6.3 using $r = 2$, $\alpha = 100 I$, $\gamma^{0+} = \gamma^{0-} = 20 \mathbf{1}$, $\gamma^0 = 40 \mathbf{1}$, $\epsilon^* = 0$, $h = 7.071068e-01$, and *fsolve*. The left plot corresponds to uh with an initial guess $u_h^{(0)} = \mathbf{0}$, and the right plot corresponds to uh with the initial guess for *fsolve* given by the approximation after 20 iterations of Algorithm 5.1 with initial guess $u_h^{(0)} = \mathbf{0}$.

We finally approximate Example 6.3 without using a numerical moment, as seen in Figure 4. When we do not have a numerical moment, *fsolve* does not converge after 25 iterations and has a residual of 103.035 with a residual of 109,660 corresponding to the initial guess. We also note that *fsolve* reports the system of equations is singular or badly scaled after the first iteration and has a residual of 28,538 after the second iteration when not using a numerical moment. When using the numerical moment, our initial guess for *fsolve* has a residual of 37,158 and after 10 iterations converges with a residual of 1.33044e-26. Thus, we see in this example that using a numerical moment and preconditioning the initial guess for a Newton solver by first using Algorithm 5.1, we were able to approximate a degenerate problem that appears singular when using a straightforward discretization with a Newton solver.

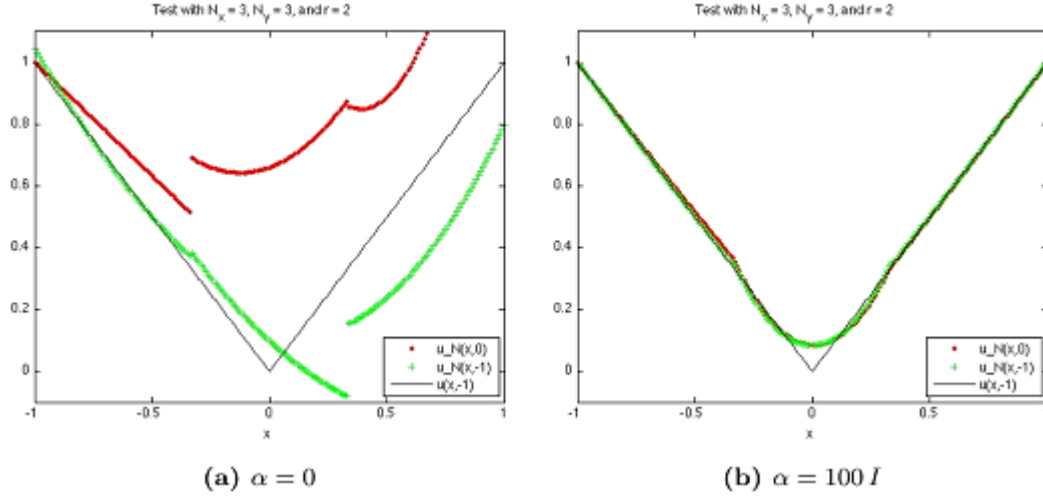


Figure 4. Computed solutions for Example 6.3

using $r = 2$, $\gamma^{0+} = \gamma^{0-} = 20 \mathbf{1}$, $\gamma^0 = 40 \mathbf{1}$, $\epsilon^* = 0$, $h = 9.428090e-01$, and initial guess $u_h^{(0)} = \mathbf{0}$. The left plot uses *fsolve* on the mixed system for only u_h and P_{2h} , and the right plot uses *fsolve* after 15 iterations of Algorithm 5.1.

From the above tests, we see that the numerical moment plays two major roles; it allows the scheme to converge for a wider range of initial guesses, especially when paired with the proper solver, and it enables the scheme to address the issue of the conditional uniqueness of viscosity solutions. Given the form of the numerical moment, $\alpha : (P_h^+ - 2P_h + P_h^-)$, these benefits are even more substantial given the way in which P_h^+ , P_h , and P_h^- are formed. The three variables only differ in their jump terms, and the entire numerical moment can be hard-coded using the jump only representation derived in Section III C. When $\gamma_{0+} = \gamma_0 = \gamma_{0-}$, the three different choices for the numerical fluxes (or jump terms) are all equivalent at the PDE level for classical solutions, and often the various jump formulations are presented as interchangeable when discretizing linear and quasilinear PDEs using the IP-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.

We end with a couple of remarks:

Remark 6.1.

- a. The discretization techniques for fully nonlinear PDEs and the choice of solver for the resulting nonlinear systems of equations should not be considered entirely independent. We see in many tests that the addition of a numerical moment yields a system of equations that is better suited for generic Newton solvers, especially when considering the 1D numerical tests in [3]. However, the tests in Section VI further indicate that the numerical moment has a much greater impact for approximating fully nonlinear PDEs when used in concert with an appropriate solver.

- b. We see from the above tests for Example 6.3 that the numerical moment has potential to serve as an indicator function for adaptivity due to the fact it appears largest in areas where the viscosity solution is not regular.

References

- 1 X. Feng, R. Glowinski, and M. Neilan, Recent developments in numerical methods for fully nonlinear second order partial differential equations, *SIAM Rev* **55** (2013), 205–267.
- 2 X. Feng, C. Kao, and T. Lewis, Convergent finite difference methods for one-dimensional fully nonlinear second order partial differential equations, *J Comput Appl Math* **254** (2013), 81-98. 10.1016/j.cam.2013.02.001 2013.
- 3 X. Feng and T. Lewis, Mixed interior penalty discontinuous Galerkin methods for one-dimensional fully nonlinear second order elliptic and parabolic equations. to appear in *J. Comp. Math.*, downloadable at arxiv.org/abs/1212.0259.
- 4 X. Feng and T. Lewis, Local discontinuous Galerkin methods for one-dimensional second order fully nonlinear elliptic and parabolic equations, *J Sci Comput* to appear. 10.1007/s10915-013-9763-3 (2013).
- 5 L. Caffarelli and X. Cabré, *Fully nonlinear elliptic equations*, Am Math Soc Coll Pub, Providence, RI, Vol. **43** 1995.
- 6 D. Gilbarg and N. Trudinger, Elliptic partial differential equations of second order, *Classics in Mathematics*, Springer-Verlag, New York, 2001. Reprint.
- 7 G. Barles and P. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asymptotic Anal* **4** (1991), 271–283.
- 8 M. Crandall, H. Ishii, and P.-L. Lions, User's guide to viscosity solutions of second order partial differential equations, *Bull Am Math Soc (N.S.)* **27** 1992, 1–67.
- 9 P. Ciarlet, *The finite element method for elliptic problems*, Vol. **40**, Classics Appl Math SIAM, Philadelphia, PA, 2002. Reprint.
- 10 B. Rivière, *Discontinuous galerkin methods for solving elliptic and parabolic equations*, Vol. **35**, Front Appl Math SIAM, Philadelphia, PA, 2008.
- 11 J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abh Math Sem Univ Hamburg* **36** (1971), 9–15.