Journal of Scientific Computing (2022) 90:35
https://doi.org/10.1007/s10915-021-01714-6

Convergent Finite Difference Methods for Fully Nonlinear Elliptic Equations in Three Dimensions

Brittany Froese Hamfeldt[1] · **Jacob Lesniewski**[1]

Received: 16 March 2021 / Revised: 17 September 2021 / Accepted: 28 September 2021

Abstract
We introduce a generalized finite difference method for solving a large range of fully nonlinear elliptic partial differential equations in three dimensions. Methods are based on Cartesian grids, augmented by additional points carefully placed along the boundary at high resolution. We introduce and analyze a least-squares approach to building consistent, monotone approximations of second directional derivatives on these grids. We then show how to efficiently approximate functions of the eigenvalues of the Hessian through a multi-level discretization of orthogonal coordinate frames in $\mathbb{R}^3$. The resulting schemes are monotone and fit within many recently developed convergence frameworks for fully nonlinear elliptic equations including non-classical Dirichlet problems that admit discontinuous solutions, Monge–Ampère type equations in optimal transport, and eigenvalue problems involving nonlinear elliptic operators. Computational examples demonstrate the success of this method on a wide range of challenging examples.

Keywords Fully nonlinear elliptic equations · Generalized finite difference methods · Viscosity solutions · Three dimensions

Mathematics Subject Classification 35J15 · 35J25 · 35J60 · 35J66 · 65N06 · 65N12 · 65N22 · 65N25

1 Introduction

In this article, we introduce and implement a convergent finite difference method for solving a large class of fully nonlinear elliptic partial differential equations (PDEs) on general three-

The first author was partially supported by NSF DMS-1619807 and NSF DMS-1751996. The second author was partially supported by NSF DMS-1619807.

✉ Brittany Froese Hamfeldt
bdfroese@njit.edu

Jacob Lesniewski
jl779@njit.edu

[1] New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

dimensional domains. The method we develop encompasses a range of challenging problems including Pucci's maximal and minimal equations, obstacle problems, prescribed curvature equations, Monge–Ampère type equations arising in optimal transport, and eigenvalue problems involving nonlinear PDEs.

1.1 Background

Fully nonlinear elliptic partial differential equations (PDEs) appear in a variety of applications including optimal transport, seismology [12], astrophysics [18], mathematical finance [17], materials science [38], and molecular engineering [3]. These problems are challenging because they often include discontinuous or sharp jumps in the data, involve intricate domains, and may require data to satisfy a solvability condition that is not known *a priori*.

In recent years, the numerical solution of these equations has received a great deal of attention, and several new methods have been developed including finite difference methods [4,16,27,34,37], finite element methods [1,6,9,35], least squares methods [11], and methods involving fourth-order regularization terms [14]. However, these methods are not designed to compute weak solutions. When the ellipticity of the equation is degenerate or no smooth solution exists, methods become very slow, are unstable, or converge to an incorrect solution.

A couple convergence frameworks have emerged in recent years. The approach of [13] introduces the concept of generalized monotonicity, similar in flavor to the fourth-order regularization of [14], to produce convergent methods for a class of Hamilton–Jacobi–Bellman equations. Another powerful framework, which informs the method described in the present article, is the approach of Barles and Souganidis [2], which shows that consistent, monotone methods converge if the limiting PDE satisfies a comparison principle. A variety of methods have been developed within this framework [15,21,22,28,33]. Moreover, these convergence proofs have recently been extended to non-classical Dirichlet problems [23], optimal transport type boundary conditions [24], and eigenvalue problems involving nonlinear PDEs [25].

1.2 Contributions of this Work

The method we describe applies to nonlinear equations that depend on various second directional derivatives,

$$F(\mathbf{x}, u(\mathbf{x}), u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}); \boldsymbol{\nu} \in \mathcal{A} \subset \mathbb{S}^2) = 0, \tag{1}$$

where the admissible set $\mathcal{A}$ is used to characterize a finite set of unit vectors in $\mathbb{R}^3$. We also consider functions of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the Hessian matrix,

$$F(\lambda_1(D^2u(\mathbf{x})), \lambda_2(D^2u(\mathbf{x})), \lambda_3(D^2u(\mathbf{x}))) \equiv G\left(\sum_{j=1}^{3} \phi(\lambda_j(D^2u(\mathbf{x})))\right) = 0, \tag{2}$$

where ϕ is a concave function and G is non-increasing and continuous. We note that this encompasses a range of elliptic operators including PDEs of Monge–Ampère type and various curvature equations. Finally, we consider eigenvalue problems of the form

$$F(\mathbf{x}, D^2u(\mathbf{x})) = c \tag{3}$$

where both the function u and the constant $c \in \mathbb{R}$ are unknown. Moreover, we show how to enforce a range of boundary conditions including Dirichlet conditions, Neumann conditions, and the second type boundary condition $\nabla u(\Omega) \subset \bar{\Omega}^*$ arising in optimal transport.

The starting point of our method is the generalized finite difference methods of [21], which produced consistent, monotone schemes for a large class of nonlinear elliptic equations using very general two-dimensional point clouds. While much of the convergence theory applies to general bounded domains in $\mathbb{R}^n$, the transition to three dimensions introduces several new challenges that are not present in two dimensions.

A first challenge that we face is the discretization of the boundary of three-dimensional domains. In order to preserve both consistency and monotonicity in the entire domain, it is necessary to over-resolve the boundary in a precise way. This precludes the use of many standard structure grids such as Cartesian meshes. We describe a new approach to discretizing our domains that preserves a great deal of structure (which is needed for efficient evaluation of the nonlinear operators), while fitting within the precise requirements needed to construct monotone schemes.

A second challenge is the approximation of second directional derivatives. In two dimensions, this can be accomplished explicitly even on very complicated meshes. In three dimensions, explicit formulas are no longer possible in general. Instead, we describe an optimization approach that provably yields a consistent, monotone approximation. We describe a particularly efficient implementation of this idea that utilizes simple centered differences in the interior of the domain, while employing this optimization procedure in a small band of points near the boundary.

A third major challenge is discretizing general functions of the eigenvalues of the Hessian. In two dimensions, these two eigenvalues can be represented as the maximum and minimum possible second directional derivatives. Three dimensions introduces a third eigenvalue, and this simple approach does not easily generalize in a way that preserves monotonicity. We introduce an alternate approach that allows for monotone approximation of a large range of functions of the eigenvalues of the Hessian.

Finally, we note that the shift to a higher dimension brings the practical concerns of memory and processing speed to the forefront. There are two challenges here: (1) efficient construction and evaluation of the discrete systems of algebraic equations that approximate equations (1)–(3) and (2) efficient methods for solving the resulting systems of nonlinear algebraic equations. Our focus in this paper is the first issue; we propose a robust method for solving the resulting nonlinear algebraic system, but defer the development of efficient solvers to future work. Our ultimate goal is to produce a monotone scheme that can serve as foundation for convergent higher-order filtered methods [19]. In two-dimensional experiments, filtered methods derive their stability from an underlying monotone scheme, but frequently provide accuracy that is independent of the formal accuracy of the monotone component. Consequently, the particular implementation described in the present paper prioritizes efficiency over accuracy. Since a naive implementation of our proposed discretization can be computationally intractable, we introduce techniques derived from the structure of the underlying problem to describe an efficient method for producing a consistent, monotone approximation of many fully nonlinear elliptic PDEs in three dimensions. In particular, we propose a new multi-level procedure for approximating functions of the eigenvalues of the Hessian, which provides a dramatic reduction in computational cost.

1.3 Contents

In Sect. 2, we review the theory of generalized finite difference approximations for fully nonlinear elliptic equations. In Sect. 3, we describe the three-dimensional discretization of (1)–(3). In Sect. 4, we discuss practical considerations relating to the efficient construction of our discretization. In Sect. 5, we provide computational results for a large range of challenging problems. Finally, in Sect. 6, we provide conclusions and perspective.

2 Approximation of Elliptic Equations

In this section, we briefly review relevant results on the construction and convergence of numerical methods for solving fully nonlinear elliptic equations.

2.1 Elliptic Equations

The PDE operators we consider in this work are degenerate elliptic.

$$F(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), D^2 u(\mathbf{x})) = 0, \quad \mathbf{x} \in \bar{\Omega} \subset \mathbb{R}^3. \tag{4}$$

Definition 1 (*Degenerate elliptic*) The operator $F : \bar{\Omega} \times \mathbb{R} \times \mathcal{S}^2 \to \mathbb{R}$ is *degenerate elliptic* if

$$F(\mathbf{x}, u, X) \leq F(\mathbf{x}, v, Y)$$

whenever $u \leq v$ and $X \geq Y$.

We note that the definition of the operator is extended onto the boundary of the domain, and includes the relevant boundary conditions.

The PDE operators (1)–(2) that we consider in this work are degenerate elliptic if they are non-decreasing functions of the argument u and non-increasing functions of all subsequent arguments (which involve second directional derivatives).

Since degenerate elliptic equations need not have classical solutions, solutions need to be interpreted in a weak sense. The numerical methods developed in this article are guided by the very powerful concept of the viscosity solution [10].

Definition 2 (*Upper and lower semi-continuous envelopes*) The *upper and lower semi-continuous envelopes* of a function $u(\mathbf{x})$ are defined, respectively, by

$$u^*(\mathbf{x}) = \limsup_{\mathbf{y} \to \mathbf{x}} u(\mathbf{y}), \quad u_*(\mathbf{x}) = \liminf_{\mathbf{y} \to \mathbf{x}} u(\mathbf{y}).$$

Definition 3 (*Viscosity solution*) An upper (lower) semi-continuous function u is a *viscosity subsolution (supersolution)* of (4) if for every $\phi \in C^2(\bar{\Omega})$, whenever $u - \phi$ has a local maximum (minimum) at $\mathbf{x} \in \bar{\Omega}$, then

$$F_*^{(*)}(\mathbf{x}, u(\mathbf{x}), D^2\phi(\mathbf{x})) \leq (\geq) 0.$$

A function u is a *viscosity solution* of (4) if u^* is a subsolution and u_* a supersolution.

An important property of many elliptic equations is the comparison principle, which immediately implies uniqueness of the solution.

Definition 4 (*Comparison principle*) A PDE has a *comparison principle* if whenever u is an upper semi-continuous subsolution and v a lower semi-continuous supersolution of the equation, then $u \leq v$ on $\bar{\Omega}$.

2.2 Approximation

In order to construct convergent approximations of elliptic operators, we will rely on the framework introduced by Barles and Souganidis [2] and further extended by Oberman [32] and the authors of this work [23–25].

We consider finite difference schemes that have the form

$$F^h(\mathbf{x}, u(\mathbf{x}), u(\mathbf{x}) - u(\cdot)) = 0 \tag{5}$$

where h is a small parameter relating to the grid resolution.

The convergence framework requires notions of consistency and monotonicity, defined below.

Definition 5 (*Consistency*) The scheme (5) is *consistent* with the equation (4) if for any smooth function ϕ and $x \in \bar{\Omega}$,

$$\limsup_{h\to 0^+, \mathbf{y}\to\mathbf{x}, \xi\to 0} F^h(\mathbf{y}, \phi(\mathbf{y}) + \xi, \phi(\mathbf{y}) - \phi(\cdot)) \leq F^*(\mathbf{x}, \phi(\mathbf{x}), \nabla\phi(\mathbf{x}), D^2\phi(\mathbf{x})),$$

$$\liminf_{h\to 0^+, \mathbf{y}\to\mathbf{x}, \xi\to 0} F^h(\mathbf{y}, \phi(\mathbf{y}) + \xi, \phi(\mathbf{y}) - \phi(\cdot)) \geq F_*(\mathbf{x}, \phi(\mathbf{x}), \nabla\phi(\mathbf{x}), D^2\phi(\mathbf{x})).$$

Definition 6 (*Monotonicity*) The scheme (5) is monotone if F^h is a non-decreasing function of its final two arguments.

Schemes that satisfy these two properties respect the notion of the viscosity solution at the discrete level. In particular, these schemes preserve the maximum principle and are guaranteed to converge to the solution of the underlying PDE under a range of interesting settings.

Another important property of schemes is stability, which allows discrete solutions to be bounded uniformly.

Definition 7 (*Stability*) The scheme (5) is stable if there exists some $M \in \mathbb{R}$ such that if u^h is any solution of (5) then $\|u^h\|_\infty \leq M$.

Under mild conditions relating to the well-posedness of the limiting PDE and continuity of (5) in its final two arguments, consistent and monotone schemes are automatically stable [23, Lemmas 35-36].

The convergence framework of Barles and Souganidis applies to well-posed PDEs that satisfy a comparison principle.

Theorem 1 (Convergence [32]) *Let F be a degenerate elliptic operator with a comparison principle and suppose that the PDE (4) has a unique viscosity solution u. Let the approximation F^h be consistent, monotone, and stable and u^h any solution of the scheme (5). Then u^h converges uniformly to u as $h \to 0$.*

Recently, convergence results have also been obtained for a variety of equations that do not have a traditional comparison principle. This includes non-classical Dirichlet problems that admit discontinuous solutions [23], Monge–Ampère type equations equipped with optimal transport type boundary constraints [24], and eigenvalue problems of the form (3) that require solving for the solution of both a fully nonlinear PDE and an unknown scalar constant [25].

2.3 Generalized Finite Difference Methods

The results discussed in the previous section provide a powerful tool for proving convergence of monotone schemes. However, they do not offer insight into the actual construction of monotone methods. Indeed, it is well known that there are linear elliptic operators for which no finite difference stencil of bounded width admits a consistent, monotone discretization [26, 29].

Wide stencil methods on Cartesian grids have been developed for a variety of fully nonlinear elliptic PDEs [28,33]. However, these can fail to preserve both consistency and monotonicity at points near the boundary, particularly in the absence of Dirichlet boundary conditions. For insight into the construction of globally consistent and monotone schemes on general domains, we turn to the two-dimensional meshfree finite difference methods developed in [21].

We begin by introducing some notation, which applies equally well to higher dimensional settings.

(N1) $\Omega \subset \mathbb{R}^d$ is a bounded domain with Lipschitz boundary $\partial\Omega$.
(N2) $\mathcal{G} \subset \bar{\Omega}$ is a point cloud consisting of the points $\mathbf{x}_i$, $i = 1, \ldots, N$.
(N3) $h = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{y}\in\mathcal{G}} |\mathbf{x} - \mathbf{y}|$ is the spatial resolution of the point cloud. In particular, every ball of radius h contained in $\bar{\Omega}$ contains at least one discretization point $\mathbf{x}_i$.
(N4) $h_B = \sup_{\mathbf{x}\in\partial\Omega} \min_{\mathbf{y}\in\mathcal{G}\cap\partial\Omega} |\mathbf{x} - \mathbf{y}|$ is the resolution of the point cloud on the boundary. In particular, every ball of radius h_B centered at a boundary point $\mathbf{x} \in \partial\Omega$ contains at least one discretization point $\mathbf{x}_i \in \mathcal{G} \cap \partial\Omega$ on the boundary.
(N5) $\delta = \min_{\mathbf{x}\in\Omega\cap\mathcal{G}} \inf_{\mathbf{y}\in\partial\Omega} |\mathbf{x} - \mathbf{y}|$ is the distance between the set of interior discretization points and the boundary. In particular, if $\mathbf{x}_i \in \mathcal{G} \cap \Omega$ and $\mathbf{x}_j \in \partial\Omega$, then the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is at least δ.
(N6) ϵ is a search radius associated with the point cloud.

Using this notation, we can pose some necessary hypotheses on the point cloud and related discretization parameters.

Hypothesis 2 (Hypotheses on point cloud) *We require the discretization $\mathcal{G}$ of $\bar{\Omega}$ to satisfy:*

(H1) The boundary resolution satisfies $h_B/\delta \to 0$ as $h \to 0$.
(H2) The search radius satisfies both $\epsilon \to 0$ and $h/\epsilon \to 0$ as $h \to 0$.

Remark 1 The particular choice of scales satisfying this hypothesis strongly impacts the cost and accuracy of the resulting numerical method. See Sect. 4.1 for a detailed discussion of the choices employed in the present implementation.

Suppose we wish to approximate the second directional derivative $u_{\boldsymbol{\nu}\boldsymbol{\nu}}$ at some point $\mathbf{x}_0 \in \mathcal{G} \cap \Omega$. Let $\boldsymbol{\nu}^\perp \in \mathbb{R}^2$ be any vector orthogonal to $\boldsymbol{\nu}$. We begin by considering as candidate neighbors all nodes within a distance r of $\mathbf{x}_0$. From this we select four points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \in \mathcal{G} \cap B(\mathbf{x}_0, \epsilon)$, one in each of the four quadrants defined by the axes $\mathbf{x}_0 + \boldsymbol{\nu}t$ and $\mathbf{x}_0 + \boldsymbol{\nu}^\perp t$, that are as well-aligned as possible with the line $\mathbf{x}_0 + \boldsymbol{\nu}t$. See Fig. 1.

Using these four neighbors, we seek an approximation of the form

$$-u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0) \approx -\mathcal{D}^h_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}_0) = -\sum_{j=1}^{4} a_j (u(\mathbf{x}_j) - u(\mathbf{x}_0)).$$

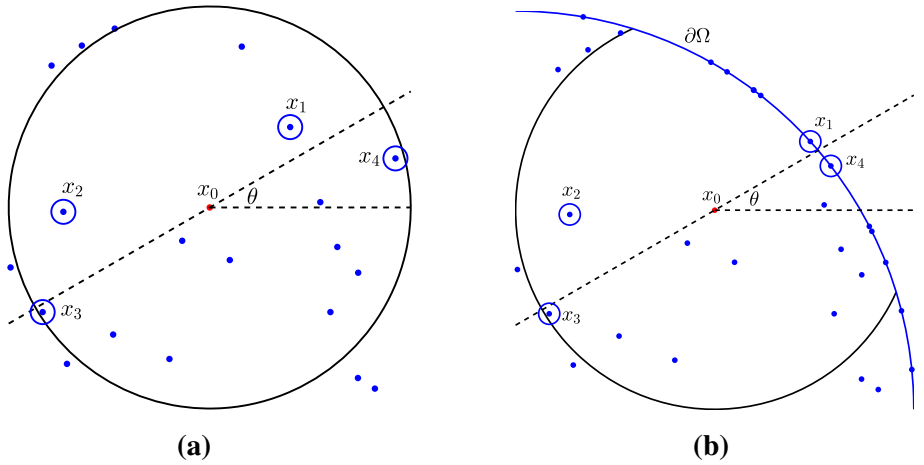

Fig. 1 A finite difference stencil chosen from a point cloud **a** in the interior and **b** near the boundary (Color figure online)

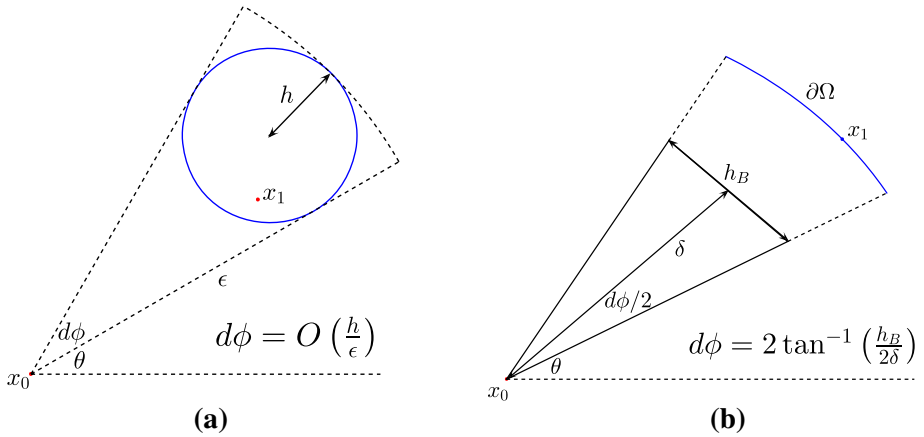

Fig. 2 The angular resolution of a generalized finite difference stencil (Color figure online)

Here monotonicity requires that each of the $a_j \geq 0$. In two dimensions, it is possible to find an explicit form of the coefficients a_j that yields a consistent, monotone approximation under Hypothesis 2.

We remark that the discretization error of the resulting scheme depends on two parameters: the effective spatial resolution ϵ (the maximum distance between $\mathbf{x}_0$ and the points $\mathbf{x}_j$ used in the finite difference stencil) and the angular resolution $d\phi$ (the maximum angle between the axis aligned with $\boldsymbol{\nu}$ and the vector $\mathbf{x}_j - \mathbf{x}_0$). These components of the error are not independent of each other, and we find that the angular resolution is bounded by $d\phi = \max\{\mathcal{O}(h/\epsilon), \mathcal{O}(h_B/\delta)\}$. See Fig. 2.

Once any second directional derivative can be approximated, it is easy to substitute these into nonlinear operators of the form (1). Functions of the eigenvalues of the Hessian (2) are

also easily approximated in two dimensions via the Rayleigh quotient characterization,

$$\lambda_1(D^2u) = \min_{|\boldsymbol{v}|=1} u_{\boldsymbol{vv}}, \quad \lambda_2(D^2u) = \max_{|\boldsymbol{v}|=1} u_{\boldsymbol{vv}}. \tag{6}$$

3 Discretization

3.1 Construction of the Grid

In order to build a grid or point cloud on which monotone schemes can be constructed efficiently for general domains, we seek to extend the framework used in [21] and summarized in Sect. 2.3. Structured grids provide certain advantages in building the stencils quickly, but as in two dimensions, the boundary needs to be more resolved than the interior and it is necessary to preserve an appropriate gap δ between interior and boundary points. In two dimensions, where the boundary is a one-dimensional curve, this is fairly straightforward. However, it is much more difficult to find an optimal sampling of boundary points in three dimensions.

We begin by identifying interior points. Our strategy is to begin with a uniform discretization of a cube C covering the domain ($\Omega \subset C$), then reduce to only the interior points. Denote the grid by $\mathcal{G}$. Define $\mathbf{x}_{ijk}$, $i, j, k = 0, \ldots, n$ to be the nodes of the discretized cube C and let h be the space between adjacent nodes.

Next we define the signed-distance function to the boundary of the domain Ω,

$$G(\mathbf{x}) = \begin{cases} \text{dist}(\mathbf{x}, \partial\Omega) & \mathbf{x} \notin \Omega \\ -\text{dist}(\mathbf{x}, \partial\Omega) & \mathbf{x} \in \Omega \\ 0 & \mathbf{x} \in \partial\Omega. \end{cases}$$

As in two dimensions, we will require that there be some separation δ between the interior and the boundary in order to consistently resolve directional derivatives near the boundary. Thus, the interior points in $\mathcal{G}$ are chosen to be

$$\mathbf{x}_{ijk} \in C \text{ s.t. } G(\mathbf{x}_{ijk}) + \delta < 0.$$

This ensures that there is a distance of at least δ between the boundary and any interior point. A two-dimensional visualization of this process is shown in Fig. 3a.

Note that although we start with the discretization of a cube, this restriction can be applied to arbitrarily complicated three-dimensional regions.

Next we describe the discretization of the boundary, which must have an effective resolution $h_B \ll \delta \leq h$ in order to produce consistent schemes near the boundary.

At each candidate interior point near the boundary, we focus on the following small cubes

$$C_{ijk} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}], \tag{7}$$

which we define for any i, j, k such that $C_{ijk} \cap \partial\Omega$ is non-empty. Here $x_{ijk} = (x_i, y_j, z_k)$, and $x_{i+1} = x_i + h$. To identify these, we seek any such cube such that at least one corner $\mathbf{x}_- \in C_{ijk}$ satisfies $G(\mathbf{x}_-) < 0$ and one corner $\mathbf{x}_+ \in C_{ijk}$ satisfies $G(\mathbf{x}_+) > 0$.

Then boundary points are added to the point cloud by further discretizing these boundary cubes and using the projection of points sufficiently close to the boundary of the domain. Let C_{ijk} be a boundary cube. We introduce the discretization

$$D_{ijk} = \left\{ (x_i + \tilde{i}h_B, y_j + \tilde{j}h_B, z_k + \tilde{k}h_B) \text{ s.t. } 0 \leq \tilde{i}, \tilde{j}, \tilde{k} \leq n_B \right\} \tag{8}$$

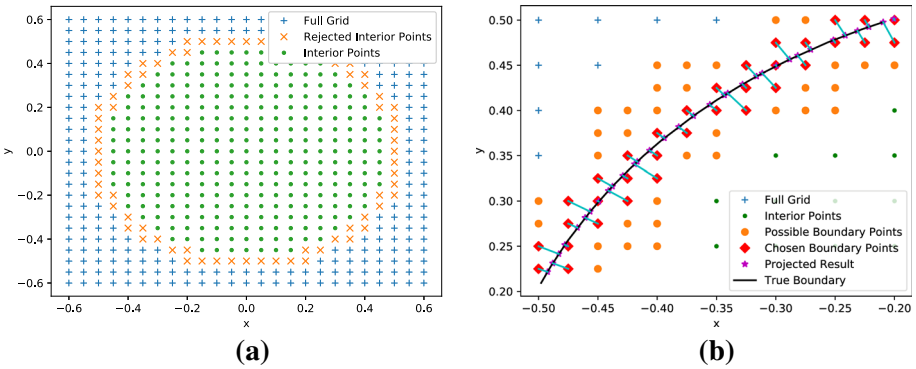

Fig. 3 A two-dimensional visualization of the construction of the grid. **a** Candidate interior points, with points too close to the boundary rejected. **b** Candidate boundary points, with selected boundary points projected onto the true boundary (Color figure online)

where the boundary resolution $h_B = \mathcal{O}(h/n_B)$.

As candidates for boundary points, we select any points $\mathbf{x} \in D_{ijk}$ such that $G(\mathbf{x}) < \frac{h_B}{2}$. We then project each of these candidates onto the true boundary and include the results $\text{Proj}_{\partial\Omega}(\mathbf{x})$ in our point cloud $\mathcal{G}$. The projection is computed using the signed distance function $G(x)$ to the boundary of the domain, with the gradient of G giving us the direction of the projection. This allows us to represent the projection as

$$\text{Proj}_{\partial\Omega}(\mathbf{x}) = \mathbf{x} - \eta \nabla G(\mathbf{x}).$$

The value of $\eta \in \mathbb{R}$ is then obtained by using a bisection method to solve

$$G(\mathbf{x} - \eta \nabla G(\mathbf{x})) = 0.$$

A two-dimensional visualization of this process is given in Fig. 3b.

A consequence of the specific sampling procedure utilized in this implementation is that the ratio $\frac{h}{h_B} = n_B$ must be an integer. As the grid is refined, consistency requires that $\frac{h}{h_B} \to \infty$. However, this will not happen continuously. This can translate to sudden increases in accuracy and computational cost as the grid is refined, rather than a smooth dependence on the grid resolution.

We emphasize again that Hypothesis 2 requires this procedure to lead to an over-resolution of the boundary ($h_B \ll h$), which is necessary (as in 2D) in order to preserve both consistency and monotonicity up to the boundary. There are $\mathcal{O}(n^2)$ boundary cubes, each of which contains $\mathcal{O}(n_B^3)$ points. Of these, we select $\mathcal{O}(n_B^2)$ to project onto the boundary. This leaves us with a total of $\mathcal{O}(n^2 n_B^2)$ boundary points in $\mathcal{G}$, as compared with $\mathcal{O}(n^2)$ boundary points in a traditional three-dimensional Cartesian grid.

3.2 Approximation of Second Derivatives

Next we describe a process for constructing a (negative) monotone approximation of the second directional derivative $u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0)$. We look for schemes of the form

$$u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0) \approx \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u_0 = \sum_{j=1}^{m} a_j (u(\mathbf{x}_j) - u(\mathbf{x}_0)) \tag{9}$$

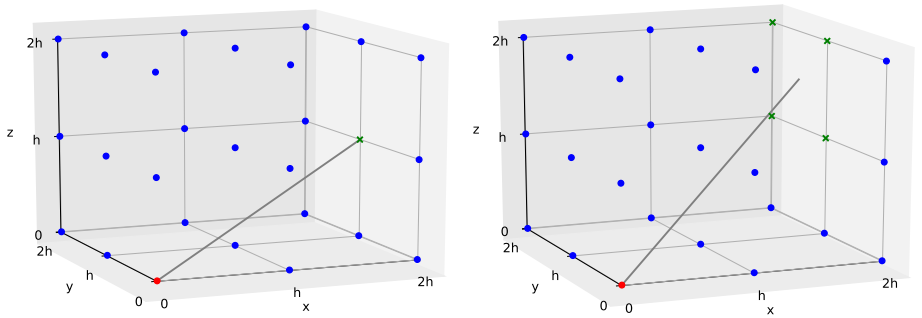

Fig. 4 Examples of **a** a perfectly aligned neighbor along the vector $\mathbf{x}_0 + \boldsymbol{\nu} h$ and **b** four non-aligned neighbors along this direction (Color figure online)

where each $\mathbf{x}_j \in \mathcal{G} \cap B(\mathbf{x}_0, \epsilon)$ is a nearby grid point and all $a_j \geq 0$ for monotonicity.

In the simplest setting, where the direction $\boldsymbol{\nu}$ is grid aligned (Fig. 4a), we can simply use a standard centered difference discretization. That is, suppose that $\boldsymbol{\nu} \in \mathbb{Z}^3$ with $|\boldsymbol{\nu}| h < \epsilon$ and $\mathbf{x}_0 \pm \boldsymbol{\nu} h \in \mathcal{G}$. Then we define

$$\mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u_0 = \frac{u(\mathbf{x}_0 + \boldsymbol{\nu} h) + u(\mathbf{x}_0 - \boldsymbol{\nu} h) - 2u(\mathbf{x}_0)}{|\boldsymbol{\nu}|^2 h^2}, \tag{10}$$

which satisfies all the requirements outlined above.

This is the approach taken by traditional wide stencil schemes. However, there are some clear situations where this simple scheme is not available: (1) if $\boldsymbol{\nu}$ is not grid-aligned, (2) if $\boldsymbol{\nu}$ is grid-aligned but requires a stencil wider than our chosen search radius ϵ, and (3) at points $\mathbf{x}_0$ near the boundary where one or both of $\mathbf{x}_0 \pm \boldsymbol{\nu} h$ can lie outside the domain.

3.2.1 Generalized Finite Difference Schemes

At points where the simple centered scheme is not available, we will follow the approach of [21] and develop generalized finite difference schemes. For clarity of exposition, we begin by considering the approximation of $u_{xx}(\mathbf{x}_0)$.

From the reference point $\mathbf{x}_0$ (which is treated as the origin), we define an orthogonal coordinate system using the standard Cartesian coordinate axes $\hat{x}, \hat{y}, \hat{z}$. Let $O_i, i = 1, \ldots, 8$ denote the eight octants defined by these axes. We can also use this coordinate frame to define spherical coordinates (r_j, θ_j, ϕ_j) corresponding to any grid point $\mathbf{x}_j \in \mathcal{G}$.

In order to derive an appropriate scheme for u_{xx} we rely on Taylor expansion. That is, we seek a scheme of the form

$$\begin{aligned}\mathcal{D}_{xx}u(\mathbf{x}_0) &= \sum_{j=1}^{m} a_j (u(\mathbf{x}_j) - u(\mathbf{x}_0)) \\ &= \sum_{j=1}^{m} a_j \Big[u_x(\mathbf{x}_0)(x_j - x_0) + u_y(\mathbf{x}_0)(y_j - y_0) + u_z(\mathbf{x}_0)(z_j - z_0) \\ &\quad + u_{xx}(\mathbf{x}_0)\frac{(x_j - x_0)^2}{2} + u_{yy}(\mathbf{x}_0)\frac{(y_j - y_0)^2}{2} + u_{zz}(\mathbf{x}_0)\frac{(z_j - z_0)^2}{2} \\ &\quad + u_{xy}(\mathbf{x}_0)(x_j - x_0)(y_j - y_0) + u_{xz}(\mathbf{x}_0)(x_j - x_0)(z_j - z_0) \\ &\quad + u_{yz}(\mathbf{x}_0)(y_j - y_0)(z_j - z_0) + \mathcal{O}(|\mathbf{x}_j - \mathbf{x}_0|^3) \Big]\end{aligned} \tag{11}$$

To simplify this expansion, we want to choose neighbors $\mathbf{x}_j$ that are well-aligned with the x-axis so that $|y_j - y_0|, |z_j - z_0| = o(|x_j - x_0|)$. We recall that in a perfectly grid-aligned scheme, we would have $y_j - y_0 = z_j - z_0 = 0$. Inspired by the two-dimensional approach of [21], we seek one point in each octant O_j such that $\mathbf{x}_j - \mathbf{x}_0$ is as well-aligned as possible with the direction $\hat{x}$. That is, we define

$$\mathbf{x}_j = \operatorname{argmin} \left\{ \theta^2 + (\phi - \pi/2)^2 \mid \mathbf{x} \in O_j \cap \mathcal{G} \cap B(\mathbf{x}_0, \epsilon) \right\}. \tag{12}$$

See Fig. 4a for the selection of four neighbors well-aligned with the vector $\boldsymbol{\nu}$; the four neighbors along the direction $-\boldsymbol{\nu}$ are chosen similarly.

Notice that with slight modification, the spherical coordinates θ_j and $\phi_j - \pi/2$ of these neighbors now play the role of the angular resolution $d\phi$ introduced in Sect. 2.3. In particular, we expect that $\theta_j, \phi_j - \pi/2 = \mathcal{O}(h/\epsilon + h_B/\delta)$ as in two dimensions, which is summarized in the following lemma.

Lemma 1 (Angular resolution) *Let $\mathcal{G}$ be a point cloud satisfying Hypothesis 2. Then the points $\mathbf{x}_j$ defined by (12) exist and have polar coordinates satisfying $\theta_j, \phi_j - \pi/2 = \mathcal{O}(h/\epsilon + h_B/\delta)$.*

Proof Without loss of generality, we consider $j = 1$. We first suppose that the search ball $B(\mathbf{x}_0, \epsilon)$ is contained in the domain Ω.

Now we define the point $\tilde{\mathbf{x}}$ with spherical coordinates $\tilde{r} = \epsilon - h$, $\tilde{\theta} = \sin^{-1}(2h/\epsilon)$, $\tilde{\phi} = \pi/2 - \sin^{-1}(2h/\epsilon)$. By design, the small ball $B(\tilde{\mathbf{x}}, h) \subset B(\mathbf{x}_0, \epsilon)$. Moreover, from the definition of h, this small ball contains at least one discretization point $\tilde{\mathbf{x}}_1 \in \mathcal{G} \cap B(\mathbf{x}_0, \epsilon)$.

In Cartesian coordinates, the center of this small ball is given by

$$\tilde{x} = \frac{(\epsilon - h)(\epsilon - 2h)(\epsilon + 2h)}{\epsilon^2} = \epsilon + o(\epsilon), \quad \tilde{y} = 2h + o(h), \quad \tilde{z} = 2h + o(h).$$

This ensures that for sufficiently small h, the point $\tilde{\mathbf{x}}_1$ satisfies

$$0 < \epsilon - \frac{3}{2}h < \tilde{x}_1 < \epsilon + \frac{3}{2}h, \quad \frac{1}{2}h < \tilde{y}_1, \tilde{z}_1 < \frac{3}{2}h.$$

In particular, $\tilde{\mathbf{x}}_1$ resides in the first octant and the grid point $\mathbf{x}_1$ defined in (12) is guaranteed to exist.

Finally, we can verify that the spherical coordinates of this point (and thus also the minimizer in (12)) satisfy

$$\tilde{\theta}_1 = \mathcal{O}\left(\frac{\tilde{y}_1}{\tilde{x}_1}\right) = \mathcal{O}\left(\frac{h}{\epsilon}\right), \quad \frac{\pi}{2} - \tilde{\phi}_1 = \mathcal{O}\left(\frac{\tilde{z}_1}{\tilde{r}_1}\right) = \mathcal{O}\left(\frac{h}{\epsilon}\right).$$

The other possibility is that the search ball $B(\mathbf{x}_0, \epsilon)$ is not entirely contained in the domain Ω, and that the above construction does not yield existence of the point $\mathbf{x}_1$. Then we can repeat the above construction, replace h with h_B and ϵ with δ, to obtain $\theta_1, \pi/2 - \phi_1 = \mathcal{O}(h_B/\delta)$. □

As desired, we obtain the relationships

$$\begin{aligned}
|x_j - x_0| &= |\mathbf{x}_j - \mathbf{x}_0| \cos\theta_j \sin\phi_j = |\mathbf{x}_j - \mathbf{x}_0| + \mathcal{O}(|\mathbf{x}_j - \mathbf{x}_0|\, d\phi^2),\\
|y_j - y_0| &= |\mathbf{x}_j - \mathbf{x}_0| \sin\theta_j \sin\phi_j = \mathcal{O}(|\mathbf{x}_j - \mathbf{x}_0|\, d\phi),\\
|z_j - z_0| &= |\mathbf{x}_j - \mathbf{x}_0| \cos\phi_j = \mathcal{O}(|\mathbf{x}_j - \mathbf{x}_0|\, d\phi).
\end{aligned}$$

Returning to the Taylor expansion in (11), we now neglect all terms that are $o(|x_j - x_0|^2)$. This leaves us with

$$\begin{aligned}
\mathcal{D}_{xx}u(\mathbf{x}_0) = \sum_{j=1}^{8} a_j \Big[& u_x(\mathbf{x}_0)(x_j - x_0) + u_y(\mathbf{x}_0)(y_j - y_0)\\
& + u_z(\mathbf{x}_0)(z_j - z_0) + u_{xx}(\mathbf{x}_0)\frac{(x_j - x_0)^2}{2} + \mathcal{O}(|x_j - x_0|^2\, d\phi + |x_j - x_0|^3)\Big].
\end{aligned} \tag{13}$$

Consistency of the approximation, combined with the monotonicity requirement, leaves us with the following system of equations for the coefficients a_j:

$$\begin{cases}
\sum_{j=1}^{8} a_j (x_j - x_0) = 0\\
\sum_{j=1}^{8} a_j (y_j - y_0) = 0\\
\sum_{j=1}^{8} a_j (z_j - z_0) = 0\\
\sum_{j=1}^{8} a_j \frac{(x_j - x_0)^2}{2} = 1\\
a_j \geq 0.
\end{cases} \tag{14}$$

We note that the positivity condition allows us to easily obtain bounds on any solution of this system via the estimate

$$0 \leq a_i = \frac{1}{(x_i - x_0)^2/2} a_i \frac{(x_i - x_0)^2}{2} \leq \frac{1}{(x_i - x_0)^2/2} \sum_{j=1}^{8} a_j \frac{(x_j - x_0)^2}{2} = \frac{2}{(x_i - x_0)^2}.$$

The Taylor expansion (13) then guarantees a discretization error of

$$\sum_{j=1}^{8} a_j \mathcal{O}(|x_j - x_0|^2\, d\phi + |x_j - x_0|^3) = \sum_{j=1}^{8} \mathcal{O}(d\phi + |x_j - x_0|) = \mathcal{O}(h/\epsilon + h_B/\delta + \epsilon).$$

In two dimensions, an explicit solution of the system resulting from this procedure could be obtained. This is not straightforward in three dimensions, and explicit formulas are often

computationally intractable in practice. Instead, we solve the system (14) numerically through a least squares procedure. That is, we notice that this system has the form

$$\begin{cases} M\mathbf{a} = \mathbf{b} \\ \mathbf{a} \geq 0 \end{cases}$$

where M is a 4×8 matrix. Thus, we can easily obtain a solution by solving the low-dimensional linearly constrained least squares problem

$$\begin{cases} \text{minimize} & \frac{1}{2}\|M\mathbf{a} - \mathbf{b}\|_2^2 \\ \text{subject to} & \mathbf{a} \geq 0. \end{cases} \tag{15}$$

In our implementation, this is accomplished using the Python Scipy implementation of the Bounded-variable least-squares algorithm of [36]. This is an active set method, which requires the number of iterations comparable to the number of variables. However, as the size of these optimization problems is independent of the grid resolution, this is not a time-consuming component of the method and the particular choice of optimization routine is not critical to the performance of the overall method.

We remark that this procedure is easily adapted to the approximation of more general second directional derivatives $u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0)$. To accomplish this, we introduce any two vectors $\boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ such that $\boldsymbol{\nu}, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ form an orthonormal set. We use these as our coordinate axes, centered at the point $\mathbf{x}_0$, and introduce the change of coordinates

$$\begin{cases} \bar{x}_j = (\mathbf{x}_j - \mathbf{x}_0) \cdot \boldsymbol{\nu}, \\ \bar{y}_j = (\mathbf{x}_j - \mathbf{x}_0) \cdot \boldsymbol{\nu}_2, \\ \bar{z}_j = (\mathbf{x}_j - \mathbf{x}_0) \cdot \boldsymbol{\nu}_3. \end{cases}$$

The coefficients a_j in the approximation $\mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0)$ in (9) are then obtained by solving the system (14) using these new coordinates in place of x_j, y_j, z_j.

3.2.2 Existence of a Positive Solution

Our procedure for generating consistent, monotone generalized finite difference schemes in 3D hinges on finding a solution of (14) via a least squares procedure. However, it is by no means obvious that a solution satisfying the positivity requirement ($a_j \geq 0$) actually exists in general. Fortunately, this is guaranteed by our careful choice of neighboring points $\mathbf{x}_j$ lying in different octants.

The proof of this relies on Farkas' Lemma [39].

Lemma 2 (Farkas' Lemma) *Let $M \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^{m\times 1}$. Then exactly one of the following two conditions holds:*

- *There exists $\mathbf{a} \in \mathbb{R}^{n\times 1}$ such that $M\mathbf{a} = \mathbf{b}$ and $\mathbf{a} \geq 0$;*
- *There exists $\mathbf{y} \in \mathbb{R}^{m\times 1}$ such that $M^T\mathbf{y} \geq 0, \mathbf{y}^T\mathbf{b} < 0$.*

This allows us to prove the existence of a solution to the scheme (14), which immediately yields existence of a consistent and monotone scheme for $u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0)$.

Lemma 3 (Existence of positive solution) *A positive solution to the system of equations (14) exists if the eight neighbors $\mathbf{x}_j$ lie in different octants as required by (12).*

Proof Since each of the points $\mathbf{x}_j$ is chosen to lie in the octant O_j, we can assign a definite sign to each of the $x_j - x_0$, $y_j - y_0$, and $z_j - z_0$. Moreover, since the $\mathbf{x}_j$ are chosen to minimize $\theta^2 + (\pi/2 - \phi)^2$ and $\mathbf{x}_j \neq \mathbf{x}_0$, we are assured that each $x_j \neq x_0$. The system (14) then takes the form

$$\begin{cases} M\mathbf{a} = \mathbf{b} \\ \mathbf{a} \geq 0 \end{cases}$$

where

$$M = \begin{bmatrix} c_{11} & -c_{12} & -c_{13} & c_{14} & c_{15} & -c_{16} & -c_{17} & c_{18} \\ c_{21} & c_{22} & -c_{23} & -c_{24} & c_{25} & c_{26} & -c_{27} & -c_{28} \\ c_{31} & c_{32} & c_{33} & c_{34} & -c_{35} & -c_{36} & -c_{37} & -c_{38} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} & c_{47} & c_{48} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{16}$$

and all of the $c_{ij} \geq 0$ with $c_{4j} > 0$.

Following Farkas' Lemma, suppose there exists some $\mathbf{y}$ such that $A^T\mathbf{y} \geq 0$ and $\mathbf{b}^T\mathbf{y} < 0$. Notice that this second condition implies that $y_4 < 0$. There are eight possible combinations of signs for the remaining components y_1, y_2, and y_3.

Consider, for example, the case where $y_1, y_2, y_3 \geq 0$. Then we would have

$$(M^T\mathbf{y})_7 = -c_{17}y_1 - c_{27}y_2 - c_{37}y_3 + c_{47}y_4 < 0.$$

Similarly, we can verify that any other possible combination of signs in y_1, y_2, y_3 will require at least one component of $M^T\mathbf{y}$ to be negative.

We conclude that there is no $\mathbf{y}$ such that both $M^T\mathbf{y} \geq 0$ and $\mathbf{y}^T\mathbf{b} < 0$. By Farkas' Lemma, we infer the existence of a solution to (14). □

We should also remark that this same strategy can be used using one perfectly aligned neighbor along the direction $\boldsymbol{\nu}$ (the setting of Fig. 4a) and four non-aligned neighbors along the direction $-\boldsymbol{\nu}$ (the setting of Fig. 4a). This situation can easily arise near the boundary, involving one perfectly aligned interior neighbor and four non-aligned boundary neighbors. The proof of Lemma 3 is unchanged since it does not require the assumption that the points $\mathbf{x}_1, \ldots, \mathbf{x}_8$ belonging to the eight octants are pairwise distinct. In particular, a perfectly aligned neighbor simply belongs to 4 such octants.

Corollary 3 (Existence of a consistent, monotone scheme) *Under the assumptions of Hypothesis 2, let $\boldsymbol{\nu}$ be any unit vector in $\mathbb{R}^3$. Then the procedure described in Sect. 3.2.1 yields a consistent, monotone approximation of the negated second directional derivative $-u_{\boldsymbol{\nu}\boldsymbol{\nu}}(\mathbf{x}_0)$. Moreover, the formal discretization error is $\mathcal{O}(\epsilon + h/\epsilon + h_B/\delta)$.*

3.3 Approximation of Nonlinear Operators

We can utilize these monotone approximations of the second directional derivatives to solve a wide class of fully nonlinear elliptic PDEs. In this case of equations of the type (1) that explicitly depend on directional derivatives over a finite subset of directions, an appropriate discretization is immediate.

$$F^h(\mathbf{x}, u(\mathbf{x}), u(\mathbf{x}) - u(\cdot)) = F\left(\mathbf{x}, u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}}u(\mathbf{x}); \boldsymbol{\nu} \in \mathcal{A}\right), \quad \mathbf{x} \in \mathcal{G} \cap \Omega. \tag{17}$$

Monotonicity follows immediately from the fact that the elliptic operator F is a non-increasing function of the second directional derivatives, which are approximated with a negative monotone discretization.

We are also interested in functions of the eigenvalues of the Hessian (2). The Rayleigh quotient formulation that was used in two dimensions (6) does not immediately provide an expression for all three of the eigenvalues that we need to consider in three dimensions. Straightforward modifications of this formula, involving minimization over an appropriate subspace of $\mathbb{R}^3$ to obtain the middle eigenvalue $\lambda_2(D^2u)$, do not lead to monotone approximations for $\lambda_2(D^2u)$.

Instead, we use properties from linear algebra to provide a new formulation that will allow for an appropriate monotone approximation of nonlinear functions of $\lambda_1(D^2u)$, $\lambda_2(D^2u)$, $\lambda_3(D^2u)$. To accomplish this, we first consider all sets of orthonormal coordinate frames in $\mathbb{R}^d$ (where we are particularly interested in $d = 3$ in this work).

$$V = \{(\boldsymbol{v}_1, ..., \boldsymbol{v}_d) \mid \boldsymbol{v}_j \in \mathbb{R}^d, \|\boldsymbol{v}_j\|_2 = 1, \boldsymbol{v}_j \perp \boldsymbol{v}_i \forall i \neq j\}. \tag{18}$$

Lemma 4 (Functions of eigenvalues) *Let $G : \mathbb{R} \to \mathbb{R}$ be non-increasing, $\phi : \mathbb{R} \to \mathbb{R}$ concave, and A a symmetric real-valued $d \times d$ matrix. Then*

$$G\left(\sum_{j=1}^{d} \phi(\lambda_j(A))\right) = \max_{(\boldsymbol{v}_1, \boldsymbol{v}_2, ...\boldsymbol{v}_d) \in V} G\left(\sum_{j=1}^{d} \phi(\boldsymbol{v}_j^T A \boldsymbol{v}_j)\right). \tag{19}$$

Proof Since A is a real-valued symmetric matrix, we can find d orthonormal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$. Any $(\boldsymbol{v}_1, \boldsymbol{v}_2, ...\boldsymbol{v}_d) \in V$ can be expressed as a linear combination of these eigenvectors:

$$\boldsymbol{v}_j = \sum_{k=1}^{d} c_{jk} \mathbf{v}_k = \sum_{k=1}^{d} (\boldsymbol{v}_j^T \mathbf{v}_k) \mathbf{v}_k.$$

Since $\boldsymbol{v}_j$ and $\mathbf{v}_j$ are both orthonormal, we can also compute

$$\sum_{k=1}^{d} c_{jk}^2 = \left(\sum_{k=1}^{d} c_{jk} \mathbf{v}_k\right)\left(\sum_{l=1}^{d} c_{jl} \mathbf{v}_l\right) = \boldsymbol{v}_j^T \boldsymbol{v}_j = 1,$$
$$\sum_{j=1}^{d} c_{jk}^2 = \mathbf{v}_k^T \left(\sum_{j=1}^{d} \boldsymbol{v}_j \boldsymbol{v}_j^T\right) \mathbf{v}_k = \mathbf{v}_k^T I \mathbf{v}_k = \mathbf{v}_k^T \mathbf{v}_k = 1.$$

Now for any unit vector $\boldsymbol{v}_j$, we can use Jensen's inequality to estimate

$$\phi(\boldsymbol{v}_j^T A \boldsymbol{v}_j) = \phi\left(\sum_{k=1}^{d} c_{jk}^2 \lambda_k\right) \geq \sum_{k=1}^{d} c_{jk}^2 \phi(\lambda_k).$$

Summing these concave functions yields

$$\sum_{j=1}^{d} \phi(\boldsymbol{v}_j^T A \boldsymbol{v}_j) \geq \sum_{j=1}^{d} \sum_{k=1}^{d} c_{jk}^2 \phi(\lambda_k) = \sum_{k=1}^{d} \phi(\lambda_k)$$

with equality if the $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ coincide with the eigenvectors $(\mathbf{v}_1, \ldots, \mathbf{v}_d)$ of A.

Since G is non-increasing, we conclude that

$$G\left(\sum_{j=1}^{d} \phi(\lambda_j(A))\right) = \max_{(\boldsymbol{v}_1, \boldsymbol{v}_2, ...\boldsymbol{v}_d) \in V} G\left(\sum_{j=1}^{d} \phi(\boldsymbol{v}_j^T A \boldsymbol{v}_j)\right).$$

□

Remark 2 This also applies if G is non-decreasing and ϕ convex, with the maximum replaced with a minimum.

This formulation immediately suggests a consistent, monotone approximation of the functions of the eigenvalues of the Hessian D^2u since $\boldsymbol{v}^T(D^2u)\boldsymbol{v}$ is identical to the second directional derivative $u_{\boldsymbol{vv}}$. That is, for equations of the form (2),

$$F(\lambda_1(D^2u), \lambda_2(D^2u), \lambda_3(D^2u)) \approx \max_{(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_d) \in V} G\left(\sum_{j=1}^{d} \phi(\mathcal{D}_{\boldsymbol{v}_j \boldsymbol{v}_j} u)\right). \tag{20}$$

However, this is not computationally feasible as it requires computing a maximum over an infinite set of orthogonal coordinate frames.

Instead, we must consider some finite subset V^h of the possible orthogonal frames V. We begin with a finite subset $E^h \subset \mathbb{S}^2$ of unit vectors in $\mathbb{R}^3$. Then we let

$$V^h = \{(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d) \in E^h \mid \boldsymbol{v}_j \perp \boldsymbol{v}_i \forall j \neq i\}.$$

We can define the angular resolution of this subset to be

$$d\theta = \max_{(\mathbf{v}_1, \ldots, \mathbf{v}_d) \in V} \min_{(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d) \in V^h} \max_i \cos^{-1}(\mathbf{v}_i \cdot \boldsymbol{v}_i). \tag{21}$$

That is, for each frame in V, we first find the frame in V^h that minimizes the worst case angle between $\mathbf{v}_i$ and $\boldsymbol{v}_i$. Then, we maximize over all possible frames in V to find the worst case $d\theta$.

We remark that V is a manifold of dimension $D = \frac{d(d-1)}{2}$, known as the orthogonal group. Thus in three dimensions, a necessary condition for achieving an angular resolution $d\theta$ is that the number of elements in V^h satisfies $\#(V^h) \geq c(d\theta)^{-D} = c(d\theta)^{-3}$ for some fixed $c > 0$.

Many suitable choices of V^h are possible, and this immediately leads to an appropriate discretization.

Lemma 5 (Consistent, monotone approximation) *Consider a grid $\mathcal{G}$ satisfying Hypothesis 2 and a finite set $V^h \subset V$ chosen so that the angular resolution $d\theta \to 0$ as $h \to 0$. Let G be continuous and non-increasing and ϕ be concave. Then*

$$F^h(\mathbf{x}, u(\mathbf{x}) - u(\cdot)) = \max_{(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_d) \in V^h} G\left(\sum_{j=1}^{d} \phi(\mathcal{D}_{\boldsymbol{v}_j \boldsymbol{v}_j} u(\mathbf{x}))\right), \quad \mathbf{x} \in \mathcal{G} \cap \Omega \tag{22}$$

is a consistent, monotone approximation of (2).

Remark 3 If the functions G and ϕ are Lipschitz continuous, we can use the Lipschitz continuity of the maximum operator and the formal consistency error from Corollary 3 to deduce a formal consistency error of $\mathcal{O}(\epsilon + h/\epsilon + h_B/\delta + d\theta)$.

3.4 Boundary Conditions

We now turn our attention to the approximation of boundary conditions. Dirichlet boundary conditions are straightforward. However, we are also interested in constructing monotone schemes for Neumann or Robin boundary conditions, as well as the nonlinear second type (optimal transport) boundary condition $\nabla u(\Omega) \subset \bar{\Omega}^*$.

3.4.1 Approximation of First Derivatives

We begin by describing the approximation of first directional derivatives in directions $\mathbf{n}$ exterior to the domain. That is, letting $\mathbf{n}_\mathbf{x}$ be the unit outward normal to the domain at the point $\mathbf{x} \in \partial\Omega$, we discretize derivatives $u_\mathbf{n}(\mathbf{x}_0) = \nabla u(\mathbf{x}_0) \cdot \mathbf{n}$ for directions $\mathbf{n}$ satisfying $\mathbf{n} \cdot \mathbf{n}_{\mathbf{x}_0} > 0$.

As with the interior, simple schemes can be used if the direction $\mathbf{n}$ is well-aligned with the grid (locally at the boundary point $\mathbf{x}_0$). That is, if $\mathbf{n} \in \mathbb{Z}^3$ and $|\mathbf{n}|\, h < \epsilon$, we could utilize the upwind scheme

$$\mathcal{D}_\mathbf{n}(\mathbf{x}_0) = \frac{u(\mathbf{x}_0) - u(\mathbf{x}_0 - \mathbf{n}h)}{|\mathbf{n}|\, h}. \tag{23}$$

However, given that the boundary is highly resolved relative to the interior and that complicated domains are possible, we do not expect this simple approximation to be possible in general.

In the interior, we were able to construct monotone schemes by choosing neighbors in different octants relative to the direction and the point of interest. On the boundary, a similar approach yields monotone schemes for the first directional derivatives using only interior neighbors.

Taylor expanding as before, we get

$$\begin{aligned}\mathcal{D}_\mathbf{n} u(\mathbf{x}_0) &= \sum_{j=1}^{m} a_j (u(\mathbf{x}_j) - u(\mathbf{x}_0)) \\ &= \sum_{j=1}^{m} a_j \left[u_x(\mathbf{x}_0)(x_j - x_0) + u_y(\mathbf{x}_0)(y_j - y_0) + u_z(\mathbf{x}_0)(z_j - z_0) + \mathcal{O}(|\mathbf{x}_j - \mathbf{x}_0|^2) \right].\end{aligned} \tag{24}$$

Consistency is achieved by equating the coefficients of the various first partial derivatives to the components n_1, n_2, n_3 of the unit direction $\hat{\mathbf{n}}$. Coupled with the (positive) monotonicity requirement, we obtain the system

$$\begin{cases} \sum_{j=1}^{m} a_j (x_j - x_0) = n_1 \\ \sum_{j=1}^{m} a_j (y_j - y_0) = n_2 \\ \sum_{j=1}^{m} a_j (z_j - z_0) = n_3 \\ a_j \le 0. \end{cases} \tag{25}$$

As a simple way of selecting appropriate neighbors, we let C_{ijk} be the first small cube (7) entered by the ray $\mathbf{x}_0 - t\mathbf{n}$. We choose as neighbors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ the four vertices of the face through which this ray enters the small cube.

Lemma 6 (Existence of a negative solution) *A negative solution to the system of equations (25) exists if $\mathbf{x}_0 - t\mathbf{n}$ lies in the convex hull of the four vertices $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ of a square for some $t > 0$.*

Proof Since $\mathbf{x}_0 - t\mathbf{n}$ lies in the convex hull of the four corners of a square, then it also lies in the convex hull of three of these points. Without loss of generality, let these be $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$.

Then there exist $\lambda_1, \lambda_2, \lambda_3 \in [0, 1]$ with $\lambda_1 + \lambda_2 + \lambda_3 = 1$ such that

$$\mathbf{x}_0 - t\mathbf{n} = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \lambda_3 \mathbf{x}_3.$$

Now we let $v(\mathbf{x})$ be the piecewise linear interpolant of the values of $u(\mathbf{x})$ at the points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^3$. Since v is linear, we can compute its first directional derivative in the direction $\mathbf{n}$ via

$$\begin{aligned} \frac{\partial v}{\partial \mathbf{n}} &= \frac{v(\mathbf{x}_0) - v(\mathbf{x}_0 - t\mathbf{n})}{t} \\ &= \frac{v(\mathbf{x}_0) - \lambda_1 v(\mathbf{x}_1) - \lambda_2 v(\mathbf{x}_2) - \lambda_3 v(\mathbf{x}_3)}{t}. \end{aligned}$$

Then we can easily verify that

$$a_1 = -\frac{\lambda_1}{t},\ a_2 = -\frac{\lambda_2}{t},\ a_3 = -\frac{\lambda_3}{t},\ a_4 = 0$$

is a solution of (25). □

Corollary 4 (Existence of a consistent, monotone scheme) *Consider a grid $\mathcal{G}$ satisfying Hypothesis 2 and let $\mathbf{n}$ be any vector in $\mathbb{R}^3$ exterior to the domain Ω at the point $\mathbf{x}_0 \in \partial\Omega$. Then the procedure described in Sect. 3.4.1 yields a consistent, monotone approximation of the first directional derivative $u_{\mathbf{n}}(\mathbf{x}_0)$.*

3.4.2 Approximation of Optimal Transport Conditions

In optimal transport and many geometric PDEs [8], a traditional boundary condition is replaced by the so-called second type boundary condition $\nabla u(\Omega) \subset \bar{\Omega}^*$ where $\Omega^* \subset \mathbb{R}^3$ is convex and the solution u is also required to be convex.

This global constraint can be re-expressed as a nonlinear Hamilton–Jacobi equation on the boundary

$$H(\nabla u(\mathbf{x})) = 0, \quad \mathbf{x} \in \partial\Omega \tag{26}$$

where H is the signed distance function to the boundary of the target set Ω^*. By utilizing the Legendre-Fenchel transform, it is possible to rewrite this in the form

$$\sup_{\mathbf{n}\cdot\mathbf{n}_{\mathbf{x}}>0} \left\{ \nabla u(\mathbf{x}) \cdot \mathbf{n} - H^*(\mathbf{n}) \right\} = 0, \quad \mathbf{x} \in \partial\Omega. \tag{27}$$

This is established in [5, Lemma 2.4], and is due to the fact that the solution u we seek is the gradient of a convex function.

This immediately allows us to construct an appropriate discretization using our approximations for first directional derivatives $u_{\mathbf{n}}$ and the finite subset E^h of unit vectors in $\mathbb{R}^3$.

Lemma 7 (Consistent, monotone approximation) *Consider a grid $\mathcal{G}$ satisfying Hypothesis 2 and a finite set $E^h \subset \mathbb{S}^2$ chosen so that the angular resolution $d\theta \to 0$ as $h \to 0$. Then*

$$H^h(\mathbf{x}, u(\mathbf{x}) - u(\cdot)) = \max \left\{ \mathcal{D}_{\boldsymbol{n}} u(\mathbf{x}) - H^*(\mathbf{n}) \mid \mathbf{n} \in E^h,\ \mathbf{n} \cdot \mathbf{n}_{\mathbf{x}} > 0 \right\}, \quad \mathbf{x} \in \mathcal{G} \cap \partial\Omega \tag{28}$$

is a consistent, monotone approximation of (27).

4 Implementation

The preceding section shows how to define a consistent, monotone approximation for a wide range of fully nonlinear elliptic operators in three dimensions. However, naive evaluation of these approximation schemes may be computationally intractable in three dimensions. In this section, we describe the details of our implementation that allow us to evaluate these schemes efficiently. This discussion can be separated into two parts: discussion of a particular discretization and discussion of a particular solution method.

In Sect. 4.1, we propose a particular choice of discretization parameters that is designed to enhance the efficiency of evaluating the finite difference scheme. The resulting scheme fits within the framework of Sect. 3 and relevant convergence results apply.

In Sects. 4.2–4.3, we propose a combination of an active set method with a Gauss-Seidel iteration for solving the system of nonlinear algebraic equations resulting from our discretization. This requires many evaluations of the nonlinear operator. As this can be expensive, we utilize a multi-level approach for efficiently estimating the value of this nonlinear operator. The techniques in these sections are proposed without proof of convergence. However, these methods perform well in practice (see Sect. 5) and we can easily verify that the solution we obtain does in fact satisfy the desired system of nonlinear algebraic equations.

4.1 Discretization Parameters

There are many valid choices for the parameters used to construct our numerical discretization. The particular choices used in our implementation are motivated by the need for efficiency that is brought to the forefront in three dimensions.

We begin with the parameters used to define the grid in Sect. 3.1, particularly the boundary resolution h_B and the gap to the boundary δ. We recall that the total number of boundary points scales like $\mathcal{O}(n^2 n_B^2)$ while the total number of interior points scales like $\mathcal{O}(n^3)$. While the boundary has to be more highly resolved than in a traditional finite difference grid, we still desire the number of boundary points to be less than the number of interior points to prevent this high resolution from significantly impacting computational cost. With this in mind, we choose $n_B \approx n^{1/4}$, so that the total number of boundary points is $\mathcal{O}(n^{5/2}) \ll n^3$. Note that the boundary resolution is then $h_B = \mathcal{O}(h/n_B) = \mathcal{O}(h^{5/4})$, which is asymptotically less than h as required by Hypothesis 2. In order to satisfy the condition $h_B \ll \delta$, we choose $\delta = h/2$.

The procedure for approximating second directional derivatives also requires us to define a search radius $\epsilon \gg h$. We recall that (as long as we are not too close to the boundary), the discretization error of these approximations depends on both the effective spatial resolution ϵ and the angular resolution $d\phi = \mathcal{O}(h/\epsilon)$ (Corollary 3). Motivated by the desire to balance these two components of the error, we choose $\epsilon = \sqrt{h}$ for an overall discretization error of $\mathcal{O}(\sqrt{h})$. We note that the discretization error near the boundary will be slightly larger in this implementation as it scales like $\mathcal{O}(h_B/\delta) = \mathcal{O}\left(h^{5/4}/h\right) = \mathcal{O}(h^{1/4})$. However, this occurs in only a narrow band near the boundary and need not necessarily affect the scaling of the overall error in the computed solution.

If we are considering functions of the eigenvalues of the Hessian (2), we also need to define a discretization V^h of orthogonal coordinate frames. In two dimensions, this is very straightforward. However, in three dimensions, the number of possible coordinate frames can quickly become very large as the resolution $d\theta$ of V^h is improved. In light of the need to conserve computational resources, we would like to make use of the simple centered

scheme (10) as much as possible. For this reason, we restrict our attention to grid aligned directions remaining within our search radius and define

$$E^h = \left\{ \boldsymbol{\nu} \in \mathbb{Z}^3 \mid \|\boldsymbol{\nu}_j\|_\infty \leq \epsilon/h \right\}. \tag{29}$$

and

$$V^h = \left\{ (\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \mid \boldsymbol{\nu}_j \in E^h, \boldsymbol{\nu}_i \perp \boldsymbol{\nu}_j \forall i \neq j \right\}. \tag{30}$$

Then the more complicated generalized schemes only need to be constructed in a band of width ϵ around the boundary. We provide a simple construction showing that this leads to a consistent scheme, though we do not pursue an optimal bound on the angular resolution $d\theta$.

Lemma 8 (Discretization of coordinate frames) *Let $(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in V$. Then the angular resolution of the discretization V^h in (30) is bounded by $d\theta = \mathcal{O}\left((h/\epsilon)^{1/6}\right)$.*

Proof We first define the integer $m = \lfloor \left(\frac{\epsilon}{\sqrt{31}h} \right)^{1/3} \rfloor$. Since Hypothesis 2 requires $h \ll \epsilon$, we have $m \geq 1$ for small enough h. Now we seek an element $(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in V^h$ such that

$$\boldsymbol{\nu}_i \cdot \hat{\boldsymbol{\nu}}_i \geq 1 - \frac{C}{m}$$

for every $i = 1, 2, 3$ and for some fixed $C > 0$.

Let $v_{i,1}, v_{i,2}, v_{i,3}$ be the three components of $\boldsymbol{\nu}_i$. We suppose without loss of generality that $v_{1,3} \geq 1/\sqrt{3}$ and propose the following choice:

$$\begin{aligned}
\boldsymbol{\nu}_1 &= \left(\lfloor m v_{1,1} \rfloor, \lfloor m v_{1,2} \rfloor, \lfloor m v_{1,3} \rfloor \right) \\
\boldsymbol{\nu}_2 &= \left(\lfloor m v_{1,3} \rfloor \lfloor m v_{2,1} \rfloor, \lfloor m v_{1,3} \rfloor \lfloor m v_{2,2} \rfloor, -\lfloor m v_{1,1} \rfloor \lfloor m v_{2,1} \rfloor - \lfloor m v_{1,2} \rfloor \lfloor m v_{2,2} \rfloor \right) \\
\boldsymbol{\nu}_3 &= \boldsymbol{\nu}_1 \times \boldsymbol{\nu}_2.
\end{aligned}$$

Note that by construction these vectors have integer components and are mutually orthogonal.

We now produce estimates on the relevant norms and inner products by relying on the orthonormality of $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ and the following bounds on the floor function:

$$m v_{i,j} - 1 \leq \lfloor m v_{i,j} \rfloor \leq m v_{i,j}.$$

Since $m \to \infty$ as $h \to 0$, we may also express this asymptotically as $\lfloor m v_{i,j} \rfloor = m v_{i,j} + \mathcal{O}(1)$.

We can then rewrite the first two vectors as

$$\begin{aligned}
\boldsymbol{\nu}_1 &= \left(m v_{1,1} + \mathcal{O}(1), m v_{1,2} + \mathcal{O}(1), m v_{1,3} + \mathcal{O}(1) \right) \\
\boldsymbol{\nu}_2 &= \left(m^2 v_{1,3} v_{2,1} + \mathcal{O}(m), m^2 v_{1,3} v_{2,2} + \mathcal{O}(m), -m^2 v_{1,1} v_{2,1} - m^2 v_{1,2} v_{2,2} + \mathcal{O}(m) \right) \\
&= m^2 v_{1,3} \left(v_{2,1} + \mathcal{O}(1/m), v_{2,2} + \mathcal{O}(1/m), v_{2,3} + \mathcal{O}(1/m) \right).
\end{aligned}$$

Here the simplification in the last line is due to the fact that $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are orthogonal, which allows us to write

$$v_{2,3} = -\frac{1}{v_{1,3}} (v_{1,1} v_{2,1} + v_{1,2} v_{2,2}).$$

Note that the asymptotic bounds we obtain are independent of the vectors $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3$ since all components are less than one in magnitude and $v_{1,3}$ is bounded away from zero.

A more careful estimate of the errors produced by the floor function quickly leads to the bound

$$\|\boldsymbol{\nu}_1\|_2^2 \leq m^2 (v_{1,1}^2 + v_{1,2}^2 + v_{1,3}^2) = m^2.$$

Applying the same procedure to $\boldsymbol{v}_2$, we find that

$$\begin{aligned}\|\boldsymbol{v}_2\|_2^2 &\leq m^4(v_{1,3}^2 v_{2,1}^2 + v_{1,3}^2 v_{2,2}^2 + v_{1,1}^2 v_{2,1}^2 + v_{1,2}^2 v_{2,2}^2) + 2\lfloor m v_{1,1}\rfloor\lfloor m v_{2,1}\rfloor\lfloor m v_{1,2}\rfloor\lfloor m v_{2,2}\rfloor \\ &= m^4 v_{1,3}^2\|\boldsymbol{v}_2\|_2^2 + 2\left(\lfloor m v_{1,1}\rfloor\lfloor m v_{2,1}\rfloor\lfloor m v_{1,2}\rfloor\lfloor m v_{2,2}\rfloor - m^4 v_{1,1}v_{2,1}v_{1,2}v_{2,2}\right).\end{aligned}$$

Taking into consideration the different possibilities for the signs of $v_{1,1}$, $v_{2,1}$, $v_{1,2}$, $v_{2,2}$, and recalling that these terms must be less than one in magnitude, we obtain the bound

$$\begin{aligned}\|\boldsymbol{v}_2\|_2^2 &\leq m^4 v_{1,3}^2 + 2\left((m+1)^4 - m^4\right) \\ &\leq m^4 + 30m^3 = m^4\left(1 + \frac{30}{m}\right) \leq 31m^4.\end{aligned}$$

We conclude that

$$\|\boldsymbol{v}_1\|_2 \leq m, \quad \|\boldsymbol{v}_2\|_2 \leq m^2 v_{1,3} + \mathcal{O}(m) \leq \sqrt{31}m^2.$$

Because these are orthogonal, we also obtain

$$\|\boldsymbol{v}_3\|_2 = \|\boldsymbol{v}_1\|_2\|\boldsymbol{v}_2\|_2 \leq \sqrt{31}m^3.$$

This immediately implies that for all $i = 1, 2, 3$ we have

$$\|\boldsymbol{v}_i\|_\infty \leq \sqrt{31}m^3 \leq \epsilon/h$$

so that indeed $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) \in V^h$.

Relying on the fact that the vectors $\boldsymbol{v}_i$ are orthonormal, we can also estimate the relevant inner products as

$$\boldsymbol{v}_1 \cdot \boldsymbol{v}_1 = m + \mathcal{O}(1), \quad \boldsymbol{v}_2 \cdot \boldsymbol{v}_2 = m^2 v_{1,3} + \mathcal{O}(m).$$

This leads to the bounds

$$\begin{aligned}\boldsymbol{v}_1 \cdot \hat{\boldsymbol{v}}_1 &\geq \frac{m + \mathcal{O}(1)}{m} = 1 + \mathcal{O}\left(\frac{1}{m}\right) \\ \boldsymbol{v}_2 \cdot \hat{\boldsymbol{v}}_2 &\geq \frac{m^2 v_{1,3} + \mathcal{O}(m)}{m^2 v_{1,3} + \mathcal{O}(m)} = 1 + \mathcal{O}\left(\frac{1}{m}\right).\end{aligned}$$

Next, we use orthogonality to compute

$$\begin{aligned}\boldsymbol{v}_3 \cdot \boldsymbol{v}_3 &= (\boldsymbol{v}_1 \times \boldsymbol{v}_2) \cdot (\boldsymbol{v}_1 \times \boldsymbol{v}_2) \\ &= (\boldsymbol{v}_1 \cdot \boldsymbol{v}_1)(\boldsymbol{v}_2 \cdot \boldsymbol{v}_2) - (\boldsymbol{v}_2 \cdot \boldsymbol{v}_1)(\boldsymbol{v}_1 \cdot \boldsymbol{v}_2) \\ &= \|\boldsymbol{v}_1\|_2\|\boldsymbol{v}_2\|_2\left[(1 + \mathcal{O}(1/m))\,(1 + \mathcal{O}(1/m)) - \mathcal{O}(1/m)\mathcal{O}(1/m)\right].\end{aligned}$$

From this we verify that

$$\boldsymbol{v}_3 \cdot \hat{\boldsymbol{v}}_3 = \frac{\|\boldsymbol{v}_1\|_2\|\boldsymbol{v}_2\|_2\,(1 + \mathcal{O}(1/m))}{\|\boldsymbol{v}_1\|_2\|\boldsymbol{v}_2\|_2} = 1 + \mathcal{O}\left(\frac{1}{m}\right).$$

These bounds on the inner products lead immediately to a bound on the angular resolution $d\theta$ via (21). □

Since the second type boundary condition (27) is typically coupled to PDEs that depend on the eigenvalues of the Hessian, it is natural to use this same discretization of unit vectors in the approximation of this boundary condition.

4.2 Identification of Orthogonal Coordinate Frames

In this section, we discuss the evaluation of the expression in (20), which requires computing a maximum/minimum over many different orthogonal frames in order to achieve a consistent approximation of functions of the eigenvalues of the Hessian.

Identifying all possible coordinate frames occurring in (30) can be done offline. For each integer stencil width $k \in \mathbb{N}$, we can use brute force to construct and save the coordinate frames that can be constructed using this stencil width. That is, we define

$$E_k = \left\{ \boldsymbol{\nu} \in \mathbb{Z}^3 \mid \|\boldsymbol{\nu}\|_\infty \leq k \right\}. \tag{31}$$

and

$$V_k = \left\{ (\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \mid \boldsymbol{\nu}_j \in E_k, \boldsymbol{\nu}_i \perp \boldsymbol{\nu}_j \forall i \neq j \right\}. \tag{32}$$

Then for a given problem (which may involve many different choices of domains, grid resolutions, particular PDEs, etc.), we simply define $V^h = V_{k^*}$ where $k^* = \lfloor \epsilon/h \rfloor$. As these have been pre-computed, there is effectively no computational cost to identifying the relevant coordinate frames.

However, actually solving a PDE involving the eigenvalues of the Hessian can require multiple evaluations of a minimum/maximum over all possible frames. In three dimensions, this becomes very expensive; see Table 1 for an overview of the number of coordinate frames as a function of stencil width. For this reason, we propose a multi-level approach for obtaining the solution of these maximum/minimum problems. We will focus the discussion on the problem of solving systems of the form

$$\max_{(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in V_{k^*}} F(u; \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) = 0, \tag{33}$$

which immediately provides a means for approximating schemes of the form (20).

The idea of our approach is to first solve (33) over coordinate frames of maximum stencil width one. From here, we identify the twenty-five coordinate frames (of maximum stencil width two) most closely aligned with the maximizer $(\boldsymbol{\nu}_1^{(1)}, \boldsymbol{\nu}_2^{(1)}, \boldsymbol{\nu}_3^{(1)})$ of the narrow stencil problem. We once again solve (33) over this small set of possible coordinate frames. This procedure can be repeated, maximizing over twenty-five coordinate frames at a time, until we are solving the system by maximizing over frames of the desired maximum width k^*. See Fig. 5 for a visualization of this procedure.

We remark also that this procedure is only being used to seek coordinate frames of particular stencil widths. These same (potentially wide stencil) coordinate frames are valid even at points close to the boundary of the domain. However, at these points it becomes necessary to use the generalized finite difference scheme (11) rather than the simple centered schemes (10).

We begin by producing a hierarchy of possible coordinate frames, which can be generated offline and stored. To accomplish this, we explicitly identify the different components appearing in the sets of coordinate frames V_k. That is, we write

$$V_k = \left\{ (\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \mid \boldsymbol{\nu}_1 \in V_k^{(1)}, \boldsymbol{\nu}_2 \in V_k^{(2;\boldsymbol{\nu}_1)}, \boldsymbol{\nu}_3 = \boldsymbol{\nu}_1 \times \boldsymbol{\nu}_2 \right\}$$

where $V_k^{(1)} = E_k$ and for each $\boldsymbol{\nu}_1 \in V_k^{(1)}$ we define a set

$$V_k^{(2;\boldsymbol{\nu}_1)} = \{ \boldsymbol{\nu}_2 \in E_k \mid \boldsymbol{\nu}_2 \perp \boldsymbol{\nu}_1 \}.$$

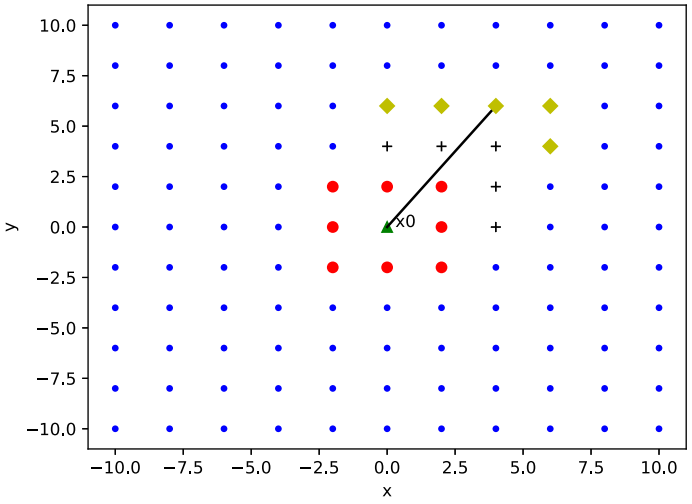

Fig. 5 A two-dimensional illustration of the multi-level process for one direction $\boldsymbol{v}_1$ in the orthogonal frame. The true direction that maximizes (33) is given by the black line. In the first level, we maximize over all the nearest (red dot) neighbors. We then identify the five (black plus) neighbors of stencil width two most closely aligned with the maximizer. After maximizing over these five neighbors, we continue the procedure by identifying the best five (yellow diamond) neighbors of stencil width three (Color figure online)

Next, we create a map from each $\boldsymbol{v}_1 \in V_k^{(1)}$ to the five most closely aligned vectors $\boldsymbol{v} \in V_{k+1}^{(1)}$. We introduce the notation

$$\alpha(\boldsymbol{v}, \boldsymbol{\mu}) = \cos^{-1}\left(\frac{\boldsymbol{v} \cdot \boldsymbol{\mu}}{\|\boldsymbol{v}\|\|\boldsymbol{\mu}\|}\right)$$

to denote the angle between the vectors $\boldsymbol{v}$ and $\boldsymbol{\mu}$. Then the map has the form

$$T_{k+1}^{(1)}(\boldsymbol{v}_1) = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4, \boldsymbol{\mu}_5\} \subset V_{k+1}^{(1)}$$

and satisfies the close alignment condition

$$\alpha(\boldsymbol{v}_1, \boldsymbol{\mu}) \leq \alpha(\boldsymbol{v}_1, \boldsymbol{\xi}) \text{ for every } \boldsymbol{\mu} \in T_{k+1}^{(1)}(v_1), \boldsymbol{\xi} \in V_{k+1}^{(1)} - T_{k+1}^{(1)}(\boldsymbol{v}_1).$$

Similarly, we produce a map from each $\boldsymbol{v}_2 \in V_k^{(2;v_1)}$ to the five most closely aligned vectors in $V_{k+1}^{(2;\boldsymbol{\mu})}$. This now has to be done for *every* $\boldsymbol{\mu} \in V_{k+1}^{(1)}(\boldsymbol{v}_1)$ since we need to obtain nearby orthogonal coordinate frames, not merely nearby vectors. That is, we define

$$T_{k+1}^{(2;\boldsymbol{\mu})}(\boldsymbol{v}_2) = \left\{\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3, \boldsymbol{\rho}_4, \boldsymbol{\rho}_5\right\} \subset V_{k+1}^{(2;\boldsymbol{\mu})}$$

satisfying the close alignment condition

$$\alpha(\boldsymbol{v}_2, \boldsymbol{\rho}) \leq \alpha(\boldsymbol{v}_2, \boldsymbol{\xi}) \text{ for every } \boldsymbol{\rho} \in T_{k+1}^{(2;\boldsymbol{\mu})}(\boldsymbol{v}_2), \boldsymbol{\xi} \in V_{k+1}^{(2;\boldsymbol{\mu})} - T_{k+1}^{(2;\boldsymbol{\mu})}(\boldsymbol{v}_2).$$

We emphasize again that all of the preceding work in building a hierarchy of maps can be accomplished offline and stored for later use in a wide variety of problems. Then the actual work of solving (33) involves solving a small number of similar problems, each involving at most twenty-five possible coordinate frames. The required online computations are summarized in the very short Algorithm 1.

Algorithm 1 Estimating the solution of (33) over orthogonal coordinate frames.

$W_1 = V_1$
for $k = 1, \ldots, k^* - 1$ **do**
  $u \Leftarrow$ solution of $\max_{(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in W_k} F(u; \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) = 0$
  $(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) = \operatorname{argmax}_{(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in W_k} F(u; \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3)$.
  $W_{k+1} = \left\{ (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3) \mid \boldsymbol{\mu}_1 \in T^{(1)}_{k+1}(\boldsymbol{\nu}_1), \boldsymbol{\mu}_2 \in T^{(2;\boldsymbol{\mu}_1)}_{k+1}(\boldsymbol{\nu}_2), \boldsymbol{\mu}_3 = \boldsymbol{\mu}_1 \times \boldsymbol{\mu}_2 \right\}$.
end for
$u \Leftarrow$ solution of $\max_{(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3) \in W_{k^*}} F(u; \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3)$.

4.3 Solution Methods

We also describe the techniques used in this implementation to solve the discrete system of nonlinear equations arising from our approximations. In three dimensions, it is not practical to explicitly build Jacobian matrices due to their prohibitively large sizes. Moreover, the schemes constructed in this paper are not differentiable with respect the unknown u because they involve a maximum or minimum. Finally, many of the PDEs we consider are degenerate and/or have singular solutions. Thus, Newton's method is not immediately suitable for these problems.

Our approach here is to use a combination of an active set approach [7], which has excellent stability properties for the nonlinear systems we consider, and a simple Gauss-Seidel iteration. Depending on the particular PDE of interest, the solver may collapse into only one of these methods or it may involve a combination. In the future, this approach could be accelerated using a nonlinear multigrid method.

Recall that we are trying to solve systems of the form

$$\max_{\mathcal{A} \in \Xi(\mathbf{x})} F(\mathbf{x}, u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}} u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}); \boldsymbol{\nu} \in \mathcal{A}) = 0, \quad \mathbf{x} \in \mathcal{G}. \tag{34}$$

Above, the $\mathcal{A}$ consist of finite combinations of admissible directions $\boldsymbol{\nu}$ and the set $\Xi(\mathbf{x})$ indicates the different combinations of directions that need to be considered. For example, for approximations of the form (17) involving a given set of admissible unit vectors $\mathcal{A}$, the set $\Xi(\mathbf{x}) = \{\mathcal{A}\}$ is a singleton. For approximations of the form (22) involving the eigenvalues of the Hessian, the set $\Xi(\mathbf{x}) = V^h$ and each $\mathcal{A} \in \Xi(\mathbf{x})$ consists of an orthogonal triplet. At boundary points involving approximation of the transport boundary conditions (28), each $\mathcal{A} = \{n\}$ consists of a single unit vector and the set

$$\Xi(\mathbf{x}) = \left\{ \{n\} \mid n \in E^h, n \cdot n_{\mathbf{x}} > 0 \right\}.$$

The basic approach is to iterate through a two step process. First, for a given input u, we identify the directions $\mathcal{A}(\mathbf{x}) \in \Xi(\mathbf{x})$ that maximize (34) at each point $\mathbf{x}$ in the computational domain. Secondly, we fix this direction and seek an approximate solution of

$$F(\mathbf{x}, u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}} u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}); \mathbf{x} \in \mathcal{A}(\mathbf{x})) = 0.$$

In order to implement the second part of the procedure, we recall that our finite difference systems can be written in the form

$$F(\mathbf{x}, u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}} u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}); \mathbf{x} \in \mathcal{A}) = G_{\mathcal{A}}(\mathbf{x}, u(\mathbf{x}), u(\cdot)).$$

As an example, we consider a standard centered difference approximation of Poisson's equation. In this case, $\mathcal{A} = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ and the finite difference operator is given

by

$$F(\mathbf{x}, u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}} u(\mathbf{x}), \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}); \mathbf{x} \in \mathcal{A}) = -\sum_{\boldsymbol{\nu}\in\mathcal{A}} \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u(\mathbf{x}) + f(\mathbf{x}),$$

which is equivalent to

$$G_{\mathcal{A}}(\mathbf{x}, u(\mathbf{x}), u(\cdot)) = -\frac{1}{h^2}\sum_{\boldsymbol{\nu}\in\mathcal{A}} (u(\mathbf{x}+h\boldsymbol{\nu}) + u(\mathbf{x}-h\boldsymbol{\nu})) + \frac{6}{h^2} u(\mathbf{x}) + f(\mathbf{x}).$$

We design a Gauss-Seidel iteration for this by solving for the reference value $u(\mathbf{x})$ in terms of the values at the remaining grid points. That is, we identify a function $G_{\mathcal{A}}^{-1}(\mathbf{x}, u(\cdot))$ such that

$$G_{\mathcal{A}}\left(\mathbf{x}, G_{\mathcal{A}}^{-1}(\mathbf{x}, u(\cdot)), u(\cdot)\right) = 0.$$

In some cases (for linear or simple nonlinear operators), this function $G_{\mathcal{A}}^{-1}$ can be identified explicitly. For example, in the example of Poisson's equation this function is given by

$$G_{\mathcal{A}}^{-1}(\mathbf{x}, u(\cdot)) = \frac{1}{6}\sum_{\boldsymbol{\nu}\in\mathcal{A}} (u(\mathbf{x}+h\boldsymbol{\nu}) + u(\mathbf{x}-h\boldsymbol{\nu})) - \frac{1}{6} f(\mathbf{x}) h^2.$$

In more complicated examples, this inverse can be obtained (or approximated) through several iterations of a nonlinear solver such as a scalar Newton's method.

The resulting solution method is described in Algorithm 2. This solver is simple to implement and memory efficient since there is no need to construct the Jacobian matrix. In practice, we can initialize the method with the solution computed on a less refined grid.

Algorithm 2 Solution method for (34)

while `Residual > Tolerance` **do**
  for $\mathbf{x} \in \mathcal{G}$ **do**
    $\mathcal{A}(\mathbf{x}) = \underset{\mathcal{A}\in\Xi(\mathbf{x})}{\operatorname{argmax}}\, G_{\mathcal{A}}(\mathbf{x}, u(\mathbf{x}), u(\cdot))$.
  end for
  for $k = 1, \ldots, 10$ **do**
    for $\mathbf{x} \in \mathcal{G}$ **do**
      $u(\mathbf{x}) = G_{\mathcal{A}(\mathbf{x})}^{-1}(\mathbf{x}, u(\cdot))$.
    end for
  end for
end while

4.4 Eigenvalue Problems

Our framework can also be used to solve eigenvalue problems involving fully nonlinear elliptic PDEs. These take the form

$$\begin{cases} F(\mathbf{x}, D^2 u(\mathbf{x})) = c, & \mathbf{x} \in \Omega \\ H(\mathbf{x}, \nabla u(\mathbf{x})) = 0, & \mathbf{x} \in \partial\Omega \\ u(\mathbf{x}_0) = 0. \end{cases} \tag{35}$$

Here the constant $c \in \mathbb{R}$ is unknown *a priori*. These arise from PDEs that require data to satisfy a solvability condition, which may not be known explicitly, may not be satisfied exactly by noisy data, or may not be satisfied at the discrete level even if the original continuous problem is well-posed. Examples that can be cast in this form include the Neumann problem for Poisson's equation, Monge–Ampère type equations in optimal transport, and the problem of computing minimal Lagrangian graphs [8].

With minor modification, monotone schemes can be used to correctly compute both the eigenvalue c and the solution u [25]. Thus our discretization applies immediately to these problems. The only modification needed is to add an additional unknown c to the nonlinear system, which is also augmented by an additional equation ($u(\mathbf{x}_0) = 0$) designed to select a unique solution.

5 Computational Results

We demonstrate the effectiveness of the method by solving a variety of computational examples including a range of challenging nonlinear PDEs and different boundary conditions.

Many of our test examples are smooth (C^∞) as this makes exact solutions easier to procure and allows us to test whether the formal discretization error can be achieved in practice. However, our implementation is also designed to converge to weak solutions, as evidenced by a test case (the convex envelope equation) where the solution is only Lipschitz continuous.

Our focus in these tests is in demonstrating (1) convergence of the discrete solutions u^h to the true solution u, (2) efficiency of the proposed procedures for setting up the computational domain and finite difference stencils, and (3) practicality of our multi-level procedure for evaluating functions of the eigenvalues of the Hessian. The method proposed in Algorithm 2 is robust and allows us to easily test these properties of our schemes. However, the development of a fast solver is not a focus of the present work and actual solver times can vary wildly between different problems and grid resolutions.

In Sects. 5.1–5.6, we discuss a number of different examples with a view towards verifying convergence and accuracy of the method, which is strongly dependent on the particular details of the equation and solution regularity. In Sect. 5.7, we provide a summary of the computational cost associated with constructing and evaluating our finite difference approximations, which turns out to be essentially independent of the particular problem.

5.1 Linear Degenerate Equation

We begin by considering a linear degenerate equation posed on the unit sphere:

$$\begin{cases} -u_{\boldsymbol{\nu}\boldsymbol{\nu}}(x, y, z) = 0, & x^2 + y^2 + z^2 < 1 \\ u(x, y, z) = \sin\left(2\pi(x - \sqrt{2}y - \sqrt{3}z)\right), & x^2 + y^2 + z^2 = 1 \end{cases} \tag{36}$$

where $\boldsymbol{\nu} = \left(1, -1, \frac{-(\sqrt{3}+\sqrt{6})}{3}\right)$. The exact solution is

$$u(x, y, z) = \sin\left(2\pi(x - \sqrt{2}y - \sqrt{3}z)\right).$$

Note that the direction $\boldsymbol{\nu}$ is not aligned with any Cartesian grid. For this example, neither the grid aligned scheme we derived nor any other grid aligned scheme can be used for a consistent, monotone approximation [21,30]. Consequently, the generalized finite difference schemes must be used exclusively.

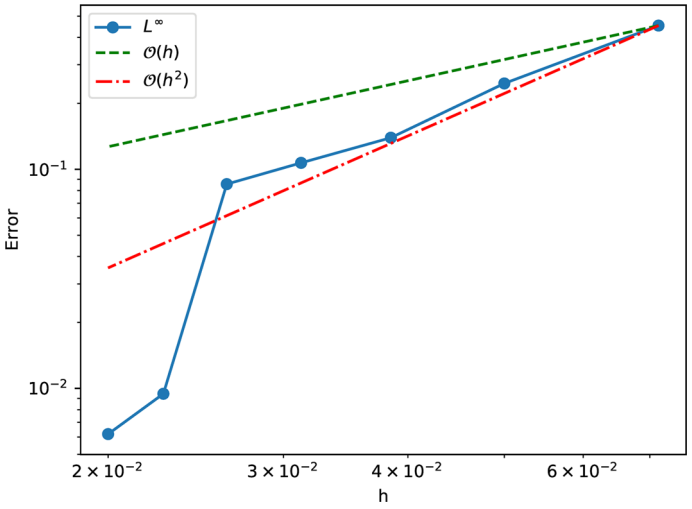

Fig. 6 A convergence plot for the linear degenerate Eq. (36) (Color figure online)

Convergence results are presented in Fig. 6 and demonstrate better than the expected $\mathcal{O}(\sqrt{h})$ accuracy. At one point, we observe a great jump in accuracy as h is refined slightly. This is due to the scaling of the ratio $n_B = h/h_B \approx n^{1/4}$, which is constrained to be an integer in our particular implementation in order to facilitate efficient discretization of the domain Ω. This causes the angular error $d\phi$ near the boundary to decrease discontinuously rather than smoothly as the grid is refined. Smooth relationships between h and h_B are also possible, and would not be expected to produce this artifact in the convergence plot. Regardless, the ability of our scheme to improve angular resolution as the grid is refined is of particular importance in this type of fully non-aligned PDE operator, for which the angular discretization error can easily dominate.

5.2 Maximum of Linear Operators

For a second example, consider the following fully nonlinear PDE:

$$\begin{cases} \max\{-u_{\boldsymbol{v}_1\boldsymbol{v}_1}, -u_{\boldsymbol{v}_2\boldsymbol{v}_2}\} = f(x,y,z) & x^2+y^2+z^2 < 1 \\ u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2 = 1 \end{cases} \tag{37}$$

where

$$\boldsymbol{v}_1 = (1,1,0), \quad \boldsymbol{v}_2 = (-1,0,1)$$

and

$$f(x,y,z) = \max\left\{-\frac{1}{2}e^{\frac{x^2+y^2+z^2}{2}}(2+x^2+2xy+y^2), -\frac{1}{2}e^{\frac{x^2+y^2+z^2}{2}}(2+x^2-2xz+z^2)\right\}.$$

The exact solution is

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}.$$

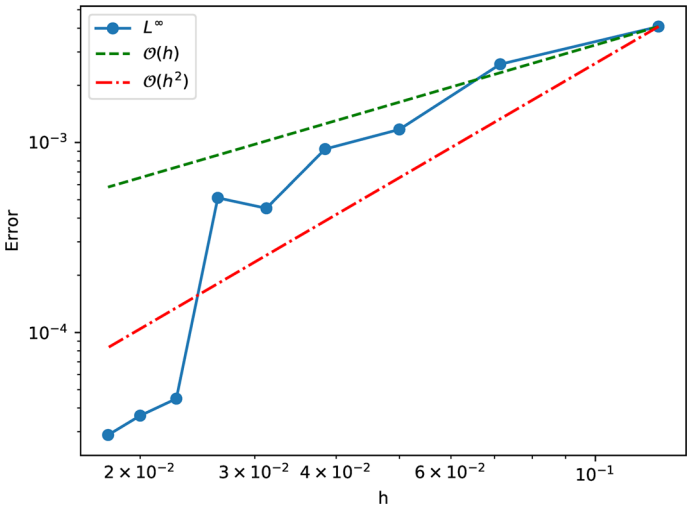

Fig. 7 A convergence plot for the two-operator problem (37) (Color figure online)

The convergence plot is presented in Fig. 7. As with the linear degenerate equation, there is a discrete jump at one point due to a discrete increase in the boundary resolution. Once again, we observe better accuracy than the expected $\mathcal{O}(\sqrt{h})$ for this fully nonlinear problem.

5.3 Convex Envelope Equation

Next, we consider a PDE for computing the convex envelope of an obstacle [31].

$$\begin{cases} \max\{-\lambda_1(D^2u), u - g\} = 0 & x^2 + y^2 + z^2 < 0.25 \\ u = 0.2 & x^2 + y^2 + z^2 = 0.25 \end{cases} \tag{38}$$

where

$$g(x, y, z) = \min\left\{2\sqrt{(x^2 + y^2 + z^2)}, 0.2\right\}.$$

In addition to being a fully nonlinear equation, the solution to this PDE is only Lipschitz continuous (but is not differentiable at the origin). Thus, it must be interpreted in a weak sense, and the use of a discretization that converges to the viscosity solution is imperative. The exact solution for this problem is the cone

$$u(x, y, z) = 0.4\sqrt{x^2 + y^2 + z^2}.$$

We remark that this PDE involves only the smallest eigenvalue of the Hessian matrix. It can therefore be characterized using the traditional Rayleigh–Ritz form and discretized as

$$\lambda_1(D^2u) \approx \min_{\boldsymbol{\nu} \in E^h} \mathcal{D}_{\boldsymbol{\nu}\boldsymbol{\nu}} u.$$

The convergence plot is presented in Fig. 8. Note that convergence is not monotone in this case. This is due to effects of variations in the alignment of the grid points for different n (by chance, some small values of n can lead to grids that are very well aligned with the singularity). This effect has previously been observed in two dimensions for problems with

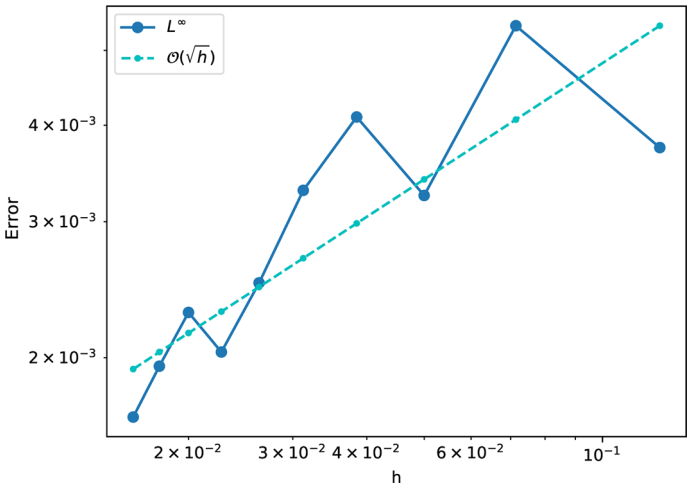

Fig. 8 A convergence plot for the convex envelope equation (38) (Color figure online)

very low regularity [23]. Nevertheless, we observe overall convergence close to $\mathcal{O}(\sqrt{h})$ even on this very singular example.

5.4 Monge–Ampère Equation

Next, we turn our attention to more general functions of the eigenvalues of the Hessian matrix. We begin with the Monge–Ampère equation:

$$\begin{cases} -\det\left(D^2 u(\mathbf{x})\right) + f(\mathbf{x}) = 0, & \mathbf{x} \in \Omega \\ u(x) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \\ u \text{ is convex.} \end{cases}$$

The determinant can be expressed as a product of the eigenvalues. Since the equation is only elliptic on the space of convex functions, we follow [20] and use the globally elliptic extension

$$-\max(\lambda_1, 0)\max(\lambda_2, 0)\max(\lambda_3, 0) - (\min(\lambda_1, 0) + \min(\lambda_2, 0) + \min(\lambda_3, 0)) + f = 0. \tag{39}$$

We notice that this can be decomposed into two different functions of the eigenvalues, each of which can be written in the form of (2). That is, let $\phi_1(x) = \log\max\{x, 0\}$, $G_1(x) = -e^x$, $\phi_2(x) = \min\{x, 0\}$, and $G_2(x) = -x$. Then we can re-express this Monge–Ampère equation as

$$G_1\left(\sum_{j=1}^{3} \phi_1(D^2 u(\mathbf{x}))\right) + G_2\left(\sum_{j=1}^{3} \phi_2(D^2 u(\mathbf{x}))\right) + f(\mathbf{x}) = 0,$$

similar to [22]. This now fits within the framework we require to produce consistent, monotone approximations.

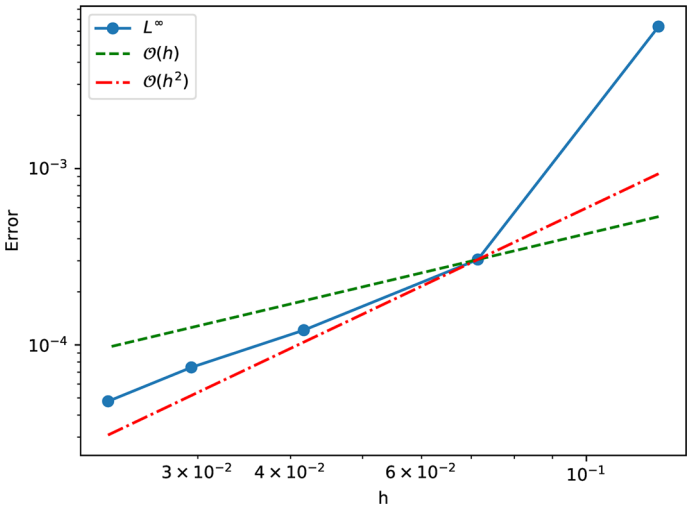

Fig. 9 A convergence plot for the Monge–Ampère equation (40) (Color figure online)

Consider the specific example

$$\begin{cases} -\det\left(D^2u(x,y,z)\right) + e^{\frac{3}{2}(x^2+y^2+z^2)}(1+x^2+y^2+z^2) = 0, & x^2+y^2+z^2 < .25 \\ u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}} & x^2+y^2+z^2 = .25 \\ u \text{ is convex.} \end{cases} \tag{40}$$

with the exact solution being

$$u(x,y,z) = e^{\frac{x^2+y^2+z^2}{2}}.$$

The results are included in Fig. 9. On this example, we also observe better than the expected $\mathcal{O}(\sqrt{h})$ convergence.

5.5 Neumann Boundary Conditions

Next, we consider Poisson's equation with Neumann boundary conditions. The point of this example is, of course, not to produce a new method for solving Poisson's equation. Instead, we use it to test our characterization of functions of the eigenvalues of the Hessian, our generalized finite difference implementation of Neumann boundary conditions, and our ability to solve eigenvalue problems.

We recall that the data must satisfy a solvability condition in order for a solution to exist. Moreover, even if the continuous problem is well-posed, the discretized problem need not be [25]. Therefore, we choose to frame this as the following eigenvalue problem:

$$\begin{cases} -\left(\lambda_1(D^2u) + \lambda_2(D^2u) + \lambda_3(D^2u)\right) = cf(x,y,z) & x^2+y^2+z^2 < 1 \\ \frac{\partial u(x,y,z)}{\partial \hat{n}} = e^{\frac{1}{2}} & x^2+y^2+z^2 = 1 \end{cases} \tag{41}$$

where

$$f(x,y,z) = (3+x^2+y^2+z^2)e^{\frac{x^2+y^2+z^2}{2}}.$$

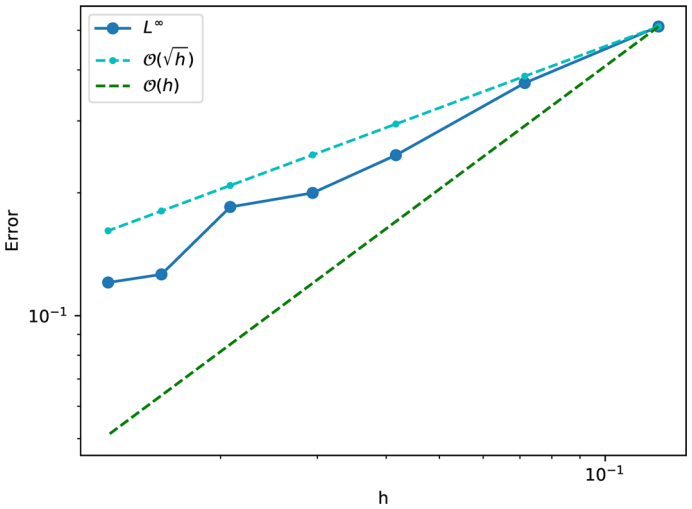

Fig. 10 A convergence plot for Poisson's equation with Neumann boundary conditions (41) (Color figure online)

The exact solution is

$$u(x, y, z) = e^{\frac{x^2+y^2+z^2}{2}}$$

with $c = 1$.

To further test our characterization of functions of the eigenvalues of the Hessian, we also note that the Laplacian does trivially have the form of (2) with $\phi(x) = x$ and $G(x) = -x$. The results are presented in Fig. 10. On average we observe the expected $\mathcal{O}(\sqrt{h})$ accuracy on this example.

5.6 Second Type Boundary Conditions

Finally, we consider the problem of computing minimal Lagrangian graphs [8]. This is an eigenvalue problem for a nonlinear PDE, equipped with the second type (optimal transport) boundary condition. Specifically, we seek a convex function u and a scalar constant $c \in \mathbb{R}$ satisfying

$$\begin{cases} \tan^{-1}\left(\lambda_1(D^2u)\right) + \tan^{-1}\left(\lambda_2(D^2u)\right) + \tan^{-1}\left(\lambda_3(D^2u)\right) = c, \quad x^2 + y^2 + z^2 < 1 \\ \nabla u(\mathbb{S}^2) \subset T(\mathbb{S}^2). \end{cases} \tag{42}$$

where $T(x, y, z) = (x + 2, y + 1, z - 1)$ is an affine shift. The exact solution is

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

In order to discretize this function of the eigenvalues of the Hessian, we first need to put the PDE operator into the form of (2). To accomplish this, we introduce a modification that agrees with (42) on the space of convex functions (which is where the desired solution lives).

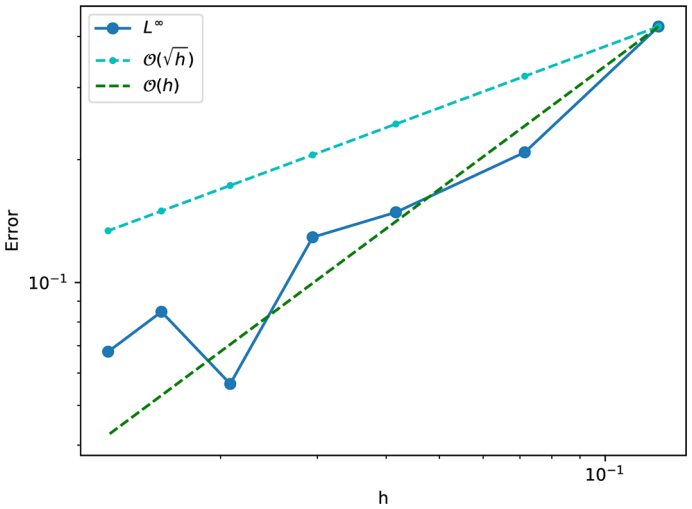

Fig. 11 A convergence plot for the computation of minimal Lagrangian graphs (42) (Color figure online)

In particular, we propose the alternate operator

$$-\sum_{j=1}^{3}\left(\tan^{-1}\left(\max\{\lambda_j,0\}\right)+\min\{\lambda_j,0\}\right),$$

which fits immediately into the required form. The optimal transport type boundary constraint is discretized as described in Sect. 3.4.2.

The computed results are shown in Fig. 11. Once again, we observe slightly better than the expected $\mathcal{O}(\sqrt{h})$ accuracy overall.

5.7 Computational Cost

We now summarize the computational cost required to construct and evaluate the finite difference approximations (17) or (22) at every point in the computational domain $\mathcal{G}$. In the following discussion, N represents the total number of grid points. All computations were performed in Python on a 2012 Windows 7 laptop with 8GB of RAM and a 4 core AMD A6 vision processor.

We first present the time required to set-up the domain and finite difference stencils for each example. This in itself is a complex problem, which involves a delicate procedure for discretizating the boundary and the need to construct many wide stencil finite difference approximations. Nevertheless, the cost of our implementation is essentially linear in the total number of grid points. See Fig. 12a.

We also report the computation time required to evaluate the finite difference approximations (17) or (22) at every point in the computational domain. See Fig. 12b. Once again, we observe a cost that is essentially linear in the total number of grid points. This is particularly telling for the examples involving the eigenvalues of the Hessian matrix (Monge–Ampère equation, Poisson's equation, and Lagrangian equation), which were computed using the multi-level procedure of Algorithm 1.

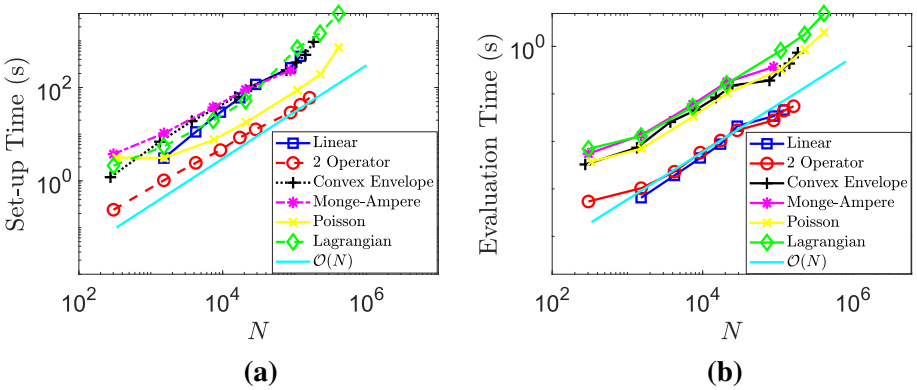

Fig. 12 **a** Set-up times and **b** evaluation times for the examples in Sects. 5.1–5.6 (Color figure online)

Table 1 Stencil widths (given by round$(1/\sqrt{h})$) and number of coordinate frames required to evaluate (22) using either a brute force approach or the multi-level approach of Algorithm 1

h	N	Stencil width	# Coordinate frames	
			Brute force	Multi-level
0.125	303	3	632	78
0.071	1520	4	1528	103
0.042	7377	5	3720	128
0.029	20,799	6	6264	153
0.023	87,979	7	12,176	178

To further elucidate the cost savings of Algorithm 1, we also display the stencil width and corresponding number of coordinate frames that need to be checked using either the brute force procedure or the multi-level procedure for our computational examples; see Table 1. The multi-level procedure has dramatically reduced this cost. In fact, the computational cost of the brute force procedure made it impossible for us to carry out these computations for a stencil width greater than 2 on the finest grids we consider. Thus this procedure should be viewed as essential to our ability to construct these finite difference schemes in three dimensions.

6 Conclusion

In this paper, we introduced a new monotone finite difference method for solving a wide variety of fully nonlinear elliptic equations in three dimensions. Because the resulting schemes are monotone, they are guaranteed to converge via the Barles-Souganidis convergence framework [2] and generalizations of these to non-classical Dirichlet problems [23], optimal transport problems [24], and eigenvalue problems involving nonlinear PDEs [25].

In particular, we described a new technique for discretizing general three-dimensional domains, which produces the higher boundary resolution needed to preserve both consistency and monotonicity throughout the entire domain. We also introduced and analyzed a simple least-squares method for generating consistent, monotone approximations of second directional derivatives. Moreover, we showed how to use these to construct monotone approximations of a large range of fully nonlinear elliptic operators. Finally, we produced

generalized finite difference approximations for a range of different boundary conditions including Dirichlet, Neumann, and the nonlinear second type (optimal transport) boundary condition.

This paper focused primarily on efficiently constructing these approximations in three dimensions, which is much more expensive than the analogous problem in two dimensions. Because our grids inherit much of the structure of a Cartesian grid, constructing the finite difference stencils is straightforward throughout most of the domain. A more serious computational challenge in three dimensions is evaluating nonlinear operators that require computing a maximum/minimum over many different orthogonal coordinate frames; this is needed for many fully nonlinear operators. We proposed a multilevel approach to this evaluation of the nonlinear operators, which converted the problem from one that is completely intractable in 3D to a very efficient process.

In future work, we intend to leverage this new scheme, which can be evaluated very efficiently, to produce numerical methods that are both efficient and higher-order. In particular, we will develop faster solvers for the discretized systems that utilize the underlying structure of the monotone approximations. We will also use these as a foundation for convergent, higher-order filtered methods [19].

Funding This research was supported by NSF DMS-1619807 and NSF DMS-1751996

Availability of Data andMaterials Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Code Availability Not applicable.

Conflict of interest Not applicable.

References

1. Awanou, G.: Standard finite elements for the numerical resolution of the elliptic Monge–Ampère equation: classical solutions. IMA J. Numer. Anal. **35**(3), 1150 (2015)
2. Barles, G., Souganidis, P.E.: Convergence of approximation schemes for fully nonlinear second order equations. Asym. Anal. **4**(3), 271–283 (1991)
3. Bates, P.W., Wei, G.W., Zhao, S.: Minimal molecular surfaces and their applications. J. Comp. Chem. **29**(3), 380–391 (2008)
4. Benamou, J.D., Froese, B.D., Oberman, A.M.: Two numerical methods for the elliptic Monge–Ampère equation. Math. Model. Numer. Anal. **44**(4), 737–758 (2010)
5. Benamou, J.D., Oberman, A., Froese, B.: Numerical solution of the second boundary value problem for the elliptic Monge–Ampère equation (2012)
6. Böhmer, K.: On finite element methods for fully nonlinear elliptic equations of second order. SIAM J. Numer. Anal. **46**(3), 1212–1249 (2008)
7. Bokanowski, O., Maroso, S., Zidani, H.: Some convergence results for Howard's algorithm. SIAM J. Numer. Anal. **47**(4), 3001–3026 (2009)
8. Brendle, S., Warren, M.: A boundary value problem for minimal Lagrangian graphs. J. Diff. Geom. **84**(2), 267–287 (2010). https://doi.org/10.4310/jdg/1274707314
9. Brenner, S.C., Gudi, T., Neilan, M., Sung, L.Y.: C^0 penalty methods for the fully nonlinear Monge–Ampére equation. Math. Comp. **80**(276), 1979–1995 (2011)
10. Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Am. Math. Soc. (N.S.) **27**(1), 1–67 (1992)
11. Dean, E.J., Glowinski, R.: Numerical methods for fully nonlinear elliptic equations of the Monge–Ampère type. Comput. Methods Appl. Mech. Eng. **195**(13–16), 1344–1386 (2006)

12. Engquist, B., Froese, B.D.: Application of the Wasserstein metric to seismic signals. Commun. Math. Sci. **12**(5), 979–988 (2014)
13. Feng, X., Lewis, T.: A narrow-stencil finite difference method for approximating viscosity solutions of fully nonlinear elliptic partial differential equations with applications to Hamilton–Jacobi–Bellman equations. arXiv preprint arXiv:1907.10204 (2019)
14. Feng, X., Neilan, M.: Vanishing moment method and moment solutions for fully nonlinear second order partial differential equations. SIAM J. Sci. Comput. **38**(1), 74–98 (2009)
15. Finlay, C., Oberman, A.: Improved accuracy of monotone finite difference schemes on point clouds and regular grids. SIAM J. Sci. Comput. **41**(5), A3097–A3117 (2019)
16. Finn, J.M., Delzanno, G.L., Chacón, L.: Grid generation and adaptation by Monge–Kantorovich optimization in two and three dimensions. In: Proceedings of the 17th International Meshing Roundtable, pp. 551–568 (2008). https://doi.org/10.1007/978-3-540-87921-3_33
17. Fleming, W., Soner, H.: Controlled Markov Processes and Viscosity Solutions. Sto. Model. Appl. Prob. Springer, New York (2006). https://books.google.com/books?id=4Bjz2iWmLyQC
18. Frisch, U., Matarrese, S., Mohayaee, R., Sobolevski, A.: A reconstruction of the initial conditions of the Universe by optimal mass transportation. Nature **417**, 260–262 (2002). https://doi.org/10.1038/417260a
19. Froese, B., Oberman, A.: Convergent filtered schemes for the Monge–Ampere partial differential equation. SIAM J. Numer. Anal. **51**(1), 423–444 (2013). https://doi.org/10.1137/120875065
20. Froese, B.D.: A numerical method for the elliptic Monge–Ampère equation with transport boundary conditions. SIAM J. Sci. Comput. **34**(3), A1432–A1459 (2012)
21. Froese, B.D.: Meshfree finite difference approximations for functions of the eigenvalues of the Hessian. Numer. Math. **138**(1), 75–99 (2018)
22. Froese, B.D., Oberman, A.M.: Convergent finite difference solvers for viscosity solutions of the elliptic Monge–Ampère equation in dimensions two and higher. SIAM J. Numer. Anal. **49**(4), 1692–1714 (2011)
23. Hamfeldt, B.: Convergent approximation of non-continuous surfaces of prescribed Gaussian curvature. Commun. Pure Appl. Anal. **17**(2), 671–707 (2018)
24. Hamfeldt, B.: Convergence framework for the second boundary value problem for the Monge–Ampère equation. SIAM J. Numer. Anal. **57**(2), 945–971 (2019). https://doi.org/10.1137/18M1201913
25. Hamfeldt, B.F., Lesniewski, J.: A convergent finite difference method for computing minimal Lagrangian graphs. Commun. Pure Appl. Anal. (in press) (2021)
26. Kocan, M.: Approximation of viscosity solutions of elliptic partial differential equations on minimal grids. Numer. Math. **72**(1), 73–92 (1995)
27. Loeper, G., Rapetti, F.: Numerical solution of the Monge–Ampère equation by a Newton's algorithm. C. R. Math. Acad. Sci. Paris **340**(4), 319–324 (2005)
28. Mirebeau, J.M.: Discretization of the 3d Monge–Ampere operator, between wide stencils and power diagrams. ESAIM Math. Model. Numer. Anal. **49**(5), 1511–1523 (2015)
29. Motzkin, T.S., Wasow, W.: On the approximation of linear elliptic differential equations by difference equations with positive coefficients. J. Math. Phys. **31**(1–4), 253–259 (1952)
30. Motzkin, T.S., Wasow, W.: On the approximation of linear elliptic differential equations by difference equations with positive coefficients. J. Math. Phys. **31**(1–4), 253–259 (1952). https://doi.org/10.1002/sapm1952311253
31. Oberman, A.: The convex envelope is the solution of a nonlinear obstacle problem. Proc. Am. Math. Soc. **135**(6), 1689–1694 (2007)
32. Oberman, A.M.: Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi equations and free boundary problems. SIAM J. Numer. Anal. **44**(2), 879–895 (2006)
33. Oberman, A.M.: Wide stencil finite difference schemes for the elliptic Monge–Ampère equation and functions of the eigenvalues of the Hessian. Disc. Cont. Dynam. Syst. Ser. B **10**(1), 221–238 (2008)
34. Saumier, L.P., Agueh, M., Khouider, B.: An efficient numerical algorithm for the L2 optimal transport problem with periodic densities. IMA J. Appl. Math. **80**(1), 135–157 (2015)
35. Smears, I., Suli, E.: Discontinuous Galerkin finite element approximation of Hamilton–Jacobi–Bellman equations with Cordes coefficients. SIAM J. Numer. Anal. **52**(2), 993–1016 (2014)
36. Stark, P., Parker, R.: Bounded-variable least-squares: an algorithm and applications. J. Comput. Stat. **10**, 129 (1995)
37. Sulman, M., Williams, J.F., Russell, R.D.: Optimal mass transport for higher dimensional adaptive grid generation. J. Comput. Phys. **230**(9), 3302–3330 (2011)
38. Thomas, E.L., Anderson, D.M., Henkee, C.S., Hoffman, D.: Periodic area-minimizing surfaces in block copolymers. Nature **334**(6183), 598 (1988)
39. Williamson, D.: Lecture notes in mathematical programming I (2008)